한국정보보호학회
Korea Institute of Information Security & Cryptology

CISC-W'17

Conference on Information Security and Cryptography 2017 Winter

한국정보보호학회 동계학술대회

일 시 : 2017년 12월 9일 토요일

장 소 : 고려대학교 안암캠퍼스



> 주 최 : 한국정보보호학회 Korea Institute of Information Security & Cryptology

> 주 관 : 고려대학교 KOREA UNIVERSITY

> 후 원 : 국가정보원 National Intelligence Service Korea, 과학기술정보통신부 Ministry of Science and ICT, KISA 한국인터넷진흥원, ETRI 한국전자통신연구원 Electronics and Telecommunications Research Institute, NSR 국가보안기술연구소, IITP 정보통신기술진흥센터 Institute for Information & communications Technology Promotion, 사이버보안연구센터, NCSOFT, EEDIOM, Penta SECURITY, 보안프로젝트 WWW.BOANPROJECT.COM, SAINT SECURITY, aws, kakao

초대의 글

2017년도 한국정보보호학회 동계학술대회를 맞이하여 회원 여러분 모두의 무궁한 발전을 기원드립니다.

본 동계학술대회는 우리 학회에서 개최하는 주요 학술대회 중 하나로서 회원 여러분의 지대한 관심과 적극적인 참여에 힘입어 1990년 학회 창립 이래 계속해서 발전해 오고 있는 대표적인 행사입니다.

금년 동계학술대회는 고려대학교 안암캠퍼스에서 영어 발표 세션, 고교생 세션을 포함한 25개 구두 발표 세션과 포스터 세션에서 총 134편의 논문이 발표됩니다. 또한, 정보통신기술진흥센터 주관의 정보보호 분야 한-미공군 과학연구실 공동연구 발표와 정보보호 R&D 데이터 챌린지 세션이 진행됩니다.

본 동계학술대회에서 발표될 논문들은 정보보호의 초석이라 할 수 있는 제반 암호 이론 뿐만 아니라 IoT 보안, 시스템 보안 등 다양한 응용환경에서 요구되는 정보보호기술, 그리고 관련 정책 및 표준들이 포함되어 있습니다. 이와 더불어 최근 이슈가 되고 있는 블록체인 세션 등을 마련하여 연구 결과에 대한 실무자간 토론 및 친목을 도모할 수 있는 자리를 마련하였습니다.

정보보호 분야는 몇 년 전부터 국내의 새로운 성장 동력으로 다양한 분야들과 융합을 하면서 점점 활동 영역을 넓혀가고 있는 가운데, 그 중요성과 위상이 더욱 공고히 되어가고 있습니다.

본 동계학술대회는 ICT 방송통신 융합과 정보보호를 주제로 학계, 산업계, 연구기관의 연구자뿐만 아니라, 정보보호 관련 정부기관, 민간기관 소속 전문가들 간의 학문적 교류 및 관련 연구를 수행하는 과정에서 도출된 새로운 연구결과를 발표하는 장으로 마련하였습니다. 여러분들의 많은 관심을 부탁드립니다.

마지막으로 이번 동계학술대회 준비를 위해서 많은 수고를 해주신 운영위원회 및 프로그램위원회 위원님들과 학회관계자, 그리고 후원기관 관계자 분들께 감사의 말씀을 드립니다.

2017. 12.

한국정보보호학회 회장 **이 동 훈**

조직도

학술대회장

한국정보보호학회 회장 이동훈 고려대학교

조직위원회

조직위원장	임종인	고려대학교			
조직위원	류재철	충남대학교	이재일	정보통신기술진흥센터
	손기욱	국가보안기술연구소	정수환	숭실대학교
	신대규	한국인터넷진흥원	진승헌	한국전자통신연구원
	이경현	부경대학교	하재철	호서대학교
	이옥연	국민대학교	홍만표	아주대학교

운영위원회

운영위원장	김휘강	고려대학교		
운영위원	강동주	고려대학교	심상규	펜타시큐리티
	곽 진	아주대학교	오형근	국가보안기술연구소
	김경곤	고려대학교	원유재	충남대학교
	김혁준	나루시큐리티	유동영	한국인터넷진흥원
	김호원	부산대학교	이경호	고려대학교
	나중찬	한국전자통신연구원	이정현	숭실대학교

프로그램위원회

프로그램위원장	임을규	한양대학교		
프로그램위원	Simon Woo	뉴욕주립대학교-송도	양대헌	인하대학교
	권태경	연세대학교	유일선	순천향대학교
	김상진	한국기술교육대학교	윤명근	국민대학교
	김용대	KAIST	윤지원	고려대학교
	김윤정	서울여자대학교	이만희	한남대학교
	김종성	국민대학교	이문규	인하대학교
	김형식	성균관대학교	이창훈	서울과학기술대학교
	김형종	서울여자대학교	임강빈	순천향대학교
	박민호	숭실대학교	조성제	단국대학교
	박영호	세종사이버대학교	조진성	경희대학교
	서정택	순천향대학교	최대선	공주대학교
	손태식	아주대학교	한동국	국민대학교
	안효범	공주대학교		

프로그램 일정표

시간	트랙 A 미래융합기술관 105호	트랙 B 미래융합기술관 110호	트랙 C 하나스퀘어 B119호	트랙 D 하나스퀘어 B120호	트랙 E 과학도서관 209호	포스터 미래융합기술관 106호	IITP / 정보보호 R&D 데이터 챌린지 미래융합기술관 601호
08:00 – 09:00	등 록						
09:00 – 10:00	A1 시스템 보안(Ⅰ)	B1 시스템 보안(Ⅱ)	C1 정보보호모델	D1 해킹 및 취약점(Ⅰ)	E1 모바일보안(Ⅰ)	포스터 1–1	IITP 정보보호 분야 한–미공군 과학연구실 공동연구 발표
10:00 – 10:10	휴 식						
10:10 – 11:10	A2 블록체인(Ⅰ)	B2 정보보호관련법 및 제도	C2 IoT보안(Ⅰ)	D2 해킹 및 취약점(Ⅱ)	E2 모바일보안(Ⅱ)	포스터 1–2	
11:10 – 11:20	휴 식						
11:20 – 12:00	**개회식** | 미래융합기술관 대강당(101호) • 개 회 사 : 학 회 장 이동훈(고려대학교) • 축 사 : 정보대학원장 이상진(고려대학교) • 학회관련안내 : 운영위원장 김휘강(고려대학교) • 우수논문 시상 및 경품추첨						
12:00 – 13:00	중식 (애기능 생활관 2층 식당)						
13:30 – 14:30	A3 자동차보안	B3 양자암호	C3 IoT보안(Ⅱ)	D3 해킹 및 취약점(Ⅲ)	E3 모바일보안(Ⅲ)	포스터 2–1	정보보호 R&D 데이터 챌린지
14:30 – 14:40	휴 식						
14:40 – 15:40	A4 클라우드보안	B4 블록체인(Ⅱ)	C4 네트워크 보안(Ⅰ)	D4 디지털포렌식	E4 암호이론(Ⅰ)	포스터 2–2	
15:40 – 15:50	휴 식						
15:50 – 17:05	A5 영어논문	B5 고교생논문	C5 네트워크 보안(Ⅱ)	D5 인증 및 키 관리	E5 암호이론(Ⅱ)	포스터 2–3	
17:05 – 17:10	휴 식						
17:10 – 18:00	총회 | 미래융합기술관 대강당(101호)						

우수 논문상

순번	시상내역	제출자 소속	논문 저자	논문 제목
1	행안부장관상	고려대학교	진홍주	난독화 방법에 따른 Clang 정적 분석 도구의 성능 영향 분석
2	KISA원장상	단국대학교	한상우, 조정모, 박창섭	DTLS 기반의 CoAP 보안 메커니즘 분석 및 성능평가
3	KISA원장상	부경대학교	정채린, 이종호, 김영우, 조은아, 성기정, 김희영, 이경현	블록체인 기반 교내 전자투표 시스템 구축 요구사항 분석
4	ETRI원장상	순천향대학교	황용운, 이임영	공개된 채널환경에서의 NTRU 기반의 RFID 상호인증에 관한 연구
5	ETRI원장상	국민대학교	박호중, 김원태, 염용진, 강주성	다중 바이트 간 독립성 검정을 이용한 소프트웨어 잡음원의 엔트로피 추정방법
6	NSR소장상	단국대학교	김규식, 심재우, 임경환, 조성제	자마린에서 C#으로 개발된 안드로이드 앱에 대한 정적 역공학 분석
7	NSR소장상	KAIST	계정오, 김은수, 김동관, 김용대	TVT: Typed Virtual Table for mitigating VTable Hijacking
8	학회최우수논문상	아주대학교, 국가보안기술연구소	배원일, 황우민, 김형천, 곽진	멀티 프로세서 환경에서 가상머신간 은닉채널 생성에 관한 연구
9	우수 논문상	KAIST	신정민, 노주환, 신호철, 김용대	초음파를 이용한 상용 드론 고도 측정 시스템 스푸핑 공격
10	우수 논문상	고려대학교, 전북대학교	박연지, 이호창, 홍득조, 홍석희	S-box기반 암호의 최적 차분경로 자동 탐색 방법(MILP 적용)
11	우수 논문상	중부대학교	지우중, 이경문, 이병천	프린터 관리 시스템의 취약점 분석
12	우수 논문상	SUNY Korea, USC, San Jose State Univ.	Simon Woo, Wenzhe Li, Hyeran Jeon	Toward Machine Generated Passwords

구두 발표 트랙 A

미래융합기술관 105호

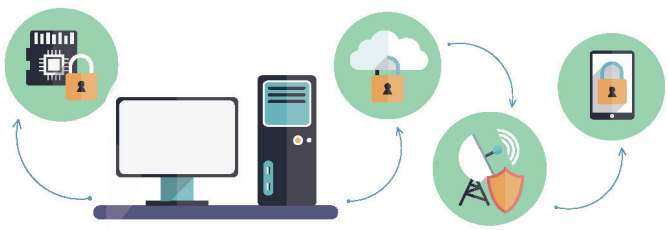
A1	발표 논문
시스템보안 (I) 09:00 – 10:00	멀티 프로세서 환경에서 가상머신간 은닉채널 생성에 관한 연구 **배원일** (아주대학교), **황우민**, **김형천** (국가보안기술연구소), **곽진** (아주대학교)
	쿠쿠 샌드박스의 정보를 이용한 랜섬웨어 특징정보 추출 **옥정윤**, **김준형**, **임을규** (한양대학교)
	Practical Plug & Play Measurement Device Independent Quantum Key Distribution System Changhoon Park, Byung Kwon Park, Min Soo Lee, Young-Wook Cho, Yong-Su Kim, Sang-Wook Han and Sung Moon (KIST), Min Ki Woo, Sangin Kim (아주대학교)
	Ponce를 활용한 기호실행 기반 악성프로세스를 발현시키는 방법 연구 **한중수**, **임을규** (한양대학교)
좌장: **강동주** (고려대학교)	

A2	발표 논문
블록체인 (I) 10:10 – 11:10	블록체인 기반 IoT 디바이스 무결성 검증 스킴 **박병주**, **곽진** (아주대학교)
	블록체인 보안 프레임워크 개발을 위한 블록체인 보안 이슈에 대한 연구 **노시완**, **이경현** (부경대학교)
	블록체인을 활용한 보안관제 적용 방안 연구 **조영민** (고려대학교)
	블록체인 기반 교내 전자투표 시스템 구축 요구사항 분석 **정채린**, **이종호**, **김영우**, **조은아**, **성기정**, **김희영**, **이경현** (부경대학교)
좌장: **장항배** (중앙대학교)	

A3	발표 논문
자동차보안 13:30 – 14:30	자율주행 차량에서 보안 위협 분석을 통한 내/외부네트워크 안전성 향상 방안 연구 **최석준**, **곽진** (아주대학교)
	자동차 보안 모델 HEAVENS 구체화를 위한 Automotive Attack DB 연구 **이세한**, **고광우**, **안광현**, **박희성**, **한철규**, **심준보** (KITRI BoB), **이현정** (페스카로)
	인터페이스 중심의 최신 자동차 공격 사례 분류 **송윤근**, **이유식** (ESCRYPT)
	차량 대상 사이버 공격의 위험 평가 동향 연구 **선현진**, **이동훈** (고려대학교)
좌장: **임강빈** (순천향대학교)	

A4	발표 논문
클라우드보안 14:40 – 15:40	OpenStack Swift의 보안 취약점과 해결방안 이동혁, 박남제 (제주대학교)
	스마트TV 취약점 분석 및 대응방안 연구 장태진, 박동수, 우지섭, 윤정준, 황윤아 (KITRI BoB)
	클라우드 기반의 지능형 영상 감시 시스템을 위한 영상 프라이버시 보호 방법 이동혁, 박남제 (제주대학교)
	토렌트를 통해 유포되는 악성코드 분석 및 대응방안 김동현, 양준호, 임명은, 염흥열(순천향대학교)
좌장: **윤지원** (고려대학교)	

A5	발표 논문
영어논문 15:50 – 17:05	A Blockchain Based Audit Record Management Bruno Joachim Kweka, Youngho Park and Kyung-Hyune Rhee (Pukyong National University)
	Toward Machine Generated Passwords Simon Woo (SUNY Korea), Wenzhe Li (USC), Hyeran Jeon (San Jose State University)
	TVT: Typed Virtual Table for mitigating VTable Hijacking Jeongoh Kyea, Eunsoo Kim, Dongkwan Kim and Yongdae Kim (KAIST)
	블록체인 기반의 신뢰성 있는 응용 서비스 설계연구 (Reliable Application Service Design based on Block Chain Platform) 박리원, 장항배 (중앙대학교)
좌장: **윤지원** (고려대학교)	



구두 발표 트랙 B

미래융합기술관 110호

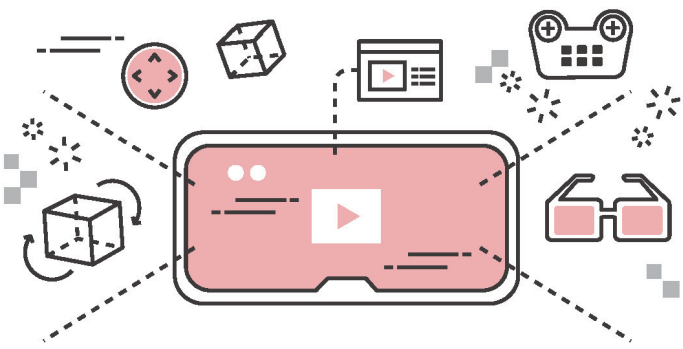
B1	발표 논문
시스템보안 (Ⅱ) 09:00 – 10:00	정적 특성 기반의 군집화를 이용한 변종 악성코드 식별 방안 김광훈, 이동근, 사경진, 조형택, 염흥열 (순천향대학교)
	이중 지불 공격 방지를 위한 트랜잭션 풀 우선순위 선정 방안 연구 박지선, 김동이, 박시현, 신상욱. 이경현 (부경대학교)
	소프트웨어 잡음원의 취약 패턴 검출 연구와 그 활용방안 박호중, 김원태, 염용진, 강주성 (국민대학교)
	동적 정보 기반 악성코드 그룹 분류 및 특징 연구 박래현, 유정빈 (연세대학교), 박준형 (한국인터넷진흥원), 권태경 (연세대학교)
좌장: **김윤정** (서울여자대학교)	

B2	발표 논문
정보보호관련법 및 제도 10:10 – 11:10	SI 프로젝트에서 아키텍트 보안활동 수행 및 효과 남혜리 (고려대학교)
	DLMS/COSEM 기반의 스마트미터링 플랫폼 수립에 대한 국제 표준화 동향 및 국내 적용 방안 강동주, 양인석 (고려대학교)
	중 · 소형 공공기관 정보보안 관리방법 체계화 연구 강창식, 권헌영 (고려대학교)
	NERC CIP 규정 분석에 근거한 전력시스템 신뢰도 관점에서 CPS 영향도 평가 방법에 관한 연구 강동주 (고려대학교)
좌장: **김기범** (경찰대학교)	

B3	발표 논문
양자암호 13:30 – 14:30	텔레그램의 IGE 모드에 대한 양자 안전성 검증 김성숙, 이지은, 김광조 (KAIST)
	Yu와 Oh가 제안한 OAuth기반의 사용자 인증 프레임워크 정형화 검증 이재경, 최홍석, 한종찬, 권오제, 유일선 (순천향대학교)
	Quantum Message Authentication Scheme using Single Qubit Measurement Yeon-Ho Choi (Korea University, KIST), Min-Sung Kang, Sang Wook Han, Sung Moon (KIST)
	공증된 conjugate coding 기반 양자 서명 강민성, 박병권 (KIST), 우민기 (아주대학교), 이민수, 김용수, 조영욱, 이상윤, 한상욱, 문성욱 (KIST)
좌장: **정익래** (고려대학교)	

B4	발표 논문
블록체인 (Ⅱ) 14:40 – 15:40	가상화폐 익명화 기술 연구 동향 **이지환, 홍영기, 허준범** (고려대학교)
	블록체인 기반 부동산 원장에 대한 법적 검토 **김미연, 황동현, 염흥열** (순천향대학교)
	비트코인 익명성 동향 분석 **김승민, 이지혜, 김윤정** (서울여자대학교)
	블록체인 기반 에너지 거래 모델 연구 **김승리, 이정혁, 구슬예, 오현옥** (한양대학교), **김지혜** (국민대학교)
좌장: **신상욱** (부경대학교)	

B5	발표 논문
고교생논문 15:50 – 17:05	챗봇을 활용한 계정탈취의 위험성 **정희태, 박민혁** (목포대학교 정보보호영재교육원), **김민수** (목포대학교)
	RFID Skimming 해킹 기법에 관한 고찰 **김건우, 김태현, 성유원, 여상준, 이진우, 이창엽, 이달원** (대구대학교 정보보호영재교육원)
	미공개 된 타원곡선 암호시스템에서의 파라미터 탐지 방법 **조현주, 최태균** (하나고등학교)
	Mac OS X의 디지털 증거 수집 방식 결정 방안 제시 **이준수** (경기과학고등학교), **신희재** (세종대학교), **조재민** (서울과학기술대학교), **강성철** (서울과학기술대학교), **박진우** (경기대학교)
좌장: **박영호** (세종사이버대학교)	



구두 발표 트랙 C

하나스퀘어 B119호

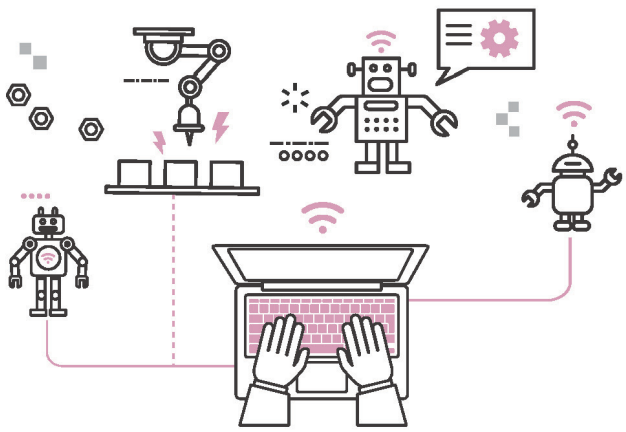
C1	발표 논문
정보보호모델 09:00 – 10:00	Network Address Mutation을 위한 시스템 아키텍처 설계 **안성범**(과학기술연합대학원대학교), **우사무엘**, **박경민**, **문대성** (한국전자통신연구원)
	차량 내부통신 보안 안전성 평가기술 기반 연구 **이선우**, **이동훈** (고려대학교)
	기업 서버의 침해 사고 대응을 위한 침해사고 준비도 모델 (Incident Response Readiness Model for Business Servers) **신현우**, **박성미**, **유운선**, **황지수** **김동현**, **한창희** (KITRI BoB), **김종현** (더존 포렌식 센터)
	보안성이 강화된 SaaS 개발 생명주기 연구 **정윤송**, **이동훈** (고려대학교)
좌장: **권헌영** (고려대학교)	

C2	발표 논문
IoT보안 (Ⅰ) 10:10 – 11:10	사물 인터넷 환경의 통신 보안에 관한 연구 **정진혁** (고려대학교)
	랜섬웨어의 동향과 사물인터넷(IoT) 보안대책 **김동휘** (고려대학교)
	IoT 악성코드 사례 분석 및 미래 위협 제시 **이지오**, **이세빈**, **염흥열** (순천향대학교)
	도메인 기반 블록체인을 이용한 IoT 접근제어 연구 **김미선**, **박경우**, **서재현** (목포대학교)
좌장: **오성준** (고려대학교)	

C3	발표 논문
IoT보안 (Ⅱ) 13:30 – 14:30	Memory Leakage Attack과 태그 가장 공격에 안전한 PUF 기반 RFID 인증 프로토콜 **권태연** (고려대학교), **변진욱** (평택대학교), **정익래** (고려대학교)
	NB-IoT 적용사례 분석을 통한 안전성 향상 방안 연구 **우시재**, **곽진** (아주대학교)
	무선 LAN 취약점에 따른 IoT 보안 **김지원**, **최효경**, **김지수**, **이세은**, **최나래**, **최은정** (서울여자대학교)
	IoT 보안 최신 동향 분석 **김지원**, **최효경**, **안혜준**, **최은정** (서울여자대학교)
좌장: **이병천** (중부대학교)	

C4	발표 논문
네트워크보안 (Ⅰ) 14:40 – 15:40	초음파를 이용한 상용 드론 고도 측정 시스템 스푸핑 공격 신정민, 노주환, 신호철, 김용대 (KAIST)
	WebOS 환경에서의 악성 애플리케이션 자동화 분석 박재유, 차상길 (KAIST)
	다크웹과 표면웹간의 연관성 분석 김다솜, 임성희, 임수진, 조은희 (KITRI BoB), 니콜라이 아카티예프 (Horangi Cyber Securi), 유현 (경찰수사연수원)
	최신 TLS 암호화 프로토콜 웹 서버 적용 의무화 방안 연구 김성진 (고려대학교)
좌장: **윤명근** (국민대학교)	

C5	발표 논문
네트워크보안 (Ⅱ) 15:50 – 17:05	SDN 베어메탈 스위치를 위한 네트워크 운영체제의 설치 취약점 연구 유봉열, 양경식, 석상윤, 유혁 (고려대학교)
	VANET 환경에서 해시기반 서명 연구 이경호, 이경현 (부경대학교)
	AEG(Automatic Exploit Generation)에 대한 연구 동향 이재용, 석재혁, 이동훈 (고려대학교)
	커넥티드카(Connected car)통신을 위한 국제 표준화 동향 연구 : 미국, 유럽 표준을 중심으로 주경호, 이동훈 (고려대학교)
좌장: **서정택** (순천향대학교)	



구두 발표 트랙 D

하나스퀘어 B120호

D1	발표 논문
해킹 및 취약점 (Ⅰ) 09:00 – 10:00	DTLS 기반의 CoAP 보안 메커니즘 분석 및 성능평가 **한상우, 조정모, 박창섭** (단국대학교)
	난독화 방법에 따른 Clang 정적 분석 도구의 성능 영향 분석 **진홍주, 박문찬, 이동훈** (고려대학교)
	프린터 관리 시스템의 취약점 분석 **지우중, 이경문, 이병천** (중부대학교)
	다크 웹 크롤링을 위한 지능형 크롤러 설계 **노인우, 권구환, 임을규** (한양대학교)
좌장: **김경곤** (고려대학교)	

D2	발표 논문
해킹 및 취약점 (Ⅱ) 10:10 – 11:10	다중 바이트 간 독립성 검정을 이용한 소프트웨어 잡음원의 엔트로피 추정방법 **박호중, 김원태, 염용진, 강주성** (국민대학교)
	기존 NTRU 전력 분석 방법 실제 적용 **안수정, 김수리. 김희석, 홍석희** (고려대학교)
	안드로이드 앱의 호출 흐름 무결성 검증 기법 **최준태, 조홍래, 하동수, 오희국** (한양대학교)
	기계학습을 이용한 행위 기반의 악성코드 판별 모델 **박의성** (목포대학교), **진수현** (동의대학교), **지한별, 이주연, 국광호** (서울과학기술대학교)
좌장: **강병탁** (에이아이스페라)	

D3	발표 논문
해킹 및 취약점 (Ⅲ) 13:30 – 14:30	심볼 기반 악성코드 분류 모델 **이은지**(아주대학교), **이태진** (호서대학교), **곽진** (아주대학교)
	침해 전파과정에서 악성행위 탐지 및 분석 **정의엽** (고려대학교), **문의성** (광운대학교), **이용민** (경찰대학교), **장진호** (호서대학교)
	NTRU 서명 시스템에 대한 오류 주입 공격 동향 및 구현에 관한 연구 **장호철**(호서대학교), **김기현** (앤앤에스피), **하재철** (호서대학교)
	원격제어 소프트웨어의 위협 도출 및 공격 시나리오 연구 **양산휘, 이아름** (세종대학교), **박인혁, 조재일** (순천향대학교), **이승명** (대구카톨릭대학교), **이영주** (선린인터넷고등학교), **신정훈** (한국정보기술연구원), **김경곤** (고려대학교)
좌장: **이경문** (중부대학교)	

D4	발표 논문
디지털 포렌식 14:40 – 15:40	사이버 범죄 예방을 위한 사이버 CPTED 모델 주민희, 박무규, 이경호 (고려대학교)
	Volume Shadow Copy의 포렌식적 활용 박은후, 이세영, 이종혁, 김청우, 이상범, 최형석, 김영효, 최건희, 김한기, 김종성 (국민대학교)
	웹 캐시의 위치 기반 서비스 정보를 활용한 위치추적 방안 연구 김채린 (고려대학교), 송인석 (세종대학교)
	IoT 디바이스의 네트워크 정보 이력관리를 통한 사고분석 활용 방안 장대일, 김태은, 김환국 (한국인터넷진흥원)
좌장: **김종성** (국민대학교)	

D5	발표 논문
인증 및 키 관리 15:50 – 17:05	멀티캐스트 CoAP 보안을 위한 키 설정 기법 조정모, 한상우, 박창섭 (단국대학교)
	OQS 프로젝트 중 격자 기반 키 교환 방식의 타이밍 등 공격 분석 안형철, 한성호, 최낙준, 김광조 (KAIST)
	IoT 환경을 위한 효율적인 그룹 인증 및 키 교환 기법 이대휘, 이임영 (순천향대학교)
	최신 PKI기반 V2X보안 기술 동향분석 이정호 (한국정보인증)
좌장: **Simon Woo** (SUNY Korea)	



구두 발표 트랙 E

과학도서관 209호

E1	발표 논문
모바일 보안 (Ⅰ) 09:00 – 10:00	공개된 채널환경에서의 NTRU 기반의 RFID 상호인증에 관한 연구 **황용운, 이임영** (순천향대학교)
	자마린에서 C#으로 개발된 안드로이드 앱에 대한 정적 역공학 분석 **김규식, 심재우, 임경환, 조성제** (단국대학교)
	Fuzzy Hash와 Dex 파일 분석 로그를 이용한 안드로이드 악성 애플리케이션 유사도 비교 **조하은, 천호진, 정수환** (숭실대학교)
	EU GDPR 사례를 통한 바이오정보보호 개선 방안 **정부금, 권헌영, 임종인** (고려대학교)
좌장: **김형종** (서울여자대학교)	

E2	발표 논문
모바일 보안 (Ⅱ) 10:10 – 11:10	NFC 스마트카드 해킹을 통한 부당이득 취득 사례 및 공격 기법에 관한 연구 **이상욱** (전남대학교), **나동빈** (한국교원대학교), **박한울** (경기대학교), **이태일** (한국교통대학교), **강종구** (유원대학교)
	엔트로피를 기반으로 한 전자거래 바이오인증 강도 산정 방법 제시 및 비교 분석 **김지혜, 사경진, 염흥열** (순천향대학교)
	악성코드 분석을 위한 안드로이드 DEX 바이너리 기반 Java API 추출 방법 **안진웅, 박정수, 정수환** (숭실대학교)
	지속 인증을 위한 사용자 행위 정보 기반특징 요소 별 성능 분석 연구 **구예은, 이호연, 권태경** (연세대학교)
좌장: **노건태** (서울사이버대학교)	

E3	발표 논문
모바일 보안 (Ⅲ) 13:30 – 14:30	안드로이드 보안 솔루션 검증 결과 변조를 통한 기능 우회 취약점의 대응방안 연구 **김규헌, 한미정, 유재욱, 장준영, 진호용, 지한별** (KITRI BoB), **김경곤** (고려대학교), **신정훈** (세종대학교)
	안드로이드 어플리케이션 자동 분석 보고서 제작 프로그램 설계 및 구현 **천호진, 강성은, 정수환** (숭실대학교)
	소프트웨어 권한 분리에 대한 연구 **방인영, 이영한, 안선우, 백세현, 백윤흥** (서울대학교)
	온라인 게임 보안을 위한 매크로의 우회 기법과 대응방안 연구 **박영서, 정담지, 김지은, 김나경, 김연주, 전정훈** (동덕여자대학교)
좌장: **이문규** (인하대학교)	

E4	발표 논문
암호 이론 (Ⅰ) 14:40 – 15:40	S-box기반 암호의 최적 차분경로 자동 탐색 방법(MILP 적용) 박연지, 이호창(고려대학교), 홍득조 (전북대학교), 홍석희 (고려대학교)
	부호 기반 암호 연구동향 조사 박병규, 김수리, 김희석, 홍석희 (고려대학교)
	암호데이터 중복제거 환경에서 데이터 블록간의 연계를 이용한 소유권 관리 기술 김원빈, 이임영 (순천향대학교)
	SPN 구조를 갖는 Block Cipher에서 Key Schedule이 차분 성질에 미치는 영향 김현우, 김성겸, 강형철 (고려대학교), 홍득조(전북대학교), 홍석희 (고려대학교)
좌장: **염용진** (국민대학교)	

E5	발표 논문
암호이론 (Ⅱ) 15:50 – 17:05	내적연산을 지원하는 함수암호 연구동향 김훈기, 이영경, 이동훈 (고려대학교)
	RTOS가 동작하는 Cortex-M4보드의 암호알고리즘 소비 전력 측정 송행권, 이옥연 (국민대학교)
	표준 모델에서 증명 가능한 LWE 기반 BGN-타입 ID기반 동형 암호 우주, 김종현, 이동훈 (고려대학교)
	동적 암호 기반 Card OTP 모듈을 사용한 복제방지 RFID Card 제안 권성민, 박기현, 김재성, 원요한, 이동현, 이예지, 지한별, 박의성 (KITRI BoB), 이종호 (고려대학교)
좌장: **안효범** (공주대학교)	



포스터발표 트랙 1

미래융합기술관 106호

세션	번호	발표 논문
09:00 – 10:00	1	고출력 전자기파 위협에 대한 원전의 통신시스템 보호 방안 연구 송동훈 (한국원자력통제기술원)
	2	소프트웨어 취약점 분석 자동화 도구에 관한 연구 나건배, 전지수, 김태은, 김환국 (한국인터넷진흥원)
	3	Monitor Mode를 이용한 모바일 디바이스사용자 추적 방안에 대한 연구 김건용, 김형철, 이원용, 유명성, 박강민, 정재완, 오동현, 심기욱, 이경문 (KITRI BoB)
	4	원자력시설 사이버보안 강화를 위한 인적 보안과 시스템 및 정보의 무결성에 대한 평가 방법론 개발 송동훈 (한국원자력통제기술원)
	5	제어시스템 테스트베드를 활용한 기술적 보안조치 적용 시 고려사항 및 대처방안 연구 임현종, 송동훈, 신익현 (한국원자력통제기술원)
	6	PPTP 인증 프로토콜 취약점 동향 분석 손민우, 이상익, 배윤겸, 김한기, 김지훈, 김종성 (국민대학교)
	7	개인 건강정보 보안을 위한 소프트웨어 설계방안 강기원 (고려대학교)
10:10 – 11:10	8	안티바이러스 취약점 분석 방법 및 주요 보안 문제법 전형린, 김도연, 김도혁, 안정언, 오우진, 이상진, 이승진, 한승훈 (KITRI BoB)
	9	기계 학습을 통한 악성코드 연관성 분석 방법론 채진희(순천향대학교), 오동빈, 정해원 (경찰대학교), 김예준 (순천향대학교), 이현 (선린인터넷고등학교), 유정인 (가톨릭대학교), 장원희 (서울호서전문학교)
	10	효율적인 OS Fingerprint Signature에 관한 연구 송진호, 김용건, 원유재 (충남대학교)
	11	IoT System 개발을 위한 Secure Agile Process:MS-SDL과 스크럼 프로세스를 이용한 통합 방법론 오민석 (고려대학교)
	12	휴리스틱 백신의 탐지기법 및 전망 장서연, 김도연, 이제원, 이윤진, 이승연, 김지훈, 김종성 (국민대학교)
	13	프라이버시를 보호하는 IEC 62304 의료기기 소프트웨어 개발 생명주기 배경진, 이동훈 (고려대학교)
	14	보안 취약점 정보 공유를 위한 STIX 데이터 모델링 및 변환 기술 연구 고은혜, 김태은, 김환국 (한국인터넷진흥원)

포스터발표 트랙 2

미래융합기술관 106호

세션	번호	발표 논문
13:30 – 14:30	1	온라인 마켓플레이스를 위한 이상거래탐지시스템(FDS) 분석 및 향후 전망 장창훈 (고려대학교)
	2	Usable Security관점에서 본 포털사이트 로그인API의 취약점 문성훈, 이예진, 윤진희, 이재원 (대구대학교 정보보호영재교육원), 김호엽 (대구대학교)
	3	아파트(공동주택)에서의 개인정보보호 방안 연구 한충희, 이상준 (전남대학교), 박순태 (한국인터넷진흥원)
	4	라즈베리파이를 활용한 패킷 스니핑(Sniffing) 공격 구현 문대정, 고민우, 김인호 (목포대학교 정보보호영재교육원), 신인철 (목포대학교)
	5	금융 정보보호 관리체계(F-ISMS) 분석을 통한 기업별 정보보호 관리체계(ISMS) 연구의 필요성 주영국 (고려대학교)
	6	안드로이드 환경에서 랜섬웨어 탐지를 위한 랜섬웨어 분석 시스템 연구 박인규, 곽진 (아주대학교)
	7	정보보안성강화를 위한 공공기관 위협모델링(Threat modeling) 제안 최종현 (고려대학교)
14:40 – 15:40	8	랜섬웨어 유포방식과 유형분석 조형택, 이동근, 사경진, 김광훈, 염흥열 (순천향대학교)
	9	SPA(Single Page Application) 기반 웹 보안 취약점 연구 이복음 (고려대학교)
	10	Wi-Fi 신호 변동 기반의 IoT 디바이스 인증 기법 정혜라, 소재우 (서강대학교)
	11	금융회사의 전사적 보안협업을 위한 핵심성공요인 박지윤 (중앙대학교)
	12	개인정보보호를 위한 효율적인 암호키 교체방안 연구 이선주 (고려대학교)
	13	서버로그 상관분석을 통한 보안 침해사고 탐지 윤인재 (고려대학교)
	14	SW Security 표준의 CWE 영향도 분석 이헌기, 최진영 (고려대학교)
15:50 ~ 16:50	15	악성코드가 호출하는 API 시퀀스를 이용한 유사 악성코드 클러스터링 방안 연구 유대훈, 최보민, 강홍구, 박준형 (한국인터넷진흥원)
	16	사이버위협에 대한 소프트웨어 보안강화 방안 연구 이길호 (고려대학교)
	17	국가사이버안보법안 타당성에 관한 연구: 사이버위협정보 공유를 중심으로 정해원 (경찰대학교), 김예준, 채진희 (순천향대학교), 이현 (선린인터넷고등학교), 유정인 (가톨릭대학교), 장원희 (서울호서전문학교)
	18	IOT 기반의 블록체인패러다임과 보안 성종현, 송호령 (대구대학교 정보보호영재교육원), 김창훈 (대구대학교)
	19	초음파 공격을 고려한 인공지능 스피커 보안위협 및 대응방안 분석 최근영 (고려대학교)
	20	기계학습을 이용한 네트워크 전장정보 수집 한규석, 윤지원 (고려대학교)
	21	IoT 환경에서 전자정부 시스템 활용을 위한 행정안전부 C 시큐어코딩 가이드 보완점 연구: SEI CERT C Coding Standard 2016 적용 백승태 (고려대학교)

IITP 국제공동연구 / R&D 데이터 CHALLENGE 트랙

미래융합기술관 601호

세 션	IITP 정보보호 분야 한-미공군 과학연구실 공동연구 발표
IITP 9:00 – 11:10	IoT 환경에서 제로데이 공격에 강건한 규칙 명세 기반의 악의적인 행위 탐지 연구 **유일선** (순천향대학교)
	양자 랜덤 오라클 모델 (QaROM) 및 래티스 기반 인증키교환 프로토콜 **최락용** (KAIST)
	Wigner rotation of photons in a gravitational field. **안도열** (서울시립대)
	PUF 응용을 위한 랜덤 레이저 제작 **송영민** (GIST)
	양자정보를 위한 2차원물질 단일광자원 구현 **박홍규** (고려대)
좌장: **하선우** (IITP)	

세 션	정보보호 R&D 데이터 챌린지 수상자 우수 결과물 발표
CHALLENGE (Ⅰ) 13:30 – 14:10	차량 이상 징후 탐지
CHALLENGE (Ⅱ) 14:20 – 15:00	모바일 악성앱 탐지
CHALLENGE (Ⅲ) 15:10 – 15:50	악성코드 탐지
CHALLENGE (Ⅳ) 16:00 – 16:40	악성코드 선재대응

2017 한국정보보호학회 동계학술대회

A1 시스템 보안(I)

▌강 동 주 (고려대학교)▐

- **멀티 프로세서 환경에서 가상머신간 은닉채널 생성에 관한 연구**
  배원일(아주대학교), 황우민, 김형천 (국가보안기술연구소),
  곽 진 (아주대학교)
- **쿠쿠 샌드박스의 정보를 이용한 랜섬웨어 특징정보 추출**
  옥정윤, 김준형, 임을규 (한양대학교)
- **Practical Plug & Play Measurement Device Independent Quantum Key Distribution System**
  Changhoon Park, Byung Kwon Park, Min Soo Lee, Young-Wook Cho, Yong-Su Kim, Sang-Wook Han and Sung Moon (KIST), Min Ki Woo, Sangin Kim (아주대학교)
- **Ponce를 활용한 기호실행 기반 악성프로세스를 발현시키는 방법 연구**
  한중수, 임을규 (한양대학교)

멀티 프로세서 환경에서 가상머신간 은닉채널 생성에 관한 연구

배원일*, 황우민**, 김형천**, 곽진***

*아주대학교 컴퓨터공학과 정보보호응용및보증연구실

**국가보안기술연구소

***아주대학교 사이버보안학과

Implementation of Covert Channel on Cross-VM in Multi-Processor Environment

Won-Il Bae*, Woomin Hwang**, Hyoung-Chun Kim**, Jin Kwak***

*ISAA Lab., Department of Computer Engineering, Ajou University.

**National Security Research Institute (NSR).

***Department of Cyber Security, Ajou University.

요 약

멀티 프로세서 환경은 둘 이상의 프로세서를 이용하여 하드웨어적 관점에서 병렬 처리를 가능하게 한다. 경제적인 비용 절감의 효과로 인해 클라우드 및 모바일 등 다양한 분야에서 활용되고 있다. 최근 AMD 멀티 프로세서 환경에서 캐시 일관성 프로토콜을 이용하여 악용 가능한 은닉채널 생성에 대한 연구가 진행되었다. 하지만 클라우드 환경에서는 메모리 및 캐시와 같은 공유 자원을 이용하여 은닉채널을 생성하고 데이터 전송이 가능하지만 다양한 격리 기술을 제공하기 때문에 은닉채널 생성에 대한 어려움이 발생한다. 본 논문에서는 Intel 멀티 프로세서 환경에서 캐시 일관성 프로토콜 기반의 가상머신간 은닉채널 생성의 가능성을 증명하고자 한다.

I. 서론

멀티 프로세서 시스템은 다수의 프로세서를 통해 하드웨어적으로 프로세스 병렬처리를 가능하게 함으로써 작업의 효율을 높인다[1].

하지만, 하드웨어 측면에서 마이크로 아키텍처 구성 요소인 캐시, 메모리 버스와 같은 하드웨어 자원을 통해 비밀 정보가 누출될 수 있다[2].

기존에 알려진 마이크로 아키텍처 기반의 은닉채널로는 L1 캐시, LLC(Last Level Cache), 메모리 버스 등이 있다. 이렇게 알려진 은닉채널을 이용하여 부채널 공격에 대한 연구 또한 활발히 진행되고 있다. 그 예로, Fangfei Liu 등[2]은 LLC를 통해 은닉채널 및 부채널 공격이 가능한 Prime+Probe 공격을 제안하였다. 또한, Yuval Yarom 등[3]도 LLC를 이용한 Flush+Reload 공격을 제안함과 더불어 GnuPG 라이브러리를 사용해 RSA 후보키 탈취 실험을 진행함으로써 부채널 공격을 실증하였다.

이와 관련된 연구로 Gorka Irazoqui 등[1]은 AMD 멀티 프로세서 환경을 타겟으로 은닉채널 생성 및 부채널 공격이 가능한 Invalidate+transfer 공격을 제안하였다. Invalidate+transfer 공격은 프로세서 간 캐시 일관성 유지를 위한 캐시 일관성 프로토콜을 은닉채널로 활용한다. 이는 각 프로세서가 캐시를 독점하는 환경에서도 은닉채널 생성 및 부채널 공격이 가능함을 보였다. 이에 따라, 본 논문은 Gorka Irazoqui 등[1]이 제안한 Invalidate+transfer 공격을 기반으로 Intel 멀티 프로세서 환경에서 가상머신간 은닉채널 생성 가능성을 증명하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서 기존 멀티 프로세서 환경에서의 은닉채널 생성

방안과 캐시 일관성 및 Intel의 QuickPath Interconnect 기술에 대해 분석한다. 3장에서는 제안하는 방안을 설명하고 구현 및 실험결과에 대한 분석을 진행하며, 4장에서 결론을 맺는다.

II. 관련 연구

2.1 멀티 프로세서 환경 은닉채널

Gorka Irazoqui 등[1]은 AMD 멀티 프로세서 환경에서 은닉채널을 활용한 공격을 수행하였다. AMD 프로세서에서 캐시 일관성을 가능하게 하는 Hyper-Transport를 이용하여 DRAM을 거치지 않고 CPU1의 캐시로부터 작업과 비교적 느린 DRAM 접근을 통해 작업이 수행되는 것의 두 작업에 소요되는 시간을 측정 및 구별하면 악용 가능한 은닉채널이 될 수 있다[1]. 멀티 프로세서 환경에서 은닉채널 생성을 위한 과정은 3단계로 이루어져 있으며, 다음과 같다.

Step 1. 캐시 무효화 단계

캐시 무효화 단계는 공격자가 자신의 캐시 계층에 있는 메모리 블록을 무효화한다. 공격자의 캐시 계층이 무효화될 경우 캐시 일관성 프로토콜을 통해 다른 프로세서의 캐시 계층도 무효화된다.

Step 2. 대기 단계

대기 단계는 무효화 단계 후에 희생자가 계산을 수행하도록 일정 기간 동안 대기하는 단계이다.

Step 3. 전송 단계

마지막 단계인 전송 단계는 공격자가 무효화된 공유 메모리 블록에 대해 접근 요청하는 것으로 시작된다.

해당 시스템의 프로세서가 무효화된 메모리 블록을 캐시할 경우 디렉토리에 대한 항목이 업데이트 되어 프로브 요청이 다른 프로세서로 전송된다. 하지만 해당 메모리 블록을 사용하지 않을 경우 DRAM에 대한 접근을 요청한다.

2.2 캐시 일관성

캐시 일관성은 공유 메모리 환경에서 각 클라이언트 및 프로세서의 캐시간의 일관성을 나타낸다. 캐시 일관성 구조는 디렉토리 기반과 스누핑의 2가지로 분류할 수 있다. 디렉토리 기반 캐리 일관성 구조는 캐시의 공유 상태 등을 저장하는 디렉토리를 이용해서 관리하는 구조이며, 스누핑 캐시 일관성 구조는 버스를 주기적으로 모니터링하여 캐시에 대한 메모리 접근을 감시하는 구조이다[1].

2.3 Intel QuickPath Interconnect

Intel은 캐시 일관성을 유지하기 위해 AMD의 HyperTransport와 비슷한 기술인 Intel QPI(QuickPath Interconnect)을 사용하고 있다.

QPI는 효율적인 점 대 점 상호 연결이 가능하고 고속 링크를 통해 공유 메모리 블록을 전송함으로써 다중 프로세서 환경에서 캐시 일관성을 효율적으로 만족시키는 것이 가능하다[5].

III. 제안사항

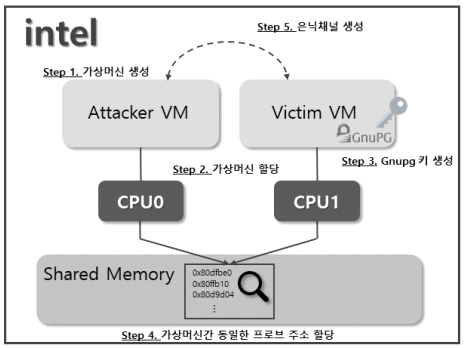
본 장에서는 멀티 프로세서 환경에서 가상머신간 은닉채널 생성 과정과 실험 결과에 대한 내용을 설명하며 실험 환경은 [표 1]과 같다.

[표 1] HOST server 실험 환경

구분	사양
CPU	Intel Xeon E5-2650 v4 × 2
OS	Ubuntu 16.04
RAM	16G

3.1 가상머신간 은닉채널 생성 구현

Intel 멀티 프로세서 환경에서 생성된 가상머신간 은닉채널 생성의 전체 시나리오는 [그림 1]과 같다.



[그림 1] 가상머신간 은닉채널 생성 시나리오

Step 1. 2개의 가상머신 생성

공격자 및 희생자 가상머신 생성을 위해 Virtual Machine Manager를 사용하며, ISO 이미지와 메모리 및 CPU를 [표 2]와 같이 설정한다.

[표 2] 2개의 가상머신 사양

구분	공격자 가상머신	희생자 가상머신
OS	Ubuntu 16.04	Ubuntu 16.04
CPU 코어	24개	24개
메모리	1024MB	1024MB

Step 2. 가상머신 CPU 할당

본 실험환경에서 Intel CPU 코어는 48개이므로, 0~23번째 코어인 CPU0을 공격자 가상머신에 할당하고 24~47번째 코어인 CPU1을 희생자 가상머신에 할당하였다.

Step 3. GPG 키 생성

본 논문에서 가상머신간 은닉채널 생성 가능성을 실험하기 위해 GnuPG-1.4.13 도구를 사용하며, --gen-key 명령어를 이용하여 암호화 종류 선택과 공개키 및 개인키를 생성할 수 있다.

Step 4. 가상머신간 동일한 프로브 주소로 할당

[그림 2]와 같이 희생자 가상머신에서 컴파일된 GPG 파일을 objdump 도구를 이용하여 각 연산에 대한 주소를 프로브 주소로 할당한다. 이때, S는 제곱 연산을 의미하고 R은 모듈러 연산, M은 곱셈 연산을 의미한다.

```
map /home/attacker/wi/gnupg-1.4.13/g10/gpg
base 0x8048000
probe 0x80fdd01 S #mpih-mul.c:270 (First cache line in mpih_sqr_n())
probe 0x80fd185 R #mpih-div.c:329 (Loop in default case in mpihelp_divrem())
probe 0x80fdcc7 M #mpih-mul.c:121 (First cache line of mul_n())
slot 2048
```

[그림 2] 프로브 주소 값

Step 5. 모니터링을 이용한 은닉채널 생성 확인

공격자 가상머신에서 스파이 프로세스를 이용하여 지수 연산, 모듈러 연산 및 곱셈 연산의 프로브 주소가 포함되어 있는 gpg.probe를 실행시켜 [그림 3]과 같이 모니터링을 수행한다.

이후, 희생자 가상머신에서 복호화 연산을 수행하면 지수 연산, 모듈러 연산 및 곱셈 연산에 해당하는 S, R, M이 출력된다.

```
Mapped at 0xb73f7000 (0x12dff8)
SRSSSSSSSRSSSSSSSSSSSSSSSSSSSRSSSSSSSSSSSSSSSSSSSRRSSSSSSSSSSSSSSSSSSSRSSSSSS
SSSSSSSSSSSSSSSSSRSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSRRSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSRSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSRSRSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSRSRS
```

[그림 3] 공격자 가상머신에서 스파이 프로세스 모니터링 수행 결과

3.2 은닉채널 생성 검증

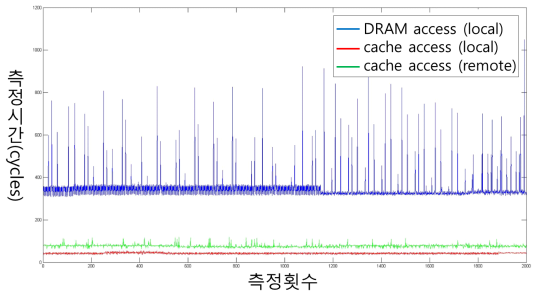
❑ 시간 측정을 통한 검증

캐시 적중 여부는 local 또는 remote 캐시 적중/미적중의 경우로 나뉠 수 있다. local 캐시 적중 여부는 하나의 가상머신에서 독립적으로 일어나는 경우이다. 반면, remote 캐시 적중 여부는 두 개의 가상머신이 통신 할 때 하나의 가상머신이 공유메모리에 접근할 경우 다른 가상머신에서 해당 접근에 따른 캐시 적중/미적중을 모니터링하는 경우이다.

이에 따라, 공격자 가상머신에서 독립적으로 local 캐시 적중/미적중되는 시간 분포를 수집한다. 또한, 희생자 가상머신에서 공유메모리로의 접근을 공격자가 모니터링하여, 이 때의 remote 캐시 적중 또는 미적중되는 시간 분포를 수집한다. 결과적으로 local 캐시 적중 또는 미적중되는 시간 분포가 remote 캐시 적중 또는 미적

중되는 각각의 시간분포와 차이가 있음을 분석함으로써 은닉채널을 통해 공격자 및 희생자가 통신함을 검증할 수 있다.

이에 따른 시간분포 차이의 결과는 [그림 4]와 같다. 실험 결과 캐시 미적중인 경우 캐시에 의한 접근보다 느리게 측정되었으며, local 캐시 적중 시간이 remote 캐시 적중 시간보다 상대적으로 빠른 것을 확인할 수 있다.



[그림 4] 시간 분포 측정 결과

❑ 비트 전송을 통한 검증

위의 시간 측정을 통한 검증 방법에서 cache의 적중/미적중 여부를 확인하였으며 remote cache의 접근을 임의로 조작할 수 있으므로, 이를 통해 [그림 5]와 같이 비트 전송을 수행할 수 있다. 측정 시간이 짧으면 cache에 적중된 상태이므로 1을 출력하고 측정 시간이 비교적 느리면 미적중 상태이므로 0을 출력한다. 이를 반복하여 비트 전송을 수행한다.



[그림 5] 비트 전송 검증

3.3 활용 방안

본 실험 결과의 모니터링 결과를 수집하여 연산의 패턴 정보를 획득할 수 있다. 지수 연산이 + 모듈러 연산의 패턴 정보를 획득할 경우, 0비트로 변환할 수 있으며 지수 연산 + 모듈러 연산 + 곱셈 연산 + 모듈러 연산의 패턴이 획득될 경우 1비트로 변환할 수 있다.

따라서 변환된 비트 정보를 통해 비밀키 후보군을 추출할 수 있다.

IV. 결론

본 논문에서는 네트워크 연결이 해제된 멀티 프로세서 환경에서 가상머신간 모니터링을 통해 프로브 주소에 대한 캐시 적중 상태를 보여주고 local 상태와 remote 상태의 시간 측정 및 비트 전송을 통해 가상머신간 은닉채널 생성을 증명하였다.

향후에는 파라미터 튜닝 과정을 통해 모니터링 과정 시, 출력 값에 대한 보정 작업과 생성된 은닉채널을 이용하여 AES 및 Elgamal 후보키 추출 공격을 수행할 예정이다.

[참고문헌]

[1] Gorka Irazoqui, Thomas Eisenbarth, Berk Sunar, “Cross Processor Cache Attacks”, ASIA CCS '16 Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, Pages 353-364, May - June, 2016.

[2] Fangfei Liu, Yuval Yarom, Qian Ge, Gernot Heiser, Ruby B. Lee, “Last-Level Cache Side-Channel Attacks are Practical”, 2015 IEEE Symposium on Security and Privacy, 2015.

[3] Yuval Yarom, Katrina Falkner, “FLUSH+RELOAD: a High Resolution, Low Noise, L3 Cache Side-Channel Attack”, 23rd USENIX Security Symposium, August, 2014.

[4] Intel QuickPath Architecture, http://www.intel.com/pressroom/archive/reference/whitepaper QuickPath.pdf.

[5] Github, “defuse/flush-reload-attacks”, https://github.com/defuse/flush-reload-attacks. October, 2016.

쿠쿠 샌드박스의 정보를 이용한 랜섬웨어 특징정보 추출

옥정윤*, 김준형**, 임을규***

*한양대학교 정보보안학과

**한양대학교 컴퓨터소프트웨어학과

***한양대학교 컴퓨터소프트웨어학부

Ransomware Feature Selection Using Cuckoo Sandbox's Information

Jeong-Yun Oak*, Junhyoung Kim**, Eul Gyu Im**

*Department of Information Security, Hanyang University

**Department of Computer and Software, Hanyang University

요 약

1989년 AID Trojan의 시초로 현재 랜섬웨어로 불리는 악성코드는 2005년에 유럽 및 북미에서 나타났다. 초기 랜섬웨어는 복구하기 쉬웠으나 시간이 갈수록 랜섬웨어가 진화하였다. 2009년 비트코인인 가상화폐의 등장과 2012년부터 비트코인 가격이 급등 하면서 점차 랜섬웨어는 현대인에게 가장 위험한 악성코드로 진화하고 있다. 이에 따라 랜섬웨어의 분석이 필요하다. 본 논문에서는 다양한 랜섬웨어(WannaCry, Cerber, Locky, Cryptxxx 등)를 분석하고 랜섬웨어가 아닌 악성코드(backdoor, virus, warm 등)와 일반 파일을 비교했다. 그 후 상관관계를 활용하여 랜섬웨어의 특징정보를 추출한다. 그 후 기계 학습 방법 중 한 방법인 랜덤 포레스트 알고리즘을 이용하여 실험을 진행했다.

I. 서론

현대 사회에서 컴퓨터의 활용도가 증가함에 따라 보안의 위험성 등 다양한 부정적인 측면도 부각되고 있다. 최근에는 공격자가 사용자의 파일들을 암호화하고 사용자에게 금품을 요구하는 '랜섬웨어(Ransomware)'라는 새로운 악성코드가 나타났다.

국내에서는 2015년부터 랜섬웨어의 피해가 발생했다. 한국 랜섬웨어 침해대응 센터의 2015년 말, 2017년 초 침해 분석 보고서에 의하면 2015년 랜섬웨어에 감염되어 1090억 원의 손해를 입었고 2016년에는 3,000억 정도의 손해를 입었다. 안랩의 2015년 보고서에 따르면 공격자가 랜섬웨어를 공격자의 이익에 도움이 된다고 인식하면 공격자는 더 진화된 랜섬웨어를 제작하여 유포하게 되고, 이는 결국 새로운 피해자를 양산하게 된다. 이러한 악순환을 끊기 위해 랜섬웨어를 탐지하여 랜섬웨어로 입는 피해를 최소화하는 것이 중요하다고 명시했다. 따라서 이번 논문에서 윈도우 7 운영체제 환경에서 정적 분석과 동적 분석이 가능한 쿠쿠 샌드박스(Cuckoo Sandbox)를 이용하여 랜섬웨어를 분석한다.

본 논문은 총 5장으로 구성되며 각 장에서는 다음과 같은 내용을 다룬다. 제 1장 서론 에서는 연구 배경, 연구 목적, 연구 방식 및 범위에 대해서 작성한다. 제 2장에서는 관련 연구에 관하여 알아보고, 제 3장에서는 랜섬웨어 분석 모델을 제안한다. 제 4장에서는 3장에서 제안한 모델에 대해 실험을 진행하고, 제 5장에서는 결론 및 향후 연구에 대해서 작성한다.

II. 관련 연구

Y. Liao는[1] PE(Portable Executable)헤더 값을 비교하여 악성코드와 일반파일을 구분하는 방식을 제안하였다. 해당 연구에서 PE 헤더 값들을 분석한 결과 체크섬(checksum), 섹션 이름(section name), 초기화된 데이터 크기(initialized data size) 그리고 DLL(Dynamically Linked Libraries) 특성을 악성코드의 특징 정보로 선정하였다. 선정된 특징 정보로 악성코드와 일반파일을 분석한 결과 99.5%의 정확도와 0.16%의 거짓 양성(false-positive)을 얻었다.

E. Gandotra 외 2명은[2] 제로데이(Zero Day) 악성코드를 탐지하기 위해 정적, 동적 분석 방법으로 특징을 정의하고 기계 학습을 이용하여 악성코드를 탐지하는 방안을 제안하였다. 쿠쿠 샌드박스로 분석된 결과들을 정보 이득(Information Gain) 방법을 사용하여 악성 코드의 특징 정보들을 추출하였다. 그 후 다양한 기계 학습 알고리즘을 이용하여 분석한 결과 랜덤 포레스트(Random Forest) 기계학습 알고리즘이 99.97%의 정확도를 얻었다.

Qian Chen 외 1명은[3] 쿠쿠 샌드박스로 워너크라이 랜섬웨어(Wannacry Ransomware)를 분석하여 발생한 로그(host log)를 정보 연관방식에 가중치를 제공하는(term-weight) 방식인 TF-IDF(Term Frequency - Inverse Document Frequency)를 이용하여 워너크라이 랜섬웨어를 판별하는 방식을 제안하였다. 즉 분석한 워너크라이 랜섬웨어의 로그와 일반파일의 로그를 비교하여 특정 로그에 가중치를 부과하는 방식을 제안하였다.

D. Sgandarra 외 3명은[4] 랜섬웨어의 동적 분석 결과를 이용하여 특징 정보를 선정 후 다양한 기계학습 알고리즘을 통해 랜섬웨어를 분류하는 'EldeRan' 방식을 제안하였다. 쿠쿠 샌드박스로 동적 분석한 결과인 API(Application Programming Interface) 호출, 레지스트리(registry) 키 동작, 파일 시스템 동작, 파일 확장자 변경 등을 가지고 연구한 결과 API 호출과 레지스트리 키 동작이 랜섬웨어의 특징 정보임을 발견하였다. 랜섬웨어의 특징 정보들을 가지고 다양한 기계학습을 적용한 결과 정규화 된 로지스틱 회귀(Regularized Logistic Regression) 기계학습 알고리즘이 96.3%의 정확도와 1.6%의 거짓 양성으로 가장 좋은 알고리즘임을 판단하였다.

III. 랜섬웨어 분석 모델

본 논문에서는 랜섬웨어, 악성코드, 일반 파일을 쿠쿠 샌드박스로 분석한 정보들 중에 상관관계를 이용하여 랜섬웨어, 악성 코드, 일반 파일의 특징 정보를 추출한다. 그 후 기계학습 알고리즘의 한 종류인 랜덤 포레스트를 활용하여 실험을 진행한다.



그림 1. 분석 모델

3.1 쿠쿠 샌드박스(Cuckoo Sandbox)

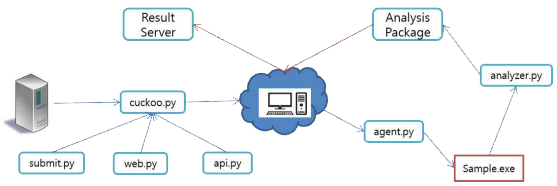


그림 2. Cuckoo sandbox 실행 프로세스

파이썬(python) 형태의 오픈소스(open source)인 쿠쿠 샌드박스는 악성코드 분석 시스템으로 의심스러운 파일, 프로그램을 격리된 환경에서 실행하여 프로그램을 동적●정적으로 분석하고 다양한 분석 결과를 제공한다.[5] 그림 2는 쿠쿠 샌드박스 실행 프로세스(process)로 호스트(host) 운영체제에서 cuckoo.py를 실행하면 가상환경이 실행되고, agent.py를 통해 악성코드 sample.exe를 실행한다. sample.exe가 작동하면 analyzer.py가 다양한 분석 방법이 존재하는 Analysis Package를 실행하여 악성코드를 분석한다. 분석이 종료되고 가상 서버를 종료한 후 결과 서버(Result Server)에 분석한 값을 전송하고 분석 보고서를 제공한다. 분석 보고서의

정보로는 프로세스의 메모리 정보, 버퍼(buffer), 네트워크 정보, 추출 파일 정보, 정적 분석 정보(PE 헤더 파일), 동적 분석 정보(프로세스 이름, 접근 파일, API 등), 로그 정보 등이 존재한다. 본 논문에서는 정적•동적 정보만을 활용했다.

3.2 상관관계(Pearson's correlation)

상관관계란 의존성을 포함하는 광범위한 통계적 관계 중 하나로 일반적으로 두 변수가 가까울수록 선형 관계에 있음을 뜻한다. 상관관계는 표준 편차가 모두 유한하고 0이 아닌 경우에만 정의된다. 즉 상관관계의 절대 값이 1을 초과 할 수 없다는 것을 의미한다.

기대 값 μx와 μy와 표준 편차 σx와 σy를 갖는 두 무작위 변수 x와 y사이의 모집단 상관계수 ρx, y는 다음과 같이 정의된다.

$$\rho x,y = \frac{cov(x,y)}{\sigma x \sigma y} = \frac{E[(x-\mu x)(y-\mu y)}{\sigma x \sigma y} \quad (1)$$

식 (1)에서 E는 기대 값 연산자이고, cov는 공분산을 의미한다. 그러므로 상관계수 결과는 0에 가까워질수록 변수 간의 상관관계가 낮다는 것을 의미하고, 반대로 계수가 -1 또는 1에 가까울수록 변수 간의 상관관계가 높다는 것을 의미한다.

3.3 랜덤 포레스트 (Random Forest)

랜덤 포레스트는[6] 앙상블 학습에서 사용되는 방법 중 하나로, 다양한 의사 결정 트리(Decision Tree) 모델을 구축하고 조합하여 사용하는 방법이다. 이 방법은 전체 학습 데이터를 임의의 여러 개의 부분 데이터로 나눈 후, 각 부분 데이터별로 의사 결정트리 모델을 학습시켜 다양한 모델을 생성한다. 그리고 이를 활용하여 입력 데이터에 대한 레이블(label)을 예측한다. 부트스트랩(bootstrap)을 통해 조금씩 상이한 훈련 데이터에 대해 기초 분류기(base learner)를 결합하는 배깅(bootstrap aggregation, bagging)을 이용한 방식과 훈련 단계에서 훈련 목적 함수를 최대로 만드는 노드 분할 함수의 매개변수의 최적값을 구하는 임의 노드 최적화를 이용한 방식이 있다. 이번 논문에서는 배깅을 이용한 랜덤 포래스트 방식을 사용한다.

IV.실험

4.1 실험 환경

앞서 설명한 분석 모델을 실제로 적용 가능함을 보이기 위하여, 실제 랜섬웨어 50개, 악성코드 90개, 일반 파일 90개를 분석하였다. 실험 환경은 아래 표2와 같다.

실제 환경	
CPU	인텔 코어i5-4세대
RAM	8GB
OS	Ubuntu 16.04 LTS
가상 환경	
CPU	인텔 코어i5-4세대
RAM	2GB
OS	Window 7(64-bit)

표 2. 분석환경

4.2 특징 정보 선정

쿠쿠 샌드박스를 통해 얻은 정적 정보와 동적 정보를 상관관계를 통하여 분석하였다. 랜섬웨어와 악성코드의 상관관계 분석 결과 파일 열기, 파일 생성, 파일 존재, 전역 고유 식별자에 관하여 가장 적은 상관관계를 보였다. 랜섬웨어와 일반 파일 분석 결과 파일 존재, 파일 쓰기, 레지스터키 열기, 레지스터키 읽기가 가장 낮은 상관관계를 보였다. 랜섬웨어와 악성코드, 일반 파일의 상관관계 결과는 표 4에 작성했다. 그 결과 랜섬웨어의 특징정보로 파일 생성(file_created), 파일 열기(file_opened), 파일 존재(file_exists), 파일 쓰기(file_written), 전역 고유 식별자(guid), 레지스터키 열기(regkey_opened), 레지스터키 읽기(regkey_read)로 정의했다.

	vs Malware	vs Benign
1) file_created	0.029	0.177
2) file_opened	0.026	0.109
3) file_written	0.135	0.053
4) file_exists	0.034	0.021
5) guid	0.06	0.204
6) regkey_opened	0.266	0.087
7) regkey_read	0.125	0.094

표 3. 랜섬웨어와 악성코드, 일반 파일 상관관계

4.3 실험 결과

실험은 세 가지 방식으로 진행했다. 첫 번째로 악성코드, 일반파일 모두 가지고 있는 파일 존재와 악성코드에서 그 다음으로 상관관계가 낮은 파일 열기와 일반 파일에서 그 다음으로 상관관계가 낮은 파일 쓰기를 가지고 분석했다. 두 번째 실험으로 첫 번째 실험 데이터에 악성코드에서 두 번째로 낮은 상관관계를 보인 파일 생성과 일반 파일에서 두 번째로 낮은 상관관계를 보인 레지스터 키 열기를 포함하여 분석했다. 세 번째 실험으로 두 번째 실험의 데이터에 악성코드에서 세 번째로 낮은 상관관계를 보인 전역 고유 식별자와 일반파일에서 세 번째로 낮은 상관관계를 보인 레지스터 키 읽기를 포함하여 분석했다. 본 논문의 실험 데이터로 91개의 랜섬웨어, 악성코드, 일반 파일을 사용했고 Ekta[2]의 논문에서 의사 결정 트리의 수가 10개가 가장 좋은 결과를 나타낸다고 기술하여 의사 결정 트리의 수를 10으로 지정했다. 아래 표4는 각 실험 결과이다. 실험 결과 랜섬웨어와 악성코드 비교 시 탐지 율이 가장 높은 랜섬웨어의 특징정보는 파일 존재, 파일 열기, 파일 쓰기, 파일 생성, 레지스터 키 읽기다. 랜섬웨어와 일반 파일 비교 시 탐지 율이 가장 높은 랜섬웨어의 특징정보는 파일 생성, 파일 열기, 파일 존재, 파일 읽기, 전역 고유 식별자, 레지스터 키 열기, 레지스터 키 읽기다.

	vs Malware	vs Benign
	accuracy	accuracy
test1	0.85	0.82
test2	0.86	0.92
test3	0.85	0.93

표 4. 랜섬웨어와 악성코드, 일반 파일 분석 결과

V.결론

본 논문에서는 쿠쿠 샌드박스를 통해 랜섬웨어, 악성코드, 일반 파일을 분석하여 분석 정보를 얻었다. 그 후 상관관계를 이용하여 랜섬웨어의 특징 정보를 추출하고, 랜덤 포레스트 기계학습 알고리즘을 활용하여 실험을 진행하여 랜섬웨어와 악성코드와 비교 시 가장 높은 정확도는 86%이고, 일반 파일과 비교 시 가장 높은 정확도는 93%다.

향후 연구에서는 정적 분석 정보인 PE Imphash와 PE Resource 정보를 이용하여 블랙●화이트 리스트를 적용할 것이다. 또한 논문에서 쿠쿠 샌드박스의 분석 정보 중 적용하지 못한 프로세스 메모리, 추출된 파일(dump file), 네트워크 정보, 로그 정보 등을 활용하여 랜섬웨어만의 특징 정보를 추가할 것이다.

[참고문헌]

[1] Yibin. Liao. “PE-Header-Based Malware Study and Detection”, http://cobweb.cs.uga.edu/

[2] Ekta Gandotra, Divya Bansal, Sanjeev Sofat. “Zero-day malware detection”, Embedded Computing and System Design (ISED), Sixth International Symposium on. IEEE, p. 171-175. 2016.

[3] Qian Chen, and Robert A. Bridges. "Automated Behavioral Analysis of Malware A Case Study of WannaCry Ransomware." arXiv preprint arXiv:1709.08753, https://arxiv.org/ 2017.

[4] Daniele Sgandurra, Luis Muñoz-González, Rabih Mohsen, Emil C. Lupu. “Automated Dynamic Analysis of Ransomware: Benefits, Limitations and use for Detection”, arXiv preprint arXiv:1609.03020. https://arxiv.org/ 2016

[5] https://cuckoosandbox.org/

[6] Tin Kam Ho. "Random decision forests." Document Analysis and Recognition, Proceedings of the Third International Conference on. Vol. 1. IEEE, 1995.

Practical Plug & Play Measurement Device Independent Quantum Key Distribution System

Changhoon Park[1,2], Min Ki Woo[2], Byung Kwon Park[1,3], Min Soo Lee[1], Young-Wook Cho[1], Yong-Su Kim[1], Sangin Kim[2], Sang-Wook Han[1,*], Sung Moon[1]

[1]Center for Quantum Information, Korea Institute of Science and Technology (KIST),
[2]Department of Electronics engineering, Ajou university
[3]Department of Nano-Materials Science and Engineering, University of Science and Technology

[*]swhan@kist.re.kr

요 약

Measurement device independent (MDI) quantum key distribution (QKD) 은 실험적으로 구현된 QKD 프로토콜 중 가장 안전한 방법이다. Blind attack과 같이 측정 장치를 공격하는 양자해킹 공격에 원천적으로 안전한 프로토콜이기 때문이다. 하지만 서로 다른 광원에서 생성된 광자 쌍의 양자 간섭을 이용하기 때문에 키 생성 효율이 떨어지고 시스템이 복잡해지는 한계가 있다. 그래서 이러한 실용성 문제를 개선하기 위해 P&P 아키텍처가 제안되었고 기본적인 구조에 대한 실험 검증도 이루어졌다.

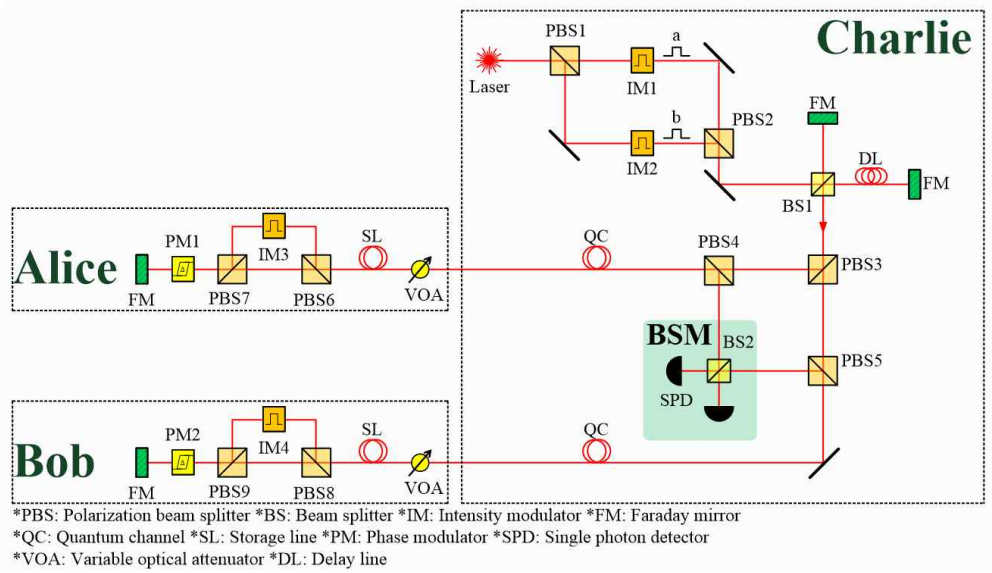
본 논문에서는 시스템 환경변화에도 동작의 안정성을 확보할 수 있는 실시간 자동보정 시스템을 사용해서 환경변화에도 안정적인 동작이 가능한 P&P MDI QKD를 구현하고 검증했다. 또한 편광기반 분할 기술을 적용함으로써 Alice와 Bob의 양자채널 길이의 차이와 무관하게 동작 가능하도록 했다.

I. 서론

양자 키 분배[1]는 양자역학에 의해 이론적으로 무조건적인 안전성이 보장된다. 하지만 시스템으로 구현할 경우 소자들이 가지는 불완전한 특성 때문에 양자해킹이 가능하게 된다.[2]

다양한 양자해킹 방법들 중 실질적으로 위협이 되는 방법은 측정장치를 타겟으로 해킹을 시도하는 경우가 많다. 측정 장치에 비정상적으로 강한 빛을 조사할 경우 단일광자검출기의 특성이 변형되어 도청자의 의도대로 검출기가 동작하게 만든다. 실제 검출기 공격을 통해 상용 양자암호 장치를 완전히 해킹하여 양자암호 키를 탈취하는 방법이 보고되어 큰 반향을 일으켰다.[9] 이에 따라 검출기 해킹 방지 방법이 많이 제안되었는데 그 중에 원천적으로 측정장치의 특성과 무관하게 안전성이 보장되는 Measurement device independent QKD (MDI QKD)가 해킹방지를 위한 방법으로 주목받고 있다.[3]

MDI QKD의 안전성 유지의 핵심은 Bell state measurement (BSM)라는 독특한 측정 방법이다. BSM은 Alice와 Bob의 광자로 이루어진 광자 쌍의 상관관계를 측정하는 기법으로, 이 측정 결과에서 키 정보를 추출하기 위해서는 Alice 또는 Bob의 인코딩 정보도 알고 있어야 한다. 따라서 어떠한 인코딩 정보도 알 수 없는 도청자는 해킹을 통해 BSM의 측정 결과를 알 수 있다고 하더라도 키 정보를 추출할 수 없기 때문에 측정 결과와 무관하게 안전성이 유지되는 것이다. 하지만 BSM은 두 광자 간섭(Two-photon interference)을 이용한 측정 방법이기 때문에 구현하기 위해서는 Alice와 Bob의 광자의 구별불가능성 유지가 필수이다. 구별불가능성 유지를 위해서는 편광, 파장, 타이밍 제



[그림1] 실험셋업

어를 위한 많은 수의 능동소자가 필요하기 때문에 시스템이 복잡해지고 실용성이 감소하게 된다. 그래서 이러한 실용성 부족을 개선하기 위해 모드매칭에 유리한 Plug & Play(P&P) 아키텍처가 제안되었다.[4] P&P 아키텍처에서는 하나의 레이저에서 Alice와 Bob의 광 펄스가 생성되기 때문에 파장이 자동적으로 일치하게 된다. 그리고 Alice와 Bob으로 전송된 빛은 Faraday mirror에 반사되어 동일한 경로를 통해 되돌아오기 때문에 전송될 때 틀어졌던 편광이 자동 보상된다. 또한, Alice와 Bob의 광 펄스가 동일한 간섭계를 통과하기 때문에 Reference frame의 크기도 자연적으로 일치된다. 이렇듯 P&P 아키텍처에서는 파장과 편광 그리고 Reference frame의 매칭이 필요 없고, Alice와 Bob의 광 펄스의 타이밍 제어만 필요하기 때문에 시스템을 단순화할 수 있다.

P&P를 적용해서 실용성을 개선시킨 MDI QKD의 연구사례에는 Free-space 상에서 실현 가능성을 보여준 연구[5]와 광섬유 기반으로 구현된 P&P MDI QKD의 연구가 있다.[6] 광섬유 기반의 연구사례는 광섬유에서의 최초구현이라는 것에 주목할 만하지만 Alice와 Bob의 양자채널의 길이가 비대칭인 경우만 동작 가능한 한계와 시각동기를 위해 다수의 광원 소자가 필요하다는 단점이 있다.

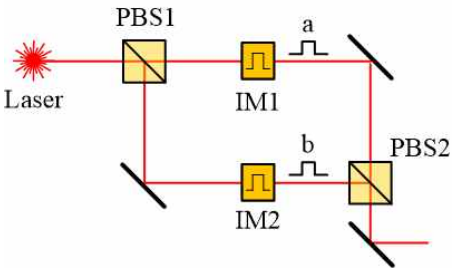
본 논문에서는 실시간 자동보정 시스템을 사용해서 온도변화와 같은 환경변화에도 안정적인 동작이 가능한 P&P MDI QKD를 제안하고 구현했다. 자동보정 시스템은 소프트웨어적으로 구현되었기 때문에 최소한의 하드웨어로 구현할 수 있었고, 또한 편광기반 분할 기술을 적용함으로써 Alice와 Bob의 양자채널 길이의 차이와 무관하게 동작 가능하도록 했다.

II. 실험

1.1 실험셋업

제안하는 실험셋업[그림1]에서는 P&P 아키텍처를 적용했기 때문에 하나의 레이저를 사용해서 Alice와 Bob의 빛을 생성한다. 이렇게 생성된 빛은 IM를 통해 3ns 펄스폭과 1MHz 속도를 가지는 광 펄스로 변조된 후 Michelson 간섭계를 통해 Time-bin(100ns)으로 생성된다. Time-bin은 25km 길이의 양자채널을 통해 Alice와 Bob으로 전송되고 Time-bin phase 인코딩 방식으로 인코딩된다. Z basis는 IM을 사용하여 Time-bin의 Fast 또는 Slow 타이밍을 선택해서 인코딩할 수 있고, X basis는 Phase modulator(PM)를 사용하여 Fast와 Slow 타이밍의 상대적 위상 차이를 0 또는 π로 변조함으로써 인코딩할 수 있다. 인코딩된 후 FM에 반

사되어 Charlie로 되돌아온 Alice와 Bob의 광자는 BSM으로 측정된다. BSM의 측정결과가 $|\psi^-\rangle$일 경우 Charlie는 고전채널을 통해 Alice와 Bob에게 측정결과를 알려주고, Alice와 Bob은 basis 교환을 통해 키를 나눠가지게 된다.



[그림2] 편광기반 분할 기술

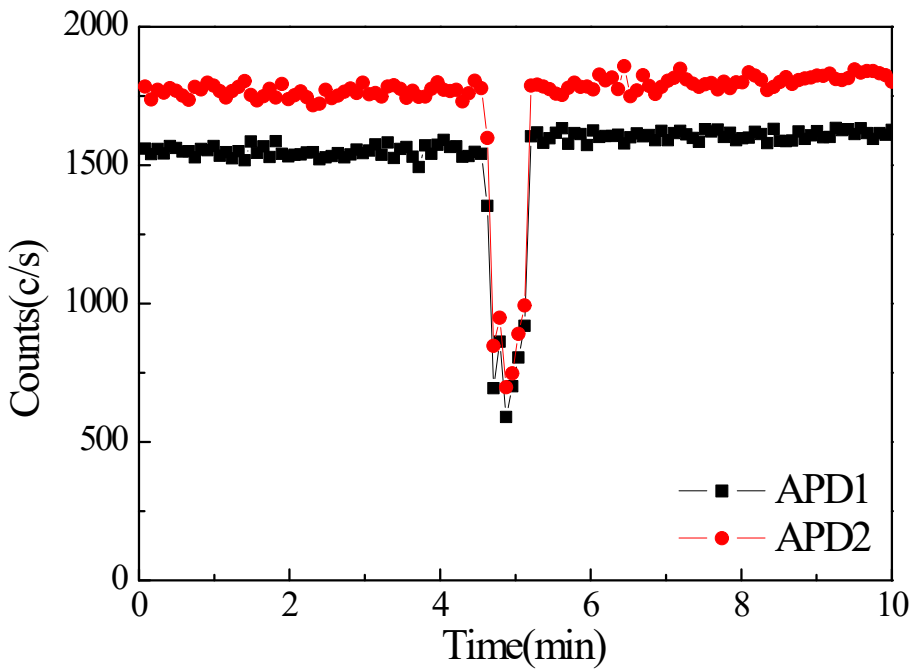
1.2 편광기반 분할 기술

제안하는 셋업은 [그림2]와 같은 편광기반 분할 기술을 적용했기 때문에 양자채널 길이의 대칭 또는 비대칭과 무관하게 동작이 가능하다. 편광기반 분할 기술은 편광에 따라 광 경로를 분할할 수 있는 기술로서, 레이저에서 나온 빛은 PBS1 (Polarization beam splitter)의 각각의 경로에 있는 IM에 의해 독립적인 타이밍에 Alice와 Bob의 광 펄스로 생성되고 PBS2에서 합쳐진다. 합쳐진 광 펄스들은 동일한 경로, 간섭계를 통과한 후 PBS3를 통해 수평편광의 광 펄스는 Alice로, 수직편광의 광 펄스는 Bob으로 분할 전송된다. 이러한 방법을 적용하면 독립적인 타이밍에 Alice와 Bob의 광 펄스를 생성할 수 있기 때문에 양자채널의 길이와 무관하게 동작이 가능하게 된다.

1.3 실시간 자동보정 시스템

광섬유 기반 QKD를 구현하기 위해서는 온도에 따라 광섬유의 유효 길이가 달라지는 문제를 해결해야 한다. 검출기는 노이즈 성분을 최소화하기 위해 광자가 입력될 타이밍에만 동작되도록 설계되어있다. 그래서 환경 변화에 따라 광섬유의 유효길이가 변화하게 되면 초기에 맞춰진 검출기와 광자의 타이밍이 틀어져서 광자가 검출되지 않는 문제가 발생한다.

실시간 자동보정 시스템은 이러한 문제를 해결할 수 있는 방법으로 검출기의 검출량을 모니터링해서 검출량이 임계값보다 낮아지지 않도록 Alice와 Bob의 타이밍을 실시간으로 제어해주는 기술이다.[7] 자동보정 시스템의 동작확인을 위해 광섬유 온도제어가 가동되지 않는 환경에서 검출기의 검출량 변화를 측정했다. 결과



[그림3] 실시간 자동보정 시스템(적용된 임계값: 1400)

는 [그림3]과 같이 검출량이 임계값보다 낮아지면 자동보정 시스템이 딜레이를 변화시키면서 검출량을 임계값 이상으로 복구 시키는 것을 확인할 수 있다.

III.결과

제안하는 셋업의 실현 가능성을 확인하기 위해 Key rate와 Quantum bit error rate(QBER)를 측정했다. 평균 광자 수는 0.5 photons/pulse 수준으로 설정했고, 측정 결과로 3%(Z basis)와 28%(X basis) 수준의 QBER과 6.42×10^{-6} bits/pulse 수준의 Key rate를 얻을 수 있었다. 레이저를 광원으로 사용한 QKD의 경우 X basis의 이상적인 QBER 값이 25%라는 것을 감안하면 두 basis의 실질적인 QBER이 3% 이하라는 것을 알 수 있다.[8]

IV.결론

본 논문에서는 MDI QKD보다 실용성과 안정성이 개선된 광섬유 기반의 P&P MDI QKD를 구현했다. P&P 아키텍처를 적용했기 때문에 모드매칭에 필요한 능동소자를 최소화할 수 있었고, 실시간 자동보정 시스템을 적용함으로써 환경변화에 대한 안정성도 높일 수 있었다. 또한 편광기반 분할 기술을 적용하여 Alice와 Bob의 양자채널 길이의 차이와 무관한 동작이 가능하도록 했다.

양자채널 길이가 25km 일 때 Key rate는 6.42×10^{-6} bits/pulse 이고 QBER은 3% 이내로 측정되었다. 낮은 효율의 검출기와 1MHz의 저속 동작주파수를 사용했기 때문에 상대적으로 키 생성 속도가 낮지만 고효율 검출기와 높은 동작 주파수를 사용할 후속연구를 통해 성능을 대폭 개선할 예정이다.

사사

본 연구는 ICT R&D MSIP/IITP program(Grant No. B0101-161355), KIST research program(Grant No. 2E27231), KIST-ETRI research program(Grant No. 2V05340)의 일환으로 수행하였음.

[참고문헌]

[1] C. H. Bennett and G. Brassard, Quantum Cryptography: public key distribution and coin tossing, in Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing, Bangalore, India (IEEE, Piscataway, 1984), p.175.

[2] V. Makarov et al., Effects of detector efficiency mismatch on security of quantum cryptosystems, Phys. Rev. A 74, 022313 (2006).

[3] H.-K. Lo et al., Measurement-Device-Independent Quantum Key Distribution, Phys. Rev. Lett. 108, 130503 (2012).

[4] A. Muller et al., "Plug and play" systems for quantum cryptography, Appl. Phys. Lett. 70, 793 (1997)

[5] Y. Choi et al., Plug-and-play measurement-device-independent quantum key distribution, Phys. Rev. A 93, 032319 (2016).

[6] Tang, G. Z. et al., Experimental asymmetric plug-and-play measurement-device-independent quantum key distribution. Phys. Rev. A 94, 032326 (2016).

[7] B. K. Park et al., "QKD system with fast active optical path length compensation," Sci. China Phys. Mech. 60, 060311 (2017).

[8] T. Ferreira da Silva et al., Proof-of-principle demonstration of measurement-device-independent quantum key distribution using polarization qubits, Phys. Rev. A 88, 052303 (2013)

[9] L. Lydersen et. al., Hacking commercial quantum cryptography systems by tailored bright illumination, Nat. Photonics 4, 686 (2010).

Ponce를 활용한 기호실행 기반 악성프로세스를 발현시키는 방법 연구

한중수* 임을규

*한양대학교 컴퓨터소프트웨어학과

한양대학교 컴퓨터소프트웨어학부

Research on triggering malicious process based on symbolic execution using Ponce

Joong-Soo Han*

Eul-Gyu Im

*Division of Computer and Software, Hanyang University

Department of Computer and Software, Hanyang University

요 약

악성코드 분석에 있어서 분석하려는 악성코드가 동작하도록 발현시키는 작업은 악성 행위분석을 위해 필수적이다. 특정한 입력 값에 대해 실행시킬 실행 프로그램 테스팅에서 쓰이는 퍼징 또는 무차별 대입 등 실제실행 방법은 전체 실행경로 탐색이 필수적이며 훨씬 광범위한 입력 범위를 요구한다. 반면 기호실행은 각 변수들을 기호 자체로서 인식하며 원하는 해당 분기가 발현되기 위해 필요한 입력값의 범위를 줄일 수 있다. 본 논문에서는 오염분석(Taint analysis) 기반 기호실행을 활용하여 본 연구를 위해 제작한 악성 프로그램을 발현시키는 방법을 보인다.

I. 서론

프로그램은 크게 두 가지 방법으로 실행된다. 일반적으로 프로그램을 실행시키는 방법은 실제실행(Concrete execution)이며 실수, 정수, 또는 문자열 등 구체적인 값들을 프로그램 실행 시 입력 받아 실행되는 방법이다. 반면 기호실행(Symbolic execution)은 기호변수(Symbolic variables)를 사용한다. 사용자 또는 외부 프로세스의 입력에 의해 결정되는 분기를 기호변수로 취급하여 각 분기의 입력을 알아내며 프로그램을 실행시킨다. 실제실행은 탐색 범위를 지정한 후, 범위 안의 값들을 무차별 대입(Brute force)하거나, 임의의 입력 값을 입력하는 퍼징(Fuzzing) 방법으로 진행된다. 기호실행은 특정 기호변수에 해당하는 분기의 제약조건을 오염분석(Taint analysis)하여 유도할 분기로 풀이하여 진행된다. 본 논문에서는 IDA Pro 플러그인의 하나인 Ponce를 이용하여 기호실행 및 오염분석을 진행한다. Ponce는 내부적으로 기호실행 라이브러리인 Triton의 API들을 사용한다. Triton API에 의해 바이너리가 로드된 후 Triton 중간 표현 형태로 변환된다. Triton 표현 형태들을 이용해 오염될 메모리 설정 및 기호실행이 가능하다. 본 논문에서 기호실행 및 오염분석에 대한 절차와 원리에 대해 살펴본다. 추가로 적절한 기호실행이 되었는지 제작한 예시 악성 프로그램을 통해 검증한다. 앞서 기술한 악성 프로그램은 특정 개시 문자열을 네트워크 패킷으로 입력받았을 때 동작하며 기호실행을 통해 해당 개시문자를 풀이한다.

II. 관련 연구

Choi, YH. 외 2인은 동적 바이너리 분석 모듈을 제작하여 프로그램 취약점 진단을 수행하였다. 기존 패턴 매칭 방식의 취약점 진단을 코드를 스스로 변화하는 악성코드들에 대응할 수 있도록 개선하였다. [2]

Bekrar 외 1인은 프로그램 취약점 진단 방법으로 오염분석을 적용한 스마트 퍼징(Smart Fuzzing) 기법을 사용하였다. 오염분석 및 기호실행을 통해 커버리지를 분석하며 퍼징 기법으로 프로그램 실행경로 범위를 축소시켰다. [3].

Wang 외 1인은 프로그램 취약점 진단 방법으로 체크섬 기반 무결성 검사기를 제작했다. 메모리 영역을 오염분석하여 정상 및 악성 입력을 구분했다. 후보 취약점들의 범위를 감소시킨 후 퍼징 기법을 적용했다. [4].

III. 기호실행 절차

기호실행의 단계는 바이너리 로드, 초기 상태 설정, 메모리 기호화 및 오염분석, 제약조건 수집 및 풀이의 단계를 거친다. Ponce 플러그인을 사용한 기호실행 방법은 Triton에서 지원하는 API들을 이용해 진행된다.

3.1 바이너리 로드

Ponce에서 바이너리를 동적 분석할 때, 첫 과정은 바이너리 로드이다. 로드된 바이너리는 시스템 구조에 따라 다양한 객체들로 데이터를 읽어오고 연산할 수 있다. Ponce 플러그인에선 Triton API를 이용하의 Triton 기본 구조체를 사용한다. IDA Pro 디버거로 바이너리 실행 시 얻어지는 어셈블리 코드들(ARM Instruction)들은 Triton 추상 자료형인 ast로 변환된다.

3.2 초기 상태 설정

바이너리는 로드된 후 Triton API에서 이용하는 IR로 변환된다. 그 후, 목적 메모리를 오염 및 실행 전 상태를 초기상태라 칭한다. 초기상태 이후 오염분석이 진행된다. 이 때 Ponce는 해당 메모리의 비트스트림 변경을 추적한다.

3.3 메모리 기호화 및 오염분석

본 논문은 기호실행 동적분석 방법 중 오염분석(Taint analysis) 기술을 사용하여 분석을 진행한다. 오염분석 기술은 바이너리가 실행되는 중 정보들이 끊임없이 복사되고 수정되는 부분에 주목하여 정보의 흐름을 분석하는 기술이다. 추적할 대상 메모리를 기호화 하면 해당 메모리에 산술적 연산을 통해 영향을 주는 모든 객체들의 집합을 추적하여 전체 코드의 실행흐름을 분석한다. 본 논문의 기호실행 분석 역시 오염분석 기술에 기반을 둔 분석이다.

3.4 제약조건 수집

목표 메모리를 오염시킨 후, 프로그램 분기의 제약조건을 수집한다. 기호화된 메모리에서 분기가 발생할 때, 각 조건 분기는 기호화된 값들 간 연산이 이루어지며 Bool 형식으로 출력되며 프로그램의 실행이 변경된다. 이때, 제약조건에 따라 Bool 형식의 출력이 결정된다. Ponce 및 Triton API는 실행되는 분기의 모든 제약조건을 수집하여 이후 풀이 연산에 활용한다.

3.5 제약조건 풀이

수집된 제약조건들은 선형 방정식으로 변환되며 선형 방정식 풀이 엔진인 SMT(Satisfiability Modulo Theories)를 이용해 풀이된다. Ponce 플러그인에서 사용하는 Triton은 z3 엔진을 사용하여 제약조건을 풀이한다.

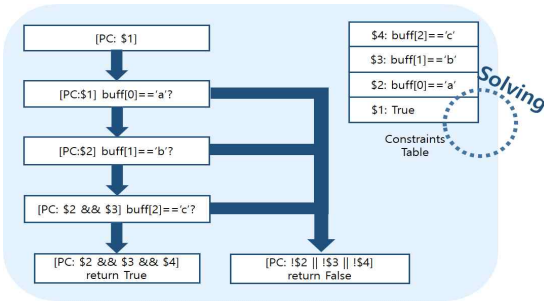


그림 1 제약조건 수집 방법 예시

IV. 기호실행을 위해 제작한 키로거

4.1 키로거 악성코드 소개

본 논문에서 악성코드의 오염분석을 위해 키로거(Key logger)를 제작하였다. 키로거는 감염 PC의 키 입력을 공격자 서버로 전송하는 프로그램이다. 본 논문의 목적상, 키로거는 외부 네트워크 패킷 입력과 내부 제약조건에 의존하여 발현되도록 작성하였다.

4.2 제약조건 구조

사용자 PC에서 동작하는 키로거 클라이언트는 윈도우 내장 소켓 버전이 유효한 버전인지 검증한 후, 서버와 통신이 성립되었는지 검증하는 단계를 거친다. 이후 서버로부터 개시 문자열 "CISC"을 전송받을시 키로거가 동작한다. 이후 Winsock 버전이 목표 버전을 만족하는지 확인 후 키로깅이 시작된다. 키로거 동작의 제약조건은 아래 그림과 같다.

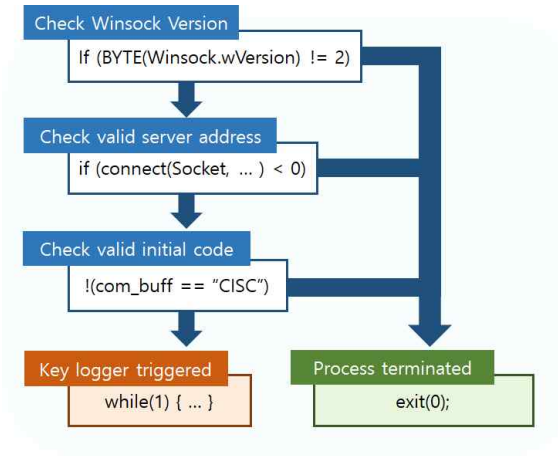


그림 2 제작 키로거 제약조건 도식

V. Ponce를 이용한 키로거 기호실행

5.1 내부 조건 입력 풀이

사용자 정의 함수를 포함한 함수들은 실행을 종료하고 결과값을 반환할 때, eax 레지스터의 하위 비트로 반환이 이루어진다. 해당 하위 비트를 al 레지스터라고 한다. 반환값이 유효한 값인지 실패하였는지 여부는 eax 레지스터 하위비트인 al 레지스터를 비교하여 구분 가능하다. Winsock 버전 확인 등 내부 상태에 영향받는 제약조건들은 al 레지스터를 수정하여 강제 분기한다.

5.2 네트워크 관련 입력 풀이

본 논문의 키로거 프로그램은 서버로부터 개시 문자열이 포함된 패킷을 전송받은 이후 동작한다. 네트워크 패킷 입력과 관련된 Win API 함수들의 경우 함수실행 전 네트워크 소켓 내부에서 버퍼 기능을 수행하는 문자열 배열 변수 선언이 이루어진다. 그림 3은 코드 상단에 길이 10바이트의 com_buff[] 배열 선언을 보여준다. 코드 하단에 앞서 서술하였듯이, _recv@16 함수로 수신한 문자열은 eax 하위 비트로 복사된 후 제약조건 연산이 진행된다.

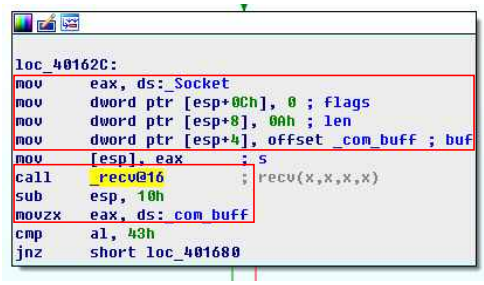


그림 3 버퍼 문자열 배열 변수 예시

앞서 Socket 관련 함수 실행 이전에 문자열 배열 변수 선언이 이루어졌다. 선언된 변수는 일반적으로 정적으로 할당된 변수들을 포함하는 영역인 .bss 안에 존재한다. 그림 4는 스택 안에서 추출한 .bss 영역 내에 존재하는 com_buff 버퍼 문자열 배열의 주소를 보여준다.

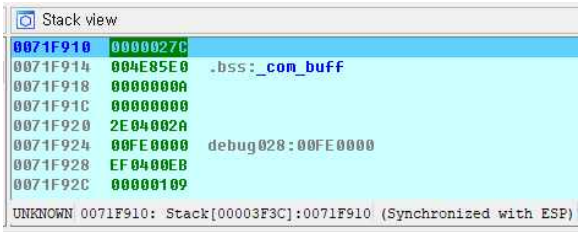


그림 4 스택에서 확인한 .bss 영역 주소

스택에서 추출한 주소를 통해 해당 버퍼 주소로 이동하여 메모리 영역을 오염시킨다. 그 후, 목표한 상태에 도달하기 위해 필요한 프로그램

실행 경로를 선택한다. Ponce는 오염된 메모리의 기존 실행 흐름과 반대 Bool 값으로 강제 분기하는 negate 기능을 지원한다. 각 분기마다 negate 기능을 적용하여 정상 패킷 수신 메모리 상태까지 진행한다.

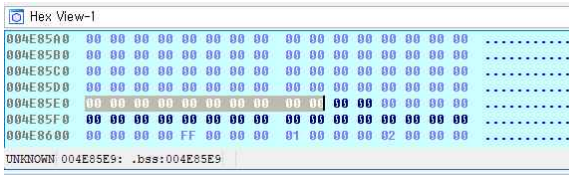


그림 5 Ponce 활용 com_buff[] 오염

목표 메모리까지 분기가 완료되었다면, 수집한 제약조건들을 풀이한다. z3 엔진은 오염된 메모리가 포함된 연산에서 적절한 분기로 진행하는 메모리 상태를 출력한다. 그림 6 및 그림 7은 풀이가 완료되어 개시문자 “CISC”가 성공적으로 출력됨을 보여준다.

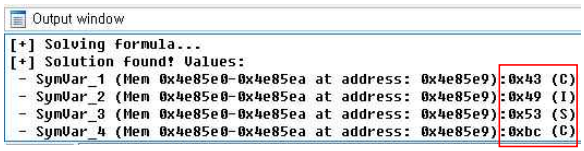


그림 6 키로거 개시 문자열 기호실행 성공 화면

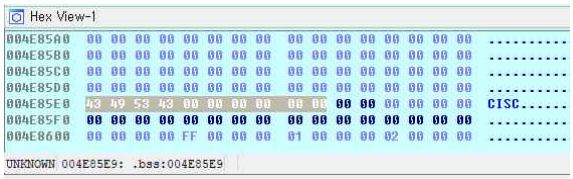


그림 7 메모리 덤프 상에서 제약조건 풀이 검증

VI. 결론 및 향후 연구

본 프로그램의 경우 목적 오염 메모리 설정 및 분기 선택은 사용자의 선택으로 진행된다. 반면 대규모의 악성코드의 경우 사용자 선택이 필요한 분기의 수가 급증한다. 정해진 시간 안에 목표한 분기를 찾아내는 작업이 필수적이다. 향후 연구에서는 유력한 네트워크 입력 패킷 혹은 내부 조건 입력을 추천하거나 사용자가 직접 지정해야 할 분기의 숫자를 감소시키는 방법에 대한 연구를 수행할 예정이다.

[참고문헌]

[1] https://github.com/illera88/Ponce

[2] Choi, YH., Park, MW., Eom, JH. et al. “Dynamic binary analyzer for scanning vulnerabilities with taint analysis” Multimed Tools Appl(2015) 74: 2301. https://doi.org/10.1007/s11042-014-1922-5 2015.

[3] Bekrar, Sofia, et al. "Finding software vulnerabilities by smart fuzzing." Software Testing, Verification and Validation (ICST), 2011 IEEE Fourth International Conference on. IEEE, 2011.

[4] Wang, Tielei, et al. "TaintScope: A checksum-aware directed fuzzing tool for automatic software vulnerability detection." Security and privacy (SP), 2010 IEEE symposium on. IEEE, 2010.

[5] Alazab, Mamoun, Sitalakshmi Venkataraman, and Paul Watters. "Towards understanding malware behaviour by the extraction of API calls." Cybercrime and Trustworthy Computing Workshop (CTC), 2010 Second. IEEE, 2010.

[6] Zhang, Bin, et al. "Detecting integer overflow in Windows binary executables based on symbolic execution." Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2016 17th IEEE/ACIS International Conference on. IEEE, 2016.

2017 한국정보보호학회 동계학술대회

A2 블록체인(I)

장 항 배 (중앙대학교)

- **블록체인 기반 IoT 디바이스 무결성 검증 스킴**
  박병주, 곽진 (아주대학교)
- **블록체인 보안 프레임워크 개발을 위한 블록체인 보안 이슈에 대한 연구**
  노시완, 이경현 (부경대학교)
- **블록체인을 활용한 보안관제 적용 방안 연구**
  조영민 (고려대학교)
- **블록체인 기반 교내 전자투표 시스템 구축 요구사항 분석**
  정채린, 이종호, 김영우, 조은아, 성기정, 김희영, 이경현 (부경대학교)

블록체인 기반 IoT 디바이스 무결성 검증 스킴

박병주*, 곽진**
*아주대학교 컴퓨터공학과 정보보호응용및보증연구실
**아주대학교 사이버보안학과

Blockchain-based IoT Device Integrity Verification Scheme

Byeong-Ju Park*, Jin Kwak**
*ISAA Lab., Department of Computer Engineering, Ajou University.
**Department of Cyber Security, Ajou University.

요 약

최근, ICT 기술의 발달과 함께 IoT 환경이 주목받으며 다양한 IoT 디바이스가 등장하고 있다. IoT 디바이스는 다양한 사용 용도만큼 다양한 성능의 칩셋이 사용되는데 다양한 기능을 제공하는 디바이스의 경우, 고성능의 칩셋을 사용하고, 단순한 기능을 제공하는 디바이스의 경우, 저성능의 칩셋을 사용한다. 하지만 저성능의 칩셋의 경우, 암호화 기능을 지원하지 않거나, 공개키 암호의 동작이 불가능한 스펙을 보유하고 있어 취약하다고 볼 수 있다. 이에 따라, 본 논문에서는 블록체인을 활용하여 저성능 IoT 디바이스에서도 동작이 가능한 블록체인 기반 IoT 디바이스 무결성 검증 스킴을 제안한다.

I. 서론

최근 ICT 기술이 발달하며 IoT 환경도 주목받으며 점차적으로 확산되고 있다. IoT 디바이스는 전등 제어, 가스 밸브 제어 등 간단한 작업부터, 홈 CCTV, 스마트 냉장고, 스마트 TV 등으로 영역을 넓히며 지속적으로 활용도가 증가하고 있다. Cisco의 발표에 따르면, 2018년에는 350억 개, 2020년에는 500억 개의 IoT 디바이스들이 인터넷에 연결될 것으로 예측된다[1].

IoT 디바이스들은 사용 목적 필요로 하는 기능에 따라, 칩셋의 성능이 다양하다. 다양한 기능과 성능을 제공하는 디바이스는 고성능의 칩셋을 사용해 대부분의 암호화 프로토콜이 동작할 수 있다. 하지만, 단순히 전등을 on/off 하거나 반복 작업 등의 기능을 제공하는 디바이스는 저성능의 칩셋을 사용하고 있어, 암호화 프로토콜을 칩셋에서 지원하지 않거나, 인증서나 공개키 암호 처리에 충분한 성능을 가지지 못해 취약하다고 할 수 있다.

이에 따라, 본 논문에서는 저성능 IoT 디바이스에서도 적용 가능한 블록체인 기반의 IoT 디바이스 무결성 검증 스킴을 제안하고자 한다.

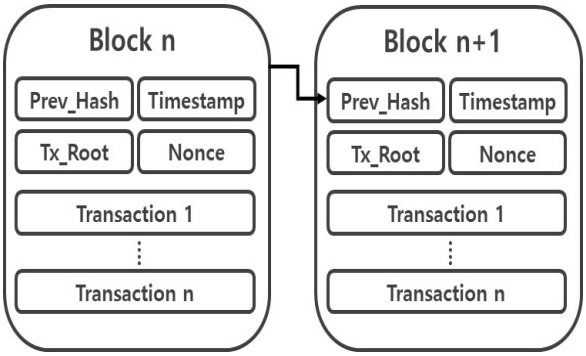
본 논문의 구성은 다음과 같다. 2장에서 블록체인과 IoT 디바이스의 성능에 대해 분석하고, 3장에서는 블록체인 기반 IoT 디바이스 무결성 검증 스킴을 제안한다. 4장에서는 제안하는 무결성 검증 스킴에 대해 안전성을 분석하고, 5장에서 결론을 맺는다.

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 대학ICT연구센터육성 지원사업의 연구결과로 수행되었음(IITP-2017-2016-0-00304)

II. 관련 연구

2.1 블록체인

블록체인은 비트 코인을 위해 개발된 기반 기술로, [그림 1]과 같이 이전 블록의 해시값, 전자서명, 현재 블록의 해시값, 타임스탬프 등으로 구성되어 체인과 같이 연결된 구조를 갖고 있어, 블록에 저장된 정보에 대한 무결성을 제공한다[2,3].



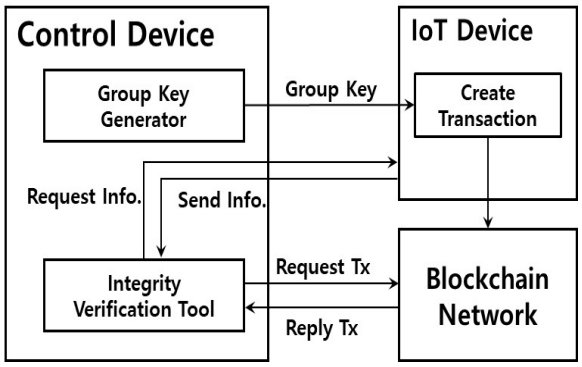
[그림 1] 블록체인 구조

2.2 IoT 디바이스 성능 분석

IoT 디바이스 인증에 사용되는 암호화 방식은 공개키 암호로 높은 연산량을 필요로 한다. IoT 디바이스들은 단순 전등 제어부터 스마트 냉장고, 스마트 TV 등에 이르기까지 사용 용도와 기능에 따라 다양한 성능을 보유하고 있다. 하지만 일부 저성능 디바이스의 경우, 공개키 암호를 지원하지 않고, CPU에 암호화가 내장되어 있지 않다. 이에 따라, 다양한 암호화 기능을 제공하지 않아 디바이스 무결성 검증 수행에 있어 취약하다고 할 수 있다. 아래 [표 1]은 IoT 디바이스 별 공개키 암호화 및 CPU 암호화 기능 지원 여부를 나타낸 표이다.

III. 제안기법

제안하는 디바이스 무결성 검증 스킴은 블록체인을 활용한 기법으로, 트랜잭션 생성 단계, 디바이스 무결성 검증 단계로 분류할 수 있다. 제안하는 스킴의 구조는 다음 [그림 2]와 같으며, 사용되는 기호 및 표기는 다음 [표 2]와 같다.



[그림 2] 제안하는 무결성 검증 스킴

[표 2] 기호 및 표기

파라미터	설 명
K_G	그룹키
Device Info.	디바이스 내부 정보
Tx	트랜잭션
$h(\bullet)$	해시 함수 (256bit)
T	타임스탬프

[표 1] IoT 디바이스 성능 분석

제품명	CPU	CPU 암호화	공개키 암호
Belkin Wemo Switch	Ralink RT5350F	**No**	Yes
Samsung Smarthings Hub	PIC32MX695f-512H 32bit	**No**	Yes
LIFX Color 1000	Kinetis K22 (ARM Coretex-M4)	**No**	**No**
Pebble Time	ST Microelectronics STM32F439ZG (ARM Coretex-M4)	Yes	**No**
Fitbit Surge	Silicon Labs EFM32 Giant Gecko (ARM Cortex-M3)	Yes	**No**
Fitbit One	ST Mircroelectronics 32l151C6 Ultra Low Power (ARM Coretex-M3)	**No**	**No**

❑ Control Device

IoT 디바이스를 제어하는 Control 디바이스는 각각의 IoT 디바이스에 그룹키를 전송하고 무결성 검증을 수행한다.

❑ IoT Device

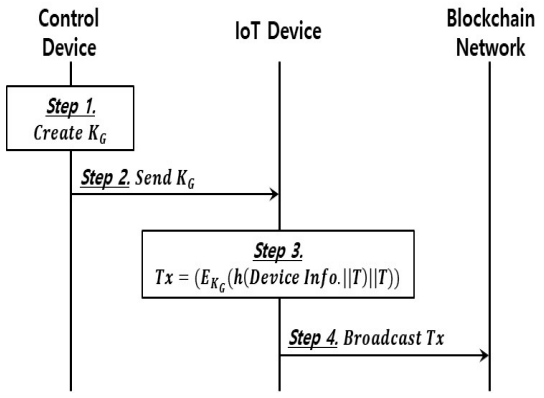
IoT 디바이스에서는 Control 디바이스로부터 전송받은 그룹키를 디바이스의 커널 코드, 데이터, 디바이스 고유 번호 등 해당 IoT 디바이스의 고유한 정보와 결합해 트랜잭션을 생성한다.

❑ Blockchain Network

블록체인 네트워크는 IoT 디바이스가 브로드캐스트한 트랜잭션을 블록 형태로 저장하고, 사용자가 Control 디바이스를 통해 무결성을 검증하고자 트랜잭션을 요청하면 트랜잭션을 제공한다.

3.1 트랜잭션 생성 단계

제안하는 스킴에서는 블록체인의 트랜잭션을 활용하여 디바이스의 무결성을 보장한다. 트랜잭션은 커널 코드, 디바이스 고유번호, 타임스탬프 등 해당 디바이스의 고유한 정보들을 사용자의 컨트롤 디바이스로부터 전달받은 그룹키로 암호화해 생성한다. 그룹키는 IoT 디바이스와 초기 연결하는 단계에서 안전한 채널을 통해 전송되며, IoT 디바이스는 트랜잭션을 주기적으로 생성해 블록체인 네트워크에 브로드캐스트한다. 트랜잭션을 생성하는 단계는 다음 [그림 3]과 같다.



[그림 3] 트랜잭션 생성 단계

Step 1. Control 디바이스를 통해 IoT 디바이스에 전송할 그룹키 K_G를 생성한다.

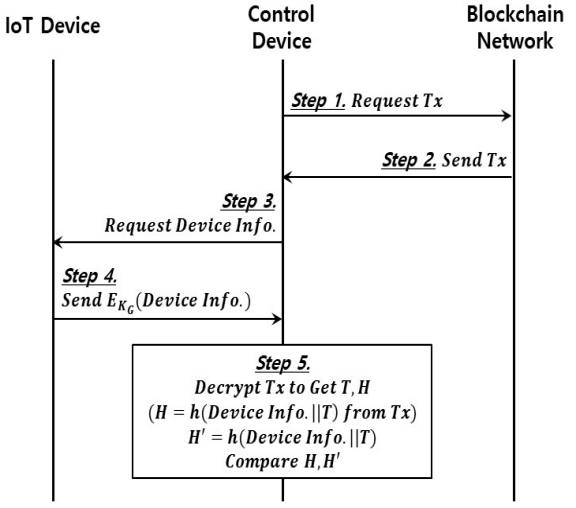
Step 2. Control 디바이스는 생성된 그룹키를 IoT 디바이스에 전송한다.

Step 3. IoT 디바이스는 디바이스의 고유한 정보인 고유번호, 커널 코드, 디바이스에 담긴 사용자의 정보 등을 타임스탬프와 연접 후, 전송받은 그룹키 K_G로 암호화해 트랜잭션을 생성한다.

Step 4. IoT 디바이스는 생성된 트랜잭션을 블록체인 네트워크에 브로드캐스트한다.

3.2 디바이스 무결성 검증 단계

제안하는 무결성 검증 스킴은 [그림 2]와 같이 3개의 엔티티로 구성된다. 무결성을 검증하기 위해서 Control 디바이스는 IoT 디바이스에는 IoT 디바이스의 정보를, 블록체인 네트워크에는 가장 최근에 등록된 트랜잭션 정보를 요청한다. Control 디바이스에서는 전달받은 각각의 정보들의 비교 검증을 통해 디바이스 무결성 검증을 수행하며, 절차는 다음 [그림 4]와 같다.



[그림 4] 무결성 검증 단계

Step 1. Control 디바이스는 무결성 검증 수행을 위해 블록체인 네트워크에 트랜잭션 전송을 요청한다.

Step 2. 블록체인 네트워크는 Control 디바이스에게 트랜잭션을 전송한다.

Step 3. Control 디바이스는 IoT 디바이스에게 트랜잭션 생성에 사용된 IoT 디바이스의 고유정보를 요청한다.

Step 4. IoT 디바이스는 디바이스 고유정보를 그룹키로 암호화해 Control 디바이스에 전송한다.

Step 5. Control 디바이스는 Tx를 복호화해 타임스탬프 T와 IoT 디바이스가 생성한 해시값 H를 얻고, 타임스탬프와 IoT 디바이스로부터 전송받은 $Device\ Info.$를 통해 해시값 H'을 생성하고, H와 H'를 비교해 무결성 검증을 수행한다.

IV. 안전성 및 효율성 분석

본 논문에서 제안하는 블록체인 기반 IoT 디바이스 무결성 검증 스킴은 IoT 디바이스의 무결성 검증을 위해 블록체인 트랜잭션의 수정 불가능한 특성을 활용한다.

4.1 안전성 분석

❑ 중간자 공격 방지

제안하는 스킴에서 무결성 검증을 위해 IoT 디바이스에서 Control 디바이스에 전송되는 $Device\ Info.$는 그룹키로 암호화되어 전송된다. 또한, Control 디바이스, IoT 디바이스, 블록체인 네트워크 간에 전송되는 트랜잭션들은 해시화된 형태로 공격자가 중간에 탈취하더라도 트랜잭션만으로는 정보를 알 수 없어 안전하다.

❑ 메모리 위변조 공격 방지

공격자가 디바이스에 접근해 커널 메모리 변조 등의 공격을 수행하게 되면 다음에 생성된 트랜잭션은 기존의 트랜잭션과 다른 값이 생성되어 디바이스의 위·변조 여부를 알 수 있어 디바이스에 대한 공격을 방지할 수 있다.

4.2 효율성 분석

제안하는 스킴에서 IoT 디바이스는 다음 수식 (1)과 같은 연산을 통해 블록체인 트랜잭션을 생성한다. 수식 (1)과 같은 단순한 연산을 통해 저성능 칩셋에서도 수행이 가능해 기존의 공개키 암호 등을 지원하지 않는 칩셋에서도 사용이 가능하다.

$$Tx = (E_{K_G}(h(Device\ Info. \| T) \| T)) \quad (1)$$

V. 결론

ICT 기술의 발달과 함께, IoT 환경도 주목받고 있으며 다양한 연구도 함께 진행되고 있다. IoT 환경은 개인정보와 밀접한 관련을 갖고 있으며, 많은 디바이스의 수와 인터넷과 연결되어 있다는 점으로 인해 디바이스 무결성에 대한 고려가 필요하다. 하지만 저성능 IoT 디바이스들의 경우, 공개키 암호 미지원 등으로 낮은 보안성을 갖고 있다. 이에 따라, 본 논문에서는 블록체인의 특성을 활용하여 안전하고 경량화된 블록체인 기반 IoT 디바이스 무결성 검증 스킴을 제안하고 이에 대한 안전성 및 효율성을 분석하였다.

향후 연구로는 IoT 디바이스 인증에 대한 연구와 함께 실증 연구를 진행하고자 한다.

[참고문헌]

[1] Cisco, “11th annual Visual Networking Index: Global IP Traffic Forecast Update,” December. 2015

[2] Satoshi Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008.

[3] Ali Dorri, Salil S. Kanhere, Raja Judrak, Praveen Gauravaram, “Blockchain for IoT security and privacy: The case study of a smart home, Pervasive Computing and Communications Workshops,” 2017 IEEE International Conference, 17. March. 2017

블록체인 보안 프레임워크 개발을 위한 블록체인 보안 이슈에 대한 연구

노시완*, 이경현**

*부경대학교 정보보호학협동과정

**부경대학교 IT융합응용공학과

A Study on the Security Issue of Blockchain for Development of Blockchain Security Framework

Si-Wan Noh*, Kyong-Hyune Rhee**

*Interdisciplinary Program of Information Security, Graduate School, Pukyong National University

**Dept of IT Convergence and Application Engineering, Pukyong National University

요 약

최초의 가상화폐인 비트코인의 등장이후 기반기술인 블록체인에 대한 관심이 지속적으로 증가하면서 블록체인을 사용한 새로운 서비스 혹은 기존 시스템의 개선 등이 활발하게 이루어지고 있다. 하지만 아직 표준에 관한 논의가 이루어져있지 않아 블록체인의 보안 프레임워크의 연구의 필요성이 인식되고 있다. 본 논문에서는 보안 프레임워크 개발의 선결과제로서 블록체인의 주요 특성에 관한 공격들에 대해 조사하고 대응방안을 소개한다.

I. 서론

2008년 최초의 가상화폐인 비트코인[1]의 등장 이후 이더리움, 라이트코인 등, 2017년 기준 약 900개 이상의 가상화폐가 개발되어 전 세계적으로 사용되고 있다. 이들은 블록체인이라는 탈중앙화된 데이터구조를 기반으로 은행과 같은 중앙 신뢰기관의 참여 없이 Peer-to-Peer(P2P) 네트워크를 구성하여 상호간의 검증을 통해 거래를 하는 것이 가능하고 이전 블록의 해시 값을 포함하는 새로운 블록이 주기적으로 모든 네트워크 사용자들의 합의를 통해 사용자 개인의 블록체인 사본에 추가되어지므로 사용자들은 모두 동일한 체인을 보유할 수 있다.

비트코인의 등장 이후 블록체인에 대한 관심이 지속적으로 증가하면서 금융·의료·물류관리 등 다양한 분야에서 기존 시스템에 블록체인을 적용하거나 블록체인을 기반으로 하는 새로운 서비스가 등장하고 있다[2]. 하지만 비트코인에 대한 공격이 꾸준히 증가하고 블록체인기술에 대한 국제적인 표준 연구가 미흡하여 안전한 블록체인 기술의 적용을 위해서는 표준화된 블록체인 보안 프레임워크의 개발을 통해 블록체인 기반 시스템의 안전성을 보장하는 것이 필요하다. 이에 본 논문에서는 이러한 프레임워크의 연구에 앞서 현재 가장 대중적으로 사용되고 있으면서 오랜 기간 서비스되어 다양한 연구가 이루어져있는 비트코인 블록체인에 대한 공격사례를 분석하고 이에 대한 대응방안에 조사하였다.

II. 블록체인 보안 이슈

본 논문에서는 비트코인에서 발생하는 공격 중 사용자의 컴퓨터를 해킹하거나 거래소를 공격하여 비트코인 지갑의 비밀키를 탈취하는 공격하는 등 블록체인으로 인한 문제가 아닌 공격에 대해서는 다루지 않는다.

본 논문에서는 다음과 같이 비트코인만이 아

닌 블록체인 기술의 일반적인 특성을 기반으로 하는 공격과 이에 대한 분석을 수행한다.

- P2P 네트워크 : 블록체인 기술의 가장 큰 특징은 중앙 관리 기관이 존재하지 않는 탈중앙화에 있다. 블록체인 네트워크의 노드들은 연결된 이웃노드로부터 전달받은 블록을 다른 이웃노드에게 전파한다. 하지만 네트워크의 규모가 커질수록 모든 사용자에게 네트워크에서 발생한 이벤트가 전파되기까지 지연이 발생하거나 전파되지 않을 수도 있다.
- 분기(Fork) : 네트워크에서 모든 사용자들은 최초에 전체 블록체인의 사본을 생성하며 이후 네트워크에 전파되는 블록에 대한 검증을 통해 자신의 블록체인에 이를 추가할지를 결정한다. 이러한 블록이 동시에 여러 개가 전파되거나 네트워크 지연으로 인해 전파 도중 다른 블록이 발견될 경우 네트워크에는 서로 다른 여러 블록체인이 존재하고 합의 프로토콜에 의해 하나의 블록체인을 주 블록체인으로 결정하고 다른 블록체인은 폐기한다.

공격자는 위의 두 특성을 사용하여 네트워크에서 공격대상보다 특수한 우위(superiority)를 얻음으로서 공격 대상에게 피해를 입힐 수 있다.

2.1 블록 보류 공격(Block withholding)

블록은 채굴자(Miner)라 불리는 노드가 일정 시간마다 네트워크에 전파된 트랜잭션을 모아 블록을 생성한 뒤 특정한 암호학적 퍼즐의 해답과 함께 네트워크에 전파함으로서 추가된다. 비트코인에서 블록을 생성하여 주 체인에 추가할 경우 이에 대한 보상을 받을 수 있기 때문에 블록을 생성하고 네트워크에 전파하지 않는 것은 공격자에게 큰 이익이 되기 어렵다. 하지만 공격자가 네트워크의 다른 노드들보다 특수한 우위를 가지고 있을 경우 이를 사용하여 이중지불 공격(Double spending attack)의 성공확률을 크게 높이는 것이 가능하다. 다음은 비트코인의 이중지불을 위한 블록 보류 공격 중 하나인 Finney attack[3]의 시나리오이며 Fig.1은 이를 도식화한 것이다.

1) 공격자 C_d는 사전에 이중지불을 위한 트랜잭션 $T_{C_d}^{C_d}$를 포함한 블록 $B^{'}$을 생성한다.

2) 공격자는 생성한 블록을 네트워크에 전파하지 않고 정상적인 트랜잭션 $T_V^{C_d}$를 네트워크에 전파하고 블록B에 포함되어 판매자 V에게 확인될 때까지 대기한다.

3) 이후 사전에 생성한 블록 $B^{'}$을 네트워크에 전파하여 분기를 생성한다(네트워크에는 정상적인 트랜잭션 $T_V^{C_d}$을 포함한 분기 B_{fork}와 이중지불 트랜잭션 $T_{C_d}^{C_d}$을 포함한 분기 B'_{fork}가 존재한다).

4) 이후의 채굴 경쟁에서 B'_{fork}가 가장 긴 체인이 될 경우 B_{fork}에 포함된 정상적인 트랜잭션 $T_V^{C_d}$은 유효하지 않은 거래가 되어 공격자는 판매대금을 지불하지 않고도 판매물품을 얻을 수 있다.

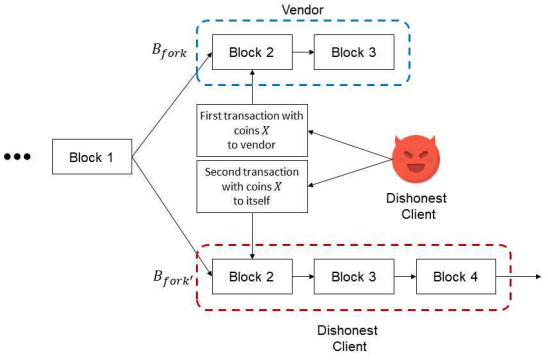


Figure 1. Double spending Attack

2.2 Eclipse 공격

비트코인에서 사용자는 이웃노드와 P2P 네트워크를 구성하여 블록체인 합의과정에 참여한다. 공격자는 공격대상의 P2P 네트워크 연결에 자신이 제어할 수 있는 노드가 선택되도록 유도하여 공격대상이 네트워크에 전파하는 모든 메시지에 대해 제어할 수 있는 Eclipse Attack[4]이 가능하다..

비트코인 클라이언트는 다음과 같은 별도의

테이블에 P2P연결을 위한 IP주소들을 저장한다.

- **tried table** : 이전에 연결한 기록이 있는 IP주소를 저장
- **new table** : 연결한 기록이 없는 새로운 IP 주소들을 저장

클라이언트는 매 시작 시 P2P 연결 노드들을 새로 선택하는데 Fig.2와 같이 p 확률로 tried table에서 주소를 선택하고 $1-p$ 확률로 new table에서 주소를 선택한다. 하지만 tried table에서의 선택은 최근 연결한 주소를 우선한다. 공격자는 사전에 공격대상과 정상적인 연결을 유지함으로써 이 확률을 증가시킬 수 있고 존재하지 않는 IP주소를 공격대상에게 전파하여 new table을 존재하지 않는 주소로 채워 특정한 IP주소가 공격대상에게 선택되도록 유도할 수 있다.

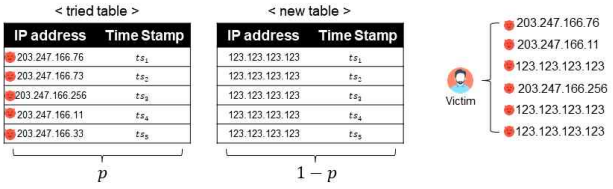


Figure 2. Eclipse Attack

III. 분석 및 대응

3.1 블록 보류 공격

블록 보류 공격은 공격자가 공격을 위해 사전에 미리 블록을 생성해두기 때문에 정상적인 트랜잭션을 포함한 블록이 생성되더라도 공격자에 의해 분기가 발생하므로 공격을 효과적으로 방지하기 어렵다. 네트워크의 노드들의 새로운 블록을 선택에 블록의 시간정보를 추가하여 최근에 생성된 블록을 우선적으로 선택하도록 하여 이에 대한 대응이 가능하다.

- **Random select[4]** : 블록을 수신한 상태에서 새로운 블록을 수신하더라도 폐기하지 않고 주변 노드에 전파한다. 노드는 수신한 블록들 중에서 랜덤하게 하나의 블록을 선택하여 체인에 추가한다.
- **Freshness preferred(FP)[5]** : 추가할 블록을 랜덤하게 선택하는 Random select 방식을 개선하여 블록에 포함된 타임스탬프가 더 최근인 블록을 추가한다. 노드들은 공격자의 블록을 먼저 수신하더라도 최근에 생성된 정상적인 블록을 우선적으로 체인에 추가한다.
- **Fork-resolving policy[6]** : 블록 전파에 걸리는 예상 시간 범위, 즉 현재 보유한 체인의 이전 블록의 생성시간 이후 새로운 블록의 생성시간의 차이의 범위 τ를 정한다. 공격에 의해 분기가 생겼을 경우 공격자의 블록의 경우 블록 생성시간이 해당 범위 외에 있을 가능성이 높으므로 해당 블록을 폐기한다.
- **ZeroBlock[7]** : 시간(Expectred time)동안 새로운 블록이 발견되지 않을 경우 ZeroBlock이라 부르는 빈 블록을 생성하여 체인에 추가한다. 네트워크의 노드들은 ZeroBlock을 우선적으로 추가하므로 공격자가 새로운 블록이 전파되기를 기다리고 있었을 경우 공격자는 공격자의 사전 채굴 블록을 무효화 시킬 수 있다.

3.2 Eclipse 공격

P2P네트워크에서 노드들은 전체 네트워크의 상황을 확인할 수 없기 때문에 자신의 전파한 메시지가 이후 네트워크에서 어떻게 다뤄지는지를 확인할 수 없다. 때문에 연결할 P2P 노드를 선택하는 과정에 있어 공격자가 개입할 여지를 줄이는 방법으로 대응할 수 있다.

- 동일한 여러 주소들은 테이블에 하나의 주소로 저장
- 최근 연결된 IP 주소의 연결을 우선하지 않음
- Ping 테스트를 이용하여 실제 존재하는 IP 주소인지 확인
- 테이블의 저장 공간을 확장

IV. 결론

본 논문에서는 블록체인 기술의 핵심적인 특징인 P2P 네트워크와 블록의 분기를 이용한 공격과 이에 대한 대응책을 살펴보았다. 이를 바탕으로 모든 블록체인 기반 서비스에 사용 가능한 보안 프레임워크의 개발을 통해 보다 신뢰성 있는 블록체인 기술의 활용을 기대한다.

Acknowledgement

이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2017-0-00156, 블록체인의 정보보호 프레임워크 및 평가 방법 개발)

[참고문헌]

[1] Satoshi Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” www.bitcoin.org/bitcoin.pdf,

[2] Cachin, C. (2016, July). Architecture of the Hyperledger blockchain fabric. In Workshop on Distributed Cryptocurrencies and Consensus Ledgers.

[3] Finney, H. (2011). Best practice for fast transaction acceptance-how high is the risk.

[4] Heilman, E., Kendler, A., Zohar, A., & Goldberg, S. (2015, August). Eclipse Attacks on Bitcoin's Peer-to-Peer Network. In USENIX Security Symposium (pp. 129-144).

[5] Eyal, I., & Sirer, E. G. (2014, March). Majority is not enough: Bitcoin mining is vulnerable. In International conference on financial cryptography and data security (pp. 436-454). Springer, Berlin, Heidelberg.

[6] Heilman, E. (2014, March). One weird trick to stop selfish miners: Fresh Bitcoins, a solution for the honest miner. In International Conference on Financial Cryptography and Data Security (pp. 161-162). Springer, Berlin, Heidelberg.

[7] Zhang, R., & Preneel, B. (2017, February). Publish or Perish: A Backward-Compatible Defense against Selfish Mining in Bitcoin. In Cryptographers' Track at the RSA Conference (pp. 277-292). Springer, Cham.

[8] Solat, S., & Potop-Butucaru, M. (2016). ZeroBlock: Preventing Selfish Mining in Bitcoin. arXiv preprint arXiv:1605.02435.

블록체인을 활용한 보안관제 적용 방안 연구

조영민*

*고려대학교 정보보호학과

A study on application of security control using Blockchain

Youngmin Cho*

*Division of information security, Korea University.

요 약

최근 가상화폐 열풍과 함께 블록체인 기술에 대한 관심이 여러 산업분야에서 나타나고 있다. 블록체인 기술은 참여자가 원장 정보를 공유하여 관리함으로써 편의성과 함께 안전성을 강조하며 활성화 되고 있다. 본 논문에서는 블록체인 개념과 기술원리, 한계점을 파악하고 이를 기업의 사이버 공격을 탐지하는 보안관제 영역에 적용할 수 있는 분야를 생각해본다. 또한 보안관제에 블록체인 기술을 적용할 경우 예상되는 문제점에 대해 살펴본다.

I. 서론

1.1 연구 배경 및 목적

최근 일상생활에서 쉽게 간편한 결제 수단의 이용이나, 인터넷 은행의 이용자 증가 소식 등을 접하면서 핀테크의 발전을 체감할 수 있다. 이러한 핀테크 중에 화두가 되고 있는 것이 바로 블록체인이다. 블록체인 기술이 이토록 화제가 되는 근본적인 배경 기술을 이해하고, 이를 보안관제 영역에 적용하고자 할 때 문제점은 없는지 주요 항목별로 살펴보고자 한다.

1.2 블록체인 개요

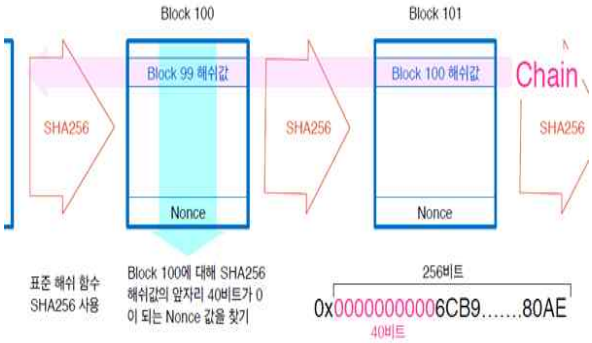
블록체인(block chain)은 거래 장부이며 원장 정보를 분산 저장하여 해킹을 어렵게하는 기술이다. 분산 데이터베이스의 한 형태로, 지속적으로 성장하는 데이터 기록 리스트로서 분산 노드의 운영자에 의한 임의 조작이 불가능하도록 고안되었다. 잘 알려진 블록체인의 응용사례는 암호화폐의 거래과정을 기록하는 탈중앙화된 전자장부로서 비트코인이 있다. 이 거래 기록은 의무적으로 암호화되고 블록체인 소프트웨어를 실행하는 컴퓨터상에서 운영된다. 비트코인을 비롯한 대부분의 암호화폐들이 블록체인 기술 형태에 기반하고 있다.[1]

최초의 블록체인은 1991년 Stuart Haber와 W.Scott Stornetta에 의해 고안되었고, 이후 많은 연구자들에 의해 진행되어오다가, 1998년 Nick Szabo가 분산디지털화폐 'Bit Gold' 매커니즘을 발표를 통해 개념이 정립되었고, 2000년에 이르러서 Stefan Konst가 현재의 일반적인 블록체인 이론(cryptographic secured chains)을 발표하며 널리 알려지게 되었다. 그러나 이 시점까지는 '블록체인'이라는 용어는 사용되지 않았으며, 현재의 비트코인을 발명한 것으로 알려진 Satoshi Nakamoto가 2008년 10월 발표자료에서 사용하면서 표준 용어로 사용되고 있다.[2]

1.3 블록체인 기술

블록체인은 'HashCash'의 개념을 토대로 발전된 기술이다. HashCash는 MD5, SHA1 등의 해시 함수를 바탕으로한 '작업 증명(Proof of

Work)'의 결과물이라고 할 수 있다. 작업증명은 문제를 해결하는데 일정 시간이 걸리는 작업을 연속해서 수행하도록 함으로써 공동작업에 참여한 사람이 선의를 가졌다는 것을 확인하는 절차를 뜻한다.[3]



[그림 1] Chain of Hash Values(출처 : 소프트웨어정책연구소 블록체인의 원리, 2016)

1.4 블록체인 방식

1) 무허가 블록체인(Permission - less Blockchain, 공용 블록체인)

거래를 증명하는 작업자들과 거래 이용자들이 특정 기관의 허락 없이 자유롭게 네트워크에 참여할 수 있는 프로토콜이며, 증명자와 이용자의 신상이 알려질 필요가 없고, 중앙 기관의 통제가 불가능하여 법적제도안에 들어가기 어렵다.(사례 : Bitcoin, Ethereum)

2) 반허가 블록체인(Semi-permissioned Blockchain, 컨소시엄 블록체인)

거래 증명자들은 네트워크의 허락 및 증명과정을 거치지만 거래는 누구나 자유롭게 사용이 가능한 프로토콜을 말하며, 한정된 탈중앙화적 요소를 가지는 동시에 법적 요구사항을 준수할 수 있다.(사례 : Ripple)

3) 허가 블록체인(Permissioned Blockchain, 사설 블록체인)

거래증명자와 송금/수신자 모두 개인정보를 제공하고 중앙기관의 허락하에 네트워크에 참여하는 방식이며, 완전히 중앙화된 방식으로 모든 거래를 컨트롤하지만 블록체인의 분산장부 장점을 이용한다.[4]

1.5 블록체인 기술과 한계

1) 암호화폐 기술(대표사례 : 비트코인)

화폐를 인터넷의 가상공간에서 보다 편리하게 사용하고자 가상화폐가 등장하였으나, 가상화폐는 운영사가 신뢰를 잃거나 서비스를 종료할 경우 함께 가치를 잃게되는 종속성을 가지게된다. 이러한 문제를 해결하고자 시도된 것이 암호화폐이다.

[암호화폐 한계]

· 가치변동성 : 수시로 가격이 변동하여 실물경제 활용이 어렵다.

· 해킹 : 비트코인은 해킹되지 않지만 비트코인을 이용한 서비스들이 해킹 대상이 된다.

2) 자산발행 기술(대표사례 : Coleredcoin)

컬러드코인은 비트코인 블록체인을 통해 현물자산을 디지털 형태로 표현하는 일종의 자산발행 레이어(Asset Registry Layer)이다.

[자산발행기술의 한계]

· 거래증명주체(validator) : 비트코인과 같은 퍼블릭 블록체인(Public blockchain)에서는 누구나 해당 거래를 증명하는 작업에 참여할 수 있다. 따라서 기업의 내부 이체정보 등이 공개될 수 있다.

· 저장공간비용 : 현재 상태를 체크하기 위해서는 자산의 최초 발행부터 모든 거래 역사를 검토할 수 있어야하기 때문에 수백기가바이트, 미래엔 테라바이트가 넘는 블록체인을 끊임없이 다운로드해야 한다.

3) 자산중심 기술(대표사례 : 리플)

리플은 분산원장 기반으로 작동하는 실시간 총액결제 시스템으로 전 세계에서 다수의 참여자가 발생시키는 대량의 결제를 빠르게 처리하는 역할을 한다.

[리플의 한계]

· 오픈소스 : 리플 블록체인은 오픈소스지만 소수 개발자만 코드 수정 권한을 갖고 있다. 이 때문에 해커가 코드를 분석해 공격

하기 쉽다.

· 중앙집중화 : 사실상 설립인과 리플랩스에서 절대 다수의 XRP를 소유하고 있다.

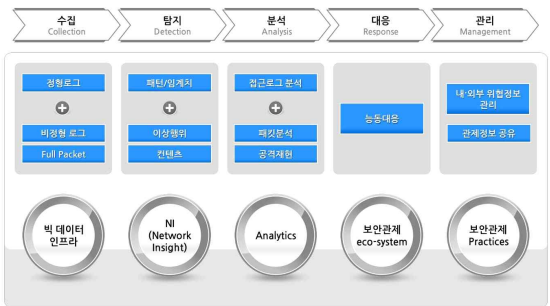
4) 확장응용 플랫폼(대표사례 : 이더리움)

비트코인은 내부적으로 스크립트 언어를 이용해 이체를 처리하는데, 의도적으로 30여 가지의 제한된 명령어만 지원하기 때문에 다양한 수준의 응용이 어려웠다. 이러한 제한성을 극복하고자 이더리움은 처음부터 튜링완전언어를 스마트컨트랙트(Smart Contract) 작성용 언어로 선택하였으며, Python과 유사한 구조의 Serpent, C++과 유사한 Solidity, Lisp과 유사한 LLL 등의 튜링완전언어를 통해 스마트컨트랙트 코딩을 지원한다. 이를 통해 일반적으로 블록체인 외부에서 구현하게 되는 많은 종류의 프로그램을 이더리움 블록체인 위에 애플리케이션으로 구현할 수 있으나, 그 만큼 경우의 수 또한 늘어나기 때문에 보안적으로 불완전하다.[5]

1.6 보안관제 개념

보안관제 업무란 관제 대상기관의 IT자원에서 발생하는 보안 이벤트, 로그, 네트워크 트래픽 등을 실시간으로 감시, 분석, 대응하여 사이버공격으로부터 보호하는 업무이다. 정보 자산에 대한 보안은 전문 집단이 수행하고, 관제 대상기관은 기관의 핵심 역량에 집중할 수 있는 선진화된 보안 서비스이다. 이것은 정보공유분석센터(ISAC, Information Sharing & Analysis Center)와 같은 의미를 갖고 있다.[6]

1.6 보안관제 프로세스 및 적용 개념



[그림 2] 보안관제 업무 프로세스(출처 : 이글루시큐리티)

보안관제에 블록체인 기술을 적용한다는 것은 기업내 임직원이 발생시키는 트래픽과 장비에서 발생되는 모든 로그를 특정 그룹(조직, 날짜, 장비분류 등)별로 원장의 개념으로 관리하여 위변조를 막고 외부로부터의 침입흔적을 보다 쉽게 발견하기 위함을 의미한다.

1.7 블록체인 기술의 보안관제 활용 분야

1) 외부 침입 인식

보안관제의 주목적은 외부 공격자로 부터의 침입 시도를 탐지하는 것이다. 이때 내부자의 정상적인 활동을 구별하고 관리할 수 있다면 쉽게 외부 침입을 인식할 수 있을 것이다. 내부자의 정상 활동내역을 블록으로 자산발행하여 분산원장으로 관리함으로써 구별이 가능하다.

2) 위변조 방지

보안관제는 내부 데이터의 위변조가 발생하지 못하도록 방어하는 역할 또한 담당하게 되는데, 이때 자산중심 기술을 통해 데이터의 위변조 여부를 감시 할 수 있을 것이다.

3) 사고 조사에 활용

외부로부터 공격이 발생하여 부득이 사고 조사로 이어지는 경우에도 활동에 대한 기록(주체, 시간, 크기 등)이 남아 이를 조사에 활용할 수 있고, 필요시 이를 증거로 제시할 수 있을 것이다.

1.8 블록체인 기술의 보안관제 적용 문제점

1) 처리 속도 문제

비트코인의 경우 1MB 정도 크기의 블록을 사용할 때 초 당 7개의 거래를 처리하는 것으로 알려져 있다. 보안관제를 위해서는 기업에서 발생하는 모든 트래픽, 로그 데이터를 처리하기 위해 기업 규모에 따라 수십기가바이트에서 수십테라바이트까지 이르는 데이터를 실시간 처리해야하는데 블록이 만들어지고 처리하는데 생기는 문제는 실시간 탐지 업무에 지장을 초래할 수 있다.

2) 정보 공개 문제

블록체인은 정보의 투명성을 추구하므로 블록체인에 개인정보가 평문으로 기록된다. 이는 기업의 정보 유출 또는 프라이버시 면에서는 심각한 문제가 아닐 수 없다. 보안관제는 기업 임직원의 대외비 데이터를 다룰 수 있으므로 반드시 암호화 기반의 구간 통신이 요구된다.

3) 보안성 문제

블록체인은 10년이 되지 않는 짧은 역사를 보유하고 있으며 최근들어 급격한 관심과 적용이 이루어지고 있다. 따라서 블록체인 소프트웨어 인프라에 대한 보안검증이 충분하다고 볼 수 없다. 최근에는 비트코인 거래소가 해킹을 당해 사용자가 피해를 입는 등의 부가적인 보안문제도 심각한 고려사항이다.

이를 보안관제에 적용하기 위해서는 블록체인을 활용하게되는 소프트웨어 설계 단계부터 보안 컨셉을 적용하여 보안성이 높은 결과물을 구현할 수 있어야하며, 이를 사용하게 되는 인프라 환경내의 추가적인 취약 요소는 없는지 살펴보는 선행작업이 필수적으로 요구된다.

4) 중앙집중화 현상 문제

블록체인이 추구하는 핵심 개념중의 하나가 분산 데이터베이스인데, 가트너 보고서에 따르면 중국의 4개 컴퓨팅 풀이 전체의 80%를 차지한다. 분산 데이터베이스를 추구하고 있으나 현실적으로는 중앙집중형 시스템을 그대로 따르고 있는 양상을 보이고 있다. 기업내 보안관제 데이터를 블록체인으로 관리할때 특정 그룹으로 컴퓨팅 풀이 집중될 경우 공격 포인트가 될 수 있으며 활용시 수행 성능 측면에서도 문제를 일으킬 수 있다.[7]

II. 결론

본 논문에서는 4차 산업혁명 시대를 선도하는 주요 기술 중의 하나인 블록체인 기술을 살펴보았다. 블록체인은 해쉬캐시의 아이디어를 발전시켜온 것으로 참여자의 기록을 분산 원장에 관리하여 배분함으로써 기본적으로 탈중앙화를 통해 해킹으로부터 안전성을 추구하고 있다. 세부적으로는 암호화폐 기술, 자산발행 기술, 자산중심 기술, 확장응용 플랫폼으로써의 기술활용 분류를 살펴보았으며 각각에 따른 한계점도 살펴보았다.

이러한 블록체인 기술을 보안관제에 적용하여 기업내 발생하는 이벤트와 트래픽 정보들을 모두 블록체인 원장 기술로 분산관리함으로써 위변조를 어렵도록하고 나아가 외부의 침입 흔적도 쉽게 파악하는데 적용한다면 그 효과는 매우 클 것이라고 예상된다. 다만, 이를 적용하기 위해서는 처리속도, 정보공개, 보안성, 중앙집중화 현상이라는 부분의 문제점들에 대한 선행 조치가 우선되어야 함을 파악할 수 있었다. 향후 이러한 선행조치 항목들에 대한 추가연구를 통해 블록체인 기술을 보안관제에 적용할 수 있는 심층연구를 진행하고자 한다.

[참고문헌]

[1] 미래금융연구센터, 블록체인의 이해, 2016

[2] “블록체인”, <네이버 지식백과>, 용어로 보는 IT, 2017

[3] 김석원, 소프트웨어정책연구소 블록체인의 원리, 2016

[4] 한전경제경영연구원, KEMRI 전력경제 Review(블록체인 방식 구분), 2017

[5] Finector, 블록체인 기술의 발전과정과 이해, 2016

[6] 이현도, 공공분야 보안관제 업무에 대한 평가지표 개발 연구, 2012, 5페이지

[7] 블록체인 아직 시기상조다, <디지털타임즈>, 2017.3.1

블록체인 기반 교내 전자투표 시스템 구축 요구사항 분석

정채린*, 이종호*, 김영우*, 조은아*, 성기정*, 김희영*, 이경현*

**부경대학교 IT융합응용공학과

Analysis of requirements for construction of electronic voting system based on blockchain

*Dept. of IT Convergence and Application Engineering,

Pukyong National University.

요 약

최근 4차 산업에 관심이 고조되면서 블록체인 기술은 금융거래를 넘어서서 다양한 분야에서 관심을 가지고 연구되고 있다. 중앙기관 없이 P2P방식을 이용한 블록체인은 마이너가 블록을 독립적으로 검증하고 보유하기 때문에 높은 신뢰성을 가진다. 또한 각 블록이 이전 블록의 해시를 참조하기 때문에 한 블록을 변경하고자 한다면 모든 블록을 변경해야 하므로 변조가 매우 어렵다. 이를 전자투표시스템에 적용 시 투표결과에 대한 투명성과 무결성이 확보된다. 본 논문에서는 블록체인을 사용한 교내전자투표 시스템 모델에서의 요구사항을 제시하고 관련된 개인정보 문제와 투표의 원칙 및 기반 위협을 분석하여 이에 대한 대응방안을 제시하였다.

Ⅰ. 서론

전자투표는 현재 많은 관심을 가지고 연구되고 있으며[1][2] 실제로 한국예탁결제원에서는 블록체인 기반의 전자투표시스템을 내년에 시범적으로 도입한다. 그러나 블록체인 기반의 투표시스템이 대규모 국가 선거에 적용되기에는 적용 선례가 적고 인터넷 투표의 모든 문제점을 다루지는 못하지만[3] 소규모 블록체인 기반의 전자투표시스템의 경우 기존 전자투표 시스템의 문제점인 무결성과 공개 검증 문제를 해결하였으며 현재 확장 가능한 시스템으로 소개되었다.

본 논문에서는 교내 전자투표시스템 구축 시 발생할 수 있는 신뢰성과 무결성, 공개검증 문제를 해결하기 위해 블록체인 기반의 교내 전자투표 시스템을 제안하며 시스템에서의 개인정보 문제 선거 원칙을 고려하여 보안 위협과 대응방안을 제시하고자 한다.

Ⅱ. 관련 연구

FollowMyVote[3]는 온라인상에서 구현되는 블록체인 기반의 전자투표 시스템이다. 이 시스템은 응용 프로그램을 다운받아 그들의 신원을 인증 받으면 투표권을 얻는다. 이 시스템은 투표자 익명의 유지를 위해 타원-곡선 암호화 방식을 사용하여 두개의 키 쌍을 사용한다. 하나는 신원 증명을 위한 것이고, 나머지는 투표를 위해 사용되었다.

BitCongress[3]는 VOTE 토큰이라 불리는 개인의 토큰을 활용해 후보자나 법 제정에 투표할 때 사용되며 선거가 끝나면 개인은 토큰을 반환하는 방식을 사용한다.

Ⅲ. 요구사항

3.1 사용자 개인정보 익명화 및 암호화

우리나라 정보통신망법에서 개인정보는 생존

하는 개인에 관한 정보로서 다른 정보와 쉽게 결합하여 알아볼 수 있는 정보를 포함한다. EU의 개인정보보호법 GDPR에서는 익명화된 정보는 법률적으로 '개인정보가 아닌 것'으로 취급한다. 대표적인 코인의 익명화 조치로는 코인 Mix서비스[4]와 개인정보 익명화 조치인 k-anonymity[5] 가 있으며 개인정보는 반드시 암호화하여 저장하여야 한다[6]. 또한 데이터 마스킹[7]등의 적용이 가능하다. 이상의 기법들을 고려하여 개인정보에 대한 익명화와 암호화를 교내 전자투표시스템에 적용하여야 한다.

3.2 Unlinkability

블록체인 기반의 비트코인의 경우 가명을 가지는 주소를 사용한다. 하지만 주소에 대한 추적이 가능[8]하므로 추가적인 익명화 조치가 필요하다. MIX 서비스[4]의 주요 아이디어는 바로 두 구성원(송신자, 수신자)간의 익명의 의사소통을 보장하여 추적을 어렵게 만드는 것이다. 하지만 Mix서비스는 third party를 사용하기 때문에 단일 실패 지점이 발생한다. 이러한 문제를 해결하기 위해 사용 되는 기법으로는 익명네트워크 Tor을 사용하는 coinjoin기법이 있다. 이에 추가로 DoS위험에 대한 보장 능력을 제공하는 기법이 mix network에서 주소를 분산시키는 coinshuffle 기법이다[8]. 위와 같은 기법을 포함하여 투표권자와 투표를 의미하는 트랜잭션 전송의 관계의 Unlinkabilty를 보장하는 기법을 고려하여야 한다.

Ⅳ. 블록체인 기반 교내전자투표

4.1 용어정의

[표1] 교내전자투표 시스템에서 용어정의

투표	투표권은 송금할 수 있는 화폐 및 소유권 가치를 가지는 코인 및 토큰이며 이후 논문에서는 비트코인의 최소금액을 투표권으로 정의하고, 투표 행위를 특정 후보자 소유의 비트코인 지갑으로 최소금액의 코인을 지불하는 트랜잭션을 발생시키는 행위로 정의한다. 득표수는 송금한 지갑 주소에서 발생한 하나 이상의 트랜잭션에 대해 하나의 득표로 집계한다.
투표 절차	투표권을 부여받기 전 투표자는 반드시 일정한 신원검증 절차를 거쳐야 한다. 신원을 교내에서 인증하는 기관을 "교내 인증기관" 이라 한다.
노드	투표권 보관을 위한 사용자의 개인 디바이스는 "노드"로 표시하며 노드에는 모바일 디바이스, PC등이 있다. 본 논문에서는 노드를 개인에게 귀속된 모바일 디바이스로 가정한다.
지갑	개인키를 생성하고 이에 대응하는 비트코인 주소를 가지는 실체로, 투표권 보관을 위해 사용하며 노드에 저장된다.
합의 알고리즘	합의알고리즘은 블록체인 기반 네트워크를 안전하게하기 위한 일련의 합의 규칙으로써 현재 투표시스템을 해킹 등의 공격으로부터 안전하게 하는 것이다. 논문에서는 외부 비트코인 채굴자를 이용하여 투표결과를 합의한다고 가정한다.
채굴	채굴이란 "합의 알고리즘"을 바탕으로 투표의 결과를 불록에 포함시키는 과정이며 "합의 알고리즘"과 동일하게 외부 비트코인 채굴자가 코인 베이스를 목적으로 채굴한다고 가정한다.

그 외 언급하지 않은 기본적인 용어는 비트코인 기반시스템의 역할과 기능에 따른다고 가정한다.

4.2 보안 요구사항 및 대응방안

◆ 지갑 생성을 위한 안전한 알고리즘 사용

투표를 위해 투표자는 후보자의 공개된 주소를 제공받으면 투표자는 단순히 자신의 개인키를 지갑에서 생성한다. 생성된 키는 투표의 용지로 간주하여 사용한다[9]. 개인키는 일반적으로 256비트 난수로 구성되고 실제로 Jenkins(1996)

가 설계한 ISAAC을 기반으로 한 CSPRNG (Cryptographically Secure Pseudo-Random Number Generator)를 이용해 생성되며[10], 이와 같은 방법으로 안전한 키 생성 알고리즘이 사용되어야 한다.

◆ 지갑 내 키 분실 및 도난

디바이스 내 지갑에 저장된 키 쌍은 악성코드로부터의 개인키 탈취, 장비의 물리적 분실과 도난 등에 취약하다. 위에 대해 키의 백업이나 해킹 등에 대한 보안이 요구된다. 교내 전자투표 시스템에서 학생들의 디바이스에 개인키 탈취 방지를 위해 다음과 같은 보안 시스템을 사용할 수 있다. Anti-Malware software[1], Firewall[2], 그리고 키 분실 시 복구를 위해 QR코드 혹은 종이 지갑, Trezor와 같은 하드웨어 지갑도 있다[11]. 또한 “지갑 가져오기” 사용[12]이 고려되어야 한다.

◆ 네트워크 상 안전성

투표자의 노드에서 투표를 위해 블록체인으로 보내는 트랜잭션은 반드시 신뢰된 망을 통해 전송되어야 하며 교내 전자 투표를 위해 인가된 AP나 안전한 사설망을 통해 투표를 진행하는 것이 고려되어야 한다.

◆ 민감한 개인정보 보호

투표를 위해 교내 학생들은 본인을 인증해야 한다. 이 과정에서 교내 인증기관 외에는 개인정보를 알 수 없어야 하며 인증기관의 경우도 최소한의 정보를 통해 투표자를 식별해야 한다. 이를 위해 투표 전 인증을 위한 개인정보 식별과 보호조치 방안[13]이 필요하다.

◆ 감사 및 투표부정 적발

교내 전자 투표 시스템 구현 시 교내라는 폐쇄적 특징 때문에 상위 집계기관의 투표수 조작 등 부정행위가 발생할 가능성이 있다. 이에 대응하여 투표자의 투표기록을 블록체인에 기록함으로써 투표의 무결성을 보장할 수 있어야 한다.

◆ 지갑주소와 투표자간의 익명성

지갑의 소유자가 신원인증 후 투표권(비트코인)을 부여받거나, 투표후보자에게 투표권을 행사하는 경우 각 참여자는 역할에 따라 투표정보 혹은 인증정보 중 일부만을 확인 가능하여 지갑주소와 투표자간의 관계에 대해 식별이 불가능하도록 하여야 한다.[4][8]

◆ 투표권에 대한 외압 혹은 부정 청탁방지

투표권자는 투표현장에서 반드시 외압에 관여 받지 않고 투표가 가능하여야하며 금전적 목적을 위해 자신의 의견과는 다른 후보자를 투표하지 않도록 본인인증과 서명이 가능한 암호학적 프리미티브나 부정청탁의 이익보다 상대적으로 큰 투표 인센티브 시스템을 고려하여 구성하여야 한다.

Ⅴ. 결론

본 논문에서는 블록체인 시스템을 교내 투표 시스템에 적용하였을 때 수반될 수 있는 문제점과 그에 대한 대응방안을 제시하였다. 보안에 민감한 학생들의 개인정보의 처리를 고려하였으며 분실, 도난 될 수 있는 지갑 내 키 관리 문제에 대해서는 안전한 알고리즘 사용을 요구한다. 또한 투표결과 조작 방지를 위해 외부 채굴자에게 투표결과 집계를 위임하였다. 일반적인 전자투표시스템에 비해 블록체인 기반의 전자투표는 신뢰성, 무결성이 높으며 공개검증도 가능하다. 향후 본 논문에서 고려된 사항을 바탕으로 블록체인 기반의 교내 전자투표 구축에 활용할 수 있을 것이다.

1) 실시간 환경에서 매우 효과적으로 작동하고 외부의 위협을 검색하며 서명 인증을 통해 malware 감염을 차단

2) 일련의 규칙을 기반으로 네트워크 전송을 허용하거나 거부하는 장치 또는 소프트웨어이며 합법적인 전송되도록 하면서 무단 액세스로부터 네트워크를 보호

[참고문헌]

[1] Y. Wang and A. Kogan, ”Designing

Privacy-preserving Blockchain based Accounting Information Systems", SSRN1, May, 2017
[2] 장준희, "'블록체인 활용 전자투표' 주요사례 및 시사점",NIA-2017-5,National Information Agency, April 2017.
[3] F.ciazzo and M.chow,"A Block-chain implemented voting system", 2016
[4] E. Heilman, F. Baldimtsi, and S. Goldberg , "Blindly Signed Contracts : Anonymous On-Blockchain and Off-Blockchain Bitcoin Transactions", International Conference on Financial Cryptography and Data Security, pp.43-60, Feb. 2016
[5] 양현철, 빅 데이터 활용을 위한 개인정보 기술 안내서(ver1.0), 미래창조과학부
[6] 개인정보의 암호화 조치 안내서, 한국인터넷진흥원, Jan 2017
[7] Iain W.Fergusson.,"SYSTEM AND METHOD FOR DYNAMIC DATA MASKING", US 2006/0074897 A1, June. 2006
[8] T. Ruffing, P. Moreno-Sanchez, and A. Kate, "CoinShuffle: Practical Decentralized Coin Mixing for Bitcoin", Computer Security-ESORICS 2014, LNCS 8713, pp.345-364, 2014
[9] Kibin Lee, J.I. James, T.G. Ejeta, Hyoung Joong Kim, "Electronic Voting Service Using Block-Chain", Journal of Digital Forensics, Security and Law, vol.11, no.2, pp.123-136, 2016.
[10] Antonopoulos, Andreas M., Mastering Bitcoin : Unlocking Digital Cryptocurrencies. Oreilly & Associates Inc, Aug. 2014
[11] Eric Mu (15 October 2014). "Meet Trezor, A Bitcoin Safe That Fits Into Your Pocket". Forbes Asia. Forbes. Archived from the original on 24 October 2014. Retrieved 31 Oct. 2014.
[12] S. Eskandari, D. Barrera, E. Stobert, and J. Clark, "A First Look at the Usability of Bitcoin Key Management", Workshop on Usable Security(USEC), 2015
[13] Frost & Sullivan, "World Data Leakage Prevention Market", ND34D-74, Frost & Sullivan, United States, 2008.

2017 한국정보보호학회 동계학술대회

A3 자동차보안

임 강 빈 (순천향대)

- **자율주행 차량에서 보안 위협 분석을 통한 내/외부네트워크 안전성 향상 방안 연구**
  최석준, 곽진 (아주대학교)
- **자동차 보안 모델 HEAVENS 구체화를 위한 Automotive Attack DB 연구**
  이세한, 고광우, 안광현, 박희성, 한철규, 심준보 (KITRI BoB), 이현정 (페스카로)
- **인터페이스 중심의 최신 자동차 공격 사례 분류**
  송윤근, 이유식 (ESCRYPT)
- **차량 대상 사이버 공격의 위험 평가 동향 연구**
  서현진, 이동훈 (고려대학교)

자율주행 차량에서 보안 위협 분석을 통한 내/외부네트워크 안전성 향상 방안 연구

최석준*, 곽 진**

*아주대학교 컴퓨터공학과 정보보호응용및보증연구실

**아주대학교 사이버보안학과

A Study on the Improvement of Security through Analysis of Security Threats on Connected Vehicle

Suk-June Choi*, Jin Kwak**

*ISAA Lab., Department of Computer Engineering, Ajou University.

**Department of Cyber Security, Ajou University.

요 약

최근 4차 산업의 핵심 기술로 자율주행차량에 대한 기술이 활발히 연구가 되고 있다. 자율주행 차량이 실현되기 위해서는 차량과 차량, 주변 인프라 등의 통신으로 주변 도로 상황과 같은 정보를 주고 받으며, 차량 내부적으로 주고받는 도로상황에 대한 데이터 처리가 필수적이다. 하지만 공격자로부터 주고받는 메시지에 대한 변조 등이 이루어질 경우, 차량은 잘못된 처리를 하여, 결과적으로 심각한 인명피해가 발생할 수 있다. 따라서, 자율주행 차량을 위해 안전한 통신이 필요하며, 이는 차량 내/외부 네트워크에 대한 통합적인 보안이 고려되어야 한다. 본 논문에서는 차량에 사용되는 메시지에 대한 무결성, 기밀성 등을 보장해주기 위해 필요한 차량 보안 기술에 대해 나열하고, 차량 내/외부에서 안전한 통신을 위한 보안 요구사항을 도출한다.

I. 서론

4차 산업 혁명으로 인한 기술 발전으로 인해 최근 자율주행 차량에 대한 연구가 활발히 진행되고 있다[1]. 자율주행 차량이 실현되기 위해선 기본적으로 주변 도로상황에 대한 정보 등의 데이터를 빠르게 전송하고 처리하는 것이 핵심이라 할 수 있다. 또한 자율주행 차량 서비스의 가용성을 보장해주기 위해선, 차량 내/외부 네트워크로부터 발생할 수 있는 보안위협들에 대한 안전성을 보장하는 것이 최우선이라 할 수있다. 만약 도로주변상황에 대한 데이터가 보장되지 않는다면, 보안위협만 아니라, 심각한 인명피해가 발생할 수 있다. 따라서 본 논문에서는 2장에서는 차량 보안 기술 등과 같은 관련 연구에 대해 분석을 진행하고, 3장에서는 보안위협 분석을 진행한 후, 4장에서 차량 내/외부에서 필요한 보안 요구사항에 대해 제안하고, 5장에서 결론으로 끝을 맺는다.

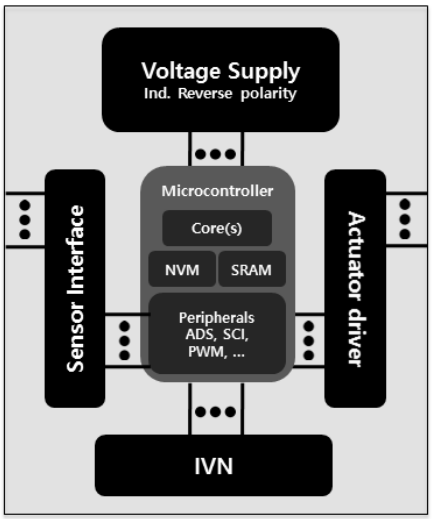
본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 대학ICT연구센터육성 지원사업의 연구결과로 수행되었음(IITP-2017-2015-0-00403)

II. 관련 연구

2.1 차량 내부네트워크구조(IVN)

차량의 내부네트워크는 IVN(In-Vehicle Networking)이라고도 한다. 차량 내부에는 다수의 ECU(Electronic Control Unit)가 포함되어 있으며, 각종 차량 장비를 제어하는 ECU는 [그림 1]과 같다. 이와 같이 각각의 ECU마다 전력을 제공하는 Voltage Supply가 존재하며, ECU에 할당된 차량 장비를 제어를 하는 Actuator driver와 차량 센서에 대한 Sensor Interface가 연결되어 있다. 또한 차량은 CAN통신을 이용해 내부 버스에 브로드 캐스팅으로 메시지를 전송하는 방식이며, 자신에 해당하는 메시지만

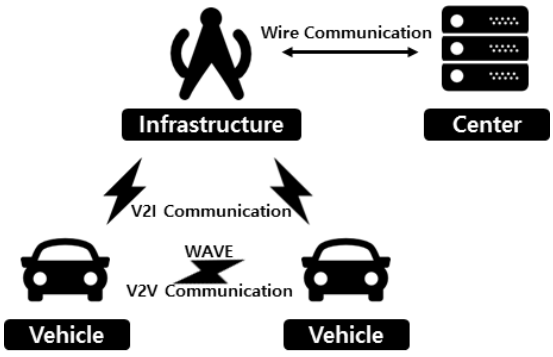
ECU 받아 처리하는 방식이다. 이를 통해 차량의 각종 장비(엔진, 자동변속기, ABS 등)를 제어하게 된다[2].



[그림 1] 차량 내부네트워크

2.2 차량 외부네트워크구조

차량 외부네트워크는 [그림 2]와 같이 2 가지로 V2V(Vehicle to Vehicle), V2I(Vehicle to Infrastructure)가 존재한다. 차량과 차량은 무선통신 기술(WAVE)을 통해 차량의 위치 등의 정보를 주고받으며, 인프라로부터 주변 도로상황 등의 정보를 주고받는다.



[그림 2] 차량 외부네트워크

III. 차량 보안 위협 분석

3.1 차량 내부네트워크 보안 위협

차량 내부 네트워크에서 발생할 수 있는 보안위협을 차량 내부 네트워크, 차량과 차량 간 통신, 차량과 주변인프라 간 통신과 같이 분류하여 나타냈으며, [표 1]과 같다.

[표 1] 내부네트워크 보안위협

No	분류	보안위협
T1	IVN	- 차량 내부의 OBD-II를 통해 차량 내부의 악의적인 메시지를 전송하여 각각의 ECU를 제어할 수 있다. (인증)
T2	IVN	- OBD-II, 차량 내부 버스에 대한 대량 메시지를 주입하여, 서비스에 대한 거부 공격을 할 수 있다. (가용성)
T3	IVN	- 차량 점검 ECU(타이어 기압 정보 등)에 대한 위·변조 메시지를 보내어, 사용자로 하여금 불필요한 차량정기점검을 유발할 수 있다. (무결성)

3.2 차량 외부네트워크 보안 위협

차량 외부 네트워크에서 발생할 수 있는 보안위협을 분석하였으며, [표 2]와 같다[3].

[표 2] 외부네트워크 보안위협

No	분류	보안위협
T4	V2I	- 주변 도로상황과 관련된 메시지를 위·변조하여 잘못된 정보를 제공함으로써, 차량을 제어할 수 있다. (무결성)
T5	V2V	- 차량과 차량이 주고받는 메시지를 위·변조하여 잘못된 정보(위치, 속도 등)를 전달할 수 있다. (무결성)
T6	V2I	- 공격자는 특정 차량에 대한 메시지 위·변조, 서비스 거부 등의 공격이 가능하다. (인증)

IV. 차량 보안 요구 사항

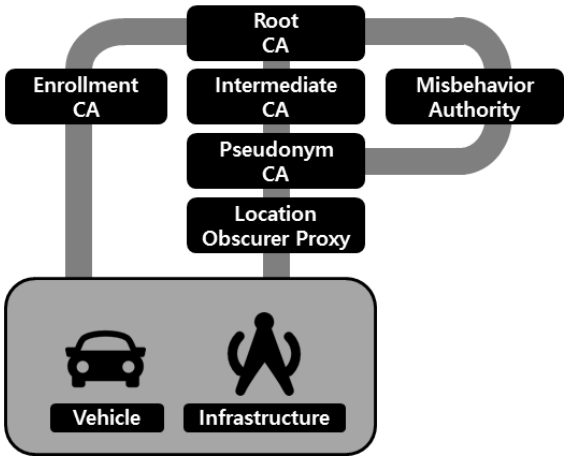
4.1 차량 내/외부네트워크 보안기술

❑ HSM(Hardware Security Module)

HSM은 차량에 탑재되는 보안 모듈로, 내부 네트워크 내의 메시지 변조 등과 같은 하드웨어공격을 방지하여 차량 내부네트워크에 대한 무결성을 제공하는 기능을 한다. NHTSA(미국 도로교통부)에서는 차량의 모든 장비에서 해당 모듈을 사용하도록 요구한다[4].

❑ SCMS(Security Credential Management System)

SCMS는 차량의 외부네트워크인 V2X, V2I에 대한 안전성을 제공해주는 시스템을 말한다. SCMS의 구조는 [그림 3]과 같다[5].



[그림 3] SCMS 구조

❑ SCMS를 통한 차량 보안 기술

Step 1. 등록인증서 발급

차량은 등록 인증기관으로부터 등록인증서를 발급받는다. 이를 통해 정상적인 차량임을 인증받을 수 있다.

Step 2. 익명인증서 발급

차량은 인증서를 통해 익명인증서를 발급 받는다.

Step 3. 오작동 보고

만약 비정상행위가 발견된다면 익명인증서를 폐기한다.

Step 4. 전자서명된 메시지 전송

해당 익명인증서를 통해 전자서명된 메시지를 전송하며, 이를 통해 차량에 대한 익명성, 무결성 등을 보장할 수 있다.

❑ SCMS를 통한 인프라 보안 기술

Step 1. 등록인증서 발급

인프라는 등록인증기관에 대한 승인을 받기 위해 등록인증서를 발급 받는다.이를 통해 정상적인 인프라임을 인증 받을 수 있다.

Step 2. 전자서명된 메시지 전송

인프라는 응용 인증서를 통해 무선 메시지에 대한 서명을 할 수있으며, 이는 메시지에 대한 무결성을 보장할 수 있다. SCMS의 인터페이스

[표 3] 차량 내/외부네트워크 보안요구사항 분석

보안요구사항 / 모듈	T1→R1	T2→R2	T3→R3	T4→R4	T5→R5	T6→R6
HSM 기술 적용	✓	✓	✓			
SCMS 기반 차량 보안기술 적용					✓	
SCMS 기반 인프라 보안기술 적용				✓		✓

로부터 등록인증서를 발급하여 통신을 진행하며, 인프라에 대한 인증을 할 수 있다.

[표 3]은 차량 내/외부 네트워크 보안위협을 대응하기 위해 필요한 보안 요구사항을 도출하였다.

4.2 차량 내부네트워크 안전성 분석

❑ R1. 사용자 인증과 접근제어

HSM을 통해, OBD를 통해 내부로 들어오는 메시지에 대한 인증을 제공할 수 있으며, 이를 통해 악의적인 메시지가 ECU에 접근이 불가능하다. 따라서 악의적인 메시지에 대한 안전성을 보장할 수 있다.

❑ R2. 가용성 보장

HSM을 통해, 외부로부터 접근이 가능한 정상적이지 경로를 통한 메시지를 차단이 가능하며, 이를 통해 대량 메시지에 대한 가용성을 보장할 수 있다.

❑ R3. 무결성 보장

HSM을 통해 위·변조 메시지에 대한 메시지가 해당 ECU에 접근하지 못하도록 무결성을 보장받을 수 있다.

4.3 차량 외부네트워크 안전성 분석

❑ R4. 무결성 보장

SCMS 기반 보안기술을 적용할 경우, 메시지를 전송할 때 전자서명을 하여 메시지를 전송하기 때문에 정상적이지 않은 인프라에 대한 인증이 가능하다. 따라서, 정상적으로 서명되지 않은 메시지에 대한 무결성을 보장할 수 있다.

❑ R5. 무결성 보장

SCMS 기반 보안기술을 적용할 경우 차량은 익명인증서를 통해 메시지에 대한 전자서명을 하기 때문에 위·변조된 메시지에 대한 식별이 가능하다. 이를 통해 무결성을 보장할 수 있다.

❑ R6. 사용자 인증

SCMS 기반 보안기술을 적용할 경우, 비정상적인 인프라의 경우 등록인증기관으로부터 인증을 받을 수 없다. 또한, 차량은 익명인증서를 이용해 메시지를 전송하기 때문에 특정 차량을 식별할 수 없다. 따라서, 비정상적인 인프라에 대한 무결성을 보장할 수 있다.

V.결론

자율주행차량이 실현되기 위해서는 서비스를 제공하기 이전에 차량 내/외부적인 네트워크에 대한 보안기술 적용이 선행되어져야한다. 본 논문에서는 자율주행차량에서 발생할 수 있는 차량 내/외부 네트워크에서 발생할 수 있는 보안위협을 분석하였다. 또한 차량 내/외부네트워크에서 안전한 통신을 보장해주기 위한 보안 기술에 대한 분석을 진행하였으며, 이를 통해 차량 보안위협에 대응할 수 있도록 차량 보안 기술을 적용하였다. 또한 차량 내/외부네트워크에서 발생할 수 있는 보안위협에 대응방안으로 차량 내/외부네트워크 보안기술을 적용한 보안 요구사항을 도출하여, 안전성 분석에 대한 진행을 하였다.

향후에는 해당 보안기술을 적용하여, 차량 내/외부 네트워크에 대한 보안 통합프레임워크를 설계할 것이며, 이에 대한 안전성 분석에 대한 연구를 진행할 예정이다.

[참고문헌]

[1] R. Coppola, M. Morisio, “Connected Car: Technologies, Issues, Future Trends”, ACM Computing Surveys, October. 2016.

[2] P. Kleberger, Pierre, T. Olovsson, and E. Jonsson. “Security aspects of the in-vehicle network in the connected car”, Intelligent Vehicles Symposium (IV), IEEE, 2011.

[3] S. Peter, B. Norbert, K. Frank, D. Stefan, P. Panos, S. Christian, “Security Requirements of VSA”, Preserve, June. 2011.

[4] L. Evans, J. Phillip, L. Arden, Bement, “FIPS PUB 140-2 : Security Requirements for Cryptographic Modules“, Federal Information Processing Standards Publication, Janaury. 1994.

[5] J. Petit, F. Schaub, M. Feiri, F. Kargl, “Pseudonym Schemes in Vehicular Networks: A Survey”, IEEE Communication survey & Tutorials, 2015.

자동차 보안 모델 HEAVENS 구체화를 위한 Automotive Attack DB 연구

이세한*, 고광우*, 안광현*, 박희성*, 한철규*, 심준보*, 이현정**

*KITRI Best of the Best

**페스카로 보안플랫폼개발팀

A Study on Automotive Attack DB for Concretization of HEAVENS, A Car Security Model

Se-Han Lee*, Kwang-Woo Go*, Gwang-Hyun Ahn*, Hee-Sung Park*, Cheol-Kyu Han*, Jun-Bo Shim*, Hyun-Jung Lee**

*Best of the Best, KITRI

**Security Platform Development Team, Fescaro

요 약

최근 Connected Car가 등장하면서 자동차가 정보통신 기술에 의존하는 경향이 커지고 있다. 하지만 2013년 Charlie Miller와 Chris Valasek이 DEFCON21 컨퍼런스에서 “Adventures in Automotive Networks and Control Units” 발표와 2015년 원격 자동차 해킹 Demo 공개로 인해 크라이슬러가 취약점 발생 차량에 대한 리콜을 실시하는 등 더 이상 자동차도 해킹의 안전지대가 아님이 밝혀졌다. 이로 인해 자동차의 기능적 안전뿐만 아니라 보안적인 요소도 고려된 HEAVENS(HEAling Vulnerabilities to ENhance Software, Security and Safety, 소프트웨어, 보안 및 안전성 향상을 위한 취약점 점검) 모델이 등장하였다. 그러나 HEAVENS 모델에는 간략한 프로세스만 있을 뿐 프로세스에 대한 상세한 절차 및 Activity가 존재하지 않아 실제 차량의 보안 취약점 점검에 적용하기에는 어려움이 있다. 따라서 본 논문은 HEAVENS 모델을 구체화하기 위해 Automotive Attack DB를 제시하고자 한다.

I. 서론

최근 자동차 산업이 발전하고, 자동차에 수많은 IT 기술들이 같이 이용되면서 Smart Car가 등장하고, 인공지능이 접목된 Connected Car가 등장하게 되었다.

그러나 2013년부터 최근까지 자동차에 대한 원격 해킹 기술 및 다양한 자동차 해킹 공격 시나리오들이 등장하면서 자동차도 더 이상 해킹 공격으로부터 안전하지 않다는 것이 밝혀졌으며 수많은 자동차 제조회사는 이를 해결하기 위해 자동차의 기능적 안전뿐만 아니라 보안도 고려하여 설계를 하고자 연구를 진행하고 있다[1].

자동차의 기능적 안전과 보안을 종합적으로 고려한 취약점 점검 모델인 HEAVENS가 등장하였지만[2], 간략한 프로세스만 존재하고 각 프로세스에 대한 상세한 절차 및 Activity가 존재하지 않는다.

본 논문에서는 아직 구체적이지 않은 HEAVENS 모델에 대하여 구체화를 위한 Automotive Attack DB를 연구하고 제시함으로서 향후 HEAVENS 모델의 발전 및 자동차 보안 연구에 도움을 주고자 한다.

II. 본론

2.1 Automotive Attack DB 정의

Automotive Attack DB는 자동차 내/외부에서 발생할 수 있는 취약점에 대한 Attack Vector/Point와 공격기법, 공격 프로세스에 따

라 공격을 수행한 결과 및 대응방안 등을 체계적으로 정리한 자동차 보안 취약점 점검 데이터베이스이다.

2.2 Automotive Attack DB 구성



그림 1. Automotive Attack DB 일부

Automotive Attack DB는 다음과 같은 항목들로 구분되어 있다. 각 항목들은 번호, 구분, 공격대상, 취약점, Attack Vector/Point, 공격기법, 공격프로세스, 공격 데이터, 결과, 연식(모델)에 따른 성공/실패 여부, 출처, 사용한 장비, 대응방안, 사고사례로 구분되어있다. Automotive Attack DB의 가로줄 한 줄은 각각 독립된 한 개의 Attack Case로 각 구분자에 관한 설명은 아래와 같다.

번호는 Automotive Attack DB의 각 한 개의 Attack case에 대한 구분 번호이다.

구분은 각 Attack Case의 공격벡터에 대한 분류로 키, 통신, USB, QNX, CAN으로 각 Attack Case별 해당하는 공격벡터 이름으로 분류한다.

공격 대상은 Attack Case의 공격벡터의 대상에 대한 분류이다. 위의 구분에 비해 좀 더 세분화된 분류를 제공한다.

취약점은 각 Attack Case에서 공격 대상을 공격하는데 사용된 취약점이다.

Attack Vector/Point는 Attack Case에서 실제 취약점이 발생하는 컴포넌트로 찰리 밀러와 크리스 발라섹의 Remote Exploitation of an Unaltered Passenger Vehicle에서 사용된 Attack Vector/Point 분류를 기반으로 하여 취약점이 발생하는 컴포넌트를 분류한다.

공격기법은 Attack Case에서 Attack Vector/Point의 취약점을 공격하는데 사용된 공격기법이 기술된다.

공격 프로세스는 자세한 공격의 과정을 기술하는 부분이다.

공격 데이터는 Attack Case에서 실제 공격이 이루어 질 때 사용되는 페이로드 및 각종 정보를 기술한다. (예: CAN Injection 공격에서 CAN StdID: 0x579 CAN Data: 0x57 0xFF 0x00 0x00 0x45 0x34 0x77 0x66)

결과는 Attack Case의 공격 프로세서 실시 후 결과를 기술한다.

연식(모델)에 따라 성공/실패 여부는 각 제조사별 자동차 모델에 따라 Attack Case에서 실시되는 공격의 성공여부를 기록한다.

출처는 Attack Case의 정보를 어떠한 경로를 통해서 획득하였는지 기록한다.

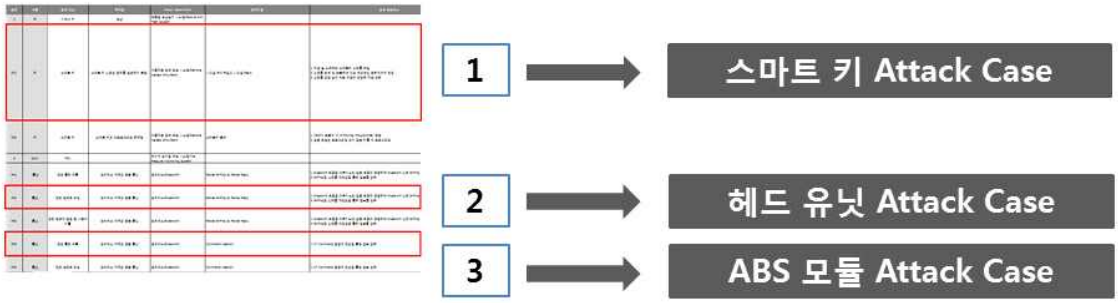
사용한 장비는 Attack Case에서 공격 및 정보 수집 등에 사용된 장비 이름을 기록하고 각 장비들이 공격 프로세스의 어떤 부분과 어떤 과정에 사용 되었는지 기술한다.

대응 방안은 Attack Case에서 실시된 공격에 대한 대응 방안을 기술한다.

사고사례는 실제 Attack Case에서 사용된 방식과 유사한 혹은 같은 방식의 침해 사례를 인터넷이나 신문기사 등을 통해 실제 사고 사례를 수집한다.

Attack Vector/Point
헤드유닛 및 AVN(Audio, Video, Network) 시스템 및 인포테인먼트 시스템
파일시스템 및 서비스(File System and Services) - Uconnect 유닛에서 NAND 플래시에 포함된 여러 파일 시스템
QNX 환경(QNX Environment) - 지프 체로키의 ARM 프로세서에 사용되는 운영체제
CAN 네트워크

그림 2. Attack Vector/Point 일부



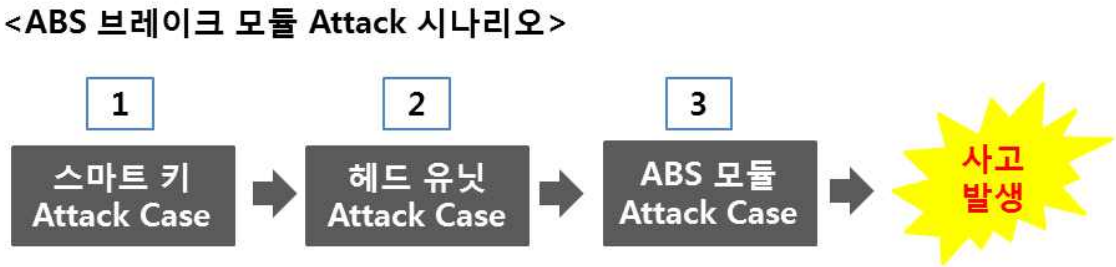


그림 3. Attack Case 조합을 이용한 Attack 시나리오 예시

Attack Surface란 [그림 2]과 같은 허가받지 않은 사용자가 악용 가능한 여러 Attack Vector/Point들의 집합으로 Automotive Attack DB에서 Attack Vector/Point의 세로축 조합으로 이루어져 있다.



번호	구분	Attack Vector/Point	공격기법	
8-3)	USB	헤드유닛 및 AVN(Audio, Video, Network) 시스템 및 인포테이먼트 시스템	USB Port Fuzzing	1. Kernel레벨에 전환이 가능에 2. fuzzing파일 3. 구현한 fuzz

그림 4. Attack Case 일부

Attack Case란 [그림 4]와 같이 하나의 독립된 공격 케이스로 Automotive Attack DB에서 번호로 구분되는 가로줄 한 줄을 가리키며, Attack Vector/Point를 대상으로 취약점을 공격기법을 이용하여 공격프로세스를 수행하는 하나의 Case를 나타낸다.

Attack 시나리오란 Attack Case 조합을 통해 도출할 수 있는 일련의 연속된 공격과정을 시나리오처럼 구성한 것을 의미하며 Attack Case의 조합 방법 및 순서 혹은 개수에 따라 다양한 방법으로 조합이 가능하다.

[그림 3]을 통해 보면 스마트키 Attack Case, 헤드 유닛 Attack Case, ABS 브레이크 모듈 Attack Case이렇게 3가지의 Attack Case를 조합하여 하나의 시나리오를 구성하였다. 먼저 스마트키를 공격 하여 차량 내부로 진입한 후 차량의 헤드 유닛을 공격하여 헤드 유닛의 제어권한을 탈취한다. 그 이후 헤드 유닛과 차량의 CAN Network 연결부를 통해 진단용 CAN 메시지 Injection 공격을 통해 ABS 브레이크 모듈을 off 상태로 전환시킨다. 그리고 공격 목표인 운전자가 공격당한 차량에 탑승한 후 운전을 하면 헤드 유닛의 라디오 볼륨을 갑자기 최대로 올려서 운전자를 당황하게 만들고 off상태로 꺼져있는 ABS 브레이크 모듈에 의해 브레이크가 정상적으로 작동하지 않아 사고를 유발하는 시나리오를 도출 할 수 있다.

2.3 HEAVENS 구체화를 위한 Automotive Attack DB 활용

HEAVENS 프로젝트는 소프트웨어 집약적인 자동차 시스템의 보안 취약점들을 확인하고 소프트웨어 보안 테스트를 수행하는 도구들을 다루는 방법론을 정의한다[2]. 그 방법론으로 제시된 HEAVENS 보안 모델은 평가 목적을 정의하고, 그 뒤에 위협 분석, 위험성 평가, 보안 요구사항 식별의 세 과정을 거친다[3]. 하지만

HEAVENS 보안 모델은 각각의 과정에 필요한 것이 무엇인지와 과정의 산출물이 무엇인지만을 설명할 뿐 구체적으로 어떤 과정을 통해서 해야 하는지는 설명하지 않는다. 그래서 HEAVENS 프로세스의 각각의 단계를 Automotive Attack DB를 활용하여 구체화 하고자 한다. 자세한 방법은 다음과 같다.

2.3.1 잠재적 위협 분석 구체화

잠재적 위협 분석 단계는 평가 목적 정의 단계에서 TOE(Target of Evaluation)가 설정된 이후, 해당 TOE에 해당하는 잠재적 위협을 분석하는 단계이다.

이를 실천하기 위해서 해당 TOE에 해당하는 공격대상, 공격기법, 공격 프로세스 등이 포함된 항목들을 Automotive Attack DB에서 찾는다. 그 뒤 해당 항목을 살펴 그에 해당하는 취약점, 결과 항목 등을 통해 잠재적 위협을 분석한다. 이것이 본 논문에서 제시하는 구체적 실행 방안이다.

2.3.2 위험성 평가 구체화

위험성 평가 단계는 위협 분석 과정에서 산출된 내용에 위협 수준과 파급성을 고려하여 위험성을 평가하는 단계이다.

이를 실천하기 위해서 Automotive Attack DB의 결과, 사고사례 항목 등을 활용할 수 있다.

2.3.3 보안 요구사항 식별 구체화

보안 요구사항 식별 단계는 앞서 위협 분석 과정에서 찾은 위협과 보안 적 속성, 그리고 위험성 평가 과정을 통해 판단한 위험성을 통해 보안 요구사항을 식별하는 단계이다.

이 과정에서 Automotive Attack DB의 대응 방안 항목이 그 자체로 보안 요구사항이 될 것이다. 또한 Automotive Attack DB의 공격 데이터 항목을 이용하여 이 요구사항이 제대로 이행되었는지 테스트할 수도 있다.

2.3.4 HEAVENS 구체화 예시

예시를 통해 알아보자. [그림 1]을 활용해 설명하기 위해 Gateway ECU를 TOE로 설정했다. 공격대상이 Gateway ECU인 것을 Automotive Attack DB에서 찾는다. Command Injection, malicious upload, CAN Packet Injection 등의 잠재적 위협을 확인할 수 있다. 그 위협들의 위험성을 평가할 차례인데, 그림에 나오지 않은 부분에 있는 결과, 사고사례 등을 위험성 평가에 활용할 수 있다. 예시 그림에는 보이지 않지만 대응방안 항목을 살펴본 결과 Gateway ECU를 7.1 이상 버전으로 업데이트하여 해결할 수 있다는 것을 확인할 수 있다. 이렇게 보안 요구사항을 식별할 수 있다. 공격 데이터 항목을 활용하면 테스트도 할 수 있다.

III. 결론

본 논문에서는 HEAVENS 구체화를 위한 Automotive Attack DB에 대해 연구하고 이를 활용할 수 있는 방안에 대해 소개하였다.

HEAVENS를 구체화하기 위해 Automotive Attack DB를 연구하고 제시하였지만 이보다 더 나아가 자동차 보안 산업에도 도움이 될 수 있도록 변경하여 사용할 수 있다.

추후 연구를 통해 Automotive Attack DB를 통한 다양한 시나리오를 구성하고 각 시나리오에 대한 DFD(Data Flow Diagram)을 통해 어떤 취약점을 도출할 수 있는지를 알 수 있는 방향을 제시하고 시뮬레이션을 진행하고자 한다.

[참고문헌]

[1] Charlie Miller, Chris Valasek. "Remote Exploitation of an Unaltered Passenger Vehicles". Black Hat USA, 2015

[2] HEAVENS, https://www.sp.se/en/index/research/dependable_systems/heavens/sidor/default.aspx

[3] A. Lautenbach, M. Islam. “HEAVENS Deliverable D2 – Security models”, ver. 1.0, 2014

인터페이스 중심의 최신 자동차 공격 사례 분류1)

송윤근, 이유식*
*ESCRYPT

Classification of the latest car attack cases based on interfaces

Yunkeun Song, Yousik Lee*
*ESCRYPT

요 약

오늘날 자동차는 스마트 폰에 비견될 정도로 다양한 소프트웨어와 인터페이스를 갖춘 IT시스템으로 변모 중이다. 운전자에게 다양한 편의와 안전을 제공하기 위해 전장 부품이 크게 확대 적용되었고, 이제는 전형적인 기계적 장치를 벗어나 첨단화된 전기/전자/IT 시스템이 된 것이다. 이로 인하여 자동차 시스템은 더 다양한 공격 벡터(attack vector)와 기존 IT 시스템이 가지는 보안 취약점을 그대로 수반하게 되었다. 그 뿐만 아니라 자율주행 시스템과 커넥티드(connected) 카가 폭발적으로 발전됨에 따라 더욱 다양한 보안 위협에 직면하게 될 것으로 예측된다. 자동차는 일반 IT환경과는 다르게 생명과 직결될 수 있으므로 보안 위협에 대한 적절한 대응책이 없다면, 보안 (security) 뿐 만 아니라 안전성(safety)까지 훼손될 수 있다. 본 논문에서는 자동차 시스템의 아키텍처 모델을 살펴보고, 사이버 공격 대상이 될 수 있는 자동차의 내/외부 인터페이스를 분류한다. 그리고 실제 자동차 보안사고 사례에 대한 분석하고, 인터페이스를 중심으로 분류한 후 대응책을 알아본다.

I. 서론

지난 수십 년간 자동차의 진화는 기계기계학에서 시작하여 전기기계학, 전자공학, 그리고 소프트웨어 영역으로까지 발전하였으며, 이와 함께 자동차 기능의 초점이 점차 소프트웨어로 옮겨가게 되었다. 실제로 프리미엄 세단의 경우 70~100개 가량의 ECU(Electronic Control Units)가 사용되고, 1억 라인에 달하는 소스 코드가 탑재되고 있다. 뿐만 아니라 자동차는 다양한 ECU를 연결하는 네트워크와 유/무선 인터페이스를 가지고 있다. [1][2][3]

그럼에도 불구하고 현대의 자동차 아키텍처는 보안을 고려하여 설계된 것이 아니기에 공격에 매우 취약하다.

최근에는 공격자에 의해 생명과 직결되는 시스템인 에어백(air bag)과 ABS(Anti-lock Braking System)를 비활성화 시키거나, 운전 중에 Braking System을 조작할 수 있음을 보여 주었다. [4][5]

이렇듯 자동차 사이버 공격을 위한 다양한 공격 벡터(attack vector)가 존재 하지만, 대응은 여전히 부족한 것이 현실이다. 본 논문에서는 최신 공격 사례를 살펴보고, 공격 사례를 인터페이스를 중심으로 분류하여 대응책을 알아본다.

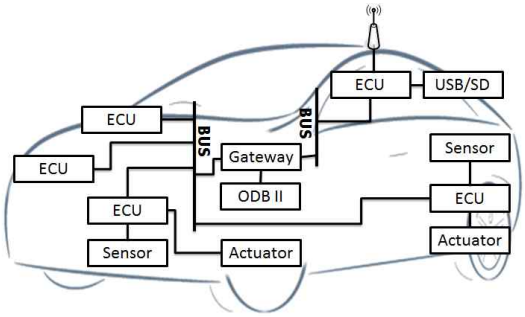
II. 배경

본 장에서는 자동차 시스템 아키텍처 모델을 서술한다. 즉, 자동차를 구성하는 시스템과 해당 시스템에 존재 할 수 있는 데이터, 그리고 시스템 간의 커뮤니케이션에 사용되는 프로토콜에 대해서 알아보고, 자동차가 가지는 인터페이스에 대해서도 설명한다.

일반적으로 현대의 자동차는 [그림 1]과 같이 네트워크 버스, ECU, 센서, 액추에이터, 유/무

1) 본 연구는 국토교통부 및 국토교통과학기술진흥원의 연구비지원(17TLRP-B117133-02)으로 수행된 연구임.

선 인터페이스를 포함하고 있다.



[그림 1] 자동차 시스템 아키텍처 모델

2.1 On-Board IT Systems

현대식 자동차의 On-Board IT System은 ECU, 센서, 액추에이터로 구성되어 있다.

대부분의 ECU는 엔진 및 제동 제어 등 단일한 목적을 가지고 있지만, 인포테인먼트 및 헤드 유닛 등 다양한 목적을 가지는 복잡한 고성능 ECU도 존재한다. ECU는 센서가 수집한 데이터를 처리하고 그에 따라 필요한 액추에이터를 제어한다. 다양한 센서는 자동차 속도, 종·횡가속도 등의 자동차 상태 정보 및 바퀴 속도, 조향 각, 스로틀 위치 등의 자동차 구성요소 정보 그리고 교통 상황 및 온도 등의 자동차 환경 정보를 수집한다. 센서가 자동차를 관찰한다면 제동장치, 조명 및 파워 윈도우 등 액추에이터는 자동차에 물리적인 영향력을 가한다. 일반적으로 센서와 액추에이터는 안전과 편의에 직접적인 영향을 미친다.

2.2 On-Board IT Data

ECU는 주로 ECU 펌웨어 및 매개변수로 구성되어 있으며, 경우에 따라 키, 핀 및 인증서와 같은 암호 데이터를 포함하기도 한다. 많은 ECU는 에러 로그, 통신 로그 및 유지 보수 로그 등의 중요한 데이터를 추가로 기록한다. 또한 GPS 내비게이션 시스템용 지도, 멀티미디어 데이터, 통신 데이터 및 심지어 3rd party 애플리케이션용 데이터에 이르기까지 보다 많은 데이터가 자동차 내에 저장되는 추세이다. 마지막으로 운전자는 개인 전화번호부의 주소 및 전화번호나 자주 방문하는 목적지 혹은 맞춤형 세팅 등의 개인적인 데이터를 저장하고자 할 수도 있다.

2.3 On-Board IT Communication

다양한 안전(safety) 관련 기능은 여러 ECU와 센서, 그리고 액추에이터가 서로 긴밀히 연결 된다. 예를 들어 ABS(Anti-lock Braking System)는 개별 바퀴의 속도 측정 센서에 접근하여 제동 압력을 제어할 수 있어야 한다.

이처럼 각 요소를 상호연결하기 위해서는 자동차의 내부에 네트워크가 필요하다. 대부분의 자동차의 내부 네트워크는 기능적으로 네 개의 분리된 버스 시스템, 즉 파워트레인, 샤시, 바디, 인포테인먼트로 나눌 수 있다. 파워트레인은 CAN(Controller Area Network) 프로토콜을 주로 사용하고, 샤시는 CAN 프로토콜을 사용하지만, FlexRay, RF(Radio Frequency)를 사용하기도 하며, 바디는 CAN, LIN((Local Interconnect Network), RF를 사용한다. 높은 대역폭을 사용하는 인포테인먼트는 MOST(Media Oriented System Transport)를 사용하지만, Bluetooth, Wi-Fi 또는 Ethernet을 사용하기도 한다. [6][7]

앞서 언급된 버스들은 대개 특수한 게이트웨이 브릿지를 통해 연결되는데, 텔레매틱스 및 통신 기능을 갖춘 ECU인 TCU(Telematics control unit)를 사용한다. 이 때 TCU는 경우에 따라서 파워트레인, 바디, 샤시 혹은 인포테인먼트 버스에 연결된다[1]. 마지막으로 모든 버스는 자동차 내부 진단 목적으로 사용하는 외부 포트와 게이트웨이를 통하여 연결될 수 있다.

III. 자동차 인터페이스 분류

[표 1]은 자동차 인터페이스를 인터페이스 클래스로 분류하고 이에 맞는 예시를 제시한다. 인터페이스 클래스는 자동차에서 사용하는 각 기술들의 특성을 기준으로 분류 한 것이고, 자동차에 접근하는 수준(Level of access)은 인터

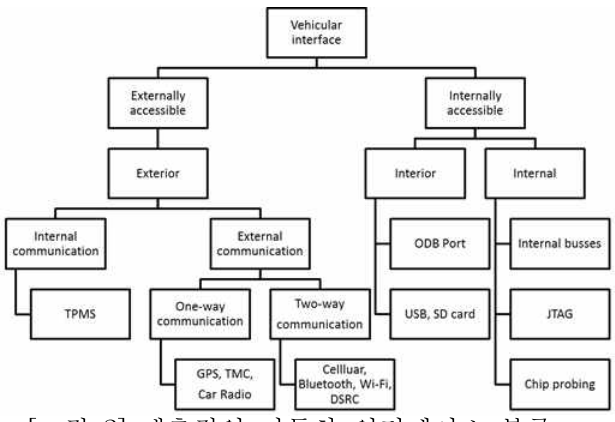
페이스에 접근하기 위한 환경적인 요소이다.

모든 인터페이스는 3가지로 분류 할 수 있는데, 차량 외부(exterior), 차량 내부(interior) 혹은 차량 네트워크 내부(internal)로 분류 할 수 있다.

Interface class	interface	Level of access
Low-range wireless	Bluetooth, TPMS	Interior, Exterior
Mid-range wireless	Wi-Fi, DSRC	Exterior
Wide-range wireless	GSM, LTE	Exterior
One-way Wide-range wireless	GPS, TMC, Car radio,	Exterior
Wired interior	SD Card, USB, CD	Interior
Wired internal	OBD II	Interior
Debug	JTAG	Internal
Physical	chip probing	Internal

[표 1] 자동차의 공격 대상 인터페이스

[그림 2]는 자동차의 공격 대상 인터페이스를 공격자의 접근 방법과 인터페이스의 특성에 따라 분류하여 자동차 인터페이스의 계층을 개괄적으로 보여준다.



[그림 2] 계층적인 자동차 인터페이스 분류

IV. 자동차 보안사고 사례 분석과 대응방안

2010년을 기점으로 본격 적으로 자동차를 대상으로 한 보안 사고가 발생하고 있다. 본 장에서는 그 중 잠재적 위험성이 큰 대표적인 사례를 나열하고, 대응 방안을 알아본다.

4.1 공격 사례

[표 2]는 보안사고 사례와 공격 대상 인터페이스를 정리 한 것이다.

보안사고 사례	interface
안드로이드 앱 리패키징을 이용하여 자동차 강제 제어 공격: engine, RPM, dashboard	OBD II
AVN (Uvo)을 취약점을 이용한 공격: door	GSM, LTE
테슬라 차량 내 웹 브라우저 취약점을 이용한 공격: break system, seat, top roof	Wi-Fi
아웃랜더 PHEV Wi-Fi PSK 취약점을 이용한 공격: door, alarm, light, HAVC	Wi-Fi
차량의 IP만으로 Jeep cherokee 해킹 공격: audio, break, engine	GSM, LTE
GM Onstar 해킹: door, engine	Wi-Fi
도요타 프리우스 CAN 패킷 수집/분석 후 리플레이 공격: all	OBD II
Blank 키에 정보 주입을 통한 BMW 도어락 해킹: door	OBD II
악성코드가 담긴 음악 파일을 통한 공격: AVN system	CD, USB
TPMS 도청을 통한 공격: 차량 추적(privacy), dashboard	TPMS

[표 2] 자동차 보안사고 별 인터페이스 분류

대부분의 공격이 Interior와 Exterior에 존재하는 인터페이스를 통해서 이루어지며, 그 중에서도 비교적 접근이 용이한 ODB II를 통한 공격이 많음을 알 수 있다.

4.2 대응 방안

자동차 내부(Interior)의 공격은 ODB II를 통하여 ECU에 CAN 패킷을 주입하거나, 펌웨어를 변조하기도 한다. 이러한 공격은 OBD II 사용 시 종단 간 인증 수단을 사용하고, 유효한 패킷만 수용하도록 필터링을 적용해야 한다.

자동차 외부(Exterior)의 공격은 취약한 프로

토콜 또는 키 공유 방식을 사용하기 때문에 발생한다. 뿐만 아니라 OEM의 기본 패스워드 정책이 단순하게 설정 되어 있는 것도 문제이다. 보안 사고를 방지하기 위해서는 충분히 강력한 프로토콜, 알고리즘과 함께 기본 설정 까지도 신경 써야 한다.

모든 공격의 최종 대상인 ECU에는 개발 시 사용하던 Debug port를 제거하여야 하며, 내부 펌웨어 분석 방지를 위해서 난독화하고, ECU 펌웨어 변조를 막기 위해서 secure boot, secure flash 기술을 적용하여야 한다.

V. 결론

본 논문에서는 자동차의 공격 대상 인터페이스를 분류하고, 자동차 보안사례를 인터페이스 중심으로 분석하였다. 대부분의 공격은 복수의 인터페이스를 통해 공격이 이루어진다. 그렇기 때문에 하나의 보안 대책이 무력화되어도 자동차 전체 시스템의 안전은 위협받지 않도록 해야 한다. 이를 위해서 계층적으로 분류된 인터페이스를 통한 방어뿐만 아니라 자동차 도메인을 분리 하고, 차량 내부 네트워크의 통신 데이터 보안과 ECU에 대한 보안 까지 심층 방어(defense-in-depth)개념을 적용 한다면 자동차의 보안성은 한층 더 강화 될 것으로 기대된다.

[참고문헌]

[1] K. Koscher et al., "Experimental Security Analysis of a Modern Automobile" 2010 IEEE Symposium on Security and Privacy, pp. 447-462. May 2010

[2] R. Charette. "This car runs on code", https://spectrum.ieee.org/transportation/systems/this-car-runs-on-code, February 2009

[3] McCarthy, C., Harnett, K., and Carter, A. "Characterization of potential security threats in modern automobiles: A composite modeling approach", DOT HS 812 074, National Highway Traffic Safety Administration., October 2014

[4] Palanca A., Evenchick E., Maggi F. and Zanero S. and Stefano Z., "A Stealth, Selective, Link-Layer Denial-of-Service Attack Against Automotive Networks.", , DIMVA 2017, LNCS 10327, pp. 185-206, June 2009

[5] Sen Nie, Ling Liu and Yuefeng Du, "FREE-FALL: HACKING TESLA FROM WIRELESS TO CAN BUS", https://www.blackhat.com/docs/us-17/thursday/us-17-Nie-Free-Fall-Hacking-Tesla-From-Wireless-To-CAN-Bus-wp.pdf blackhat USA 2017, 2017

[6] ENISA "Cyber Security and Resilience of smart cars", European Union Agency for Network and Information Security, January 2017

[7] SAE International, "LIN Network for Vehicle Applications", J2602, November 2012

차량 대상 사이버 공격의 위험 평가 동향 연구

선현진*, 이동훈*
*고려대학교 정보보호학과

Trend research for risk assessment of vehicle threats

Hyun-Jin Sun*, Dong Hoon Lee*
*Graduate School of information security, Korea University.

요 약

차량에 다양한 전자 장치가 장착되면서 차량의 IT 시스템에 대한 취약점을 이용한 사이버 공격들이 제안되고 있다. 이러한 공격에 효율적으로 대처하기 위해 위험에 대한 정량적 평가방법이 선행되어야 한다. 본 논문에서는 차량에 대한 사이버 공격의 위험을 정량화하는 연구를 비교 분석한다. 그리고 현재의 평가방법의 한계점에 대해 서술하고 앞으로의 연구 방향을 제안한다.

I. 서론

차량의 ADAS(Advanced Driver Assistacnce Systems)기술이 발전하면서 다양한 전자 장치가 차량에 장착되었다. 이러한 기술은 운전자에게 편의기능을 제공하고 발생할 수 있는 위험을 알려주어 안전하게 운전할 수 있도록 도와준다. 그러나 이러한 기술을 제공하는 전자장치는 공격자에게 새로운 attack surface를 제공한다. 공격자는 전자 장치에 직접 공격을 수행할 수 있고, 장치간의 통신을 이용해 공격을 수행할 수도 있다. C. Yan et al은 Tesla Model S 차량의 탑재된 센서가 공격에 취약하다는 점을 실험적으로 증명하였다[1]. K. CHO et al은 차량 내부 네트워크의 취약점을 악용해 공격을 수행하였다[2]. 다양한 공격이 제안되면서 차량에 대한 사이버 공격을 정량적으로 분석할 필요성이 대두되었고, 관련 연구가 진행되었다[3][4][6]. 그러나 평가 방법별로 차이점이 존재하고, 표준화된 위험 평가 방법이 아니기 때문에 동일한 공격에 대해 심각도가 다르게 산출된다는 한계점이 존재한다. 본 논문에서는 차량 대상 사이버 공격에 대한 위험 평가 방법들을 비교·분석하고 현재의 위험 평가 방법의 한계점에 대해 예시를 들어 서술한다.

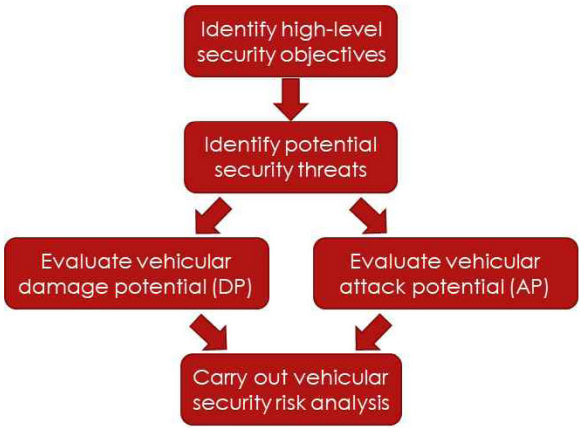
본 연구는 국토교통부 및 국토교통과학기술진흥원의 연구비지원(17TLRP-B117133-02)으로 수행된 연구임.

II. 차량의 사이버 공격에 대한 위험 분석 평가 방법에 대한 연구

이번 절에서는 차량에 대한 사이버 공격의 위험 평가 방법을 분석한다.

2.1 wolf-sheibel's method

M. wolf et al은 IT 시스템의 위험 분석 방법을



[그림 1] 위험 평가 프로세스 I

차량 도메인에 적용하는 방법에 대한 연구를 **수행했다**[3]. 해당 논문의 위험 평가 프로세스는 [그림 1]과 **같다**. 해당 연구의 위험 평가 **프로세스**는 보안목표 **식별**, 보안 위협 **식별**, 위험 **분석** 단계로 **이루어진다**. 보안 목표에 대한 위협 식별 **후**, 공학 분야에서 사용하는 **risk**의 **정의(1)를** 통해 차량의 위험을 수치화 **한다**.

$$risk = (probability\ of\ an\ accident) \times (expected\ losses\ through\ that\ accident) \quad (1)$$

식 (1) 에서 **probability of an accident**는 공격 발생 확률을 **의미하여**, 공격을 수행하기 위해 공격자에게 필요한 조건을 **의미하는 AP**와 **반비례** 하는 **값이다**. **expected losses through that accident**는 **DP(damage potential)**로 대응되며 **DP**는 해당 공격으로 인하여 발생하는 피해의 정도를 **의미한다**. **AP**는 **elapsed time, specialist expertise, knowledge of the target, access, equipment**에 각각 가중치를 부여한 후 각각의 요소마다 **AP**를 계산 후 그 **합(2)**을 **구한다**.

$$AP = AP_{time} + AP_{expertise} + AP_{knowledge} + AP_{access} + AP_{equipment} \quad (2)$$

구한 합을 통해 **AP**를 **basic**에서 **beyond high**까지 **분류한다**. **DP**는 **safety severity, finance severity, operational severity**에 각각 가중치를 부여한 후 각각의 요소마다 **DP**를 계산 후 그 **합(3)**을 **구한다**.

$$DP = DP_{safety} + DP_{financial} + DP_{operational} \quad (3)$$

구한 합을 통해 **DP**를 **insignificant**에서 **catastrophic**까지 **분류한다**. 위와 같이 **AP**와 **DP**를 계산한 이후 각각의 값마다 범위를 정해 분류한 후 **위험을 negligible**부터 **inacceptable**까지 5 단계로 **분류한다**.

2.2 HEAVENS's method

HEAVENS(HEAling Vulnerabilities to ENhance Software Security and Safety) project는 차량의 **전기/전자** 시스템에 대한 **위협**을 분석하고 위험평가를 위한 방법 및 도구 개발에 중점을 두고 **연구했다**[4]. 해당 연구의 위험 평가 **프로세스**는 [**그림2**]와 **같다**. 첫 번째 단계는 위협 분석으로 위협에 대응되는 자산과 **보안속성**을 **식별한다**. 식별 방법으로 **Microsoft의 SPRIDE**[5]**를 이용한다**. 두 번째 단계는 **위험평가** 단계로 **TL(Threat Level)**과 **IL (Impact Level)**을 정량적으로 계산하여 **SL (Security Level)**을 **측정한다**. **TL은** 위협이 발생하기 **위한** 필요조건의 **추정치이다**. **expertise, knowledge about TOE, equipment, window of opportunity**를 **각각 평가하여** 각 요소마다 0~4의 값을 **부여한다**. 각 요소의 값을 합하여 **TL**을 **None**에서 **Critical**까지 분류하고 **TL** 값을 0~4까지 **할당한다**. **IL(Impact Level)**은 위협이 발생하였을 때 미칠 수 있는 피해에 대한 **추정치이다**. **safety, financial, operational, privacy and legislation**으로 **나눈다**. 각 요소마다 값을 계산하고 가중치를 부여한 후 합하여 **IL**을 **No impact**에서 **Critical**까지 분류하고 **IL** 값을 0~4까지 **할당한다**. 최종적으로 **TL**과 **IL**을 **이용해서 SL**을 **구한다**. **SL**은 위험의 심각도를 **나타내며 QM(Quality Management)~Critical**로 **분류된다**. **TL**과 **IL**이 높을수록 **Critical**에 **가까워진다**.

2.3 EVITA's method

EVITA(E-safety Vehicle Intrusion protected Applications)는 사이버 보안 관련 구성요소가 변조로부터 보호되고 중요한 데이터가 **손상으로부터** 보호되는 차량 **on-board** 네트워크 용 아키텍처를 **설계**, 검증하려는 목표를 가지고 **연구했다**[6]. **EVITA**는 위험을 **cybersecurity**



[그림 2] 위험 평가 프로세스 II

objective, attack probability, controllability 3가지 관점에서 **평가했다.** cybersecurity objective는 safety, operational, privacy, financial 4단계로 나눈 후 각각의 요소의 심각도를 S0부터 S4까지 5단계로 **분류하였다.** S0는 영향이 존재 하지 않음을 의미하고 S4는 가장 심각한 영향을 **의미한다.** 각각의 cybersecurity를 분류할 때 위험 식별과 위험 분류를 **한다.** 위험 식별 **단계에**서 공격 동기를 **식별**·분류하고 공격자의 **능력**을 **평가한다.** attack probability는 IT security evaluation에서 사용하는 attack potential의 **개**념을 이용해 계산한 후 계산된 값의 범위를 **나**누어 beyond high부터 basic까지 총 5개로 **분**류 하고 각각 1~5의 값을 **할당한다.** 그리고 **잠**재적인 사고를 피하기 위해 사람이 해야 하는 행위를 controllability로 정의하고 C1에서 C4까지 **분류하였다.** C1은 전문지식이 없는 일반적인 운전자의 대처로 피할 수 있는 위험을 **의미하고,** C4는 어떤 행동으로도 피할 수 없는 위험을 **의미한다.** 3개의 기준이 각각 행과 열로 구성된 matrix를 만들어 최종적으로 위험의 심각도를 **분류했다.**

III. 위험 평가 방법의 차이점 분석

이번 절에서는 앞에서 분석한 위험 평가 **요소**와 평가 방법의 차이점을 분석하고 이로 인해 발생하는 현재의 위험 평가 **방법의** 한계점에 대해 **서술한다.**

3.1 위험 평가항목의 차이점

위의 3가지 평가방법에 평가항목은 [표 1]과 **같다.** wolf-Scheibel's method은 DP를 계산할 때 다른 방법과 달리 privacy를 고려하지 **않았고,** AP를 계산할 때 window of opportunity를 고려하지 **않았다.** HEAVENS's method에서는 TL를 계산할 때 다른 방법과 **달리** elapsed time을 고려하지 **않았다.**

3.2 위험 평가방법의 차이점

HEAVENS's method에서는 TL의 각 요소에 가중치를 주지 **않았다.** 예를 들어 wolf-Scheibel's method는 elapsed time의 최고치를 7로 하고 access의 최고치는 10으로 했지만 HEAVENS's method에서는 TL을 계산할 때 모든 요소가 동일하게 4라는 최대치를 **가진다.** 또한 HEAVENS's method에서 IL을 계산할 때 financial과 safety 둘 다 0~1000의 값을 **가지**지만 wolf-Scheibel's method에서 safety는 0~

[표 1] 위험 평가 방법별 평가 **항목**

	wolf-Scheibel's method[3]	HEVEANS's method[4]	EVITA's method[6]
objective (DP or IL)	• safety • finance • functionality	• safety • finance • functionality • privacy and legislation	• safety • finance • functionality • privacy
requirement (AP or TL)	• elapsed time • expertise • access • equipment • knowledge of target	• expertise • window of opportunity • equipment • knowledge of toe	• elapsed time • expertise • equipment • window of opportunity • knowledge

10000의 값을 가지고 finance는 0~1000의 값을 가진다. 그리고 HEAVENS's method는 privacy를 고려하기 때문에 IL에 0~100의 값이 추가로 더해진다.

3.3 현재 위험 평가방법의 한계점

현재 차량을 대상으로 하는 위험 평가 방법에 대한 표준이 존재하지 않는다. 이로 인해 각 평가방법 별로 위험 평가 항목, 가중치가 달라 동일한 위험에 대해 심각도의 차이가 큰 경우가 발생한다. [표 2]의 예와 같이 안전에 심각하지만 공격에 필요한 요구조건이 많은 동일한 시나리오1에 대해 wolf-Scheibel's method는 위험이 가장 심각한 inacceptable 등급으로 분류된다. 하지만 같은 시나리오에 대해 HEAVENS's method의 경우 위험이 가장 낮은 QM 등급으로 분류된다.

[표 2]시나리오1에 대한 평가 방법별 위험 비교

방법	항목		시나리오 1
wolf-scheibl's method	DP	safety	1000(life-threat)
		class	catastrophic
	AP	expertise	8(multi-expert)
		equipment	9(multi-bespoke)
		knowledge	11(critical)
		class	beyond high
	risk		inacceptable
HEAVENS's method	IL	safety	100(life-threat)
		class	high
	TL	expertise	3(multi-expert)
		equipment	3(multi-bespoke)
		knowledge	3(critical)
		class	None
	risk		QM

IV. 결론

본 논문에서는 차량 대상 사이버 공격의 위험 분석평가 방법의 연구를 비교·분석하고 현재의 위험 분석평가 방법의 한계점을 구체적인 예를 통해 서술했다. 현재의 위험 분석평가 방법은 공인된 기준이 없으므로 동일한 위험에 대한 평가가 방법별로 다르게 분류 되는 한계점을 가진다.

향후에는 동일한 위험에 대해 평가 방법별 위험 심각도 평가 결과의 편차를 줄일 수 있는 방법을 연구할 예정이다.

[참고문헌]

[1] C. Yan, W. Xu and J.H. Liu, Can You Trust Autonomous Vehicles: Contactless Attacks against Sensors of Self-driving Vehicle, DEF CON 24 Hacking Conference, August, 2016.

[2] K. Cho and K. Shin, Error Handling of In-vehicle Networks Makes Them Vulnerable, Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pages 1044-1055, October, 2016

[3] M. Wolf and M. Scheibel, A Systematic Approach to a Quantified Security Risk Analysis for Vehicular IT Systems, Gesellschaft für Informatik, Bonn, pages 195-210, 2012.

[4] M. Islam et al., Heaven Project, Release 1. December, 2014.

[5] Wikipedia, STRIDE (security), https://en.wikipedia.org/wiki/STRIDE_(security), Accessed: 2017-10-25

[6] Ruddle, Alastair, Ward, David et al, EVITA Project, Security requirements for automotive on-board networks based on dark-side scenarios Version 1.1, December, 2009.

2017 한국정보보호학회 동계학술대회

A4 클라우드보안

윤 지 원 (고려대)

- **OpenStack Swift의 보안 취약점과 해결방안**
  이동혁, 박남제 (제주대학교)
- **스마트TV 취약점 분석 및 대응방안 연구**
  장태진, 박동수, 우지섭, 윤정준, 황윤아 (KITRI BoB)
- **클라우드 기반의 지능형 영상 감시 시스템을 위한 영상 프라이버시 보호 방법**
  이동혁, 박남제 (제주대학교)
- **토렌트를 통해 유포되는 악성코드 분석 및 대응방안**
  김동현, 양준호, 임명은, 염흥열 (순천향대학교)

OpenStack Swift의 보안 취약점과 해결방안

이동혁*,**, 박남제*,**

*제주대학교 대학원 컴퓨터교육전공, **제주대학교 초등교육연구소

OpenStack Swift Security Vulnerability and Solution

Donghyeok Lee*,**, Namje Park*,**

*Graduate School, Jeju National Univ., **Dept. of Com. Edu., Jeju National University

요 약

객체 스토리지 시스템은 클라우드에서의 대규모 데이터 관리 및 전달을 위한 차세대 서비스 저장소 기술이며, 이 가운데 OpenStack Swift가 객체 저장 스토리지로 널리 사용되고 있다. OpenStack Swift는 데이터 관리 및 검색을 위한 여러 가지 다양한 구성 요소를 갖추고 있다. 그러나, OpenStack Swift를 통한 데이터 전송시 REST 통신 과정에서 데이터가 노출될 수 있고, 서버에서 암호화 수행시, 서버상 평문 데이터가 노출될 수 있다는 단점이 있다. 본 논문에서는 별도의 키관리 서버를 통한 안전한 파일 보안 관리 구조를 제안하고, 프로토콜을 설계하였다.

I. 서론

최신 기술이 등장하고, 사진/영상 편집 기술의 발달, SNS와 같은 소셜네트워크 서비스의 활발한 사용 등으로 텍스트, 이미지, 오디오, 비디오 스트림 등과 같은 비정형 데이터가 기존에 비해 크게 증가하고 있다. IDC에서는 이러한 예측할 수 없는 비정형 데이터가 2020년까지 최대 40제타바이트까지 도달할 수 있다고 경고하고 있다. 이러한 대량의 대용량 데이터를 처리하기 위해 클라우드 환경에서의 객체 저장 기술이 차세대 스토리지 플랫폼으로 부상하고 있다. 클라우드 환경에서의 객체 스토리지 기술은 대량의 비정형 데이터를 보관/관리하는데 적합한 기술이다. 현재 객체 스토리지 기술 가운데 OpenStack Swift가 흔히 사용되고 있다. OpenStack Swift는 OpenStack 프레임워크에서의 데이터 저장소 역할을 담당하고 있으며, 데이터 관리 및 검색을 위한 여러가지 다양한 기능을 제공하고 있다. 그러나, OpenStack에는 보안상의 이슈가 존재하고 있다. 특히, 클라우드의 특성상 여러 사용자가 동일한 리소스를 공유하게 되므로, 클라우드 환경에서는 보안 침해가 발생할 소지가 크다. Openstack Swift 기술은 사용자 이름, 암호 및 계정으로만 데이터의 유효성을 검사하는 TempAuth 미들웨어를 사용하여 보안 인증을 처리하고 있다. 그러나, 높은 수준의 데이터 보안을 달성하려면 암호화가 반드시 필요하다. 암호화가 수행되는 위치에 따라 서버측 암호화와 클라이언트 측 암호화로 분류된다. 서버측 암호화는 데이터를 안전하게 보호하는 반면 클라이언트 측 암호화는 전송 중에 데이터를 보호한다는 특징이 있다. 그러나 Swift에서 데이터를 암호화하고 키를 구성할 수 있는 자동화된 메커니즘이 없다는 문제가 있어 해결책이 요구된다. 본 논문에서는 키 관리 서버를 별도로 두고 OpenStack 통신 과정에서의 암호화 및 파일 기밀성 보장을 위한 구조를 제안하고자 한다.

II. 관련 연구

2.1 기존 객체기반 스토리지 시스템

현재 일반적으로 사용되는 객체 기반 스토리지 시스템에는 Openstack Swift, Amazon S3 및 Windows Azure 등이 있다. 이러한 시스템은 무단 액세스로부터 데이터를 보호하기 위한 다양한 보안 메커니즘을 제공하고 있다. Keystone은 OpenStack에서 제공하는 Identity 관리 및 인증 서비스이다. Keystone은 사용자 이름 및 암호, LDAP 및 TLS 클라이언트 인증을 포함한 다중 요소 인증을 지원하고 있다. Amazon S3는 AWS 키 관리 서비스를 사용하여 객체를 암호화하는 데 사용되는 키를 만들고 관리한다. 또한 사용자는 사용 정책을 설정/수정할 수 있으며 감사를 모니터링하여 어떤 사용자가 데이터에 액세스하기 위한 키사용 여부를 확인할 수도 있다.

2.2 OpenStack Swift

OpenStack Swift는 오픈소스 객체 저장 시스템이다. Swift는 OpenStack을 위한 분산된 궁극적으로 일관된 가상 객체 저장소를 제공한다. 스위프트는 여러 노드에 분산된 수십억 개의 개체를 저장할 수 있다. Swift는 파일, 이미지 및 비디오와 같은 멀티미디어 데이터, 웹 콘텐츠, 백업, 가상 머신 스냅샷 및 기타 비정형 데이터를 대규모로 저장하도록 설계되었다. 스위프트에는 스위프트 클러스터에서 각 객체를 세 번 복제하는 내장 중복 기능이 있으며, 파일 사이즈와 개체수 측면에서 매우 확장성이 뛰어나다는 장점이 있다.

III. OpenStack Swift 보안성 개선방안

3.1 클라우드 환경에서의 보안 취약점

클라우드 스토리지 서비스를 사용할때는 공개형 API를 사용하여 데이터를 클라우드 환경에 업로드하게 되며, 이 과정에서 데이터의 실제 객체와 소유자는 분리된다는 단점이 있다. 이러한 점은 클라우드 환경의 치명적인 취약성이다. 과거 많은 클라우드 서비스 제공 업체에서 다양한 보안 문제가 발생한 바 있다. 클라우드 환경에서의 데이터 보안 문제는 클라우드 스토리지 환경을 활성화하는데 큰 걸림돌이 되고 있다. 이러한 경우, 데이터 암호화가 해결책이 될 수 있다. 그러나, 데이터 암호화의 주체가 누구인지, 클라이언트 측에서 암호화를 하게 될 경우 서버상에서 관리상의 문제, 검색 문제 등 다양한 해결과제가 존재한다. 또한, 데이터를 여러 사용자와 공유하는 경우도 문제가 될 수 있으며, 이러한 경우 데이터의 소유권을 어떻게 정할 것인가에 대한 이슈가 발생한다.

또한, 클라우드 서비스는 일반적으로 중복 제거 모델을 적용하고 있다. 이 개념은 클라우드 서버에 중복된 데이터를 저장하지 않으므로 네트워크 대역 폭을 줄임과 동시에 스토리지 용량을 획기적으로 절감할 수 있다는 장점이 있다. 그러나, 중복 제거의 기능 구현시, 중복 제거의 키가 되는 해쉬값을 클라이언트 프로토콜 상에서 제공하는 경우, 해커가 고의로 훼손된 파일 해쉬값을 만들어 올리거나, 데이터를 악의적으로 변조할 경우, 해당 파일과 동일한 파일을 가진 사용자는 정상적으로 클라우드 스토리지 서비스를 사용할 수 없게 된다. 또한, 중복 제거된 파일 삭제하면 해당 파일을 소유한 모든 계정의 파일이 논리적으로 사라진다는 문제점도 존재한다.

3.2 키관리 기반의 접근제어

Swift의 보안 문제는 접근정책을 통해 어느 정도까지는 해결할 수 있다. 그러나, 클라우드 사용자간에 파일 공유시 문제가 발생할 수 있다. 일반적으로, 서로 다른 파일 접근 권한이 있는 사용자간에 동일한 파일을 공유하는 과정에서 다양한 문제가 발생할 수 있다. 따라서, 서로 다른 권한을 가진 자에 의한 파일 공유를 위한 접근제어 정책을 고려할 필요가 있다.

또한, 클라우드 서비스 제공 업체는 클라우드에서 보안을 제공하기 위해 여러 가지 서버 측 암호화 도구를 구축하고 있다. 그러나, 실질적으로 클라우드 환경에서의 암호화 시점은 서

버에 파일이 올라갔을때 암호화가 되며, 이 경우 REST로 전송되는 과정에서는 평문이 그대로 노출될 수 있고, REST 종단간 암호화를 적용한다 하더라도, 결국 서버측에서 재암호화를 위해서 평문을 그대로 볼 수 있다는 보안 문제가 존재한다. 따라서, 본 논문에서는 클라이언트 측에서 파일을 암호화하여 올리는 방식을 고려하였다. 클라이언트 측의 암호화를 통하여 데이터를 보호할 수 있으며, 서버 측에서는 임의로 암호화된 파일을 확인할 수 없다.

일반적으로 클라우드에는 다양한 종류의 파일이 업로드 될 수 있으며, 그 중에서 특히 강력한 보안이 필요한 파일이 있을 수 있다. 이 과정에서, 각각의 클라이언트 별로 서로 다른 암호키로 적용하여 파일을 업로드하는 경우, 파일 공유 시 문제가 발생할 것이다. 예를 들어, 특정 클라이언트에서 특정키로 암호화하여 올린 데이터의 경우, 다른 클라이언트에 파일을 공유하려면 결국 암호키를 알려줘야 한다는 문제가 있다. 이러한 문제를 해결하기 위하여, 키 관리 서버를 적용하였다. 본 논문에서는 (그림 1)에서 제안하는 방식과 같이 별도의 키 관리 서버를 통하여 키를 관리할 수 있는 방법을 제안하고자 한다.



[그림 1] OpenStack Swift와 키 관리 서버

(그림 2)의 프로토콜 설명은 다음과 같다.

① 사용자는 키관리 서버(KMS)에 사용자 정보와 함께 키를 요청한다.

② 키 관리 서버는 사용자 정보를 확인하고, 사용자에 권한에 대응하는 암호화된 전달한다.

③ 사용자는 클라이언트의 키와 키 관리서버에서 전달받은 키를 XOR하여 암호키를 생성한다.

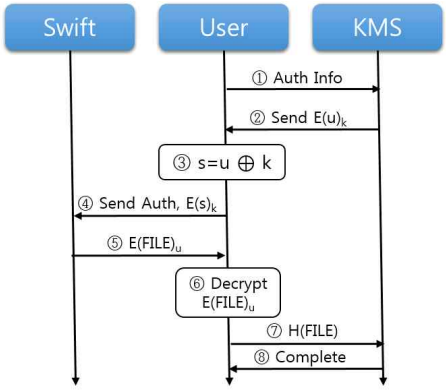
④ 사용자는 Swift 서버에 파일을 요청한다.

⑤ Swift 서버는 사용자의 신원 확인 후, 암호화된 파일을 사용자에게 전달한다.

⑥ 사용자는 앞서 생성한 암호화 키를 통하여, 파일을 복호화하고, 원본 파일을 획득한다.

⑦ 파일의 해쉬값을 KMS서버에 전송하여, 소유권의 정당성 확인 및 감사증적에 활용한다.

⑧ KMS 서버는 확인이 완료됨을 알린다.



[그림 2] Swift 키 생성 프로토콜

IV. 결론

본 논문에서는 OpenStack Swift의 객체 스토리지 환경에서서의 클라이언트 측면의 암복호화 구조와 프로토콜을 제안하였다. 제안된 구조 및 프로토콜은 클라이언트 측의 오버 헤드가 다소 발생할 수 있으나, Swift 서버와의 REST 통신과정에서 어떠한 정보도 노출시키지 않으며, 서버측의 암호화 비용을 절감시킬 수 있다는 장점이 있다. 또한, 키관리 서버를 통하여 개체간 파일 공유가 가능하다는 장점이 있다.

ACKNOWLEDGMENT

이 논문은 2017년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No.2017-0-00207,클라우드 기반 지능형 영상보안 인큐베이팅 플랫폼 개발). 교신저자 (박남제, namjepark@jejunu.ac.kr).

[참고문헌]

[1] Surekha Thota Email author, Ravi Prakash Reddy Induri, Raghavendra Kune, "Split key management framework for Open Stack Swift object storage cloud", CSI Transactions on ICT, pp.1-10, 2017.

[2] Ayache M, Erradi M, Freisleben B (2015) Access control policies enforcement in a cloud environment: openstack. In: IEEE 11th international conference on information assurance and security (IAS)

스마트TV 취약점 분석 및 대응방안 연구

장태진* 박동수* 우지섭* 윤정준* 황윤아*

*Best of the Best

Analysis of smartTV vulnerability and Research on countermeasure

Tae-Jin Jang*Dong-Su Park* Ji-Sub Woo* Jung-Joon Yoon* Yun-Ah Hwang*

*Best of the Best

요 약

스마트TV는 아날로그TV와 인터넷이 접목된 것으로, TV시청 외에도 사용자의 목적에 따라 어플리케이션을 설치하여 다양한 서비스를 이용할 수 있다. 그러나 여러 기능들의 추가로 인해 사용자 편의성이 증가함과 동시에 다양한 해킹공격이 가능하게 되는 부작용이 발생하였다. 본 논문에서는 국내의 스마트TV를 분석하고 취약점을 이용한 해킹기법들을 수행하여 발생할 수 있는 침해사고 위협들을 사전에 방지하고자 대응방안에 대해 연구하였다. 이를 통해 발전하고 있는 스마트TV의 보안위협들을 인지하고 제거하여 사용자들에게 보다 안전한 서비스를 제공함으로써 스마트TV시장의 발전을 기대한다.

Abstract

Smart TV is a combination of analog TV and internet. In addition to watching TV, smart TV can be installed various applications according to user's purpose. As user convenience has been increased, however, various side effects such as various hacking attacks have been generated. In this paper, we analyze smart TV in Korea and study countermeasures to prevent potential threats of infringement by perform the penetration test. We expect the development of smart TV market by recognizing and eliminating the security threats of developing smart TV.

I. 서론

스마트TV는 기존의 TV기능에 인터넷을 더해 사용자의 목적에 따라 어플리케이션을 설치하여 다양한 컨텐츠들을 이용할 수 있다. 또한 네트워크를 통해 스마트기기를 연결해주는 스마트허브로써의 역할도 수행할 수 있다. 이러한 이점들을 지닌 스마트TV에 대해 IoT시장을 선도하고자 하는 S사, A사, L사 등 다양한 기업들이 이 시장에 참여하고 있다. 그중 국내 기업들이 세계적으로 가장 많이 스마트TV를 수출하고 있다 [1]. Fig. 1.은 전 세계적으로 스마트TV의 수출량을 그래프 화 한 것이다. 이는 국내 스마트TV의 취약성이 발견될 시 세계적으로 큰 영향이 미칠 수 있음을 알려준다.



Fig. 1. 세계 스마트TV시장 국가별 점유율

최근 발전되고 있는 스마트 홈에서도 스마트TV의 도입이 늘어나고 있다. 스마트 홈 환경에 도입된 여러 IoT중에서도 스마트TV가 가장 많

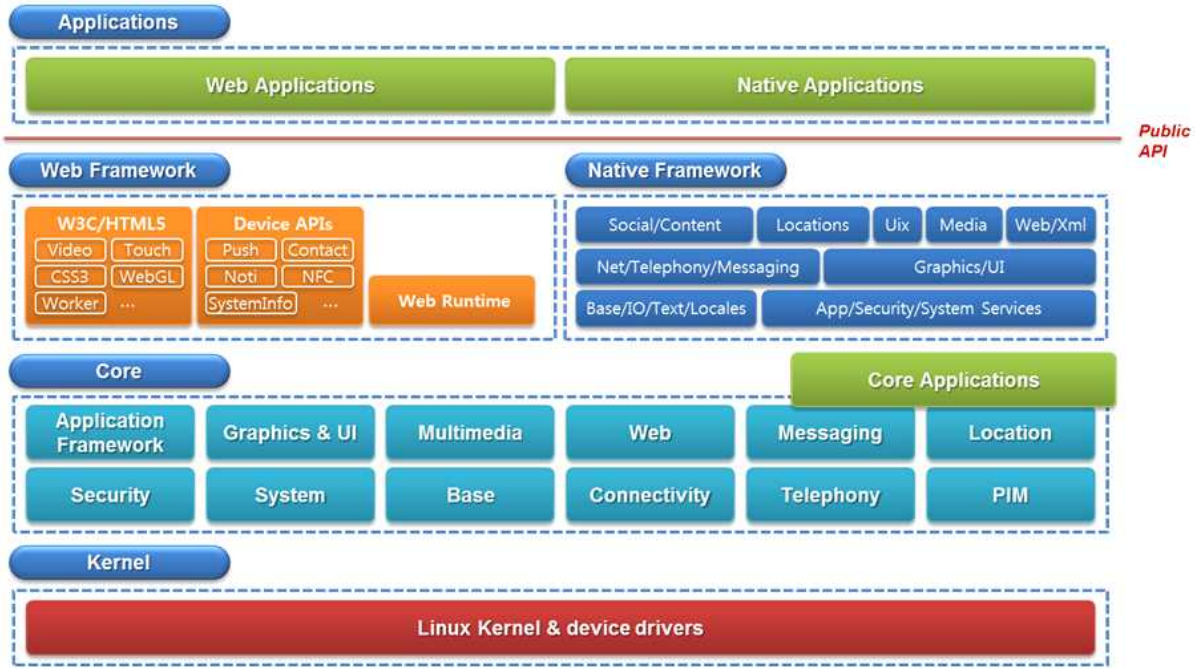


Fig. 2. Tizen OS Architecture

이 보급되었으며 스마트서비스의 핵심요소로 자리 잡고 있다 [2]. Fig. 3.은 스마트 홈 환경에서 스마트TV의 보급 수치를 그래프 화 한 것이다.

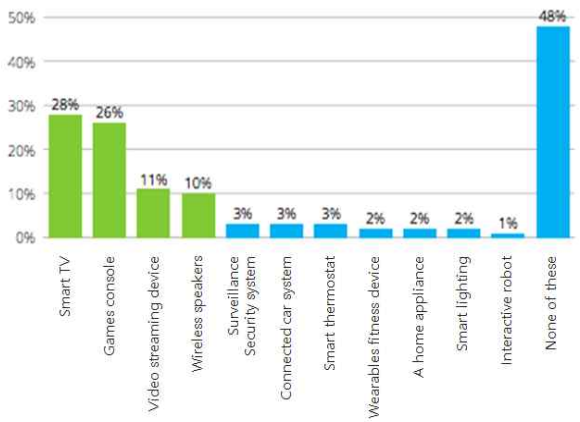


Fig. 3. IoT most used in smart home

스마트 TV에서 여러 기능들의 추가로 인해 사용자 편의성이 증가함과 동시에 다양한 공격이 가능하게 되는 부작용이 발생하였다. 스마트TV에서 유료로 제공하는 서비스들을 결제 없이 이용할 수 있고, 해적방송이나 악의적인 이미지를 출력하여 사용자의 서비스를 방해할 수 있다. 실제로 CIA와 MI5가 제작한 악성코드를 통해 스마트TV로부터 대화를 도청한 사례가 존재하며[3], TV전원이 꺼진 상태를 가장해서 집안 내부 촬영, 개인정보 유출 등 많은 문제가 발생할 수 있다[4]. 이에 따라 발전하고 있는 스마트 홈, 스마트 호텔과 같은 스마트TV가 도입된 곳은 모두 프라이버시 침해 위협이 발생할 수 있다.

본 연구는 S사 스마트TV를 대상으로 취약점분석을 진행하며 스마트TV의 보안성 향상을 위해 어플리케이션, 네트워크 관점으로 취약점 분석을 수행하였다. 그 결과 스마트TV와 서버 간의 통신 과정을 공격하여 실험을 통해 화면에 악의적인 이미지를 출력할 수 있었고, 스트리밍서비스를 지원하는 미디어 관련 어플리케이션에서 유료로 사용하는 서비스들을 결제 없이 사용할 수 있는 문제점이 나타난 것을 볼 수 있다. 이러한 문제점들을 바탕으로 일어날 수 있는 해킹위협에 대해 대응방안을 제시하고자 한다.

II. 분석환경

분석은 국내 S사의 스마트TV중에서 2013년 모델 UN32F6400와 2016년 모델 UN60KS8000을 대상으로 진행하였다. 펌웨어 버전은 각각 T-MST12AKUC, T-JZMAKUC이며 OS는 Tizen OS로 2.2, 2.4버전이 탑재되어있다.

S사 스마트TV의 Fig. 2에서 Tizen OS는 Linux커널을 기반으로 운영되며 HTML5, 자바스크립트, CSS3을 기반으로 하는 웹 어플리케이션과 C++을 기반으로 하는 네이티브 어플리케이션, 이 둘을 합친 하이브리드 어플리케이션을 모두 지원한다 [5]. 특히 Tizen은 웹 어플리케이션기술을 최대한 지원하고 있으며 높은 HTML5 호환성을 지니고 있으며 상대적으로 개발이 쉬워 진입장벽이 낮다. 이러한 이유로 S사의 스토어에서는 네이티브 어플리케이션보다 웹 어플리케이션이 대부분을 차지한다. 본 연구는 웹 어플리케이션과 네트워크를 대상으로 분석을 진행하였다.

III. 해킹공격

3.1 악의적인 이미지 변조

스마트TV는 웹 브라우저를 사용하여 웹 서비스를 이용할 수 있다. 사용자가 스마트 서비스를 이용하면서 화면에 출력되는 대부분의 이미지들은 서버로부터 전송받는다. 그러나 사용자와 서버 간의 통신과정에서 전송되는 이미지가 모두 암호화가 되어있지 않고 노출되어있다. Sniffing툴을 이용해서 패킷을 확인할 수 있으며 중간자 공격을 위한 MITMF툴로 Arp Spoofing 공격을 수행했다. 먼저 사용자가 서비스를 이용하기 위해 서버에게 요청을 보내고 이에 대한 서버의 응답패킷을 가로챈다. 이미지에 해당하는 부분을 악의적인 이미지로 변조하여 사용자에게 전송하여 스마트TV에 정상적인 이미지가 아닌 해커로부터 악의적인 이미지로 변조되어 출력된다. Fig. 4.는 스마트TV 웹브라우저의 이미지가 변조된 것을 보여준다.

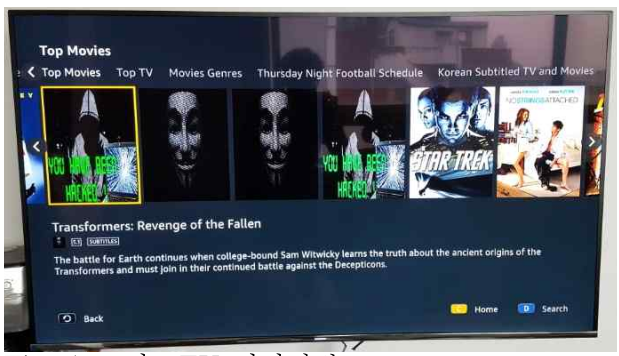


Fig. 4. 스마트TV 이미지변조

3.2 해적방송

스마트TV 사용자와 서버간의 네트워크 통신 과정에서 MITM공격이 가능하다는 것을 확인한 후 TV의 주요기능인 방송채널을 악의적으로 조작하여 해적방송을 유포하는 위험이 발생할 수 있다. 기존의 아날로그TV는 유선으로 연결하여 주파수를 통해 방송을 시청하지만 S사의 스마트TV의 경우 무선으로도 방송을 시청할 수 있다. 무선으로 TV를 시청할 때 IPTV어플리케이션을 사용하며 채널정보를 서버로부터 받아서 시청을 한다. 통신 과정에서 네트워크 공격으로 채널정보를 조작하여 해적방송을 유포할 수도 있다.

3.3 웹 어플리케이션 취약점

S사의 스마트TV에 기본으로 설치되어 있는 노래방 어플리케이션은 암호화가 되어있지 않은 통신으로 인해, MR Streaming URL을 확인할 수 있다. 해당 URL로 접속하게 되면 별 다른 인증 없이 Straming서비스를 이용하거나 음원을 다운받을 수 있다. 이를 통해 결제를 하지 않고 노래방 서비스를 이용할 수 있으며 Streaming으로만 제공되어야 하는 음원을 다운로드할 수 있다. 또한 IP로 방송을 볼 수 있는 어플리케이션의 경우 해당 방송에 대한 URL이 노출되어 약간의 변조를 통해 인가되지 않은 방송을 시청할 수 있다.

IV.결론

본 논문에서는 스마트TV를 대상으로 네트워크 및 웹 어플리케이션 취약점을 분석하였다. 네트워크의 경우 암호화가 되어있지 않은 패킷들로 통신하는 대부분의 스마트기기들에 대해서도 적용이 가능하다. 네트워크 통신과정에서 SSL사용을 권장하여 서비스를 이용하는 사용자들의 프라이버시를 보호하고 결제우회나 서비스를 방해하는 악성 행위를 예방하기 위해 암호화를 수행해야한다. 사용자가 안전한 프로토콜을 이용할 수 있도록 기업과 개발자들은 보안 위협에 대해 고려하고 시큐어코딩이 적용된 서비스를 제공해야 한다.

스마트TV시장에서 서비스는 계속해서 발전해 나아갈 것이며 이에 따라 보안위협요소도 증가할 것이다. 서비스 제공자들은 이러한 위협들로 인해 피해가 발생하지 않도록 개발해야 하며 제시한 문제점들을 신속히 대응하여 안전한 서비스를 제공함으로써 스마트TV시장이 발전하길 기대한다.

[참고문헌]

[1] Digital Times News (http://www.dt.co.kr/contents.html?article_no=2017033002100832816001)

[2] Smart home market size – Trends and projections (http://www.ironpaper.com/webintel/articles/smart-home-market-size-trends-projections/)

[3] Ohmynews (http://www.ohmynews.com/NWS_Web/View/at_pg.aspx?CNTN_CD=A0002305182&PAGE_CD=N0002&CMPT_CD=M0112)

[4] Seung-jin (beist) Lee and Seung-joo Kim, "Hacking, Surveilling, and Deceiving the Audience with Smart TV," Black Hat Briefings (2013), Jul. 2013.

[5] Android VS Tizen Preseantation - Basavaraj Shetty (https://www.slideshare.net/BasavarajSavalagi/android-vs-tizen-preseantation)

클라우드 기반의 지능형 영상 감시 시스템을 위한 영상 프라이버시 보호 방법

이동혁*,**, 박남제*,**

*제주대학교 대학원 컴퓨터교육전공, **제주대학교 초등교육연구소

Privacy Protection Method for Intelligent Video Surveillance System based on Cloud Environment

Donghyeok Lee*,**, Namje Park*,**

*Graduate School, Jeju National Univ., **Dept. of Com. Edu., Jeju National University

요 약

빅데이터 기반의 영상분석 기술이 발전함에 따라, CCTV 지능치안 환경이 도입되고 있으며, 이에 따라 CCTV 영상 처리의 중요성도 커지고 있다. CCTV 영상을 효율적으로 보관하기 위해서는 클라우드 서비스의 활용이 필수적이다. 그러나, 클라우드 스토리지를 활용하게 되면 여러 다양한 보안 취약점과 프라이버시 문제가 발생할 수 있다. 본 논문에서는 클라우드 환경에서의 CCTV 영상데이터를 취급하는 가운데 발생할 수 있는 영상 프라이버시 침해를 안전하게 보호할 수 있는 방법을 제안한다.

I. 서론

최근 클라우드 기술이 일반화되고 있다. 특히, 빅데이터 기반의 딥러닝 기술의 발전 및 데이터 축적건수의 증가에 따라 영상정보 처리 기술 또한 비약적인 발전을 거듭하고 있다. 최근의 영상 객체 인식률은 매우 높은 수준에 도달해 있다.

영상 분석 기술을 활용하기 용이한 분야 가운데 하나로 CCTV 기반의 지능형 감시 환경이 있다. 이러한 기술은 스마트시티, 치안 감시 등의 목적으로 활용될 수 있으며, 정확도 높은 얼굴 식별 기술에 따라서 향후 더욱 안전한 치안 서비스를 제공받을 수 있게 될 것이다.

그러나, 영상분석 기술이 치안환경에 적용되려면 프라이버시 문제 해결이 매우 시급하다. CCTV에 촬영된 영상은 일종의 개인정보로 간주되며, 여기에는 개인의 성향, 라이프스타일, 생활 패턴, 위치정보, 습관 등 다양한 개인정보가 직접적으로 노출될 수 있어 보안상의 문제가 매우 심각할 수 있다. 그러나, 보안을 위해 CCTV에 촬영된 파일의 암호화를 수행한다면, 암복호화시 걸리는 부하 뿐만 아니라, 원활한 검색이 어렵다는 특징이 있다. 예를 들어, 홍길동이란 사람이 촬영된 데이터를 암호화된 파일에서 찾으려 한다면 모든 영상 데이터를 복호화한 후에 찾아야 하므로 서버측의 부하가 매우 커질 수 있다. 따라서, 영상을 암호화하는 것은 CCTV 기반 치안 감시 환경에서는 성능상 문제가 매우 클 수 있다.

본 논문에서는 영상에 나타난 얼굴정보는 마스킹하고, 검색에 필요한 메타정보는 비식별화하는 방법을 제안하였다. 제안한 방법의 특징으로, 마스킹된 영상에 대한 검색이 용이하며, 원본 파일 및 메타정보가 모두 비식별화 처리되어 객체의 프라이버시를 보장할 수 있다.

II. 기존 연구

영상 데이터에서 최소한의 정보를 공개하는

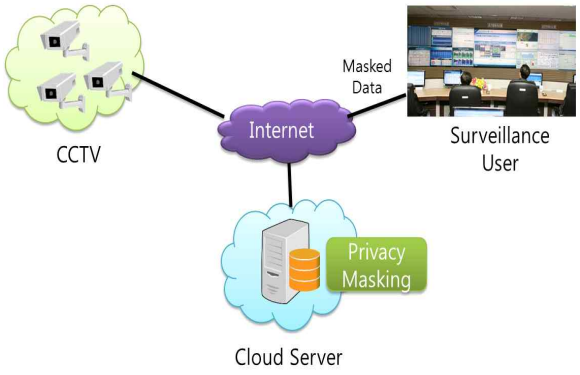
방법으로, 얼굴 마스킹 방법이 존재한다. 이 방법은 화면에 나타난 얼굴을 마스킹 처리하여, 개인을 식별할 수 없도록 할 수 있는 기술로써, 영상정보보호에 가장 일반적으로 사용되는 기술이다. 그러나, 이러한 경우 마스킹된 정보를 복원할 수 없다는 단점 또한 존재한다. 따라서, 마스킹된 상태에서 개인이 식별이 필요할 경우, 원본 영상데이터가 별도로 필요하게 될 수 있다. 이를 보완할 수 있는 방법으로, 얼굴영상 부분 암호화 기법이 존재하며, 이 방법은 얼굴 영상 정보만 암호화 시키는 방법으로, 암호화 키를 알면 원본으로 복호화가 가능하다는 장점이 있다. 그러나, 복호화 키가 노출되었을 경우, 결국 다시 프라이버시 침해로 이어질 수 있다는 문제도 존재한다.

현재 CCTV 영상정보를 안전하게 관리하는 다양한 제품이 국내외에 출시되어 있다. IBM의 Intelligent Video Analytics, Smart Surveillance System 등 다양한 제품이 존재하며, 미국 뉴욕시는 마이크로소프트사와 공동개발한 DAS(Domain Awareness System) 시스템을 구축하여 운용중에 있다. CCTV 영상 처리에서는 영상 객체의 프라이버시 보호가 필수적이며, 지금까지 CCTV상의 영상정보를 보호하기 위한 다양한 방법이 제안되어 있다.

한편, 국방부는 CCTV 및 드론의 촬영영상을 빅데이터로 분석하여 대응하는 D-Net 프로젝트를 진행중이고, 미국의 DARPA에서는 영상데이터에서의 특정행위를 자동 인지하는 VIRAT, 상황 및 행동을 분석하여 다음에 발생할 상황에 대해 예측하는 Mind's Eye 프로젝트가 진행중에 있다. 본 연구에서는 CCTV 영상 데이터를 보다 효율적이고 안전하게 보호할 수 있는 방법에 대해 제안하고자 한다. 제안한 방법에서 영상정보와 메타정보를 평문으로 보관하지 않고, 각각 마스킹 및 비식별화 처리를 통해 영상 객체의 개인정보를 보호할 수 있다.

III. 제안 기법

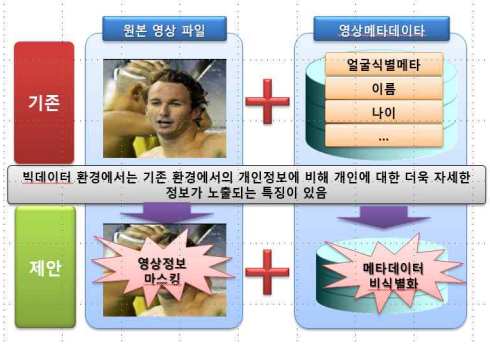
본 연구에서 제안하는 방식의 개요는 (그림 1)과 같다. 기본적으로, 클라우드 서버는 CCTV에서 촬영된 영상 정보와 검색에 필요한 메타정보를 가지고 있다. CCTV에서 영상이 촬용되면, 클라우드 서버에 안전한 경로를 통하여 데이터 영상을 전송한다. 이 경우, 서버는 CCTV에서 수신한 영상데이터에 대하여 빅데이터 기반의 메타정보 추출 후, 영상 정보를 안전하게 마스킹 처리한다. 메타정보는 영상정보의 검색에 사용된다. 따라서, 마스킹된 영상정보도 메타정보를 기반으로 용이하게 검색이 가능하다.



(그림 1) 제안 방식 개요

한편, 영상을 마스킹 처리했다고 하더라도, 메타데이터를 기반으로 원본 영상데이터에 대한 일정부분의 정보를 추정할 수 있다. 따라서, 영상에 사용된 메타정보는 (그림 2)와 같이 비식별화 처리를 수행한다. 비식별화는 기존의 비식별화 알고리즘이 사용될 수 있으며, 분석이 필요시 재식별화를 통하여 분석할 수 있다. 또한, 비식별화 방식은 직접적으로 통계정보 등에 사용할 수 있다는 장점이 있어 검색 및 집계 등에 효율성을 보인다. 결론적으로, 원본 영상 파일에 대한 이미지/비디오 정보는 마스킹 처리되고, 영상 메타데이터는 비식별화처리 되어, 평문 정보는 어디에도 존재하지 않는다. 따라서, 개인의 프라이버시를 노출되지 않은 상태에서 검색, 집계 등 질의를 수행할 수 있다.

이러한 방법을 통하여 비식별화된 영상메타 및 영상정보를 가진 클라우드 환경에 해커가 침입하고, 루트 권한을 획득하더라도 마스킹 처리된 영상 파일과 비식별화된 메타데이터를 기반으로는 실제 원본에 대한 영상 정보를 추정하기가 매우 어렵게 되어 안전하다.



(그림 2) 영상정보 비식별화를 통한 보호

한편, 객체의 프라이버시는 개인의 공개 정도 여부가 차등화되어 결정될 수 있다. 여기에서 차등화는 객체의 정보를 얼마만큼 공개할 것인가에 대한 여부가 주요 관점이 된다. 기존의 마스킹 기법은 마스킹 처리를 한 경우와 하지 않은 경우의 두가지로 구분되며, 실질적으로 이러한 점은 좀더 세분화된 접근제어 정책을 적용하기 어렵다. 즉, 영상에 대해 아주 약간의 정보를 제공하고자 하는 경우가 있을 수 있으며, 이러한 경우는 기존 마스킹 기법으로는 해결할 수 없다. 즉, 치안 환경에서 특정 객체의 위험도, 혹은 사용자의 동의사항 수준에 따라 마스킹 정보를 차별화하여 관리하는 구조가 필요하다. (그림 3)에서는 프라이버시 보호 레벨에 따른 마스킹 레벨 차등화 단계를 나타낸다. 기존의 얼굴 마스킹 방식은 일괄적으로 수행되었던 측면이 있어 마스킹 기법 자체에 프라이버시 보호 레벨을 세밀하게 지정할 수 없었으나, 본 논문에서는 (그림 3)과 같이 프라이버시 보호 레벨 기반의 마스킹 차등화 방법을 제안한다.



(그림 3) 프라이버시 레벨 기반 마스킹 차등화

IV. 분석

제안 방식을 두가지의 관점에서 분석한다.

(1) 효율성 측면 : 기존의 암호화 방식을 적용하면, 검색 시 전체 암호화 데이터를 복호화 해야 한다는 단점이 있다. 이러한 점은 성능에 큰 영향을 미칠 수 있어 바람직하지 않다. 본 논문에서는 영상정보마스킹과 메타데이터 비식별화 기법을 사용하여 데이터에 대한 직접 검색이 가능하다.

(2) 안전성 측면 : 클라우드 서버상의 영상 데이터 및 영상메타정보가 악의를 가진 공격자에 의해 노출되는 경우가 발생하더라도, 마스킹된 데이터 및 비식별화된 메타정보만을 획득 가능하고, 실제 평문정보는 어디에도 보관되지 않으므로 개인정보를 식별할 수 없다.

V. 결론

향후의 지능치안 환경은 다양한 서비스를 제공할 것이며, 이를 기반으로 수준높은 지능치안 서비스를 제공받을 수 있을 것이다. 본 논문에서는 안전한 지능치안 환경을 위한 CCTV 영상보안 기법에 대하여 살펴보았다. 이를 위해, 영상데이터의 마스킹 기법과 영상메타정보의 비식별화 기법을 제안하였으며, 프라이버시 보호 레벨에 따른 차등화된 접근제어 구조를 제안하였다. 향후 CCTV 기반의 영상개인정보보호 분야에 대한 중요성이 더욱 커질 것으로 보이며, 이에 대한 지속적인 연구가 필요할 것이다.

ACKNOWLEDGMENT

이 논문은 2017년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No.2017-0-00207,클라우드 기반 지능형 영상보안 인큐베이팅 플랫폼 개발). 교신저자 (박남제, namjepark@jejunu.ac.kr).

[참고문헌]

[1] 문해민, 반성범, "지능형 영상 감시 시스템에서 프라이버시 보호를 위한 De-identification 기술 분석", 한국정보기술학회논문지, 제9권 제7호, pp.189-200, 2011.7.

[2] 이동혁, 박남제, "대용량 클라우드 스토리지 환경을 위한 ROI 기반의 효율적인 영상정보 전송 기법", 한국차세대컴퓨팅학회 하계학술대회 논문집, 2017.

토렌트를 통해 유포되는 악성코드 분석 및 대응방안

김동현*, 양준호*, 임명은*, 염흥열*
*순천향대학교 정보보호학과

Analysis of malicious codes spread through the Torrrent and their countermeasures

Donghyun Kim*, Junho Yang*, Myeongeun Im*, Heung-Youl Youm*
*Department of Information Security Engineering,
Soonchunhyang University.

요 약

최근 많은 인터넷 사용자가 토렌트를 통해 파일을 주고받는다. 토렌트란 인터넷 곳곳에 있는 파일을 찾아내 내려받을 수 있게 한 프로그램으로, 한 명의 사용자에게 자료를 전송받는 P2P 방식과는 달리 해당 파일을 보유한 여러 명에게서 파일을 일부분 조각 형태로 갖고 온다는 특징이 있다. 또한, 토렌트는 파일 배포자가 만든 공유될 파일의 정보가 담긴 시드 파일만 있으면 전 세계 어디서든 간단하게 파일을 내려받을 수 있다. 또한, 대부분의 토렌트 사이트가 무료로 운영되고 있기에 많은 인터넷 사용자들이 토렌트를 이용하는 추세이다. 하지만 이러한 토렌트 파일 공유 사이트를 통해 악성코드를 유포하는 사례가 많아 본 논문에서는 토렌트를 통해 유포되는 악성코드를 수집하여 분석하고, 관련 대응 방향을 제시한다.

I. 서론

최근 인터넷 환경이 발달하고 각종 콘텐츠가 증가함에 따라 온라인상에서 많은 파일이 공유된다. 이에 발맞추어 2001년 최초 시연된 토렌트는 기존 당나귀 등의 P2P(peer-to-peer) 프로그램을 대체하면서 빠르게 대중화되었다. 기존의 P2P 프로그램과는 달리 한 명의 상대에게 파일 전체를 전송받는 것이 아니라, 다수의 사용자에게서 조각으로 나뉜 파일을 받는 일대다 공유 방식으로, 사용자가 많을수록 다운로드 속도가 빨라지는 특징을 가진다. 토렌트로 공유되는 자료는 한 사람만 완전한 파일을 가지고 있어도 지속적인 공유가 가능하기에, 누구나 쉽게 파일을 공유할 수 있다. 또한, 대부분의 토렌트 파일 공유 사이트들은 별도의 사용자 인증이나 결제 절차를 요구하지 않고 광고 수익만으로 운영되고 있다. 다수의 익명 사용자가 무료로 파일을 다운로드하고 배포할 수 있기 때문에 이용자들이 빠르게 증가하고 있고, 저작물에 대한 불법적 공유가 가능하다는 점과 음란물의 유포경로가 된다는 점 등으로 인해 논란이 되고 있다.[1] 이 뿐만 아니라 토렌트를 통해 악성코드를 배포하는 경우가 많아졌다. 공격자는 사용자들을 속일 목적으로 자극적인 제목이나 호기심을 유발하는 제목을 설정하여 파일을 업로드 한 뒤 사용자로 하여금 해당 파일을 다운로드받고 실행하도록 한다. 정상적인 파일로 위장한 악성코드를 실행하는 순간 악성코드에 감염되어 좀비 PC, 개인정보 탈취 등의 피해를 보게 된다. 이에 본 논문에서는 토렌트를 통해 유포되는 악성코드 중 대표적인 두 가지 종류의 악성코드를 분석하고, 사용된 관련 기법을 분석하며 대응 방안을 제시한다.

II. 토렌트 특징 및 동작 원리

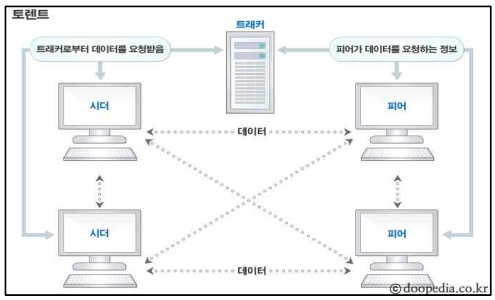
본 장에서는 토렌트의 특징과 동작 원리, 토렌트 파일을 공유하는 사이트에 대한 현황을 기술하였다.

2.1 토렌트의 정의 및 특징

비트토렌트는 미국의 소프트웨어 개발자인 Bram Cohen에 의해 창안된 P2P 프로토콜이자 이를 이용하는 소프트웨어이다. [2] 2001년에 최초로 비트토렌트 기술이 시연되면서 새로운 유형의 P2P 서비스인 토렌트가 대중화되어 사용되고 있다. 토렌트는 기존의 P2P 서비스와는 달리 한 명의 상대에게 자료나 프로그램 전체를 전송받는 것이 아니라, 다수의 사용자에게서 조각으로 나뉜 파일의 정보를 받는 '일대다(一對多)' 공유 방식이 특징이며, 사용자가 많을수록 다운로드 속도가 빨라진다.[3]

2.2 토렌트 파일 공유의 원리

토렌트 파일공유의 동작 방식은 다음 그림과 같다.

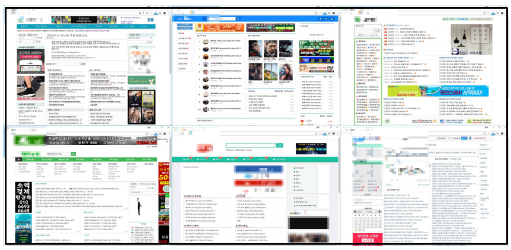


[그림 1] 토렌트 파일 공유 방식

토렌트 피어는 토렌트의 시더와 리처를 통칭해서 의미하는 용어로, 토렌트 시더는 공유 파일의 모든 조각을 가지고 있는 클라이언트를 뜻하고, 토렌트 리처는 공유파일의 일부분만 가지고 있는 클라이언트를 말한다. 토렌트 트래커는 토렌트에서 파일 공유를 위해 피어들의 정보를 관리하는 서버를 의미한다. 토렌트를 통해 파일을 다운받기 위해서는 ".torrent" 라는 확장명의 토렌트 시드 파일이 필요한데, 이는 다운로드 하고자 하는 파일에 대한 정보와 트래커에 대한 정보가 담겨있는 파일이다. 이 파일을 실행하면 토렌트 프로그램으로 연결되고, 해당 프로그램은 트래커를 통해 파일을 가지고 있는 리처나 시더에 연결을 시도하여 파일을 다운로드 받는다.[1] 파일이 어느 정도 받아지면 해당 클라이언트도 리처가 되어 공유할 수 있기에 파일 공유와 다운로드에 대한 주체 구분이 모호하다. 더욱이 완전한 파일을 가지고 있는 한 클라이언트만 존재해도 파일 공유가 지속적으로 이루어지기 때문에 파일 다운로드를 제재할 방법이 전혀 없다. 따라서 많은 공격자들이 토렌트를 통해 악성코드를 유포하고 있으며 지속적으로 토렌트 공유 사이트에 게시하여 많은 다운로드 수를 기록하고 있다.

2.3 토렌트 파일 공유 사이트 현황

토렌트는 적은 용량의 시드 파일만 존재하면 파일 공유가 가능하다는 특성 때문에 공유 사이트 운영에 대한 부담이 상대적으로 적다. 따라서 수많은 종류의 토렌트 파일 공유 사이트가 운영중이며 이들 중 대다수는 해외에 서버를 두어 불법적으로 운영 중이기 때문에 정부의 관리·감독을 받기 어렵다. 또한, 성인물 저작권이 있는 자료 또한 자연스럽게 공유되고 누구나 쉽게 다운로드 받거나 업로드 할 수 있다.



[그림 2] 불법 토렌트 공유 사이트

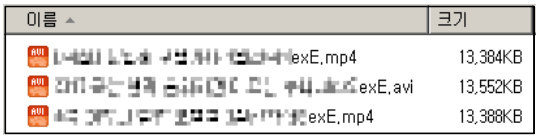
실제 한 기업의 설문조사에 따르면 대학생을 대상으로 영화화 TV 프로그램을 온라인에서 보는 방법을 질문했을 때 전체 응답자 중 27%가 토렌트를 통해 다운로드받아 시청한다고 응답했다.[4]

III. 악성코드 분석 및 결과

토렌트 공유 사이트를 통해 최신 영화나 음란물, 유틸리티 등으로 위장한 악성코드를 어렵지 않게 찾을 수 있다. 본 장에서는 토렌트를 통해 유포되는 대표적인 두 가지 종류의 악성코드를 역분석하여 사용된 기법 등을 기술하였다.

3.1 음란물 위장 악성코드

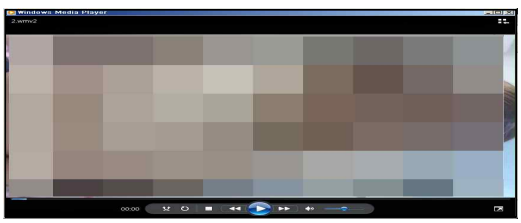
음란 동영상으로 위장하여 사용자를 속여 악성 행위를 수행하는 악성코드가 지속적으로 유포되고 있다. 해당 악성코드는 동영상 파일로 위장하기 위해 유니코드의 RLO(Right to Left Override) 기법을 사용하여 파일 확장자를 변조한다.[5] 유니코드 RLO 기법이란 오른쪽에서 왼쪽으로 유니코드를 읽게 하는 것으로 이 기법을 이용하여 파일의 실제 확장자는 '.scr, .exe'와 같은 실행파일이지만 동영상 파일의 확장자처럼 보이도록 위장하는 것이 가능하다. 또한 국내에서 널리 사용되는 동영상 플레이어의 아이콘이나 동영상의 스크린 샷 등을 악성코드의 아이콘으로 사용하여 실제 동영상 파일로 보이도록 위장하여 사용자들을 속인다.



[그림 3] RLO기법을 사용한 음란물 위장 악성코드

해당 악성코드는 용량과 이름을 다르게 하여 지속적으로 유포되고 있는 것으로 확인되며 다운로드 수 및 시더의 수를 참고했을 때 많은 사용자들이 다운로드 받고 실행하여 감염되었을 것으로 추정된다.

해당 악성코드는 실행 시 실제 음란물을 임시 폴더에 생성하여 재생시킴으로써 정상적인 음란 동영상 파일인 것처럼 행동한다.



[그림 4] 실제 음란물 재생화면

이후 레지스트리에 자동실행 등록을 하여 컴퓨터를 재부팅 하더라도 항상 악성코드가 메모리에 상주하게 하며 사용자의 감염된 PC 환경 정보를 공격자의 C&C 서버로 전송한다. 이후 C&C 서버에서 응답한 정보를 토대로 공격자가 원하는 파일을 자동으로 다운로드 하고 실행하도록 할 수 있다. 현재는 서버가 동작하고 있지 않지만, 공격자가 언제든지 자신이 원하는 실행 파일을 감염 PC에서 실행할 수 있는 일명 좀비 PC가 된다. 랜섬웨어에 감염시키거나 키로깅, 공인인증서 탈취, DDoS등 각종 이차적인 악성 행위에 노출될 수 있다.

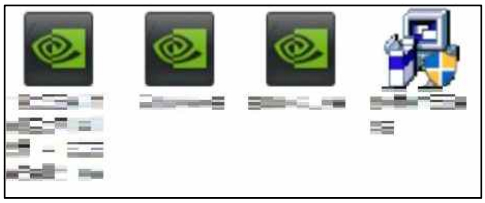


[그림 5] 파일 다운로드 및 실행 코드

3.2 유틸리티 프로그램을 위장한 악성코드

유용한 프로그램으로 위장하여 사용자들의 관심을 끌어 다운로드 및 실행을 유도하는 악성코드이다. 해당 악성코드는 윈도우 10 업데이트 파일, 시스템 정리 프로그램, 모자이크 제거 프로그램 등의 이름으로 유포되었으며 PC를 다양한 방면으로 사용하려는 이용자들을 유혹하여 감염되도록 한다.

해당 악성코드는 실행 시에 유명 그래픽카드 회사의 아이콘으로 위장하며 정상적인 기능을 수행하는 것처럼 하여 사용자가 악성코드에 감염되었다는 사실을 인지하지 못하도록 한다.



[그림 6] 유명 그래픽카드 회사 아이콘으로 위장

또한, 사용자 컴퓨터의 “C:\Windows” 폴더와 “%Temp%” 폴더에 자신을 복제하여 숨겨두고 레지스트리 및 윈도우 자동실행 파일로 등록하여 사용자가 컴퓨터를 재부팅 하더라도 악성코드가 자동으로 다시 동작할 수 있도록 한다. 이후 사용자의 키보드 입력을 기록하고 일정한 크기가 될 때마다 ZIP 파일로 압축하여 C&C 서버에 전송하며 이로 인해 사용자의 사생활 및 개인정보가 공격자에게 노출될 수 있다.



[그림 7] 키 입력 상태를 로그로 저장하는 코드

IV. 대응방안

공격자는 주로 많은 사람들이 검색하고 다운로드하는 최신 음악이나 영화 및 화제 또는 음란성 제목으로 악성코드를 업로드 하고 사용자의 심리를 자극하여 다운로드를 유도한다. 이러한 과정을 통해 감염되는 악성코드는 좀비 PC, 개인정보 유출, 원격 제어 등의 악성 행위를 하며 2차적인 공격을 위한 유용한 자원으로 쓰일 수 있다. 사용자는 신뢰할 수 없는 토렌트 파일로 다운로드 받은 파일은 함부로 실행하지 말고 항상 주의를 기울여야 한다.[3]

특히 다운로드 받은 파일 중 실행이 가능한 특정 확장자(.zip, rar, .exe, .com, .bat, .scr 등)가 포함되어 있거나, RLO 기법을 사용해 확장자를 위장한 파일이 있다면 의심해볼 필요성이 있다. 뿐만 아니라 만약 다운로드를 받아 실행했더라도 추가적인 피해를 방지하기 위해 민감한 웹사이트나 인증서의 비밀번호를 주기적으로 갱신하고 백신 프로그램 및 안티 익스플로잇 프로그램을 설치하여 최신버전으로 운용하고 윈도우나 각종 어플리케이션 또한 업데이트를 통해 최신버전으로 유지해야 한다.

V. 결론

본 논문에서는 최근 널리 사용되는 토렌트를 통해 유포되고 있는 악성코드를 수집 및 분석하였다. 분석한 샘플들은 직접적인 악성 행위는 수행하지는 않지만 공격자가 원하는 파일을 자동으로 다운받아 실행하도록 하거나 키로깅 정보를 수집하여 사용자 계정 및 암호를 탈취하여 악용하는 등의 이차적인 피해가 더 우려되는 상황이다. 토렌트를 통해 배포되는 악성코드는 토렌트 프로그램 자체의 취약점을 사용한 것이 아닌 사용자가 직접 파일을 다운받고 실행해야 감염되는 구조이다. 그 때문에 공격자는 사용자가 파일을 직접 다운받고 실행하도록 유도하기 위해 사용자의 심리를 고려하면서 점점 더 정교하게 위장할 수 있으므로, 토렌트 이용자의 각별한 주의가 필요하다.

[참고문헌]

[1] 두산백과, https://www.doopedia.co.kr/doopedia/master/master.do?_method=view&MAS_IDX=151211001517595

[2] BitTorrent, The Basics of BitTorrent, http://help.bittorrent.com/customer/en/portal/articles/178790

[3] 송민선, 비트토렌트(BitTorrent) 규제 연구 : 저작권법적 쟁점과 비트토렌트 이용의 기존 매체 대체효과를 중심으로

[4] 파이낸셜뉴스, “대학생 절반 ″토렌트·스트리밍 등 불법 콘텐츠 이용한다”, 2017년 5 월 18일

[5] The Unicode Consortium, General Punctuation, https://www.unicode.org/chats/PDF/U2000.pdf

[6] 김동현, “토렌트에서 ‘음란 동영상’ 받았다간...악성코드 감염돼 ‘좀비PC’ 전락”, 순천향대학교 SCH사이버보안연구센터, 보안뉴스, 2017년 8 월

[7] 양준호, “모자이크된 영상·이미지 복구시켜준다고? 악성코드에 감염될 뿐”, 순천향대학교 SCH사이버보안연구센터, 보안뉴스, 2017년 8월

2017 한국정보보호학회 동계학술대회

A5 영어논문

윤 지 원 (고려대)

- A Blockchain-based Audit Record Management
  Bruno Joachim, Youngho Park and Kyung-Hyune Rhee (Pukyong National University)
- Toward Machine Generated Passwords
  Simon Woo (SUNY Korea), Wenzhe Li (USC), Hyeran Jeon (San Jose State University)
- TVT: Typed Virtual Table for mitigating VTable Hijacking
  Jeongoh Kyea, Eunsoo Kim, Dongkwan Kim and Yongdae Kim (KAIST)
- 블록체인 기반의 신뢰성 있는 응용 서비스 설계연구 (Reliable Application Service Design based on Block Chain Platform)
  박리원, 장항배 (중앙대)

A Blockchain Based Audit Record Management

Bruno Joachim Kweka*, Youngho Park**, Kyung-Hyune Rhee**

*Interdisciplinary Program of Information Security, Graduate School
PukyongNationalUniversity.
**Department of IT Convergence and Application Engineering
PukyongNationalUniversity

Abstract

Information and record management in organizational level or business its important however, the transformation of handling digital records brought challenges. Challenges articulated in this paper are managing and tracking of user logs for instance who access the data, what operations they performed and when, and it is challenging when it is used in the companies that have many workers and different branches across the word. Many record management systems depend on a central database to access the data, and this can be a single point of failure and as well as be a point of conducting malicious breaching of data (hackers). The solution proposed in this paper, By using blockchain technology together with distribution hash table, and we focus to capture timestamps of users or service try to access user data and maintain them into the blockchains. because of its immutable and no central authority.

I. INTRODUCTION

Records management is involving in organizing and managing the flow of information. Includes storing, retrieving, tracking, identifying, and deleting of records. It is a field of management responsible for the efficient and systematic control of the creation, receipt, maintenance, use, and disposition of records, including the process for capturing and maintaining evidence of and information about activities and transactions in the form of records [1]. The characteristics of records are authenticity, reliability, integrity, and usability. As a lot of information being stored and volume becoming so huge, that is where a variety of challenges arose. For instance, handling of records in a decentralized environment is a challenge [2] because a lot of computers will be engaging in the transfer, creating uploading, accessing or sharing compare to centralized system hence could lead to the disaster. Another challenge in healthcare [3] records in terms of privacy of patient data that makes the clients have concerns about particular data (personal data: credit card, illness report) if those data are being accessed by the outsiders or hackers. In 214 the US Health Insurance Portability and Accountability report since 2012, security breaches have increased by 138% and the trend has continued into 2015. In their reports, they list kinds of attacks such at credential weakness (username and password), insider and privilege misuse this attack is an insider attack, could be from web administrator or software developer intentionally create a bug in the software and

cause leakage of user information.

There are many challenges faced in government level, business level, concerning records management. Therefore the aim of this paper is to try to use this novel approach called blockchain based audit record management, with the motive to overcome some of the challenges such as enhanced data integrity, provide authenticity and trustworthy, easy on conducting audit of logs and be able to maintain access logs from participants who will be engaging in accessing of shared data. This system target is to give a users features of being able to control and audit which data are stored, how are they being accessed and shared with other external participants.

II. BACKGROUND AND RELATED WORKS

A lot of models about records management have been proposed by the researchers in different varieties of fields such in health records, government records, and other areas. One model developed in health (South Africa case study) for the reason of integrating between government and private hospital institution so to enable sharing of patient record [4] but this model have several setbacks such as single point of failure because of using central database hub for stored patient data and model did not mention or give a detail about how it will guarantee user privacy.

Our approach works differently, it employs a technology called blockchain which relies on peer to peer (P2P) network (distributed environment) and this approach promote transparent of records towards the nodes because of its database nature that no one own, it is publicly shared and can be viewed by all members who use the blockchain network

2.1 Blockchain

Blockchain technology recently has attracted a large number of researchers and stakeholders, like in finance, healthcare, real estate and government sectors. The blockchain is a distributed public ledger which includes all the transactions that executed in the system [6]. It relies on P2P network whereby every node store and maintain the copy of the blockchain ledger. No central authority manages the blockchain database. Miners in blockchain receive incentives for supporting the systems with their computational power for conducting proof of work, also validating the transactions before included in the blocks. They involve in the creation of blocks. Decentralization and security properties of blockchain architecture attracted researchers to develop a various application such as smart contracts, identity management and much other non-finance application.

1.2 Distributed Hash Table

By definition, distributed hash table (DHT) is Hash tables which use (key, value) pair and hash function to map the location of the value (Value = hash (key)).The reason of use this DHT in this model is to enhance search performance (simplify resource location) also it is a convenient mechanism that can work in the distributed network. Apart from that DHT can support many services to be built up on top of its interface such as file sharing, archival storage. Database and many others which will depend on your project goal.

2.2 Onion Routing (TOR)

It is one of the technique allowing anonymous and privacy-preserving communication in the environment of the network. One of popular onion routing is Tor Usage of the onion is to prevent surveillance

and packets traffic analysis. Identity, location and network activity of each variety of nodes are protected by hiding to the external observers (hackers) both contents and routing information of the node's traffic. Because blockchain network does not provide much network security mechanism when data are being accessed via DHT then, in this model we add this layer of security to hide both communication content and location. Therefore onion routing guarantees protection of context privacy

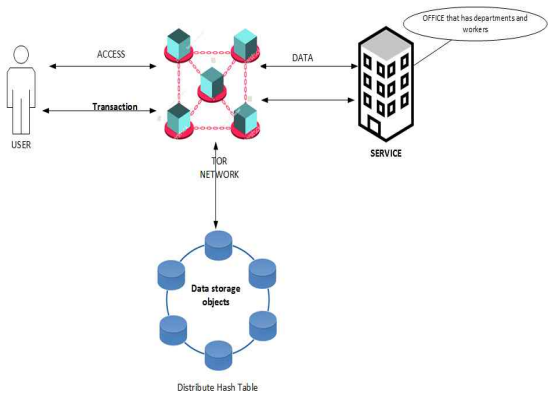


Figure 2. Conceptual Framework of Blockchain - based Audit Record Management.

First, a user can access, store or perform any action to his data stored in DHT nodes via blockchain as an intermediator, whereby in the blockchain timestamps will be captured through the transactions. Any kind of action performed, such file operations by a user will be recorded in the blockchain with timestamps. If any other services from external parties request to access user data, firstly user and the service requester wants to access the user data has to share some keys like hash pointer, that pointer point the data where it is stored in DHT node and both user and guest that request service has to sign key pairs and record the timestamps and agree upon what actions will be performed. And communication from blockchain network to DHT nodes will be anonymity because of onion routing mechanism (tor). Any sensitivity data stored in DHT must be encrypted with the well know encryption mechanism. Because in this system cannot provide the encryption of data stored in DHT

III.AUDIT RECORD ARCHITECTURE

Blockchain-based record management built on blockchain environment which will provide the ability to audit data logs, provide authenticity and trustworthy, avoid forgery of documents. Model achieves following objectives. In user aspect: In blockchain, there is no use of username or password, and only public and private address are used for a client to be authenticated and access the documents, therefore authentication mechanism is difficult compared to username and password. Tamper-proof: all nodes will maintain the copy of the records and share among the nodes. This model will record timestamped logs of all users operations. This method will detect any nodes try to misbehave. Furthermore, blockchain controls the data (key-value) store to the distributed hash table and holds only hash value as a pointer to the data. Only user and the service requester can query the data using a hash pointer to the data. Therefore this makes it easy to know what service access your data, what action did, because of the timestamp. It also makes easy for traceability if any kind of illegal activities happened. No central authority involved in manage the blockchain database. Guarantee privacy Preservation: one unique feature of blockchain is to guarantee user anonymity using the special cryptographic technique, the user can hide his identity because of the blockchain cryptographic mechanism [5] without revealing any kind of information. Data

validity: All recorded transactions are globally transparent, miners nodes do the major task to validate the transactions and create the blocks and make sure there is consistency in the block and they achieve it use consensus protocol so-called proof of work, thereafter a number of nodes provide confirmation for every block. This data method ensures data integrity.

IV. SECURITY ANALYSIS

The authentication mechanism uses public and private keys hence no authentication weakness. Privacy ensures to provide anonymity(cryptographic protocol) and also assure data ownership because private key is resided in user side. Easy to analyze and manage access logs (audit) as in blockchain any transaction (query of a data) performed by a user it is stored in the block and it is immutable, hence make it is easy to track and conduct an audit in case of any fraud happened. Onion routing protect against eavesdropper (hackers) since onion prevent traffic analysis and packet eavesdropping.

V. PERFORMANCE ANALYSIS

We perform analysis of the impact of block size by selected the block size of 1MB 150nodes, 16miners and we generated 100 blocks per different interval of time. In figure number 3 shows the block simulated varies against the time. The longer the delay can cause blockchain forking and sometimes double-spending attacks. In the bandwidth usage was fair

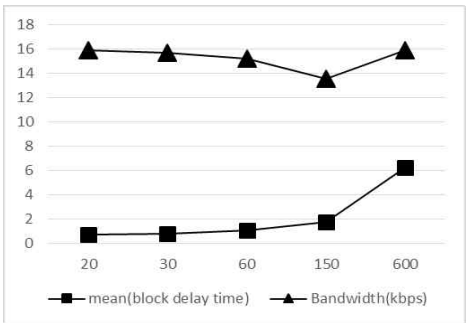


Figure 3 Block Propagation

CONCLUSION

In this paper, we present the novel model of record management based on blockchain technology for enhancing data validity, integrity, maintain the access logs of users using timestamp techniques that provides trustworthy to a user of a system and ability to control and audit his data, avoid of forgery or modification because all records stored in blockchain are immutable, and its database is public access to all node.

Acknowledgment

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (2017-0-00156, The Development of a Secure Framework and Evaluation Method for Blockchain).

References

[1] Svärd, P. (2017). Enterprise Content Management, Records Management, and Information Culture Amidst E-Government Development. Chandos Publishing, 22.

[2]Office, U. S. (July 8, 2003). ELECTRONIC RECORDS. 13.

[3] Keelan, L. B. (1 January 2012). The challenges in making electronic health records accessible to patients. Journal of the American Medical Informatics Association, Volume 19.

[4] Millham, R. (2014). An Enhanced Cloud Computing Model for. Cloud Computing in Emerging Markets (CCEM), IEEE International Conference on, 3.

[5] Saberhagen, N. v. (2013). Cryptonote v 2. 0. 13.

[6] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

Toward Machine Generated Passwords

Simon S. Woo*, Wenzhe Li-, and Hyeran Jeon+

*Computer Science Dept. SUNY Korea

-Computer Science Dept. at University of Southern California

+Computer Engineering Dept. at San Jose State University.

Abstract

As the number of online accounts users create increases, the need for generating strong and memorable passwords is also escalating. In this work, we apply a machine-learning approach to generate usable and strong passwords. In particular, we train Character-Level Recurrent Neural Networks (Char-RNNs) with publicly available leaked password dataset. We explore different neural network parameters and generates passwords stronger against offline attack. We conduct a user study with 103 participants from Amazon Mechanical Turk. Although the overall successful recall rate was 55.2% for the exact match, 79.1% of participants were able to remember the underlying structure and patterns of the assigned password. Hence, our work shows promise in generating usable and strong passwords from neural networks.

I. Introduction

Recurrent Neural Networks (RNNs) have achieved great success in computer vision, text modeling and language generation. In particular, Karpathy [1] implemented Character-Level Language Models (Char-RNNs) and demonstrated the effectiveness in generating words, sentences, and paragraphs. Melicher et al. [2] and Neef [3] used Char-RNNs for cracking passwords, and training with leaked passwords. However, there is little research in understanding passwords generated from Char-RNNs, and potential uses of Char-RNNs for generating memorable and strong passwords. We aim to explore the following question: Can Char-RNNs generate usable and secure passwords for humans? Our ultimate goal is to test the feasibility of machine generated passwords. We use Char-RNNs as shown in Fig.1. by Karpathy [1] to generate passwords. We started from analyzing generated passwords from Char-RNNs with different parameters. Then, we chose the best Char-RNN model that was

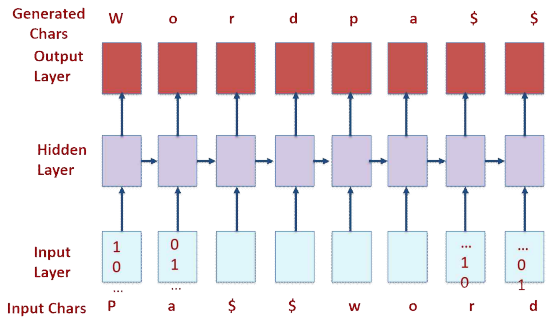


Fig. 1. Char-RNN with 1 hidden layer

similar to human created passwords, and generate strong and usable passwords as shown in Fig 2. For memorability, we evaluated the generated passwords with 103 participants from Amazon mechanical Turks (Mturks). Figure 2 describes entire our password generation and user study flow.

Our result shows that 79.1% of participants remember the gist and patterns of the machine generated passwords, which can be used for future automatic password generated system. For strength, we used the Monte-Carlo method by Dell'Amico and Filippone [4].

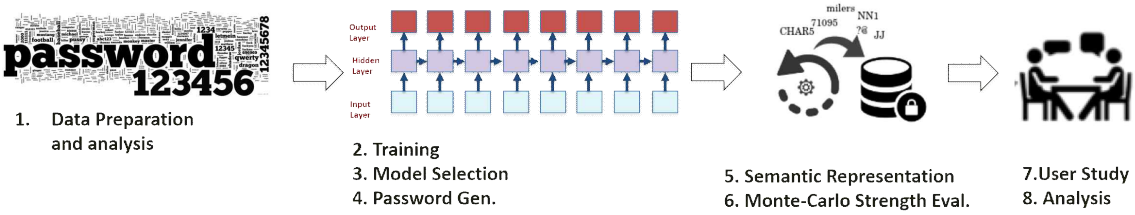


Fig. 2. Overall Password Generation and User Study Flow

II.Related Work

Neef[3] was the first to show the use of Char-RNNs for password cracking. Also, Melicher et al.[2] demonstrated LSTM can accurately estimate password strength. However, their focus was on cracking and measuring the password strength. On the other hand, we explore using Char-RNNs for usable and secure password generations.

III. Password Dataset

We use previously leaked passwords from the RockYou password dataset for training the Char-RNNs similar to other research[2]. After removing passwords that do not meet 3class8 requirement, we only use 250K passwords for training, testing, and generations.

IV. Model Selection

It is imperative to choose the accurate Char-RNN parameters, in order to generate usable and secure passwords. Hence, we extensively analyzed different Char-RNN parameters, and their impacts on generated passwords with NVIDIA Tesla K80 GPU [6]. We changed the number of layers (L) from 1 to 3, and the number of hidden cells (N) ranging from 64, 128, 512, 1024, 2048, and 4096, which control a model complexity. Further, we varied the drop rate (D=0.3, 0.5, 0.7, and 0.9), which is related to the regularization. Also, the value of temperature (T) was chosen from 0.3, 0.5, 0.7, to 0.9, which varys the smoothness in the softmax function. We limited the maximum character sequence length to be 16 during training, since we expect the maximum password length would be less than 16 characters for humans to remember. As a baseline, we used LSTM with the following parameters: ($N/L/T/D$) = (128/1/0.9/0.9) for a comparison.

1. Password Length vs. Char-RNN parameters

We varied one parameter at a time from the baseline model, and analyzed the length of the generated passwords as a metric to compare machine generated passwords vs. human generated passwords (test set).

Num. of hidden layers (L): the single layer is sufficient. With T=0.9, we varied the number of hidden layers (L) from 1 to 3, while keeping all other baseline parameters the same. We found that the baseline model with L=1 generates more similar passwords to the test set than those from 2 or 3 layer configuration. Surprisingly, unlike other applications, the single layer Char-RNNs outperforms against multi-layer Char-RNNs. And the single layer appears to be the best, yielding the lowest cross-entropy loss. For multi-layer cases, Char-RNNs tend to over-estimate and generate many passwords that are either less than 8 characters or greater than 16 characters, which are not suitable for strong and usable passwords.

LSTM Cell size (N): More cells are better. As we increase the cell size, the password length distribution became clearly closer to the test set. So, more cells generate passwords that are close to humans generates in the test set.

Temperature (T): Lower temperature helps. We varied T from 0.3 to 0.9 by 0.2, while fixing all other parameters the same from the baseline mode. As T decreases, the baseline model tend to generate more similar passwords in length compared to those from the test set.

Drop Rates (D): the drop rates has the least impact among all parameters. We varied the drop rate from 0.9 to 0.3 with the cell size N=4096. Although the drop rate impacts regularization, the drop rate appears to have the least impact among all other parameters. However, the drop rate impacts the cross-entropy loss and internal structure of passwords, where the drop rate = 0.7 yields the lowest cross-entropy loss.

2. Best model parameters

We chose the best model with the minimum overall cross-entropy loss. With baseline configuration ($N/L/T/D$) = (128/1/0.9/0.9), we obtained 3.7094 for the test set cross-entropy loss. The best model was ($N/L/T/D$) = (4096/1/0.5/0.7), yielding 3.1639 for the test cross-entropy loss. This minimum entropy loss parameters correlated with the intuitions we provided above.

V. User Study Design

We used Mturks to recruit participants. The qualification needed for each participant was at least 1,000 completed Human Intelligence Tasks (HITs) and > 95% HIT acceptance rate. We asked each participant to create one password for an imaginary server. After two days, each participant was invited to return to the study and attempt to authenticate with her password. We paid 25 cents for password creation and 30 cents for the authentication task, respectively. Participants were paid for authentication regardless of whether they recalled their password correctly or not. We allowed at most five trials to authenticate per password. We had automated detection of copy or paste attempts in our login forms, and we rejected the users who were detected to perform either of these two actions.

VI. Password Generation

We pre-generated passwords from the best Char-RNN models, and measured the strength of generated passwords using Monte-Carlo method by Dell'Amico and Filippone [4]. Among these passwords, we only use passwords that are stronger than offline attack strength limit 10^{14}. Then, we obtained 30K new passwords, which do not have any overlap with RockYou testset. When a user arrives, we randomly selected one out of 30K pre-generated passwords from the best model. We provided an option for a user to choose a new password, if a generated password was not the one a user liked.. Once users chose, they were allowed to practice twice right after the selection of their passwords. If users failed to authenticate, they were redirected and asked to choose a new passwords.

VII. Results

Total 103 participants created a password. Among them, 67 participants returned for authentication task with 65% return rate. The password recall is successful only if all characters are matched. The average success rate is 55.22% as shown in Table 1. The exact match result is not great since users are assigned to machine generated random passwords. We compared our result with other similar research [5] conducted on the same 2 day recall interval. The average recall rate was around 49%,, when users were assigned with system-generated passphrases. Hence, users recalled slightly higher in our approach than prior research[5]. Next, we manually inspected all the failed passwords and examined if users correctly remembered the underlying semantic structure of the

assigned passwords. For example, the passwords ``11SAMLE-shark'' (assigned) and ``32PSSWRD-srch'' (authenticated) are completely different. However, their semantic structure were similar such as (2digits)(chars)(1symbols)(chars). This user correctly remembered the semantic compositions of the password: starting with 2 digits, and 2 char sequences are connected with ``-''. Also, user remembered the position of the uppercase letter. However, a user failed to fill in the exact digits and characters. Although they did not provide the exact characters or digits, total 79.1% of users were able to correctly remember the underlying high level structure of the original password.

Succ. Rate	Exact Match	Structural Match
Best Model	55.22%	79.1%

Table 1. Successful recall rate after 2 days

Besides semantic structure, we analyzed the types of mistakes users frequently made. About 6.67% users used the related words (e.g., #1kidscouck vs. #1boyscouck). Total 23.33% of participants typed extra characters or missed characters (i.e, Alysse92 vs. Alysee92), and another 23.33% of participants missed digits (i.e, **haina2** vs. **haina22**). About 46.67% of participants provided completely wrong passwords. Hence, more than half of the users remembered the overall structure and the order of password segments. However, they failed to provide the exact digits or character sequences. We hypothesize that users were assigned to remember random digits or chars. So, it was difficult for them to remember exact characters. However, it is surprising to see that more than 79% of users were successful in remembering the high level password patterns. Hence, this finding can be used for password generation systems, where the system suggests structural semantic pattern suggestions to users. Once the pattern suggestions are given, users can provide their own specific digits, and chars. We believe this can improve the memorability of the passwords.

Failure Type	Percentage
Missing/adding extra chars	23.33%
Missing/adding extra digits	23.33%
Synonyms	6.67%
Completely wrong	46.67%

Table 2. Failure Analysis

VIII. Conclusion

The Char-RNNs can learn and generate passwords that are similar to human-created passwords. Since humans can better remember the overall structures of the generated passwords, it will be an interesting future research to explore neural networks for suggesting strong and memorable password patterns.

[References]

[1] A. Karpathy. Multi-layer recurrent neural networks (lstm, gru, rnn) for character-level language models in torch. 2016

[2] W. Melicher, B. Ur, S. M. Segreti, S. Komanduri, L. Bauer, N. Christin, and L. F. Cranor. Fast, lean, and accurate: Modeling password guessability using neural networks. In USENIX Security Symposium, pages 175 - 191, 2016.

[3] S. Neef. Using neural networks for password cracking. https://0day.work/using-neural-networks-for-password-cracking/, 2016

[4] M. Dell'Amico and M. Filippone. Monte carlo strength evaluation: Fast and reliable password checking. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pages 158 - 169. ACM, 2015.

[5] R. Shay, et al., Correct horse battery staple: Exploring the usability of system-assigned passphrases. In Proceedings of the eighth symposium on usable privacy and security, page 7. ACM, 2012.

[6] NVIDIA Tesla K80 GPU http://www.nvidia.com/object/tesla-k80.html

TVT: Typed Virtual Table for mitigating VTable+ Hijacking

계정오, 김은수, 김동관*, 김용대*

KAIST 정보보호대학원, *KAIST 전기 및 전자공학과

Title

Jeongoh Kyea, Eunsoo Kim, Dongkwan Kim*, and Yongdae Kim*

Graduate School of Information Security, KAIST
*Department of Electrical and Engineering, KAIST

Abstract

Object-Oriented Programming style is essential to building large program. In C++, virtual functions are used to support polymorphism, which is one of the major characteristics of OOP. However, these virtual functions can be overwritten by attackers to take the control flow of the target program, which is so-called “vtable hijacking”. Vtable hijacking is widely used for exploiting application such as a browser. In this paper, we propose a novel defense technique, TVT, to mitigate vtable hijacking using only a binary without the source code. TVT can be applied to a real-world application to prevent vtable hijacking attacks. Our experiment shows that TVT makes reasonable overhead on the Firefox browser by preventing vtable hijacking attack which cannot be prevented by previous binary level mitigations.

I. Introduction

Object-Oriented Programming (OOP) understands a program as a collection of objects, not a list of instructions. There are numerous programming languages which support OOP such as C++, Java, C#, and so on. In these languages, although a program has two function calls whose names are the same, their implementation may differ along the type of the target object or the number of the arguments. This character is called "Polymorphism". In C++, to support this polymorphism, the reserved word `virtual` is used.

+ This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2017-2015-0-00403) supervised by the IITP(Institute for Information & communications Technology Promotion)

```
 4 class animal{
 5 public:
 6         virtual void bark(){}
 7 };
 8
 9 class dog: public animal{
10 public:
11         virtual void bark(){cout << "bowbow" << endl;}
12 };
13
14 class cat: public animal{
15 public:
16         virtual void bark(){cout << "meow" << endl;}
17 };
18
19 int main(){
20         animal *a;
21         int i;
22         cout << "1. dog\n2. cat\n==>";
23         cin >> i;
24         if( i == 1 )
25                 a = new dog();
26         else
27                 a = new cat();
28
29         a->bark();
30 }
```

Figure 1. C++ polymorphism Example

In Figure 1, the cat and dog class inherit the animal class. In line 29, bark() is called. According to the dynamic type of variable `a`, bark() varies. Before calling bark(), the program cannot determine which method will be

called because the type of variable `a` is determined by user input `i`. Therefore, the compiler compiles this callsite into an indirect call instruction and makes a function table for each class, which is called as "virtual function table" (vtable). When a constructor generates an object, the object has a vtable pointer, which points to its own vtable. Using this pointer, a program can decide which function will be called.

Because a called function depends on a vtable pointer, if attackers hijack this pointer, they can change the called function. In other words, attackers can hijack the control flow. In fact, vtable hijacking is usually used to exploit applications, including browsers. To prevent vtable hijacking, there have been many mitigations proposed. These mitigations can be generally divided into two parts by the starting point, from the source code or the binary. Source-based mitigations such as gcc VTV[1], and SAFEDISPATCH[2], provide strong prevention because a source code gives detailed information of the target program. On the other hand, binary-based mitigations, such as Vfguard[3], VTint[4], and VTPin[5], has lack of information such as a type or a class hierarchy of the target. Therefore, binary-based mitigations give weaker security than source-based ones. In fact, none of the state-of-the-art binary-based mitigations can prevent Counterfeit Object Oriented Programming (COOP) [6]. Moreover, there are numerous programs without the source code or the legacy binaries, where a source-based approach is not applicable. Therefore, binary-based mitigation is essential.

In this paper, we show a new binary level mitigation, Typed Virtual Table (TVT). TVT utilizes the fact that after inserting a vtable pointer at a constructor, the pointer cannot be changed in a normal flow. From this fact, TVT records the type of each object dynamically and then checks if the type is the same at its callsite. In addition, because vtables exist in read-only memory, TVT checks if vtable address is readable. We deploy our method to tiny vulnerable binaries and a firefox browser. TVT can stop not only previously known exploits but also the COOP attack.

II. TVT Overview

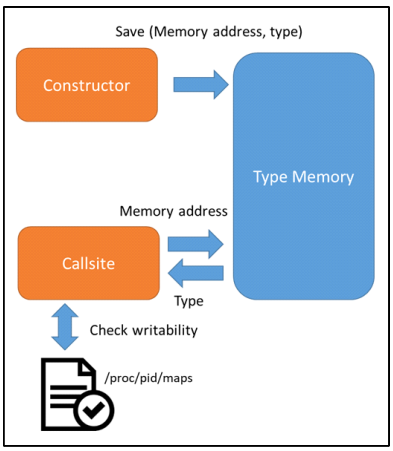


Figure 2. TVT Overview

Each object in C++ is made in a constructor function, which inserts vtable pointer into the start of the object's address. At this point, the type of the object is determined. Therefore, we save the type at the constructor and check that the type of the object is the same with the saved one at the callsite. Moreover, because vtable always located in read-only memory, a memory writability check should be conducted at the callsite. For this process, TVT finds the constructor and callsite and then instrument codes.

III. Implementation

As shown in Figure 3, TVT consists of two parts: binary patching and making runtime library. For patching binary, TVT finds vtable firstly. After then, using vtable information, TVT finds constructors and callsites. Finally, TVT patches constructors and callsites. TVT uses (1) IDAPython script for finding vtables, callsites, and constructors,

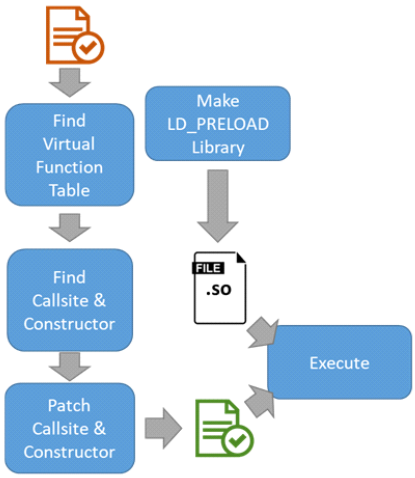


Figure 3. TVT patch flow

(2) Python for patching binaries, and (3) C code for LD_PRELOAD library.

3.1 Find Virtual Function Tables

A vtable is a collection of function pointers in read-only memory. In addition, the vtable address is referenced in the constructor code. Using this fact, we can find vtables.

First, TVT searches all values from the start address of the read-only section. If TVT meets the function address, it checks if the reference of the address is in the code section. If so, TVT determines this address as vtable address.

3.2 Find Constructor

For finding constructor, TVT uses the reference of vtable information found in Section 3.1. From the reference address of the vtable address, TVT searches forward to find the instruction which inserts the vtable value into the memory using taint analysis.

3.3 Find Callsites

Vtable callsites must be an indirect call instruction. Thus, TVT linearly searches all indirect callsites. After that, TVT filters the vtable callsites among all indirect callsites using their characteristics: 1) setting this pointer, and 2) the address which has the vtable is the same as this pointer address.

From each indirect callsite, TVT searches backward whether the indirect callsite is satisfying the characteristics.

```
.text:000000000000193F mov     rax, [rbp-68h]
.text:0000000000001943 mov     rax, [rax]
.text:0000000000001946 mov     rax, [rax]
.text:0000000000001949 mov     rdx, [rbp-68h]
.text:000000000000194D mov     rdi, rdx
.text:0000000000001950 call    rax
```

Figure 4. Example of virtual table callsite

For example, in Figure 4, the register used for the indirect call is `rax`. TVT searches backward for finding where the register `rax` comes from. At address 0x1946, the `rax` register comes from `[rax]`. TVT finds the origin of `rax` again. At address 0x1943, TVT can identify that `rax` also comes from `[rax]`. While searching backward, TVT also finds this pointer, `rdi`, which comes from `rdx`. After finding these registers, TVT checks if the origins of `rdx` and `rax` are the same. In this example, `rdx` and `rax` are the same as `[rbp-0x68]`. Therefore, TVT concludes this address is a vtable callsite.

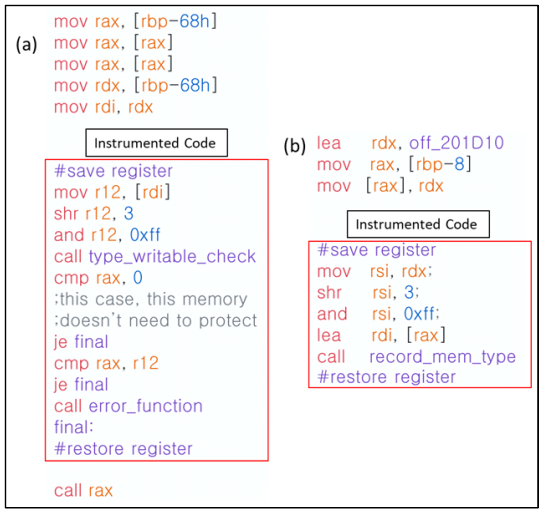


Figure 5. (a) and (b) show the instrumentation code of callsites and constructors.

3.4 Patch Constructor and Callsites

After finding constructors and callsites, TVT patches them. At the constructor, TVT instruments the code which saves the type of the object address into type memory. At the

callsites, TVT checks the type of the object address and writability. To check writability, we use `/proc/pid/maps` file. Because the vtable address is aligned by 4 or 8 bytes, we just use the type as `(object_address >> 3) & 0xff)` in x86_64. Figure 5 shows the callsite and constructor instrumented code.

IV. Evaluation

We evaluated TVT as security and performance aspects.

Binary	Vulnerability type	protected
FF 3 (cve-2011-0065)	vtable injection	YES
FF 10 (cve-2013-0753)	vtable injection	YES
Tiny Program	vtable reuse	YES

Table 1. Vulnerabilities protected by TVT

4.1 Security

We applied TVT technique to a tiny program and the Firefox browser. The tiny program has a vtable reuse vulnerability which is hard to be prevented by other binary-based mitigations. For the Firefox browser, we tested two Firefox CVEs: CVE-2013-0753 and CVE-2011-0065. For all the test cases, TVT could successfully protect them.

4.1 Performance

To check the performance overhead, we used the SPEC 2017 benchmark and a browser benchmark site+. Table 2 shows the overall result of benchmarks. For SPEC 2017, some benchmarks made over 40% overhead because these tests use an extremely large number of vtable calls. However, other cases including Firefox showed reasonable overhead.

V. Conclusion

Vtable hijacking attacks are widely used for exploiting several applications. In this paper, we proposed a novel technique, TVT to mitigate vtable hijacking attacks. TVT checks type integrity of object and vtable writability. Because TVT can be applied to a binary only without the source code, it can be used widely.

Benchmark	Origin	TVT	Overhead
SPEC 2017			
507.cactusBSSN_r	193.6169	194.1970	0.3%
508.namd_r	166.6399	166.5166	-0.07%
510.parest_r	371.2309	376.1824	1.33%
511.povray_r	326.9721	329.0326	0.63%
523.xalacbmk_r	247.1640	362.9115	46.83%
526.blender_r	218.3159	218.7828	0.21%
531.deepsjeng_r	241.0260	241.1859	0.07%
541.leela_r	394.2530	396.3287	0.53%
620.omnetpp_s	366.6408	508.7688	38.76%
Firefox 55			
ARES.6	160.50ms	209.88ms	30.76%
MotionMark	104.53pt	91.45pt	12.51%
Speedometer	39.4pt	31.9pt	19.04%
JetStream	99.415pt	89.282pt	10.19%

Table 2. TVT benchmark result for SPEC 2017 and Firefox 55

[참고문헌]

[1] TICE, Caroline. Improving function pointer security for virtual method dispatches. In: GNU Tools Cauldron Workshop. 2012.

[2] JANG, Dongseok; TATLOCK, Zachary; LERNER, Sorin. SafeDispatch: Securing C++ Virtual Calls from Memory Corruption Attacks. In: NDSS. 2014.

[3] PRAKASH, Aravind; HU, Xunchao; YIN, Heng. vfGuard: Strict Protection for Virtual Function Calls in COTS C++ Binaries. In: NDSS. 2015.

[4] ZHANG, Chao, et al. VTint: Protecting Virtual Function Tables' Integrity. In: NDSS. 2015.

[5] SARBINOWSKI, Pawel, et al. VTPin: practical VTable hijacking protection for binaries. In: Proceedings of the 32nd Annual Conference on Computer Security Applications. ACM, 2016. p. 448-459.

[6] SCHUSTER, Felix, et al. Counterfeit object-oriented programming. In: IEEE S&P. 2015.

+ http://browserbench.org/

블록체인 기반의 신뢰성 있는 응용 서비스 설계연구

박리원*, 장항배**

*중앙대학교 일반대학원 융합보안학과,

**중앙대학교 산업보안학과

Reliable Application Service Design based on Block Chain Platform

Lee-Won Park*, Hangbae Chang**

*Department of Security Convergence, Graduate School, Chung-Ang University,

**Department of Industrial Security, Chung-Ang University

Abstract

There is much hype around the Blockchain technology. The concept is innovative; a distributed ledger that allows peer-to-peer value exchange. Trust is the key that allows for such technology, and transparency and permanence is what makes the Blockchain even more valuable. However, despite what the Blockchain has to offer, the technology is not for everyone. Blind implementation of the Blockchain will not help a company gain increased efficiency, but instead may result in unwarranted costs and confusion of workflow. Thus, in this paper, we define 4 key drivers of the Blockchain technology and examine 3 companies that utilize one of four Blockchain key drivers. The process provides insight on what companies should consider for the implementation of the Blockchain.

I. Introduction

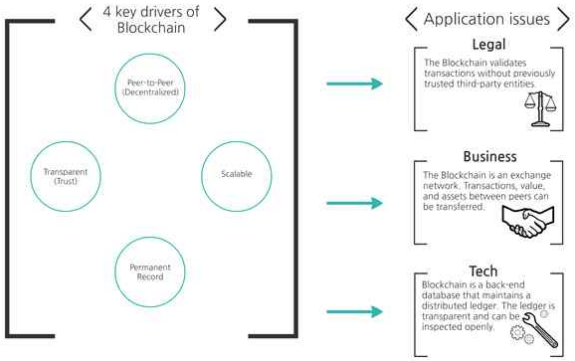
The Smart Home utilizes technologies and tools that can be integrated and applied to the home to increase user convenience and effiency. The technologies and services are applied through home networking, which aims to better the quality of living for individuals and families. The benefits of Smart Home services are that the services can be controlled by individuals, but are not strictly limited to such controls. However, like all technologies and services that allow maximum user convenience and device integration, the question of security is still on hold, as is the technology of device integration itself. The blockchain, which offers decentralized and trust-based authentication, has the potential to be applied to the Smart Home environment to enhance smart services. The blockchain is defined as the “first native digital medium for peer-to-peer value exchange. Its protocol establishes the rules – in the form of globally distributed computations and heavy duty encryption – that ensure the integrity of the data traded among billions of devices without going through a trusted third party. Trust is hard-coded into the platform. … [Blockchain] acts as a ledger of accounts, a databased, a notary, a sentry and clearing house, all by consensus.”

Bitcoin was intended to overcome the barriers of modern day currency. As Bitcoin is not controlled by any single institution, it is a decentralized distributed currency. The Bitcoin is transferable electronically, which makes the transfer quicker than most transfers made by ledger institutions such as banks and current online payment services. Since currency is not controlled by a central institution, Bitcoin transfers usually require very low transaction fees or service fees. The independency, economic feasibility, and speediness of the Bitcoin was unprecedented and rather groundbreaking. Soon, the technology of Bitcoin innovated what is known to be the blockchain

II. Business Models of 3 Blockchain Companies

2.1 The Characteristics and Status of the Blockchain

The blockchain creates a trust-based cross-certification system. Instead of having one verifier that determines whether a source can be trusted or not, the blockchain positions several verifiers that communicate with one another to determine whether the new device/information/person can be trusted and added to the chain. Because of the innovative format it offers, the blockchain can be adopted to various fields. The blockchain can mean different things in different fields. From a technical point of view, the blockchain is a back-end database that maintains a distributed ledger that can be inspected openly. In business, the blockchain is an exchange network for moving transactions, value, assets between peers, without the assistance of intermediaries. In the legal scheme, the blockchain validates transactions, replacing previously trusted



[Figure 1] 4 Key drivers of Blockchain and applied fields

entities.

Though still undergoing development, the blockchain is inarguably an innovative technology. The technology breaks down the current, more traditional system that allows one ledger to take control of an individual's information and valuables. The traditional structure disallows for individuals to have true power and ownership over their information and assets, as it allows ledgers to have power over assets and structures. The blockchain is everything. It is a platform, a decentralized database and ledger. It creates a platform for where information, data, and finances can be stored. In this platform, anyone can see that one has placed information in the blockchain, indicating that the blockchain is then wholly transparent. Though the blockchain is transparent and "open", it cannot be manipulated. Only the individual who has placed the information in place can unlock the container (public and private keys). What's more, is that all altered information within the blockchain is tracked and permanently recorded and stored in the blockchain. However, the alterations are not easily made within the blockchain. Because the blockchain is decentralized ledger, one device cannot grant access of the information in the container. Instead, various verifiers communicate and verify the validity of each transaction and new device, which creates the blockchain entity. Because of these characteristics, one key feature that brings the blockchain to the spotlight is enhanced security.

2.2 Business Models of 3 Blockchain Companies

This part of the paper will aid the design stage of creating a blockchain service model. Deloitte suggests that there are 4 stages for a company to utilize a blockchain service model. The 4 stages listed are "initiate", "design", "strengthen", and "implement." Initiating the thought for the need for a blockchain service is relatively simple. However, the stage of designing the adequate model is difficult because the company must recognize whether the service it plans to provide is in absolute need of the blockchain. In this paper, we believe that the four key drivers of the blockchain are peer-to-peer structure, transparent information, scalable database, and permanently recorded information. If a company is seeking to supplement one of the four key drivers, a blockchain service model may enhance service efficiency of the company. In order to recognize the efficiency of the Blockchain structure for companies that are in need of one of the four lockchain key drivers, below we look at some companies.

Wave is an Israel based company that has "created peer-to-peer and completely decentralized network that connects all carriers, banks, forwarders, traders and other parties of the international trading supply chain." The current trading system is rather complex and inefficient. The trading concept itself is simple. The seller suggests a deal to the buyer, and the buyer accepts, which leads to the trade of products and money. However, the process becomes complex in real life. In order for legal trade to be obtained, there needs to be an involvement of legal experts and banks to verify that the trade is being done over legal and consensual processes, wherein both parties agreed to legal conditions, payments were made concisely, and products were sent without fraud. The involvement of legal institutions is costly, and not necessarily timely or error-free. Legal organizations can take longer to write out contracts on behalf of trading companies, and may overlook errors or loopholes that can create trouble for one of the trading companies. Payment and

shipment verification is also inefficient and error-prone.

Modum is a Switzerland based company that creates "sensor devices leveraging blockchain technology to assert data immutability and public accessibility while saving costs in the pharma supply chain." Modum products monitor the temperature of individual parcels during the shipment process in order to ensure GDP regulations. Once the parcel is accepted by the consumer, data collected by the "Mod" is then transferred to the blockchain, and the product attributes are immediately and automatically assessed based on a smart contract. The data is then becomes publically accessible and unchangeable, allowing all consumers to view the data. Modum's customer value proposition is to provide accurate, real time data that can potentially influence product quality of medical supplies to customers, including doctors and distributors. Because the process is made simple by blockchain and monitoring services, the data is considerably rawer and transparent.

Healthbank, a Switzerland based blockchain company, helps to give individuals full ownership of their medical information. The idea is simple; Healthbank stores personal medical and healthcare information in the blockchain. The information is not limited to information provided by hospitals and doctors, but instead is additionally collected by IoT healthcare devices such as glucose meters, smart inhalers, fitbits, heart rate monitors, etc. Through the Healthbank, individuals are given full ownership of their medical information without turning over their rights to their information to a healthcare provider such as a hospital or physician. The individual can then choose to share their medical information, not exclusively to healthcare providers, but non-healthcare providing individuals as well. Individuals have the right to share the information with health opinion providers, and also family or friends who they wish to obtain help in managing their health. The blockchain system allows for instantaneous storage of health information collected by various organizations and devices, then storing the information so that the information is permanent, and publically accessible yet unchangeable. The huge advantage of Healthbank is that information then becomes scalable, because not just limited sources can validly store healthcare data. Whereas the current healthcare system only validates information provided by authorized physicians and hospitals, Healthbank collects data from numerous sources, exercising the scalability of blockchain technology. The information is then open to whoever the individual chooses to share with, but is permanent and cannot be manipulated by third parties that are unauthorized to change the information stored

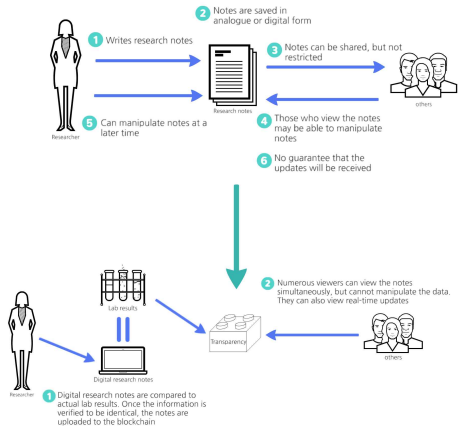
III. Reliable Application Design

In this paper, we introduced fundamental characteristics of the blockchain. Although the blockchain is a rising technology, quickly gaining hype and interest from various companies, thus far the blockchain is most widely being experimented and used by mostly financial institutions. It is important to note that the blockchain has the danger to be over-implemented even in industries and companies that do not necessarily need a blockchain platform to be efficiently functional. However, it is also untrue that the blockchain is only limitedly effective for specific industries. Thus, in order to provide insight on what companies may absolutely need a blockchain system, we categorized some key drivers that best summarize the potential of the blockchain. The four key drivers of the blockchain noted are peer-to-peer structure, transparency of information, scalable database, and permanently kept records. In other words, if a company is in need of one of the key drivers, we believe that designing and implementing a blockchain system will be most effective. Above three companies were analyzed and each of its business models were drawn out. The companies listed were in the fields of the trading industry, loan industry, pharmaceutical transport industry, agriculture manufacturing industry, and healthcare industry. Each company effectively utilized at least one of the key drivers that the Blockchain provides, thus making the Blockchain worthy.

The three companies above are meant to provide services that enhance trust and convenience for users and consumers. We believe that there is further room for improvement by collaborating smart services and the blockchain. Below, we will introduce the daily life of an individual and how a blockchain-based smart services can be utilized to maximize user experience.

At the home, smart services such as automatic pet-food dispensers, convenient book selling online, and smart rice dispensers can be developed and enhanced through the Blockchain. At work, research note service providers, and technology escrow services can be utilized based on a Blockchain system. Research notes may also benefit from the utilization of the Blockchain, since the technology offers a transparent database. Research notes, in essence, require transparency

because the information must be completely true. Also, research notes must be legible because other researchers must be able to read and comprehend the notes in order to refer to and potentially replicate the research. Currently, many research notes are written by hand, decreasing legibility and transparency. If research notes are transferred to the Blockchain instantaneously once being verified that the noted results are similar to the actual results in real-time, the research notes will become more accurate and trustworthy.



[Figure 2] Research notes to be with the Blockchain

Additionally, the Blockchain can allow specific people to view the results, with no restriction regarding how many people can access the notes simultaneously, and no concern regarding the information being manipulated. The Blockchain also has potential to be utilized as a technology escrow. The technology escrow is usually held by a third party company, and allows for midsized and smaller companies to contribute their technology to conglomerates. If a technology escrow is created, it can verify that the midsized company sold their technology to the conglomerate, and in turn, verifies that the technology does not belong to the conglomerate. However, this requires that companies join an escrow, which requires either a membership fee, annual fee, or service fee. The reason that the Blockchain can act as a technology escrow is simple; it offers permanent recording of information. With the Blockchain as a technology escrow, there is no need to actually apply for a technology escrow via a third-party organization, there is no need for a fee, and most of all, there is no concern that the information held by the third-party is leaked, destroyed, or manipulated. The Blockchain becomes the escrow itself, and records are transactions made between companies and individuals. The recorded information is permanent and unchangeable, then become proof of the transaction.

[Acknowledgement]

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2017-2014-0-00636) supervised by the IITP(Institute for Information & communications Technology Promotion)

[Reference]

[1] Don T.; Alex T., Blockchain Revolution; Brilliance Audio, 2016.

[2] What is Bitcoin?, CoinDesk, 2015.

[3] Blockchain Investments Top $1Billion Annually,Wall Street, Greenwich Associates, 2016.

[4] Jacob, B.; Jeroen, B.; Blockchain technology: 9 benefits & 7 challenges; Disrupting multiple industries. Deloitte, 2016.

[5] RJ K.; Dan H.; Mark W.; Mariya F.; Florian Q.; Dan B.; Allen N.; Kate F.; Jason K.; Adam I.; Lindsay T.; Blockchain: Opportunities for Health Care,Deloitte, 2016.

[6] Dae Il N.; Joo Hee K.; Hyun Joo A.; Ji Hye J.; Gae Won L.. Business Model Story 101. Hans Media, 2015.

[7] Tiiu K.; Kaisa V.; Evolution towards smart home environments: empirical evaluation of three user interfaces, Springer-Verlag London Limited, 2004.

[8] Rosslin J.; Tai-hoon K.; Applications Systems and Methods in Smart Home Technology: A Revie, International Journal of Advanced Science and Technology, 2010.

2017 한국정보보호학회 동계학술대회

B1 시스템보안(II)

김 윤 정 (서울여대)

- **정적 특성 기반의 군집화를 이용한 변종 악성코드 식별 방안**
  김광훈, 이동근, 사경진, 조형택, 염흥열 (순천향대학교)
- **이중 지불 공격 방지를 위한 트랜잭션 풀 우선순위 선정 방안 연구**
  박지선, 김동이, 박시헌, 신상욱. 이경현 (부경대학교)
- **소프트웨어 잡음원의 취약 패턴 검출 연구와 그 활용방안**
  박호중, 김원태, 염용진, 강주성 (국민대학교)
- **동적 정보 기반 악성코드 그룹 분류 및 특징 연구**
  박래현, 유정빈 (연세대학교), 박준형 (한국인터넷진흥원), 권태경 (연세대학교)

정적 특성 기반의 군집화를 이용한 변종 악성코드 식별 방안*

김광훈*, 이동근*, 사경진*, 조형택*, 염흥열**

*순천향대학교 융합서비스보안학과,**순천향대학교 정보보호학과

Identification of Variant malware Using Clustering based on static characteristics

Kwang-Hun Kim*, Dong-Geun Lee*, Kyeong-Jin Sa*, Hyoung-Tack Cho*, Heung-Youl Youm**

*Department of Convergence Service Security, SoonChunHyang University.

**Department of Information Security Engineering, SoonChunHyang University.

요 약

오늘날 악성코드는 날이 지날수록 기하급수적으로 증가하여 현재 모든 악성코드를 분석하는데 한계가 있다. 또 악성코드의 지능적인 기술 발전으로 인해 기존의 악성코드에 변형된 변종 악성코드 또한 증가하고 있어, 기존의 안티 바이러스 솔루션의 분석, 탐지 메커니즘으로는 이에 대응하기 어려운 상황이다. 따라서 본 논문에서는 머신 러닝을 활용하여 악성코드를 유형별로 군집화 시킴으로써, 기존과 유사한 변종 악성코드에 대응할 수 있는 방안을 마련하여 악성코드 분석가가 효율적으로 악성코드의 심층 분석을 수행할 수 있도록 지원하고자 한다.

I. 서론

오늘날 악성코드는 날이 지날수록 기하급수적으로 증가하고 있어 현재 모든 악성코드를 분석하기에는 한계가 있다. 또 악성코드의 지능적인 기술 발전으로 인해 기존의 악성코드에 변형된 변종 악성코드 또한 증가하고 있어, 기존의 안티 바이러스 솔루션의 분석, 탐지 메커니즘으로는 이에 대응하기 어려운 상황이다. 변종 악성코드에 대응하기 위한 방법 중 하나는 머신러닝 기반의 군집화를 통해 탐지하는 것이다. 악성코드에 대한 군집화가 이루어지면 기존과 유사한 악성코드를 식별 할 수 있다. 즉 기존의 악성코드에서 변형된 악성코드에 대한 탐지가 가능해진다. 이를 통해 악성코드 분석가들은 수집된 여러 샘플들을 실제 분석해보지 않고도 빠르게 해당 악성코드의 유형 파악이 가능해짐으로써 시간적, 금전적인 비용을 줄일 수 있는 효과를 얻을 수 있다. 따라서 본 논문에서는 머신러닝을 활용하여 악성코드를 유형별로 군집화 시킴으로써, 기존과 유사한 변종 악성코드에 대응할 수 있는 방안을 마련하여 악성코드 분석가가 군집화 된 결과 데이터를 기반으로 악성코드에 대해 좀 더 효율적인 심층 분석을 수행할 수 있도록 지원하고자 한다.

* 본 연구는 미래창조과학부 및 한국인터넷진흥원의 “고용계약형 정보보호 석사과정 지원사업”의 연구결과로 수행되었음

II. 관련 연구

2.1 기존 악상코드의 분석 기법

악성코드를 분석하기 위한 기법으로는 크게 악성코드의 행위에 초점을 맞춘 동적 분석과 바이너리 코드의 포맷을 기반으로 의미 있는 플래그를 분석하거나, 코드를 직접 역 분석하는 정적 분석 기법이 있다. 동적 분석은 시스템 내에서 악성코드를 실행시켜 실시간으로 악성코드에 대한 행위를 추

적, 모니터링 하는 분석 방법이다. 이와 다르게 정적 분석은 악성코드를 실행하지 않고 바이너리 코드 포맷을 기반으로 헤더의 의미 있는 플래그(예를 들어 PE 파일의 Section) 또는 특정 문자열 및 명령어 간의 호출, 연관성을 분석하여 해당 코드에 내재된 통계적 특성을 유추하는 분석 방법이다. 동적 분석은 해당 파일이 패킹, 난독화되어 있어도 분석이 가능하나 행위 파악을 위한 시간적 비용 소모가 크다는 단점이 있다. 반면 정적 분석은 빠르게 의미 분석이 가능한 반면 패킹, 난독화에 취약하다는 단점이 있다. [1]

2.2 악성코드의 특성을 이용한 분류 기법

보통 악성코드의 특성을 추출하는 방법으로는 동적 분석을 기반으로 악성코드에서 사용되는 API와 같은 행위 및 순차적인 패턴을 이용하거나 정적 분석을 기반으로 명령어의 횟수 또는 특정 문자열 및 바이트 시퀀스 패턴을 이용하는 방법이 있다. 기존 안티 바이러스 솔루션들은 이러한 악성코드의 특성을 기반으로 의미 있는 데이터 셋을 추출하여 진단명을 정의하고 통계적 방법에 따라 비교 대상과의 유사도 측정을 수행하여 해당 악성코드를 탐지한 후 유형 별로 분류하는 기법을 주로 사용한다. 분류되는 유형은 크게 바이러스, 웜, 트로이목마로 나뉘며 그 중에서도 세부 특성에 해당할 경우 다시 백도어, 스파이웨어, 랜섬웨어 등과 같은 여러 개의 소분류로 나눌 수 있다. 이러한 분류 방식은 주관적 성향이 강하기 때문에 분류 기준이 서로 중첩되거나 모호한 경우 명확하게 분류 할 수 없다는 단점을 가진다. 즉 기준에 부합되지 않는 새로운 유형의 악성코드들이 무수히 많아질 경우 수동으로 다시 새로운 기준을 재 정의해야하는 번거로움이 따른다. 이는 악성코드가 트로이목마이면서, 바이러스, 웜 모두에 해당하는 경우 이에 대한 진단명을 정의하는데 혼란을 줄 수 있다. 현재 이와 같은 혼란을 최소화 하기 위해 CME(Common Malware Enumeration)[2]라는 공동 악성코드 분류 기준을 내세우고 있으나, 이 또한 주요 악성코드에 한하여 제한적으로 수행하고 있어 악성코드의 분류 표준화에 큰 영향을 주지 못하고 있다. [1] 따라서 이를 보완하기 위해서는 잘 정의된 데이터 셋을 통해 악성코드에 대한 군집화를 수행하여 유사한 악성코드를 유형 별로 묶어 관계를 통해 식별하는 메커니즘이 추가로 필요하다.

2.3 유사 악성코드 식별을 위한 군집화 방법

군집화는 비지도학습 기반으로 지도학습인 분류와의 가장 큰 차이점은 기준 레이블이 주어지지 않는다는 것이다. 따라서 주어진 입력 데이터 셋만을 통하여 군집의 중심인 초기 클러스터를 기준으로 유사한 성질을 가지는 몇 개의 그룹으로 나누는 역할을 수행한다. 유사성은 보통 데이터 셋과 클러스터의 거리를 기준으로 표현된다. [1] 이러한 군집화 특징을 악성코드에 적용한다면 기존 악성코드에서부터 새로운 유형의 악성코드를 식별할 수 있다. 이는 군집화 된 악성코드로부터 의미 있는 특징 및 연관성을 도출하고 기존의 분류 기법에서 더 나아가 새로운 진단 기준을 군집 결과를 바탕으로 상황에 맞게 부여하여 차후 악성코드의 유형을 정확하게 판별하는 것을 가능하게 한다.

본 논문에서는 군집화를 통해 판단을 원하는 대상 파일이 대표적인 악성코드의 행위 유형에서 얼마나 관련이 있는지 식별하는 방안을 제안하고자 한다.

III. 변종 악성코드 식별 방안

3.1 개요

본 논문에서 제안하는 변종 악성코드를 식별하기 위한 군집화 과정은 다음과 같다.

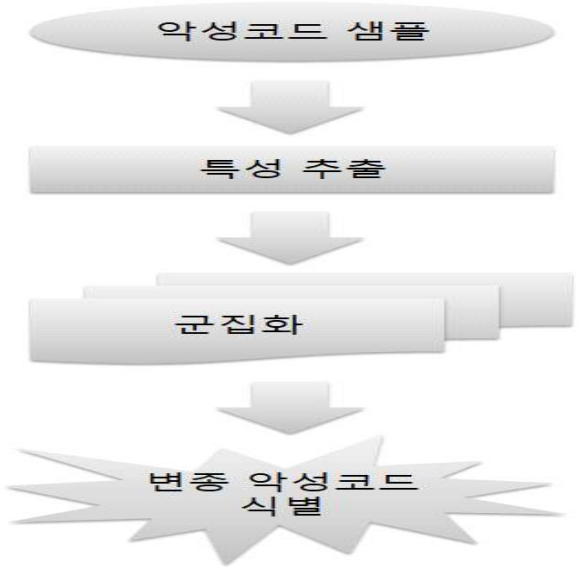


그림 1 악성코드를 식별하기 위한 군집화 과정

먼저 군집화에 사용된 악성코드 샘플은 총 24개로 Windows 실행 파일(PE) 기반의 다양한 변종 랜섬웨어(3유형 – Petaya, Cryptowall, Radement)를 대상으로 하였다.

3.2 악성코드의 특성 추출

군집화에 앞서 악성코드 특성 추출은 가장 중요한 단계다. 특성을 어떻게 추출 하냐에 따라 군집 형성에 가장 최적화된 결과를 보여 줄 수 있다. 추출한 특성은 수치화하여 군집화의 데이터 셋으로 사용된다.

본 논문에서는 정석 분석을 기반으로 하여 악성코드에 대한 특성을 추출하였다. 정적 분석을 이용하면 빠르게 해당 악성 샘플들을 파싱하여 의미 있는 데이터 셋을 추출 할 수 있다. 여기서는 대부분의 변종 악성코드가 기존의 IDT(Import Directory Table)영역을 재사용하거나 크게 변경하지 않는 점에 착안하여 IDT 영역을 기반으로 데이터 셋을 선정하였다. 따라서 파싱 영역은 IDT이며 포함된 DLL 및 API 정보를 기반으로 데이터 셋을 추출한다. 이는 통계가 가능한 데이터 셋으로 다음과 같은 구조를 가진다.

[표 1] 샘플로부터 추출할 데이터 셋 구조

데이터 셋_1 (정수 X . 소수 Y)	데이터 셋_2 (정수 X . 소수 Y)
정수 X : DLL 개수	**정수 X** : API 개수
소수 Y : DLL 개수 / 각 DLL 문자열 길이 합	**소수 Y** : API 개수 / 각 API 문자열 길이 합
D_S1 : 정수 X + 소수 Y	**D_S2** : 정수 X + 소수 Y

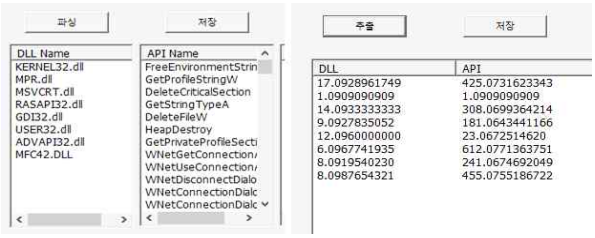


그림 2. IDT 영역 파싱 후 데이터 셋 추출

추출한 데이터 셋은 2개로 각각 정수와 소수 형태로 표현된다. 정수부 X에 해당하는 DLL 개수와 API 개수는 악성코드의 행위를 간접적으로 식별 할 수 있는 데이터이다. 소수부 Y에 해당하는 데이터는 DLL과 API 중에서도 어떤 함수를 사용하는지 문자열 길이를 통해 어느 정도 식별이 가능하도록 차별성을 부각시켰다. 이 후 추출한 데이터 셋은 군집화 모듈에서 사용하기 쉽게 DataSet.csv 파일 형식으로 저장한다.

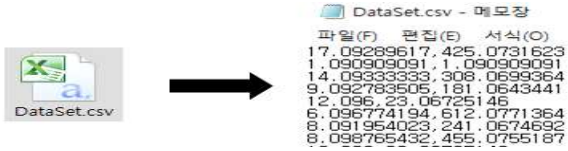


그림 3. 데이터 셋 추출 후 DataSet.csv 파일로 저장

※ 추출된 악성코드 샘플의 데이터 셋이 중복된 경우에는 같은 유형의 악성코드로 간주하고 군집화를 진행하였다.

3.3 악성코드의 군집화

3.3.1 군집화에 사용된 알고리즘

본 논문에서 사용한 군집 알고리즘은 K-means[3]로서 이는 K 개의 중심점을 기준으로 유클리드 거리 계산 방식을 이용하여 데이터 포인트 간의 거리를 구하고 가장 가까운 중심점으로 군집하는 방법이다. 해당 알고리즘의 동작 원리는 다음과 같다.

- **1. 초기 단계 :** 클러스터의 개수 K 값을 선택한다. 즉 데이터가 분포된 공간상에 클러스터 중심으로 가정할 임의의 지점을 K개 선택한다.
- **2. 할당 단계 :** 임의로 선택한 K개의 클러스터의 중심과 개별 데이터 사이의 거리를 유클리드 거리 제곱 방식을 사용하여 계산한다.

$$d(x,y)^2 = \sum_{j=1}^{m}(x_j - y_j)^2$$

그림 4. 유클리드 거리 제곱

- **3. 업데이트 단계 :** 클러스터에 속하게 된 데이터들의 평균값을 새로운 클러스터의 중심으로 둔다.
- **4 루프 단계 :** 2 ~ 3단계를 알고리즘이 수렴할 때까지 반복한다. 여기서 수렴은 클러스터의 중심이 더 이상 변화가 없게 된다는 의미이다.

3.3.2 최적의 클러스터 개수 선정

본 논문에서는 먼저 적절한 클러스터 개수의 범위를 선정하기 위해 해당 데이터 셋에 알맞은 알고리즘을 직접 만들어 사용하였다. 다음으로 클러스터링의 품질을 정량적으로 계산해주는 실루엣 기법[3][5]을 사용하여 유효성을 검증하고 최종적으로 군집에 가장 알맞은 클러스터 개수를 선정한다.

[표 2] 적절한 클러스터 범위 선정을 위한 의사코드

```
M = SUM(D_S1) / S

IF(M > S)  C = 1
ELSE C_MAX = S / M

IF(M < D_S1) UP = COUNT(D_S1)
IF(M > D_S1) DOWN = COUNT(D_S1)

IF(UP > DOWN) MIN = DOWN
ELSE MIN = UP

IF(MIN > M) C_MIN = MIN / M
ELSE C_MIN = MIN
```

S = 전체 샘플
D_S1 = 전체 샘플에 대한 데이터 셋1(DLL 개수)
M = D_S1의 총합에서 S를 나눈 평균 값
C = 클러스터 개수
C_MAX = 최대 클러스터 개수
C_MIN = 최소 클러스터 개수
UP = C_MAX 보다 큰 D_S1 개수
DOWN = C_MAX 보다 작은 D_S1 개수
MIN = UP, DOWN 중에서 작은 값

추출된 DataSet 파일로부터 위 의사코드를 적용할 경우 클러스터 범위는 1~3이다. 해당 범위로부터 최적의 클러스터 개수를 선정하기 위해 실루엣 기법을 사용하여 군집의 유효성을 검증한다.

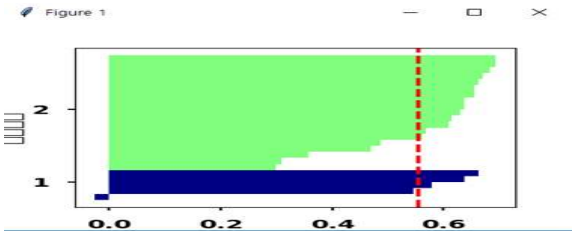


그림 5. 클러스터 개수가 2일 때 실루엣 결과

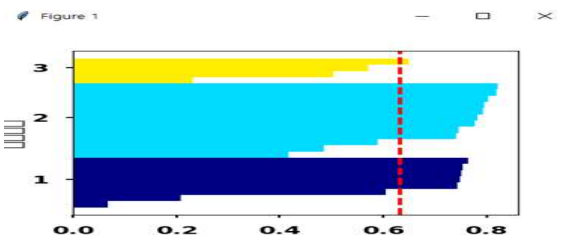


그림 6. 클러스터 개수가 3일 때 실루엣 결과

실루엣을 적용한 결과 클러스터 개수가 3일 때 빨간 점선의 평균점이 0.6 이상임을 보이는데 이는 1에 가까울수록 가장 최적화된 군집을 의미한다. 따라서 초기 클러스터의 수를 3으로 선정하여 군집화를 진행한다.

3.3.3 최적의 클러스터 변수 선정

본 논문에서는 초기 클러스터 3개에 대한 최적해를 구하기 위해 K-means++ 알고리즘[3][5]을 사용하였다.

3.4 변종 악성코드의 식별

본 논문에서는 오픈소스로 제공되는 머신러닝 기반의 파이썬 모듈 scikit_learn[4]을 사용하여 군집화를 수행하였다.
3종(Petaya, Cryptowall, Radement)의 변종 랜섬웨어 샘플 24개에 대한 군집화 수행결과는 다음과 같다.

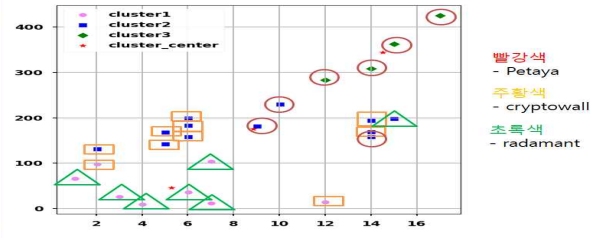


그림 7. 악성코드 샘플에 대한 군집을 시각화한 결과

시각화를 통해 변종 랜섬웨어 샘플간의 유사성을 확인 할 수 있으며, 그 결과 변종 랜섬웨어 탐지율이 약 75%(24개 중 18개 일치)정도 되는 성능을 보임으로써 변종 랜섬웨어를 식별하기에 충분한 군집화가 이루어졌음을 알 수 있다.

IV. 결론

본 논문에서는 머신 러닝 기반의 군집화를 통해 변종 악성코드를 식별하는 방안을 제안하였다. 이로서 악성코드 분석가가 더욱 효율적으로 악성코드에 대한 심층 분석을 수행할 수 있을 것으로 기대하며, 또 앞으로 계속 증가하는 신종, 변종 악성코드 수와 식별 가능한 악성코드 수의 격차를 줄여 신속한 침해 대응이 가능할 것으로 예상된다.

[참고문헌]

[1] 권오철, “퍼지 군집화 기법을 이용한 악성코드의 정량적 분류”, 고려대학교, 2008년 11월
[2] CME(CommonMalwareEnumeration), http://cme.mitre.org/
[3] "K 평균 알고리즘“, 위키백과, https://ko.wikipedia.org/wiki/K-%ED%8F%89%EA%B7%A0_%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98
[4]"Scikit-learn: Machine Learning in Python", Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.
[5] 옥수별, 2017.5.30., "[30편-31편] K-means 클러스터링" http://blog.naver.com/prologue/PrologueList.nhn?blogId=samsjang, 2017년 5월

이중 지불 공격 방지를 위한 트랜잭션 풀 우선순위 선정 방안 연구

박지선*, 김동이*, 박시현*, 신상욱**, 이경현**

*부경대학교 정보보호협동과정

**부경대학교 IT융합응용공학과

Study of Prioritization Strategy in Transaction Pool to Prevent Double-spending Attack

Ji-Sun Park*, Dong-Ee Kim*, Si-Hyeon Park*, Sang-Uk Shin**, Kyung-Hyune Rhee**

*Interdisciplinary Program of Information Security, Graduate School, Pukyong National University

**Dept. of IT Convergence and Application Eng., Pukyong National University

요 약

최근까지 여전히 비트코인(Bitcoin)은 엄청난 상승세를 유지하고 있다. 하지만 그와 동시에 비트코인의 안전성에 대한 논란도 끊이지 않고 있다. 비트코인의 대표적인 취약점 중 하나는 이중 지불 공격이며 이 공격에 대한 여러 대응책이 제안되었다. 하지만 이 대응책들도 완벽하다고 볼 수는 없다.

따라서 본 논문에서는 비트코인에서 채굴시 채굴자에게 트랜잭션 선택권을 주지 않고 트랜잭션 풀의 우선순위 선정 방식의 변경을 통해 이중 지불 공격의 새로운 대응 기법을 제안하고자 한다.

I. 서론

블록체인(Blockchain)에 대한 관심이 나날이 증가하고 있다. 블록체인은 한마디로 분산기록 데이터베이스이고 블록(block)이라고 불리는 여러 개의 레코드를 체인(chain)으로 엮어서 저장하는 형태이다. 블록체인은 모든 노드에게 그 체인이 공개되어 있기 때문에 공개장부(public ledger)라고도 불린다. 현재 글로벌 사회는 블록체인이 갖고 있는 불변성(immutability), 투명성, p2p (peer-to-peer) 네트워크, 분산 시스템 등의 이점[1]을 이용해서 여러 방면에서 블록체인을 응용하고 있다.

여러 블록체인 응용 시스템 중 하나인 비트코인은 가장 유명한 암호 화폐라고 말해도 과언이 아니다. 1 BTC(비트코인의 화폐단위) 가격은 꾸준히 증가하고 있는 추세이며 여전히 많은 거래가 이루어지고 있다. 하지만 비트코인의 유명세에 많은 우려를 표하는 이들도 많다. 그 이유는 비트코인에는 몇몇 보안 취약점과 그에 따른 위협이 존재하고 있기 때문이다. 그 중 대표적인 공격으로 이중 지불 공격(double-spending attack)을 들 수 있다. 이중 지불 공격은 공격자가 이미 사용된 화폐를 회수하거나 재지불하여 거래를 완료하는 것으로 최종적으로는 지불받지 못한 자의 손실이 발생하게 된다[2].

본 논문에서는 비트코인의 이중 지불 공격과 그에 따른 변형 공격들, 제안된 대응책에 대하여 살펴보고 트랜잭션 풀(transaction pool)에서의 우선순위를 지정하는 방법으로 이중 지불 공격에 대한 새로운 대응책을 제안한다.

II.관련 연구

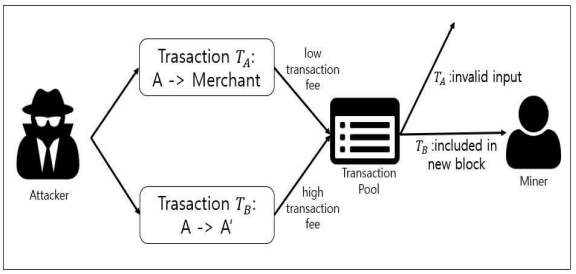
2.1 비트코인(Bitcoin)

비트코인은 2008년 Nakamoto Satoshi에 의해 제안된 첫 번째 분산 디지털 암호 화폐이다[3]. p2p 환경에서 동작하며 PoW(Proof of Work) 합의 알고리즘을 채택하고 있다. 네트워크의 모든 유저는 공개키 암호방식으로 송금 트랜잭션(transaction)을 생성한다. 채굴자(miner) 유저들은 PoW 작업을 통해 여러 개의 유효한 트랜잭션들로 블록을 생성하여 이득을 취한다(채굴). 생성된 블록들은 전체 네트워크에 broadcast되고 체인에 등록된다.

2.2 이중지불 공격(double-spending attack)

이중 지불 공격은 Race Attack이라고도 불린다[4]. 이 공격은 단일 화폐가 동시에 한번 이상 사용되는 것을 말한다. 비트코인은 가상 화폐이며 통제 및 관리 개체가 없는 분산화 된 시스템이기 때문에 이중 지불 가능성이 존재한다. 같은 입력 값에 대해 다른 두 개 이상의 출력 값이 나오게 되므로 비트코인에서는 그 중 하나의 출력 값만을 유효하다고 처리한다.

예를 들어, [그림 1]과 같이 공격자는 상품 판매자에게 일정 금액을 송금한다는 트랜잭션 T_A을 생성하여 보여주고 서비스를 제공받는다[4]. 하지만 동시에 같은 코인을 자신의 다른 계좌로 보내는 트랜잭션 T_B도 발행한다. 이 때 공격자는 T_A보다 큰 금액의 트랜잭션 수수료(transaction fee)를 T_B에 설정하여 발행하게 된다. 동시에 두 트랜잭션 T_A, T_B가 다른 채굴자 또는 채굴 풀(mining pool)의 트랜잭션 풀에 들어가게 되면 더 많은 수수료를 가진 T_B가 우선적으로 선정되어 T_A보다 먼저 블록에 채굴(mining) 된다. 그로인해 더 낮은 우선순위를 가지고 늦게 채굴되는 T_A는 입력 값이 이미 T_B에서 소비된 코인으로 검증되기 때문에 블록에 포함되지 못한다. 결국 공격자는 상품 판매자에게 서비스를 제공받지만 돈은 제공하지 않은 셈이 된다.



[그림 1] 이중 지불 공격 예

2.3 응용 공격 및 대응책

본 장에서는 이중 지불 공격에서 응용된 공격들 중 Finney Attack, Eclipse Attack에 대해 간략히 소개하고 제안된 이중지불 공격의 대응책에 대해 알아본다.

(1) Finney Attack

공격자는 직접 채굴자가 되어 공격을 시도한다. 상인을 속이기 위해 미리 같은 코인을 자신의 다른 계좌로 송금하는 트랜잭션을 생성하고 블록에 포함시킨다[2][5]. 하지만 이 블록을 broadcast하지는 않은 채 보유한다. 나중에 상인에게 정상 지불하는 같은 코인을 사용하는 트랜잭션을 만들어 채굴할 때, 미리 생성한 블록들을 빠르게 broadcast하면 비트코인 규칙에 따라 길이가 긴 분기가 선택되므로 이 트랜잭션은 무효화가 된다. 이 공격은 이중 지불 공격과 이기적인 채굴 공격(selfish-mining attack)이 결합된 형태의 공격이다.

(2) Eclipse Attack

공격자는 미리 채굴자들 중 타겟(target)을 선택하여 타겟 노드와 다른 노드와의 연결을 차단해 네트워크 연결을 독점한다[5]. 그렇게 독점한 타겟 채굴자들에게 정상 지불 트랜잭션을 전송하여 채굴하게 하지만 이들의 네트워크 연결은 공격자에 의해 조종되기 때문에 broadcast 될 수 없다. 그 때 공격자는 이중 지불된 트랜잭션을 생성하고 다른 채굴자들에게 전송한다. 결국 이 트랜잭션만 유효하게 된다.

(3) 이중 지불 공격 대응책

이중 지불 공격의 대응책으로 제안된 방법에는 다음과 같은 것들이 있다.

• Listening period

상인이 서비스를 제공하기 전에 수신하는 모든 트랜잭션에 대하여 특정 시간만큼 모니터링을 실시한다[5]. 이중 지불 시도가 없는지 확인하기 위함이다.

• Inserting observers

네트워크 전체를 모니터링 하는 observer node를 두어서 공격을 사전에 방지한다[2]. 추가 노드의 운영비용이 든다는 점이 단점이다.

그 외에 Forwarding double-spending attempt[5], Increasing the number of neighbors[5] 등의 방안이 있다.

III.제안 기법

기존에 제안된 이중 지불 공격의 대응책 외에 트랜잭션 풀 우선순위를 변경한 기법을 제안한다.

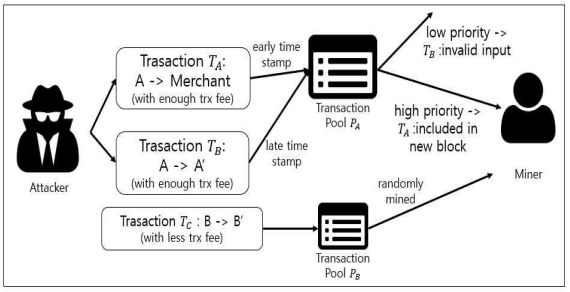
3.1 트랜잭션 풀 우선순위 선정 기법

2.2의 예시와 같은 상황에서 트랜잭션 풀에서 채굴자가 채굴할 트랜잭션을 선택할 수 있기 때문에 트랜잭션 수수료가 높은 트랜잭션을 우선으로 선택한다. 공격자는 이 점을 악용하여 공격을 시도하였다. 또한 기존의 기법은 낮은 수수료를 갖는 트랜잭션들의 무한대기를 발생시키기도 한다. 따라서 채굴자가 임의로 트랜잭션을 선택하지 못하도록 하고 새로운 우선순위 기준을 선정해야 한다.

우선순위를 결정하는 새로운 기준은 트랜잭션 생성 시간이다. 기존의 비트코인 시스템에서 트랜잭션은 tims stamp값을 가지지는 않지만 본 기법에서는 가진다고 가정한다. 먼저 발행된 트랜잭션이 높은 우선순위를 가진다. 다만 일정 금액 이상의 수수료를 갖는 트랜잭션이라는 조건이 필수적이다. 그렇지 않으면 어떤 악의적인 노드가 0 Satoshi의 트랜잭션 수수료를 갖는 가짜 트랜잭션(fake transaction) 또는 쓸모없는 트랜잭션을 여러 개 만들어 트랜잭션 풀에 DDoS (distributed denial-of-service) 공격을 시도할 수 있다. 하지만 수수료 하한 값을 설정하면 공격자는 반드시 하한 값 이상의 수수료를 지불해야하기 때문에 쉽게 공격을 시도할 수 없다. 따라서 풀 p_A에는 최소한의 수수료보다 큰 금액의 수수료를 가진 트랜잭션들만 입력이 된다.

그렇지만 반대로 정직한 노드가 0 Satoshi의 수수료를 갖는 트랜잭션을 발행했을 경우를 고려해 보았을 때, 이 트랜잭션은 풀 p_A에 포함될 수 없고 무한 대기 상태에 빠지게 된다. 이러한 점을 보완하기 위해 수수료 하한 값을 가지지 않는 또 다른 풀 p_B을 두고 적은 값의 수수료를 가진 트랜잭션들을 수용한다. 따라서 본 기법에서는 두 개의 트랜잭션 풀 p_A, p_B이 필요하다. 그리고 풀 p_B의 우선순위도 time stamp 값에 의존한다. 하지만 풀 p_A의 트랜잭션들이 주로 채굴된 후 랜덤하게 풀 p_B의 트랜잭션이 채굴되므로 풀 p_B의 채굴 확률은 현저히 낮다.

기법의 적용 예시는 [그림 2]와 같다.



[그림 2] 제안 기법이 적용된 예

3.2 분석

제안 기법에서는 채굴자의 트랜잭션 선택 권한을 없애고 적정 금액 이상의 수수료를 갖는 트랜잭션들만 트랜잭션 풀에 수용하면서 그 트랜잭션들의 생성 time stamp값을 기반으로 하여 우선순위를 선정한다. 채굴자들은 반드시 이 우선순위에 따라서 채굴을 하게 되며 가끔씩 적은 수수료를 가진 트랜잭션 풀의 트랜잭션들도 포함하여 채굴을 한다.

이 기법에서 트랜잭션 풀의 트랜잭션 수용 기준이 되는 적정금액을 정할 필요가 있다. 기본적으로 트랜잭션 수수료는 트랜잭션의 크기에 따라 결정이 된다. [6]에 따르면 (10월 16일 오후 3시 현재) 전체 트랜잭션 수수료의 중간 값은 byte당 110 Satoshi이다. 트랜잭션 풀의 수용기준은 중간 값보다는 적은 값이 적절할 것이고 중간 값은 유동적으로 변하는 값이

기 때문에 트랜잭션 수용 기준 값 역시 변경될 수 있다.

이 기법의 장점은 기존의 다른 이중 지불 공격 대응책에 비해 트랜잭션 수신자 입장에서 추가적인 관찰자 노드가 필요하지 않으며 동시에 채굴자 입장에서는 일정 금액 이상의 트랜잭션 수수료를 보장하기 때문에 경제적인 효율성을 가진다고 할 수 있다. 또한 안전성 측면에서는 트랜잭션 수수료 하한 값을 설정함으로써 악의적인 노드의 DDoS (distributed denial-of-service) 공격 시도를 방어할 수 있다. 트랜잭션 수수료 하한 값이 없는 풀도 운영하고 있기 때문에 적은 수수료를 가진 트랜잭션들의 무한대기도 방지할 수 있으므로 모든 트랜잭션들에게 형평성(fairness)을 제공한다.

IV. 결론

본 논문에서 제안한 기법은 채굴자들이 이득을 위해 높은 트랜잭션 수수료를 기준으로 채굴 우선순위를 정한다는 점을 악용한 이중 지불 공격을 방어하는 기법이다. 이 기법을 적용하면 채굴자는 채굴할 트랜잭션을 직접 선택할 수는 없으나 우선순위 목록의 모든 트랜잭션이 어떤 기준 이상의 트랜잭션 수수료를 갖고 있기 때문에 채굴에 대한 보상은 충분히 받을 수 있다. 그리고 기존의 주변 네트워크를 모니터링 하는 등의 방식이 아닌 트랜잭션 풀 내부의 규칙을 변경하는 기법으로 유저 입장에서는 추가 비용을 지불할 필요가 없다는 것이 또 다른 장점이다.

Acknowledgement

이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임. (No.2017-0-00156, 블록체인의 정보보호 프레임워크 및 평가 방법 개발)

[참고문헌]

[1] Deloitte, “Blockchain technology: 9 benefits & 7 challenges”, Retrieved from https://www2.deloitte.com/nl/nl/pages/innovatie/artikelen/blockchain-technology-9-benefits-and-7-challenges.html

[2] 이혁준, 이수미, "비트코인의 신뢰구조와 이중지불의 위협", 정보보호학회지 제26권 제2호, 4월, 2016

[3] BitcoinWiki, Bitcoin, (2016) "https://en.bitcoin.it/wiki/Bitcoin"

[4] Mauro Conti, Sandeep Kumar E, Chhangan Lal, Sushmita Ruj, "A Survey on Security and Privacy Issues of Bitcoin", Cornell University, Ithaca, NewYork, 3, 2017

[5] Ghassan Karame, Elli Androulaki, “Bitcoin and Blockchain Security”, ARTECH HOUSE, p.59-83, 2016

[6] 21, "https://bitcoinfees.21.co/"

소프트웨어 잡음원의 취약 패턴 검출 연구와 그 활용방안

박호중, 김원태, 염용진, 강주성†

국민대학교 정보보안암호수학과 / 금융정보보안학과

A study on detecting the pattern of a software noise source and its application

Hojoong Park, Wontae Kim, Yongjin Yeom, Ju-Sung Kang†

Dept. of Information Security, Cryptology, and Mathematics / Financial information security, Kookmin University

요 약

현대암호의 안전성은 안전한 난수 생성에 의존한다. 난수의 안전성을 확보하기 위해서는 잡음원으로부터 충분한 엔트로피를 수집해야 한다. 잡음원의 대부분은 하드웨어 기반이지만, 하드웨어 잡음원을 수집하기 어려운 환경에서는 소프트웨어 잡음원으로부터 엔트로피를 수집해야 한다. 본 논문에서는 현재 활용되고 있는 운영체제에 기반한 소프트웨어 잡음원을 분석하여 취약 패턴을 정의하고, 이를 검출하는 알고리즘을 새롭게 제안한다. 또한, 소프트웨어 잡음원의 고유 특성을 확인하기 위한 바이트 별 패턴 분석을 진행한다. 이 분석 결과는 운영체제 기반 소프트웨어 잡음원이 이용되는 보안시스템의 개발과 안전성 평가에 있어서 실용적인 기준으로 활용 가능하다.

I. 서론

현대암호의 안전성은 안전한 난수 생성에 달려있다. 잡음원은 예측 불가능성의 근원으로, 안전한 난수를 생성하기 위해서는 잡음원으로부터 충분한 엔트로피를 수집해야 한다. 잡음원의 대부분은 열 잡음원, 방사성 붕괴, 양자 등과 같은 하드웨어 기반이다. 하지만, 하드웨어 잡음원을 통해 엔트로피 수집이 어려운 경량 환경에서는 소프트웨어 잡음원으로부터 엔트로피를 수집해야 한다. 또한, IoT와 같은 경량 환경에서의 엔트로피 수집에 관심이 높아지면서 [1], 경량 환경에서 활용 가능한 소프트웨어 잡음원을 통한 엔트로피 수집에 대한 연구가 활발히 진행되고 있다. 대표적인 연구로는 소프트웨어 잡음원의 활용에 대한 연구 [2]와 소프트웨어 잡음원 평가에 대한 연구[3, 4]가 있다.

본 논문에서는 취약 패턴 검출을 이용하여 현재 활용되고 있는 운영체제 기반 소프트웨어 잡음원을 집중적으로 분석한다. 먼저 소프트웨어 잡음원의 특성 분석을 통해 소프트웨어 잡음원의 취약 패턴을 정의하고, 이를 검출하는 알고리즘을 제안한다. 제안하는 알고리즘은 연속적인 취약 패턴 발생 확률에 기반을 두어 설계되었으며, 잡음원 출력 특성 분석에 활용된다. 또한, 소프트웨어 잡음원의 고유 특성을 확인하기 위한 바이트 별 패턴 분석을 진행한다. 바이트 별 패턴 분석 결과는 소프트웨어 잡음원 수집에 대한 최소 수집 기준을 제시하기 때문에, 운영체제 기반 소프트웨어 잡음원이 이용되는 보안시스템 개발과 안전성 평가에 있어서 실용적인 기준으로 활용될 것으로 기대한다.

† 교신저자

II. 소프트웨어 잡음원의 취약 패턴

2.1 취약 패턴의 정의

소프트웨어 잡음원의 취약 패턴을 정의하기 위해 현재 주로 사용하고 있는 Windows, JRE(Java runtime environment), Linux 운영체제의 소프트웨어 잡음원의 특징을 분석하였다. 대부분 소프트웨어 잡음원은 현재의 프로세스 ID, 스레드 ID와 같이 운영환경의 특성을 이용하거나, 시간 변화를 이용하는 특징을 보인다. 그렇기 때문에 소프트웨어 잡음원의 출력은 [그림 1]과 같이 변화가 거의 없거나 [그림 2]와 같이 출력 값이 연속적으로 증가하는 패턴을 보였다. 고정 값을 출력하는 잡음원은 다음 출력 값을 항상 예측할 수 있기 때문에 엔트로피는 0이다. 한편, 연속적으로 증가하는 값을 출력하는 잡음원은 이를 잡음원보다는 단순히 카운터로 보는 것이 타당하다. 그렇기 때문에 본 논문에서는 잡음원의 취약 패턴을 고정된 값을 출력하는 상수 패턴과 연속적으로 값이 증가하는 증가 패턴으로 정의하였다.



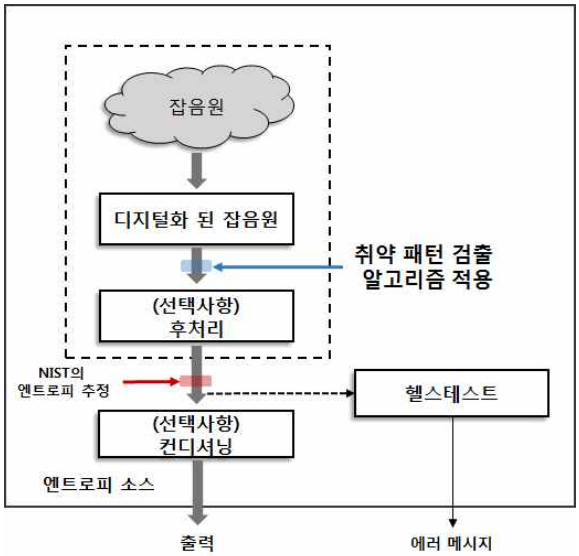
[그림 1] getProperty(JRE) - 상수 패턴의 예



[그림 2] GetTickCount(Win) - 증가 패턴의 예

2.2 취약 패턴 검출 알고리즘

기존의 잡음원 통계적 이상유·무를 확인하는 방법인 NIST SP 800-90B의 헬스테스트는 잡음원의 측정된 엔트로피에 기반을 두고 설계가 되었다[1]. 한편, 제안하는 알고리즘은 [그림 3]과 같이 잡음원의 취약 패턴을 엔트로피 측정 전에 검출하여, 사전에 취약한 잡음원의 수집을 방지하기 위한 목적이기 때문에 새로운 기준 설정이 필요하다. 이에 제안하는 알고리즘은 잡음원이 독립이고 동일한 균등분포로 부터 출력되었다는 가정 하에서 설계되었다.



[그림 3] 취약 패턴 알고리즘이 적용되는 부분

2.3 취약 패턴 검출 알고리즘의 수학적 원리

제안하는 패턴 검출 알고리즘은 잡음원의 분포가 독립이고 균등분포로부터 출력되었다는 가정 아래서 설계되었기 때문에, 패턴 검출 확률은 다음과 같이 계산된다.

크기가 k 비트인 잡음원 워드의 확률변수 X_i는 이상적인 분포(Independent and identically uniformly distributed)로부터 출력되었다고 할 때, 증가 패턴이 발생한 경우 $(X_{i+1} > X_i)$의 확률변수를 Y라 하자. 패턴이 발생한 경우를 성공 1, 아닌 경우를 실패 0으로 보면, 확률변수 Y는 [표 1]과 같은 베르누이 분포(Bernoulli distribution)를 따른다.

[표 1] 증가 패턴에 대한 확률분포

Y	0	1
$P(Y)$	$1-\frac{1}{2}\left(1-\frac{1}{2^k}\right)$	$\frac{1}{2}\left(1-\frac{1}{2^k}\right)$

증가하는 패턴이 발생할 확률은 다음과 같이 계산된다.

현재의 워드를 나타내는 확률변수를 X_1, 다음 워드를 나타내는 확률변수를 X_2라 하자. 이때, 워드의 변화에 대해 다음과 같이 증가, 감소, 동일한 경우로 쓸 수 있다.

$$P(X_1 > X_2) + P(X_1 = X_2) + P(X_1 < X_2) = 1$$

증가와 감소하는 경우는 대칭(Symmetric)이므로, 위 식을 아래와 같이 표현 할 수 있다.

$$2 \times P(X_1 > X_2) = 1 - P(X_1 = X_2).$$

이때 충돌이 발생할 확률은 $P(X_1 = X_2) = \frac{2^k}{2^{2k}}$ 이므로, 증가 패턴이 발생할 확률은 아래와 같다.

$$2 \times P(X_1 > X_2) = 1 - \frac{1}{2^k}.$$

$$\Leftrightarrow \; P(X_1 > X_2) = \frac{1}{2}\left(1 - \frac{1}{2^k}\right).$$

k가 충분히 큰 경우에 $P(X_1 > X_2)$은 $\frac{1}{2}$로 근사하는 사실을 활용하면, 연속적으로 증가 패턴이 발생할 확률은 아래와 같이 계산할 수 있다.

$$\begin{aligned}
&P(X_1 < X_2, X_2 < X_3) \\
&= \sum_{x_2=1}^{2^k} P(X_2 = x_2) \times P(X_1 < x_2 < X_3 | X_2 = x_2) \\
&= \sum_{x_2=1}^{2^k} P(X_2 = x_2) \times P(X_1 < x_2) \times P(x_2 < X_3) \\
&= \sum_{x_2=1}^{2^k} \frac{1}{2^k} \times P(X_1 < x_2) \times P(x_2 < X_3) \\
&= \frac{1}{2^{3k}} \times \sum_{i=1}^{2^k-2} i \times (2^k - 1 - i) \\
&= \frac{1}{2^{3k}} \times (2^k - 1) \times \sum_{i=1}^{2^k-2} i - \sum_{i=1}^{2^k-2} i^2 \\
&= \frac{1}{2^{3k}} \times (2^k - 1)(2^k - 2) \times \left\{ \frac{(2^k - 1)}{2} - \frac{(2^{k+1} + 1)}{6} \right\} \\
&\simeq \frac{1}{6}.
\end{aligned}$$

k가 충분히 큰 경우 연속적으로 증가하는 패턴이 검출될 확률은 $\frac{1}{6}$로 근사한다.

한편, 위의 결과를 이용하여 연속적으로 n번 증가하는 패턴이 발생할 확률을 계산하면 다음과 같다.

$$\begin{aligned}
&P(X_1 < X_2 < X_3 < \cdots < X_n) \\
&= \frac{1}{2^{nk}} \times \sum_{i=n-1}^{2^k-1} \binom{i}{n-1} \\
&\approx \frac{1}{n!}.
\end{aligned}$$

여기서 취약 패턴이 발생할 확률은 가설검정의 유의확률($p-value$)을 의미하며, 계산된 확률이 설정한 유의수준 α보다 작은 경우 이 잡음원은 취약한 패턴이 있음을 의미한다. 이때, 아 관계식을 이용하여 취약 패턴이 발생하는 횟수가 계산된다. $\alpha = 2^{-40}$로 예를 들어 설명하면,

$$\begin{aligned}
\frac{1}{n!} < 2^{-40} \;&\Leftrightarrow \frac{1}{n!} < 2^{-40} \\
&\Leftrightarrow \frac{e^n}{\sqrt{2\pi n}\, n^n} < 2^{-40} \\
&\Leftrightarrow n > 14.9387.
\end{aligned}$$

즉, 유의수준 2^{-40}에서는 연속적으로 15번이 증가하는 경우에 잡음원에 취약 패턴이 있다고 판정한다.

III. 취약 패턴 검출 실험 및 활용방안

패턴 알고리즘에 대한 실험은 다음 두 경우에 대하여 진행되었다. 첫째, 잡음원의 수집단위인 워드 단위로 패턴을 검출하는 실험을 통해 잡음원 출력의 특성을 확인하는 워드 단위 패턴 검출 실험이 진행되었고, 둘째, 잡음원 고유의 특성을 확인하기 위한 잡음원 바이트 별 패턴 검출 실험이 진행되었다.

3.1 워드 단위의 잡음원 취약 패턴 검출 실험

소프트웨어 잡음원은 워드 단위로 출력되는 특징이 있다. 워드 단위로 출력되는 잡음원의 취약 패턴 분석을 통해 현재 사용하는 잡음원 출력 자체의 취약성을 확인한다. [표 2]는 워드 단위로 소프트웨어 잡음원의 취약 패턴을 확인한 예시이다.

[표 2] 소프트웨어 잡음원의 취약 패턴 검출 예시

잡음원 명	상수 패턴	증가 패턴
JRE 운영체제		
nanoTime	X	O
getProperty	O	X
Windows 운영체제		
GetCurrentThreadId	O	X
GetTickCount	X	O

3.2 잡음원의 내부 바이트 별 패턴 검출

소프트웨어 잡음원의 바이트 별 패턴 검출 실험은 잡음원이 가지는 고유 특성을 분석하기 위해 진행되었다. 바이트 별 패턴 검출 실험은 잡음원의 각 바이트 자리의 취약 패턴을 검출하여 유의미한 소프트웨어 잡음원의 수집 길이를 도출해내는 것에 목적이 있다. [표 3]은 소프트웨어 잡음원의 바이트 별 취약 패턴이 발생한 위치와 유의미한 잡음원의 최소 수집 바이트 길이를 보여준다. [표 3]의 nanoTime을 예로 들면, nanoTime은 1, 2번째 위치에서 상수 패턴이 검출되고, 나머지 바이트 위치에서는 패턴이 검출되지 않았다. 즉, nanoTime을 이용한 잡음원 수집은 3바이트 이상을 수집하는 경우에 랜덤한 값이 출력됨을 의미하기 때문에, nanoTime을 잡음원으로써 사용하기 위해서는 최소 3바이트를 수집해야 함을 보여준다.

[표 3] 소프트웨어 잡음원 바이트 별 패턴 분석 예시

잡음원	상수패턴	증가패턴	수집바이트
JRE 운영체제			
nanoTime	**1, 2**	X	**3**
getProperty	**전체**	X	64
Windows 운영체제			
GetCurrent ThreadId	**전체**	X	4
GetTick Count	X	**3**	4
Linux 운영체제			
perfstat_disk_total	**40, 47**	X	48
psinfo	**54**	X	**55**

IV. 결론

본 논문에서는 현재 활용되고 있는 운영체제 기반 소프트웨어 잡음원의 취약 패턴 분석을 통한 잡음원의 특성 분석을 진행하였다. 먼저 소프트웨어 잡음원의 특징 분석을 통해 취약 패턴을 상수 패턴과 증가 패턴으로 정의하였고, 취약 패턴이 발생할 확률을 계산하였다. 제안한 알고리즘은 취약성을 가진 잡음원을 사전에 걸러내는 목적으로 설계되었으며, 취약 패턴 검출에는 계산된 확률이 적용되었다. 또한, 취약 패턴 검출 알고리즘을 활용한 잡음원의 바이트 별 패턴 검출을 통해 소프트웨어 잡음원의 고유한 특성을 확인할 수 있었다. 특성 분석을 통해 제시한 최소 수집 바이트 길이는 운영체제 기반 소프트웨어 잡음원이 이용되는 보안시스템 개발과 안전성 평가에 있어서 실용적인 기준으로 활용될 것으로 기대한다.

Acknowledgement

이 논문은 2017년도 ETRI 부설연구소의 지원(No.2017-075, OS 잡음원 패턴 추출을 통한 종속성 분석기법 연구)과 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No.2017-0-01857, 난수발생기 및 임베디드 기기 안전성 연구)

[참고문헌]

[1] A. Vassilev and R. Staples, “Entropy-as-a-Service: Unlocking the Full Potential of Cryptography”, Computer vol. 49.9, 2016.

[2] TTA, “운영체제별 잡음원 수집 및 응용지침”, 2013.

[3] 김예원, 염용진, 강주성, “소프트웨어 잡음원에 대한 평가방법 연구”, 2016한국정보보호학회동계학술대회, 2016.

[4] TTA, “소프트웨어 환경에서의 엔트로피 검증 알고리즘”, 2017.

동적 정보 기반 악성코드 그룹 분류 및 특징 연구*

박래현*, 유정빈*, 박준형**, 권태경*
* 연세대학교 정보대학원 정보보호 연구실
** 한국 인터넷 진흥원 (KISA)

A Study of Dynamic Feature-based Group Classification of Malware

Laehyun Park*, JungBeen Yu*, Jun-hyung Park**, Taekyoung Kwon*
*Information Security Lab, GSI, Yonsei University.
** Korea Internet and Security Agency (KISA)

요 약

악성코드 제작자들은 기존 악성코드의 탐지를 우회하기 위해 변종 악성코드를 제작하고 있다. 이러한 변종 악성코드의 수는 해마다 증가하는 가운데 기존의 정적 기반 연구는 변종 악성코드를 식별하는데 한계가 있다. 본 연구는 대량의 변종 악성코드를 식별하기 위해 동적 특징을 활용한다. 또한 머신러닝 알고리즘을 활용해 악성코드의 패밀리를 분류하고 각 패밀리 내의 악성코드를 대상으로 세부 그룹을 분류해 악성코드 식별의 정확성과 효율성을 높인다. 나아가 실험을 통해 각 악성코드 패밀리 별로 그룹화를 위한 최적의 행위 정보를 선정한다.

I. 서론

2017년 보고된 Symantec ISTR에 따르면 새로 등장한 변종 악성코드는 2014년 약 2억 7,400만 개에서 2015년 약 3억 5,500만 개, 2016년 약 3억 5,700만 개로 해마다 증가하는 추세다 [1]. 이러한 변종 악성코드들은 동일한 악성코드 제작자가 기존 악성코드의 탐지를 우회하기 위해 다양한 변형 기술을 적용시킨 것들이다. 따라서 변종 악성코드의 제작자 정보와 각 악성코드 간의 연결된 정보를 파악하기 위해 기존 악성코드와의 관계식별이 필요하다.

기존의 악성코드 식별 방식은 위와 같은 대량의 악성코드를 식별하는데 많은 시간이 소요된다. 따라서 악성코드를 효율적으로 식별하기 위해 머신러닝 기반의 연구가 진행되어 왔다. 하지만 악성코드의 바이너리나 명령어를 주로 사용하는 기존의 정적 특징 기반의 악성코드 식별 연구는 변종 악성코드 문제를 해결하는데 한계가 있다.

본 연구는 정적 특징의 한계를 해결하기 위해 먼저 악성코드의 동적 특징을 추출하여 패밀리를 분류한다. 또한 변종 악성코드가 기존의 어떤 악성코드에서 파생된 것인지 파악하기 위해서 패밀리 내 세부 그룹 분류를 통해 변종 악성코드와 기존 악성코드의 그룹화를 위한 연구를 진행한다.

II. 배경

변종 악성코드는 다형성 (polymorphic) 악성코드와 변성 (metamorphic) 악성코드로 나뉜다. 다형성 악성코드는 기존 악성코드에 패킹, 암호

* 이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No. 2016-0-00081, 악성코드 全 생명주기 통합 프로파일링 및 공격그룹 식별 기술 개발).
또한 과학기술정보통신부 및 정보통신기술진흥센터의 대학ICT연구센터육성지원사업의 연구결과로 수행되었음 (IITP-2017-2016-0-00304).

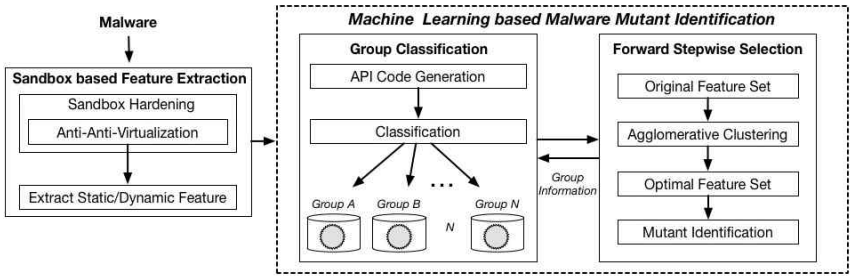


그림 1 전체 시스템 구성도

화, 난독화 기술을 적용시켜 바이너리의 형태를 변형시킨 악성코드이다. 변성 악성코드는 junk code를 추가하거나, 레지스터 재정렬 등을 통해 악성코드를 유포할 때마다 코드를 재생성하는 악성코드이다.

머신러닝은 프로그래밍 되지 않은 데이터를 컴퓨터가 학습할 수 있게 해주는 컴퓨터의 한 분야이다. 악성코드 분야에서 머신러닝 기술은 기존의 시그니처 탐지 방식이 변종 악성코드를 식별하기 어렵다는 한계를 해결하기 위해 사용된다. 악성코드 식별에 주로 사용하는 머신러닝 알고리즘은 지도학습인 분류 (classification)와 비지도학습인 클러스터링 (clustering)이 있다.

분류 알고리즘은 사전에 분류된 훈련 데이터 셋을 통해 함수를 도출해내는 방법이다. 학습을 통해 새로운 데이터 셋에 대해서도 분류할 수 있는 것을 목표로 한다.

클러스터링 알고리즘은 알려지지 않은 샘플의 정확한 특징을 계산하여 데이터 셋을 서로 분리된 구조로 나누는 데 초점을 맞춘다. 본 연구는 정확한 악성코드 식별을 위해 분류와 클러스터링 알고리즘 두 가지를 모두 사용한다.

III. 실험 설계

실험 환경은 Intel(R) Core(TM) i5-6600 CPU @ 3.30GHz, 32GB RAM, Ubuntu 16.04.2 LTS로 구성했다. 전체 시스템 구성은 그림 1과 같이 샌드박스 분석, 패밀리 분류, 동적특징 연구 및 세부 그룹 분류로 구성된다. 머신러닝은 python 기반의 scikit-learn을 활용했다 [3].

3.1 샌드박스 분석 및 악성코드 특징 추출

샌드박스는 제어된 가상 환경에서 악성코드를 직접 실행해 분석하는 소프트웨어를 말한다. 샌드박스 분석은 실행과 복구가 자유로워 호스트 시스템에 영향을 주지 않고 악성코드를 분석할 수 있다. 이 같은 장점을 활용해 Cuckoo 샌드박스를 기반으로 악성코드 분석을 진행하고 동적 특징을 추출한다 [4].

3.2 API 호출 기반 악성코드 패밀리 분류

패킹 등의 기술이 적용된 변종 악성코드를 식별하기 위해 Cuckoo 샌드박스에서 추출한 악성코드의 API 호출 정보를 활용해 악성코드 패밀리를 분류한다.

3.3 Forward Stepwise Selection 알고리즘을 활용한 악성코드 패밀리 별 동적특징 연구

분류된 각 악성코드 패밀리 별로 동일 악성코드 제작자 그룹을 식별하기 위한 실험을 진행한다. Forward stepwise selection 알고리즘을 적용해 각 패밀리 별로 세부그룹 클러스터링을 위한 최적의 특징을 선정한다.

IV. 샌드박스 분석 및 특징 추출

샌드박스는 머신러닝을 포함한 자동화된 악성코드 분석 시스템에 유용하게 사용될 수 있다. 하지만 최근에는 샌드박스의 악성코드 탐지 및 분석을 회피하기 위해 안티-가상화, 난독화 등의 샌드박스 우회 기술을 적용한 악성코드들이 등장했다 [2]. 이러한 문제를 고려하여 본 연구는 기존 샌드박스보다 정확하게 악성코드를 분석할 수 있도록 강화된 cuckoo 샌드박스를 사용 한다 [5].

강화된 샌드박스에서는 악성코드 식별을 위해 다양한 동적 특징을 추출한다. 먼저 악성코

드 패밀리 분류를 위해서 API 호출 정보를 추출한다. 이후 악성코드 패밀리 별 세부 그룹 분류를 위해 네트워크, 레지스트리 키, 뮤텍스, 서비스, 파일, 커맨드 등의 동적 특징을 추출한다.

V. 악성코드 패밀리 분류

5.1 n-gram

n-gram은 대표적인 확률적 언어 모델로 악성코드 API sequence와 같은 순서 값에도 적용 가능하다. 따라서 아래와 같은 과정을 거쳐 API 호출 정보를 2-gram한 값을 사용한다.

1) 샌드박스를 통해 악성코드의 API 호출 정보를 추출하고 API sequence를 생성한다.

$$\{\alpha, \alpha, \alpha, \beta, \gamma, \gamma, \delta\} \in malware_A$$
$$\{\alpha, \alpha, \beta, \gamma, \delta, \delta\} \in malware_B$$

2) 2-gram을 이용해 악성코드 API sequence를 인덱싱한다.

$$\{\alpha\alpha : 2, \alpha\beta : 1, \beta\gamma : 1, \gamma\gamma : 1, \gamma\delta : 1\}$$
$$\{\alpha\alpha : 1, \alpha\beta : 1, \beta\gamma : 1, \gamma\delta : 1, \delta\delta : 1\}$$

2-gram으로 가공한 특징 벡터에는 각 악성코드 API의 빈도수가 저장된다. API 특징을 머신러닝에 활용하기 위해서는 각 악성코드마다 벡터의 길이가 같아야한다. 따라서 모든 악성코드가 호출한 API 정보를 인덱스로 표현하고 호출되지 않은 API의 인덱스는 0으로 표현한다.

5.2 악성코드 패밀리 분류

앞서 추출한 악성코드 API 빈도수 정보를 바탕으로 악성코드 패밀리 분류를 진행한다. 분류 알고리즘은 random forest를 사용한다. Random forest는 decision tree에서 트리의 개수를 여러 개로 늘린 뒤 특징에 randomize 방식을 적용해 성능을 향상시킨 알고리즘이다.

VI. 악성코드 패밀리 별 동적 특징 선정 및 세부 그룹 분류

악성코드 패밀리 분류가 완료되면 각 패밀리 내 동일 공격자 그룹 분류를 위한 실험을 진행한다. 각 패밀리마다 나타나는 행위 정보가 다르기 때문에 패밀리 내 세부 그룹을 분류하기 위한 동적 특징 또한 달라질 수 있다. 따라서 각 패밀리에 forward stepwise selection 알고리즘을 적용해 세부그룹 분류를 위한 최적의 동적 특징을 추출한다.

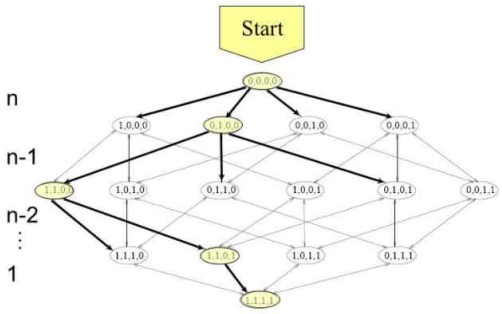


그림 2 Forward stepwise selection

Forward stepwise selection 알고리즘은 특징 선정 알고리즘의 한 종류로 특징 후보군을 형성한 뒤 그림 2와 같이 계단식으로 특징을 추가해 나가면서 성능을 측정하는 알고리즘이다. 전수조사에 비해 계산 량을 줄이면서도 다양한 경우의 수를 탐색할 수 있다는 장점이 있다.

최적의 특징 조합을 선정하기 위해서는 각 특징 후보의 점수를 측정하기 위한 방법이 필요하다. 따라서 각 특징 후보 별로 agglomerative clustering을 진행한 뒤 silhouette 점수를 계산한다. Silhouette 점수는 -1과 1 사이로, 1에 가까울수록 클러스터링 결과가 좋은 것을 의미한다. 따라서 silhouette 점수가 가장 높은 특징 조합을 최적의 특징 조합으로 선정한다.

VII. 관련 연구

악성코드 식별과 관련된 선행 연구로는 머신러닝을 이용한 악성코드 패밀리 분류와 악성코드 계통 식별 등이 있다.

2009년 Usenix LEET에서 U. Bayer et al.은 Anubis 샌드박스에서 추출할 수 있는 악성코드의 다양한 동적 특징에 대해 연구했다 [6]. 이들은 악성코드 클러스터링을 진행한 뒤 각 클러스터마다 나타나는 동적 특징의 분포를 조사했다. 조사한 특징으로는 파일 시스템 행위, 레지

스트리 행위, 네트워크 행위 등이 있다. Anubis 샌드박스는 현재 서비스가 종료된 상태이기 때문에 본 연구는 Cuckoo 샌드박스를 통해 악성코드의 동적 특징을 추출했다.

2013년 Usenix Security에서 J. Jang et al.은 소프트웨어의 혈통을 식별하기 위한 연구를 진행했다 [7]. 이들은 파일의 섹션 정보, 명령어 정보와 API 호출 정보 등을 특징으로 사용했다. 또한 hierarchical clustering을 통해 동일한 소프트웨어를 클러스터링한 뒤 각 샘플 간의 symmetric distance를 측정해 샘플의 버전에 따른 그래프를 형성했다. 이들은 다양한 버전의 소프트웨어와 악성코드를 대상으로 혈통을 식별하기 위한 실험을 진행했으나 악성코드 샘플의 개수는 84개에 불과하다는 한계점이 있다.

2015년 Usenix Security에서 M. Graziano et al.은 Anubis 샌드박스를 통해 악성코드 샘플을 수집하였고, 이들 중 개발 과정이 있는 악성코드를 탐지했다 [8]. ssdeep을 이용해 유사 샘플을 클러스터링 하였고 머신러닝을 통해 클러스터가 실제 악성코드 개발 과정인지 분류했다. 하지만 이들은 바이너리의 용이한 분석을 위해 패킹된 샘플을 제거했다. 본 연구는 패킹된 샘플을 식별하기 위해 샌드박스 기반의 다양한 동적 정보를 활용했다.

2009년 Usenix LEET에서 G. Wicherski는 악성코드의 확장성과 다형성 (polymorphic) 악성코드 문제를 해결하기 위한 연구를 진행했다 [9]. 먼저 압축 툴을 통해 샘플을 해싱하고 PE 특징을 추가했다. 이후 해당 값을 SHA1로 다시 해싱 했다. 악성코드 샘플은 해시 값이 100% 일치하는 샘플만 클러스터링 했다. 클러스터링 성능은 100%의 precision을 기록하였고, 기존의 샌드박스 분석시간 문제를 해결할 수 있었다. 하지만 클러스터 결과 중 단일 샘플만 포함한 클러스터가 대부분이었다는 한계가 존재한다.

VIII. 결론

본 연구는 강화된 샌드박스 분석을 통해 악성코드의 다양한 동적 특징을 추출할 수 있었다. 또한 샌드박스에서 악성코드 API 호출 정보를 추출해 n-gram으로 가공했고 분류 알고리즘을 통해 악성코드 패밀리를 식별했다. 마지막으로 forward stepwise selection 알고리즘을 통해 악성코드 패밀리 내 세부 그룹 분류를 위한 최적의 특징 조합을 도출했다.

추가로 세부 그룹 내에서 기존 악성코드와 변종 악성코드를 분류하기 위한 작업이 필요하다. 또한 기존의 패밀리에 속하지 않는 새로운 악성코드를 판별하고 이를 처리하기 위한 방법이 필요하다.

[참고문헌]

[1] Symantec, Internet Security Threat Report Volume 22, https://www.symantec.com/content/dam/symantec/docs/reports/istr-22-2017-en.pdf, page 38, 2017.

[2] Symantec, Internet Security Threat Report Volume 21, https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf, page 65, 2016.

[3] Scikit-learn, http://scikit-learn.org/stable/index.html

[4] Cuckoo Sandbox, https://cuckoosandbox.org/

[5] BradSpengler and 102 contributors, Cuckoo-modified, https://github.com/spender-sandbox/cuckoo-modified, 2015.

[6] U. Bayer, I. Habibi, D. Balzarotti, E. Kirda, C. Kruegel, A View on Current Malware Behaviors, In Proc. the Usenix workshop on large-scale exploits and emergent threats (LEET), 2009.

[7] J. Jang, M. Woo, D. Brumley, Towards Automatic Software Lineage Inference, In Proc. the Usenix Security Symposium, pages 81-96, 2013.

[8] M. Graziano, D. Canali, L. Bilge, A. Lanzi, D. Balzarotti, Needles in a Haystack: Mining Information from Public Dynamic Analysis Sandboxes for Malware Intelligence, In Proc. the Usenix Security Symposium, pages 1057-1072, 2015.

[9] G. Wicherski, peHash: A Novel Approach to Fast Malware Clustering, In Proc. the Usenix Workshop on Large-Scale Exploits and Emergent Threats, pages 8, 2009.

2017 한국정보보호학회 동계학술대회

B2 정보보호관련법 및 제도

김 기 범 (경찰대)

- **SI 프로젝트에서 아키텍트 보안활동 수행 및 효과**
  남혜리 (고려대학교)
- **DLMS/COSEM 기반의 스마트미터링 플랫폼 수립에 대한 국제 표준화 동향 및 국내 적용 방안**
  강동주, 양인석 (고려대학교)
- **중 · 소형 공공기관 정보보안 관리방법 체계화 연구**
  강창식, 권헌영 (고려대학교)
- **NERC CIP 규정 분석에 근거한 전력시스템 신뢰도 관점에서 CPS 영향도 평가 방법에 관한 연구**
  강동주 (고려대학교)

SI 프로젝트에서 아키텍트 보안활동 수행 및 효과

남혜리*

*고려대학교 컴퓨터정보통신대학원

The Effect of Architect's Security Process in SI project

Hye-Ree Nam*

*Korea University Graduate School of
Computer and Information Technology.

요 약

소프트웨어 개발 시 시스템구축(SI) 프로젝트에서 적극적인 보안활동을 수행함으로써 시스템의 품질을 확보하고 취약점을 제거할 수 있으나 제한된 비용과 기간 내에 보안활동을 철저히 수행하기에는 많은 어려움이 있다. 본 논문에서는 마이크로소프트의 MS-SDL 방법론과 Seven Touchpoints 모델을 바탕으로 소프트웨어 개발 라이프사이클(SDLC)의 각 단계에서 아키텍트가 수행할 수 있는 보안활동을 제시하였다. 이를 바탕으로 규모와 성격이 유사한 두 가지 프로젝트를 대상으로 보안활동을 수행하고 그 결과를 분석 비교함으로서 SI 프로젝트의 보안활동 효과와 보안 관리의 중요성을 확인하였다.

I. 서론

2017년 초 한국인터넷진흥원(KISA)은 소프트웨어 개발보안 가이드를 발표하였다. 개인정보 등을 노리는 악성사용자의 공격방식이 점차 어플리케이션 계층에 들어오는 추세에 따라 KISA는 시큐어코딩 가이드만 발표하던 기존 취지에서 [소프트웨어 개발보안], [분석/설계단계 SW보안강화 활동] 등의 내용을 포함한 전자정부 SW개발자 및 운영자를 위한 보안활동의 지침을 제시하고 있다.[1]

실제 보안활동에 있어 프로젝트 구축의 단계별로 명확한 TASK가 정의되어 있으면 프로젝트 수행자가 유용하게 활용할 수 있으므로 그 역할별 범위를 구체적으로 정의할 필요가 있다.

본 논문에서는 [소프트웨어 개발보안 가이드] 에서 설명하고 있는 보안개발 방법을 연구하고, 아키텍트 관점에서 수행할 수 있는 TASK를 도출하여, 실제 프로젝트에 적용함으로써 그 효과를 확인하고자 한다.

II. 보안개발 방법론

소프트웨어 개발 보안방법론은 마이크로 소프트에서 제시한 MS-SDL과 IEEE에서 제시하고 발전해 온 Seven Touchpoints 가 대표적이며 각 방법론의 개념은 다음과 같다.

2.1 MS-SDL

MS-SDL (Microsoft Secure Development Lifecycle) 은 SW개발보안에서 가장 실용적인 접근이 가능한 모델이며 마이크로소프트사 모든 제품의 개발과정이 필수적으로 적용되는 방법론이다. SDL방법론에서는 제품 개발단계에 따라 수행해야 하는 각종 보안활동을 정의하고

있으며 그 내용은 아래와 같다.

표 II-1. 마이크로소프트 SDL[1][2]

Stage	TASK
Training 교육	소프트웨어 개발 보안 교육
Requirements 계획/분석	소프트웨어의 품질과 버그경계 정의 (STRIDE 모델링 활용) 보안과 개인정보 위험 분석
Design 설계	보안설계검토, 방화벽 정책 준수, 위협모델링, 보안설계서 장성 및 샘플소스 보안검토 수행
Implementation 구현	최신버전의 도구 명세, 금지된 함수사용 등 제한요소 정의, 보안성 고려된 사례 가이드, 정적분석
Verification 시험/검증	동적/퍼징 테스팅 공격영역/위협모델검증 보안문서계획서검토
Release 배포/운영	사고대응계획, 수행된 보안활동에 대한 검토, 소프트웨어 인증, 기록보관
Response 대응	사고대응수행

2.2 Seven Touchpoints

Seven Touchpoints는 7가지 요소로 구성되어 있으며 보안성을 확보하기 위해 시스템을 설계하고 방어하는 활동, 외부공격이나 시스템 파괴에 대비하는 활동에 대해 정리되어 있다. 보안활동 시 Touchpoint 를 모두 채택할 필요는 없으나, 이를 적용하면 보다 안전한 소프트웨어를 구축할 수 있다. 각 요소에 대한 내용은 아래와 같다.

표 II-2. Seven Touchpoints 항목[1][3]

Task	Description
Code Review (Tools)	코드레벨에서 구현오류에 중점을 두어 소스코드에 대한 정적분석 수행 (필수적이나 보안성 확보에는 불충분)
Architectural Risk Analysis	아키텍처위험요소분석 - 통일된 보안기능 제공 - Attack Resistance 분석/모호성 분석/허점분석을 통해 가능한 공격 식별
Penetration Testing	아키텍처를 고려한 침투테스트를 통해 실제 환경에서 공격에 대한 소프트웨어의 작동을 확인할 수 있음
Risk-Based Security Testing	공격패턴, 위험분석 결과, 악용사례를 기반으로 위험기반 보안테스트를 수행
Abuse Cases	악용사례와 위험분석을 통하여 보안요구사항에 대한 정의와 명세 작성 (오용사례에 대한 정의 및 케이스 예시 등)
Security Requirements	보안에 대한 요구사항을 명시적으로 정의 (기능적인 정의와 비상상황에 대한 정의 모두 포함)
Security Operations	소프트웨어 동작 모니터링을 하며 보안운영을 통해 얻은 공격에 대한 경험과 지식은 개발자에게 다시 피드백

III. 보안활동 항목 도출

일반적인 웹 어플리케이션 구축 프로젝트에 있어서 소프트웨어 개발보안의 방법론을 적용하기 위해서는 각 수행활동의 명료화가 필요하며, 체계적으로 접근하지 않을 경우 많은 비용이 발생할 수 있다. 또한 MS-SDL의 단계와 7 Touchpoints의 수행시점이 소프트웨어 개발 생명주기(SDLC)의 단계와 명확히 연결되지 않을 수 있으며, 일부 활동은 여러 단계에 걸쳐 구성된 것도 있다.

제시 된 여러 보안활동을 목적과 실효성을 고려하여 SDLC 단계 별로 아키텍트가 SI 프로젝트에 실제 적용할 수 있는 활동 내용은 아래와 같다.

표 III-1. SDLC 단계 별 보안활동[1][2][3]

Stage	Action	TASK
분석	보안요구사항 수집/정의	개인정보 활용 여부, 외부인터페이스, 법 관련 이행사항점검 요건 확인
	제약사항정의	사용자 성격(대민/사내)과 시스템 성격(트랜젝션 수) 등을 고려한 인프라/소프트웨어 설계 (모두 실현은 불가)
설계	인프라 정책	서버계정 권한/디렉토리 권한 정책, 네트워크정책, 서버 어드민 화면 접근금지

	응용프로그램	사용자인증, 권한별 접근제어 프로그램 설계, 개인정보처리 / 데이터 암호화 방안 및 관련 샘플 제시
	적절한 솔루션 적용	영역별 전문솔루션 검토 (접근제어, 보안로그인, 암호화 솔루션 등)
구현	인프라 환경구성	보안을 고려한 최소권한, 계정 분리가 반영된 설계 적용
	프레임워크 구성	XSSFilter, 세션처리, 암호화 API 등 주요 공통컴포넌트 개발 및 가이드
검증	보안 아키텍처 평가	설계된 보안요소 정상연동 확인, 기술적으로 보안성을 확보하였는지 평가 취약점 스캔, 침투 테스트 OSS라이선스 및 OSS자체보안 검증
	추가요구사항 수집	보완 되어야하는 보안요소 도출
운영	보안개발가이드	운영단계에서 지켜져야 하는 원칙 교육

SDLC 단계별 보안활동은 분석/설계/구현단계에서 각 시스템 리소스의 보안요소를 정의하고 기술적인 해결방안과 여러 정책 등을 제시한다. 또 분석설계자와 개발자 등이 체계적인 보안활동을 수행할 수 있도록 하고, 검증/운영단계에서 안전한 시스템이 유지될 수 있도록 구성된 보안아키텍처를 평가하고 보완할 수 있다.

따라서 표 III-1에서 제시된 항목은 시스템 구축(SI) 프로젝트에서 아키텍트가 수행할 수 있는 일반적인 활동이므로 상황에 따라 항목이 생략되거나 추가되어 활용될 수도 있다.

IV. 적용 및 결과비교

아키텍트의 보안활동에 따른 효과비교를 위해 유사한 규모의 웹 어플리케이션 구축 프로젝트 A, B를 대상으로 실험을 하였다. 두 프로젝트 모두 Linux 계열의 OS에서 Apache-HTTPD Web Server 와 Apache-Tomcat Web Application Server 를 사용하였으며, 응용프로그램은 Java 기반의 Spring Framework 로 시스템을 구축하였다. 프로젝트A에서는 KISA에서 제시한 가이드를 참고하여 프로젝트 수행 시 필요한 보안활동을 하였으며, 프로젝트B 에서는 표 III-1 에서 도출한 SDLC 단계별 아키텍트의 보안활동을 바탕으로 수행하였다. 이 과정을 위해 별도의 인력이 추가되거나 비용의 차이는 발생하지 않았다.

두 프로젝트에 대한 보안평가는 모의해킹과 AppScan을 활용한 동적검사 및 오픈소스 라이선스 검사를 수행하였으며 상세한 평가항목과 평가결과는 아래와 같다.

표 IV-1. 보안평가 항목

평가 항목	등급	설명
CrossSiteScripting	H	XSS 동작 방지를 위한 c:out 태그사용 또는 html 인코딩처리
Cryptography.PoorEntropy	M	보안 결정이 필요한 경우 java.security.SecureRandom 클래스 사용
ErrorHandling.RevealDetails.Message	L	Exception 메시지가 사용자에게 노출되지 않는지 확인(사용자 정의 메시지로 출력하도록 조치)
ErrorHandling.RevealDetails.StackTrace	L	Stacktrace 삭제,필요 시 Log에 기록
Injection.HttpResponseSplitting (H)	H	Header 작성 시 사용된 변수에 개행문자 치환
Injection.SQL	H	Prepared Statement를 활용한 바인딩 변수 사용
PathTraversal	H	파일 참 조시 경로 우회에 의한 의도치 않은 파일 생성/삭제/조회 여부 확인
Quality.TestCode	L	운영 모드에서는 삭제, 필요 시 Log에 기록
Validation.EncodingRequired	L	XSS 동작 방지를 위한 html 인코딩처리 필요
Validation.Required	L	사용자 입력 값에 대한 Server Side 유효성 검증
OSS License	H	OSS 라이선스 위배 확인

위 평가항목은 빈번히 검출되면서 반드시 조치되어야 하는 것 위주로 11개의 항목을 선정하였다. 다른 항목도 함께 검사하였으나 위험도

가 상대적으로 낮은 항목은 표기를 생략하였다. 두 프로젝트에서 각 항목에 대한 위배건수는 아래와 같다.

표 IV-2. 보안평가 결과 비교

평가 항목	A PJT	B PJT
CrossSiteScripting	51	2
Cryptography.PoorEntropy	1	0
ErrorHandling. RevealDetails.Message	21	3
ErrorHandling. RevealDetails.StackTrace	6	5
Injection. HttpResponseSplitting	5	0
Injection.SQL	3	0
PathTraversal	14	3
Quality.TestCode	14	0
Validation. EncodingRequired	1	0
Validation.Required	4	1
OSS License	2	0
총 건수	**122**	**14**

비교 결과 11개의 평가 항목 중 KISA의 지침가이드를 참고하여 필요한 것으로 판단되는 보안활동을 수행한 A프로젝트는 모든 항목에서 취약점이 검출되었으며, 단계별 보안활동을 체계적으로 수행한 B프로젝트는 미검출 항목이 6개였다. 취약점의 총 건수도 각 122건과 14건으로 큰 차이를 보였다.

시스템보안성이 취약점의 유무로 평가되는 성질을 고려하면 B프로젝트의 시스템이 상대적으로 안전한 시스템이다.

V. 결론

안전한 시스템을 만들기 위해 정부의 가이드와 여러 방법론을 참고하여 보안활동을 수행하는 것은 중요하다. 하지만 실험 결과처럼 시스템보안을 위해 아키텍트가 적극적으로 기술관점에서 보안 아키텍처를 정의하고 설계하며 SDLC 단계별로 구체적인 활동을 수행한다면 더 나은 보안지표를 받을 수 있음을 확인하였다. 이는 비용 측면에서도 보안 취약성과 잘못된 구성을 조기에 식별하고 완화하여 저비용을 보안아키텍처를 실현할 수 있음을 의미한다.[4]

시스템을 구성하는 여러 요소 중 어플리케이션이 가장 위험 하다는 결과가 있다.[5] 가이드와 방법론 수준에서 더 깊이 들어가 실제적인 보안구축 방법과 단계별 상세활동이 다양하게 공유되어 쉽게 접할 수 있다면, 아키텍트 뿐 아니라 어플리케이션 개발자들에게도 실천해야 하는 활동으로 정착되는 것이 어렵지 않을 것이다. 이런 문화를 바탕으로 보안성이 우수한 어플리케이션이 구축되어 보안사고가 줄어들길 기대해본다.

[참고문헌]

[1]Korea Internet & Security Agency, Software Development Security Guide, pp.021-027, 2017

[2]MS-SDL,"https://msdn.microsoft.com/ko-kr/library/cc307748.aspx"

[3]Gary McGraw, Software Security: Building Security In, Part2 Seven Touchpoints for Software Security, 2006

[4]Lee Kyung-Hwan and Jang Seong-Yong, "A Study of Security Activities by Each Step in the Software Development Life cycle," KCI, pp.396-399, 2015

[5]Ponemon Institute, "How to Make Application Security a Strategically Managed Discipline," pp.8, 2016

DLMS/COSEM 기반의 스마트미터링 플랫폼 수립에 대한 국제 표준화 동향 및 국내 적용 방안

강동주, 양인석
고려대학교 정보보호대학원

Study on International Trend for DLMS/COSEM based Smart Metering Standards and Domestic Application Method

Dong-Joo Kang and In-Seok Yang
Division of information security, Korea University.

요 약

스마트미터는 10여년전 스마트그리드 초기 시점부터 유럽 및 북미를 중심으로 보급되어 왔으나, 기반시설로서의 전력인프라와 과금용 전력량계에 대한 강한 규제로 인해 그 활용성이 강하게 제한되어 왔다. 사물인터넷 서비스 모델에서 에너지 데이터는 핵심 서비스로 자리매김하고 있으며, 이에 따라 다른 데이터와의 융합 및 개방성 측면에서 현행 AMI 시스템은 큰 한계점을 보이고 있다. 본 논문에서는 유럽을 중심으로 한 DLMS/COSEM 기반의 스마트미터링 표준화 동향과 국내 현황을 분석하고, 향후 데이터 모델링 및 시스템 구성 측면에서 국내 적용방안을 제안한다.

I. 서론

최근 새로운 화두로 부상한 4차 산업혁명은 모든 비즈니스 모델이 빅데이터 기반으로 수렴하는 계기가 되고 있으며, 에너지 산업 역시 데이터에 기반한 소비자 패턴 분석이 핵심 가치로 부상하고 있다. 2014년부터 도입된 수요반응(Demand Response)도 소비자 데이터 분석에 기반한 서비스 모델이며, 이러한 맥락에서 스마트미터 및 AMI는 소비자의 핵심 플랫폼이 되고 있다. 그러나, 과금용 계량기가 가지는 강한 규제와 접근 제한성으로 인해 플랫폼으로서의 스마트미터는 애초 기대에 못미치는 효과를 거두고 있다. 이러한 맥락에서 기존 AMI 체계는 최근 스마트미터링이라는 보다 포괄적인 관점으로 재정의되고 있으며, IoT 기반 플랫폼으로서의 개방성을 확보하기 위해 게이트웨이 플랫폼 기반의 구조가 대안으로 제시되었다. 본 논문에서는 유럽을 중심으로 진행 중인 트렌드를 분석하고, 국내 스마트미터링 플랫폼에 대한 체계를 데이터 구조 및 시스템 구성방안 측면에서 제안하고자 한다.

II. 국제 표준화 동향

스마트미터 및 AMI는 10여년 이상 글로벌 핫이슈가 되어 왔으며, 지역별로 프로토콜 및 통신 방식에 차이가 있으나, 사물인터넷과의 접목이 증가하며 상호운용성 확보를 위한 표준화 작업이 진행되고 있다. CEN/CENELEC, ETSI의 유럽 3개 기관은 이를 위한 상호협약을 맺고 스마트미터링 표준화를 위한 협력을 진행중이며, 국제표준화기구인 ISO, IEC, ITU-T 등과의 협력도 강화하고 있다. 이들 기관들로부터 산출된 결과물은 유럽 수준에서 바로 영향을 발휘하기 때문에, 국제표준화기구에서도 우선순위로 고려된다. 또한, ISO, IEC, 3GPP와의 협력은 유럽표준화 결과물을 국제화하는 주요 지렛대가 되고 있다. ISO/IEC JTC1과 3GPP는 국제표준화기구이고, NIST(미국), KATS(한국),

JISC(일본), DKE(독일)는 국가 수준의 기구이다.

2.1 유럽의 스마트미터링 표준화 동향

유럽에서는 상기 3개 기관에 의해 모든 제품과 서비스에 대한 표준안이 개발되고 승인되고 있다: CEN(European Committee for Standardization: 유럽표준화위원회), CENELEC(European Committee for Electrotechnical Standardization: 유럽전자기술표준위원회), ETSI(European Telecommunications Standards Institute)[1]. 국제 표준화 기구 관점에서 CEN은 ISO, CENELEC은 IEC에 대응하는 것으로 이해할 수 있다. 이중 CEN과 CENELEC 사이에서 밀접한 협력이 진행 중이며, 2010년 CCMC(CEN-CENELEC Management Center)가 설립되었다. CCMC는 CEN과 CENELEC의 모든 활동들을 조율하고 프로모션하는 역할을 하며, 두 개 기관의 총회, 관리위원회, 기술위원회로부터 할당받는 모든 업무를 처리하는 책임을 지고 있다. 행정명령(M/441)을 받은 CEN, CENELEC, ETSI가 유럽 내 전기, 가스, 열, 물 분야에서 사용되는 스마트미터 기능과 통신 인터페이스에 대한 표준안을 마련하고 있으며, 이러한 맥락에서의 표준안은 자발적인 기술스펙과 일반적 기술규칙으로 구성되어 있다. M/441 목적은 스마트미터링시스템의 보급을 촉진하고, 상호운영성을 확보하고, 소비자를 보호하며, 전력시스템 신뢰도를 개선하는 것이다. 무엇보다 스마트미터링의 다음 6가지 측면이 고려되며, 지배적 표준안으로 검수 중에 있다.

- 측정과 측정치의 전송
- 계량기와 시장참여자 간 양방향 통신(빌링, 에너지 관련 서비스)
- 다양한 요금제 모델(TOU/RTP 기반의 시변동요금)과 결제 시스템을 갖춘 계량기
- 원격으로 계량기 작동정지 및 전기공급의 시작/종료 기능
- 원격 부하관리 어플리케이션을 지원하기 위한 통신 기능
- 계량데이터를 실시간으로 디스플레이 하기 위한 댁내 디스플레이와 인터페이스

2.2 DLMS/COSEM 기반의 표준화

DLMS는 CEN-CENELEC 중심의 지배적 유럽 표준이 되었고, 이후 IEC와 협력 하에 IEC 62056 표준으로 채택되었으며, 현재 전 세계적으로 AMI 데이터 구조 및 통신 관련 지배적 표준으로 자리잡고 있다. DLMS/COSEM 표준은 스마트미터 표준 기기간 통신에 사용되는 메시지(기기간 통신 언어) 규격으로서, 에너지미터와 관련한 AMR/AMI 메시지 규격과 다양한 기기들에 대한 개념적(논리적) 모델을 정의하고 있다[3].

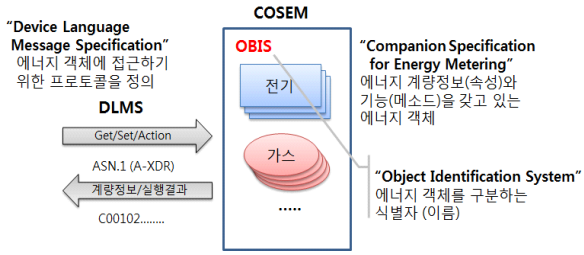


그림 1. DLMS/COSEM 표준 용어

이러한 배경에서 DLMS/COSEM은 차세대 스마트미터 통신 프로토콜로 현재 국내에서도 유력한 표준안으로 주목받고 있으며, 한전 전력량계에 채택되었고, 향후 민수용 전력량계(사례: 한전 요금을 할당을 위한 세대별 설치용)에도 적용이 고려되고 있다. 또한, 가스, 수도 등 기타 유틸리티 서비스에도 확대·적용될 것이다. DLMS/COSEM은 전기, 수도, 가스 등과 같이 사용량을 측정하는 계기와 통신하기 위해 만들어진 제어시스템 프로토콜로서 일반적인 정보통신 네트워크와 달리 시스템 및 통신 하드웨어 사양은 저사양이기 때문에 좀 더 간단한 프레임워크로 효율적인 통신을 위해 만들어진 프로토콜로 이해할 수 있다.

III. 데이터 구조 및 보안 이슈

3.1 COSEM OBIS코드 기반의 데이터 객체화

DLMS는 전력선통신(PLC), RF-Mesh 등 여러 통신매체를 이용하여 COSEM에서 정의한 에너지 객체에 대한 접근(읽기/쓰기) 및 기능에 대한 실행 어플리케이션 명령어를 정의하고 있다. 이는 우리가 사용하는 언어와 흡사하며, 인

터넷, 메신저, 전화기 등 다양한 통신매체를 사용하더라도 정보의 상호교환을 위해 사용되는 언어는 동일한 것으로 이해될 수 있겠다. DLMS를 이용하여 객체를 호출할 때 OBIS (Object Identification System)는 이들 객체를 구분하기 위한 식별자로 이용된다.

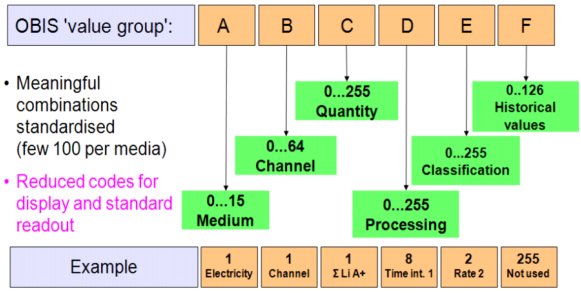


그림 2. OBIS코드 기반의 데이터 식별

OBIS 코드는 계층적 구조로 A부터 F까지 6개 그룹에 대해 각기 의미있는 값들의 조합을 설정한 것으로, 여기에는 클래스 아이디(class_id)와 버전 아이디(version_id)를 포함하여, 모든 COSEM 객체 인스턴스가 설정된다(그림 2 참조[2]). 이러한 체계는 데이터를 소비자에게 디스플레이할 경우에도 사용된다. DLMS를 통해 표준화된 객체, OBIS(Object Identification System) 데이터 타입을 적용하게 되면 하나의 데이터 구조체 내에서 다양한 데이터를 정의 가능하게 된다. 이는 서로 다른 언어를 가지고 있어 소통이 불가능하던 계량기가 같은 언어를 구사하게 되면서 원활하게 통신이 가능해졌다. OBIS는 DLMS/COSEM 표준을 준수하는 미터링 기기에서 모든 데이터에 대한 식별자를 정의한 것이다. 이렇게 OBIS코드 기반으로 객체화된 데이터는 상위에서 논리적 기기(Logical Device)로 매핑되어 하나의 물리적 기기가 다수의 논리적 기기로 접근제어될 수 있도록 활용된다.

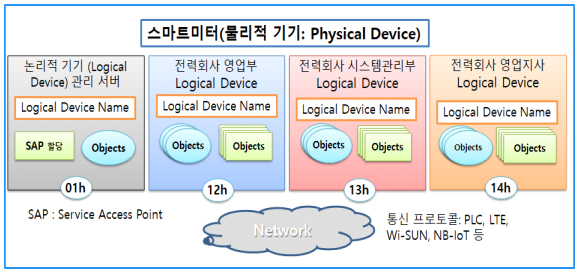


그림 3. 물리적/논리적 기기 관리 개념도

3.2 보안 이슈 및 접근 제어

다양한 통신방식 및 서비스모델에 대한 상호운용성을 확보하고, 다양한 관리자 및 사업자에 대한 차등화된 접근 제어를 구현하기 위해 다음 논리적 기기 기반의 차등화된 접근 제어를 적용할 수 잇다. 이를 기반으로 개인정보보호 및 전기, 가스, 수도, 난방 등 에너지의 통합모델과 ESCO(에너지절약사업자), D/R(수용반응) 등 에너지에 대한 제3자 서비스사업자에 효율적인 정보교환을 위한 방법이 구현 가능하다.

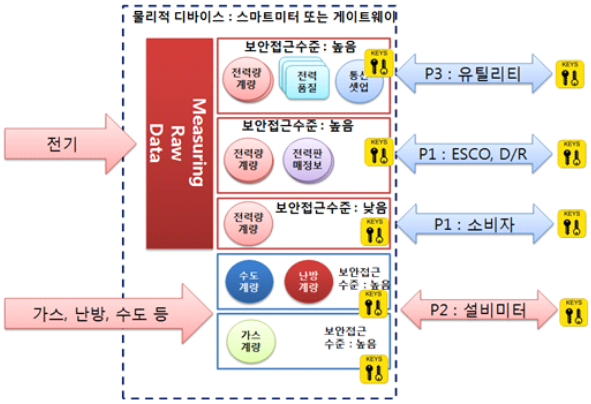


그림 4. 사용자별 차등화된 접근제어

정보보호 측면에서는 ECDSA 인증서기반 상호인증과정과 키교환과정(ECDH)을 통해 생성되는 보안 세션키 또는 사전에 정의된 보안키(AES128 또는 ARIA128)를 사업자별로 고유의 값과 보안메커니즘을 적용하여 상호인증과 통신구간에 대한 보안을 제공할 수 있으며, 이를 통해 계량정보전달에 대한 기밀성 및 무결성을 달성한다. 암호화, 디지털 서명 등이 적용된 COSEM 패킷의 헤더 부분에 다시 보안 태그가 적용되어, 시큐어 DLMS 프로토콜이 완성된다.

IV. 국내현황 분석 및 적용체계 제안

국내 한전 과금용 스마트미터는 DLMS/COSEM 규격 기반으로 스마트미터가 제조 및 보급되고 있으며, 아파트를 중심으로 한 민수용 전력량계에 있어서도 해당 규격과 호환되는 형태(해당 표준 내 최소 필요요건을 충족하는 방식)로 전환이 진행되고 있다. 민수용 전력량계 대다수는 RS-485 인터페이스 기반의 비동기식 통

신을 이용한 검침방식을 지원하며, 메시지 구조와 검침 가능한 데이터는 제조사별로 상이하다. 약 30byte 내외의 메시지 길이를 가지며, 그 구성은 시작과 끝을 나타내는 필드, 계기를 식별하기 위한 필드, 제어를 위한 필드, 검침하는 항목 값에 관한 데이터 필드, 메시지 에러 검출을 위한 체크 필드 등이다.

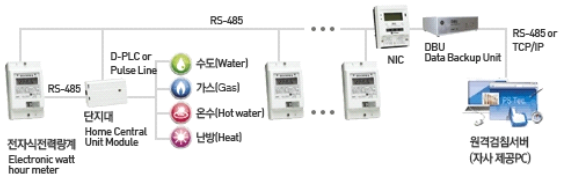


그림 5. 국내 민수용 전력량계 시스템 구성도

다음은 국내 민수용 전력량계 2개업체(A사, B사)의 데이터 구조체 사례이며, 데이터 필드의 순서만이 약간 상이할 뿐 전반적으로 유사한 형태가 사용되고 있다.

<A사>

시작	주소	시작	제어	길이	데이터 종류	데이터	에러 체크	종료
1 byte	6 byte	1 byte	1 byte	가변	2 byte	가변	1 byte	1 byte

<B사>

시작	제어	기기 ID	데이터	상태	에러 체크	종료
1byte	1byte	4byte	20byte	1byte	1byte	1byte

이를 기반으로 DLMS/COSEM 표준에서 정의된 최소한의 기본 데이터 필드가 표 1과 같이 정의될 수 있으며[4], 시스템 구성 차원에서는 전기, 가스, 열 등 다양한 데이터에 대한 융합과 접근성이 용이하게 이루어질 수 있도록 게이트웨이 기반의 통합 플랫폼 구성 방식(그림 6)을 제안하였다[5].

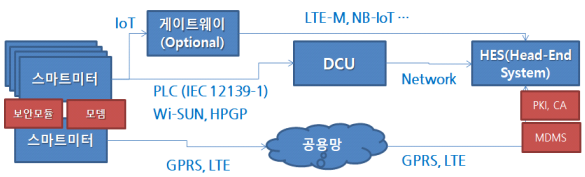


그림 6. 게이트웨이 기반의 스마트미터링 구조

표 1. 전력량계 계량 데이터 OBIS 코드

	항 목	class _id	OBIS_CODE A	B	C	D	E	F	속성	Asso.1 권한
1	현재/직전 수요시한 유효전력	5	1	1	1	4	0	255	1	R
									2	R
									3	R
									4	R
									5	R
									6	R
									7	R
									8	R/W
									9	R/W
2	최대수요전력	4	1	1	1	6	0	255	1	R
									2	R
									3	R
									4	R
									5	R
3	순방향 유효전력량	3	1	1	1	8	0	255	1	R
									2	R
									3	R
4	역방향 유효전력량	3	1	1	2	8	0	255	1	R
									2	R
									3	R
5	순방향 무효전력량 (지상)	3	1	1	5	8	0	255	1	R
									2	R
									3	R
6	순방향 평균역률	3	1	1	13	9	0	255	1	R
									2	R
									3	R

게이트웨이 기반의 접근 방식은 기존 스마트미터 방식의 약점이었던 낮은 시스템 사양과 보안 장벽을 해소하기 위해 대안으로 제시되어 독일, 네덜란드 등에서 시범사업이 진행 중이며, 향후 우리나라의 IoT 기반 스마트미터링 모델로도 도입이 검토되고 있다.

[참고문헌]

[1] https://www.cencenelec.eu/Pages/default.aspx

[2] IEC Technical Committe 13, “Equipment for Electrical Energy Measurement, Tariff and Load Control”

[3] DLMS User Association, “DLMS/COSEM-Architecture and Protocol(6th Edition)”, 2007

[4] 젤릭스주식회사, 스마트그리드 AMI 분야 민수용 전력량계 연동규격 관련 연구, 한국스마트그리드사업단, 2017.07

[5] P1 Companion Standard: Dutch Smart Meter Requirements(ver.4.2.2), March 14th, 2014

중·소형 공공기관 정보보안 관리방법 체계화 연구

강창식, 권헌영
고려대학교 정보보호대학원

A Study on Information Security Management System for Small and Medium-sized Public Organizations

Kang, Chang Sik / Kwon, Hun Yeong
Graduate School of Information Security, Korea University

요 약

본 연구는 정보보안 업무관리에 필요한 절차와 프레임워크를 조사해보고, 이를 바탕으로 우리나라 중·소형 공공기관이 국가 사이버안전관리체계 및 개인정보보호법에 따른 법·제도를 준수하면서도 정보보안 취약점 등에 효과적으로 대응함으로써 정보보안 관리목표를 달성하는 방안을 논해보고자 하였다.

Ⅰ. 서론

공공기관은 정부의 투자·출자 또는 재정지원으로 공공의 이익에 관한 업무를 수행하는데, 2017년 9월 현재 총 수는 330개에 이르고, 공기업·준정부기관·기타공공기관으로 구분된다. 이중 '기타공공기관'은 207개로서 경영 자율성 보장 필요에 따라 포함된 국립병원 등을 제외하면 자산·수입·인력이 상대적으로 적은 중·소규모 기관들이 대부분이다.

중·소규모 공공기관의 경우 정보 관련 안전관리에 완벽하게 대응하기에는 인력의 수와 전문성, 예산과 제도 등의 구비가 부족한 상태로서 정부가 실시하는 평가·점검·검토·검증 등 제도적 요구에 부응하기에도 벅찬 실정이다.

(단위 : 개)

구 분	공기업	준정부기관	기타공공기관
기관 수	35	88	207
정보보안 규정	34 (97.1%)	81 (92.0%)	126 (60.9%)
개인정보 규정	31 (88.6%)	69 (78.4%)	130 (62.8%)

<표1> 공공기관 정보보안 규정 보유 여부

본 연구는 중·소형 공공기관이 정보보안 관리체계를 원점에서부터 구축한다는 가정 하에 (1) 정부의 정보보안 거버넌스에 부합하는 관리체계 구축을 위해 필요한 기본단계를 살펴보고 (2) 기관 자체의 노력이 필요한 사항과 아웃소싱이 필요한 사항을 분석한 후 (3) 중·소형 공공기관 정보보안 체계화에 대한 정부와 민간의 지원 및 보완 필요사항을 제언하고자 한다.

Ⅱ. 정보보안 관리체계 설정단계

데얀 코스틱(Dejan Kosutic)은 조직의 정보보안 관리체계 설정에 필요한 기본적인 절차를 9단계로 구분하였다. 이를 중·소형 공공기관에 적용하는 방안은 <표2>와 같다.

구 분	설정절차	중소형 공공기관 적용(안)
1단계	법·제도 등 요구사항 파악	· 국가사이버안전관리규정 및 지침 · 개인정보보호법 및 지침 등
2단계	최고경영진의 지원 확보	· 정보보안 전략수립 조직 구성 등 기관장 관심과 지원 확보
3단계	정보보안 목표설정	· 정보보안 관리체계 강화로 기관 설립목적 달성 기여
4단계	정보보안 프레임워크 선택	· 국가사이버안전매뉴얼, G-ISMS, PIMS, 정보보호준비도평가 등
5단계	구현전략 수립	· 자체 정보보안전략수립 TF 구성 또는 아웃소싱을 통한 구현 선택
6단계	위험평가 및 관리목표 선정	· 자산분석, 위험평가, DoA 선정
7단계	정보보안 조치 실행	· 관리·물리·기술적 보완조치 시행
8단계	훈련 및 교육	· 업무책임자 및 담당자 심화교육 · 주기적 임직원 인식향상 교육
9단계	보안점검 및 피드백	· 모니터링, 내부 보안점검, 대외 점검·평가 대응, 개선사항 반영 등

<표2> 정보보안 관리체계 설정 절차

이중 본 연구에서는 중소형 공공기관이 준수해야 할 법제도 요구사항을 파악하는 1단계, 법제도 준거성을 포함하는 정보보안 전략을 수립하는 5단계, 정보보안 환경변화 및 개선사항을 반영하는 9단계를 중점으로 살펴보았다.

Ⅲ. 중소형 공공기관 정보보안 관리체계

3.1. 법·제도 요구사항

우리나라 정보보안 최상위 거버넌스는 '국가사이버 안전관리 체계'이다. 이를 위해 정부는 '국가사이버안전관리규정'에 의거, 행정기관 및 공공기관이 "예방활동-사이버공격 탐지·차단-사이버침해사고 조사·복구-사이버위기 대응" 단계별 프로세스에 대응토록하고 있다.

구 분	관련 법·제도	온라인상 공개여부
정보보안	국가 사이버안전관리규정	공개
	국가 사이버안전매뉴얼	비공개
	국가 사이버위기대응매뉴얼	비공개
	국가 정보보안기본지침	비공개
	부처별 정보보안지침	부처별로 다름
개인정보	개인정보보호법	공개
	개인정보보호지침	공개
위험관리	전자정부법	공개
	정보시스템구축운영지침	공개

<표3> 중·소형 공공기관 정보보안 관련 주요 법제도

정보보안지침, 개인정보보호지침, 정보시스템구축운영지침은 사이버 안전관리 체계를 바탕으로 각급 공공기관의 부문별 정보보안 법적 준거요소를 상세히 담고 있다. 다만, 중소형 공공기관 내부규정에 그대로 적용하기에는 실정에 부합하지 않는 사항이 다수 존재한다.

구 분	정보보안 지침	개인정보보호 지침	정보시스템 구축운영지침
전체항목	128개조	63개조	60개조
필수항목	63개조	60개조	-
선택사항	비밀관리, 보안관제, 기반시설 등	타 법과의 관계, 보칙 등	정보보안지침 조문 일부의 세부절차임

<표4> 정보보호 관련 지침의 중소형 공공기관 적용 여부

중소형 공공기관의 정보보안 관리실태를 점검하는 체계는 아래 <표5>와 같다. 점검대상 항목이 많고, 항목별로 증빙이 필요한 사항이 대부분이므로 상당한 주의와 관심이 요구된다.

구 분	주관기관	점검항목	중점 점검사항
정보보안 실태점검	주무부처	5개 분야 20개 지표 93개 항목	사이버안보활동, 단말기보안, 서버·네트워크보안, 용역업체보안, 전산장비보안 등
인터넷·업무망 보안진단	사이버안전센터	7개 분야 30개 지표 259개 항목	보안정책, 보안운영기술, 정보보호설비, 네트워크설비보안, 원도우 서버관리, DB관리
개인정보 보호 관리수준 진단평가	안전행정부	3개 분야 12개 지표 24개 항목	개인정보보호관리 프로세스, 보호정책, 보호조직, 자산분류, 정보분류 및 취급, 교육, 인적보안, 침해사고관리, 기술적보호조치, 물리적보호조치, 개인정보라이프사이클관리

<표5> 중소형 공공기관 정보보안 관련 점검·평가

3.2. 정보보안 관리체계 구현 전략

공공기관 경영정보시스템(www.alio.go.kr)에 공개된 중·소형 공공기관의 조직 및 업무분장을 살펴보면 상당수의 기관이 일반보안, 정보보안, 개인정보보호, 정보화 업무간의 구분 없이 1~2명의 실무자에게 담당토록 하고 있는 실정이다. 이러한 상태에서 정부의 법·제도 요구사항을 온전히 반영한 정보보안 관리체계를 만들고 준수하기를 기대하기는 곤란하다.

중·소형 공공기관의 정보보안 관리체계의 구현 전략은 적정한 수와 전문성을 보유한 인력의 증원이 필수적인 실정이다. 다만, 주어진 여건을 수용하는 단기적 방안으로 (1) 유사한 법제도 요구사항의 통합관리, (2) 전 임직원의 참여와 분임, (3) 정보보안 표준 인증제도의 활용을 들 수 있다.

첫째, <표4>에서 살펴본 정보보안 관련 3개 지침과 더불어 최근 정부가 권장하고 있는 클라우드 도입과 관련한 지침 등도 유사한 법제도로 볼 수 있으므로 이를 하나의 규정으로 통합하는 것이 필요하다.

둘째, 정부의 '정보보안기본지침' 제4조(책무)는 "각급기관의 장은..... 정보통신망을 보호하기 위한 보안대책을 마련하여야 하며 정보보안에 대한 책임을 진다."고 규정하고 있다. 기관장의 적극적 관심과 참여를 위해 일본 경제산업성은 '사이버보안 경영 가이드라인'(2015)을, 우리나라 금융보안원은 '경영자를 위한 정보보안 경영가이드'(2016)를 개발하였다. 아울러, 전임 담당자 외에 각 부서별로 분임 정보보호담당자를 선임하여 협업토록 할 필요가 있다. 기관장을 비롯한 전 임직원의 참여와 분임 확대는 정보보안 관리체계가 기관의 주요 과업으로 인식되는데 기여할 것이다.



<그림 1> 임직원의 참여와 분임 확대
- 김상균(2016)의 논문 36쪽 그림 일부 변용

셋째, 정보보안 책임자 또는 담당자의 업무 숙련도가 부족한 상태일 경우 정보보안 표준인 증제도를 활용할 수 있다. 다만, 이들 인증제도의 경우 정부의 정보보안지침 및 점검·평가 항목과 적지 않은 차이가 있으며, 2016년 감사원은 ISMS와 PIMS간의 유사 중복성 문제를 지적한 상태이다.

구 분	정보보호 준비도평가	ISMS	PIMS
분 야	정보보호 + 개인정보보호	정보보호	개인정보보호
평가기준	정보보호 리더십, 관리·물리·기술적 보호활동 등 30개 항목	정보보호정책, 관리체계, 위험관리 등 104개 항목	개인정보보호 관련 관리·물리·기술조치 등 124개 항목
소요기간	약 1개월	약 6개월	약 6개월

<표6> 전문기관 인증제도 비교

3.3. 주기적인 보안점검 및 피드백

대부분의 중·소형 공공기관은 서버 등 네트워크 시스템에 대한 관리를 외주 용역업체에 위탁하고 있으며, 보안관제는 주무부처의 사이버안전센터에 의지하고 있는 실정이다. 정보보안 및 개인정보보호와 관련된 침해위협이 갈수록 지능화되고 다양화하는 등 변화하고 있으므로 지속적인 보안점검과 피드백이 필요하다. 특히, 자연재해, 악의적인 공격, 내부자 및 용역업체에 의한 보안위규, 오작동 및 의도하지 않은 인적실수 등 정보보안 및 개인정보보호 관리에 반하는 위해요인이 발생할 경우에 대한 대응이 필요하다.

<표5>에 기술한 정보보안 점검·평가는 연간 각 1회씩 실시되며, 외부평가에 대한 부담으로 인해 내부의 정보보안 실태가 축소 또는 왜곡될 가능성이 있다. 따라서, 체크리스트 항목을 기반으로 보다 단주기적인 점검을 통해 자율적으로 문제점을 발굴하고 개선하는 것이 바람직하다.

3.4. 정부와 민간의 지원제도 개선

공공기관의 경영 자율성에 기반하여 정보보안 관리체계의 성숙도를 촉진하기 위해서는 부처 주관으로 실시되는 '공공기관 경영평가' 지표의 개선·반영할 필요가 있다. 정보보안 및 개인정보보호는 '경영관리' 범주에 속하므로 해당 범주 중 '업무효율' 지표의 배점을 감하고, '정보보호' 지표를 신설하는 것이 그 방안이다.

범 주	평가지표	기 존	개 선
리더십및 책임경영	1. 기관장 리더십	6	6
	2. 기관의 경영혁신	5	5
	3. 국민평가	8	8
경영 관리	1. 업무효율	7	5
	2. 재무예산관리 및 성과	5	5
	3. 보수 및 복리후생관리	14	14
	4. 정보보호	-	2
주요 사업	대표사업 3개 내외	55	55
계		100	100

<표7> 중소형 공공기관 경영평가 지표개선(안)

또한, <표6>에서 기술한 정보보호준비도평가를 중·소형 공공기관의 정보보안 관리체계와 보다 더 긴밀하게 접목시키기 위해, 공공기관의 정보보안 규범을 바탕으로 평가지표를 개선하거나, 중·소형 공공기관에 특화된 지표를 별도로 개발하여 운용할 필요가 있다.

Ⅳ. 결론

4.1. 연구 요약

본 연구는 임직원의 참여와 분임, 복잡한 제도적 요구사항의 통합관리, 민간 전문역량의 활용을 통해 정부의 법제도 요구사항에 대한 준수와 더불어 정보보안 위협요인에 효과적으로 대응하기 위한 관리체계를 구성하고 운용하는 방안에 대해 논하였다. 이를 통해 정보보안 및 개인정보보호 관련 규정이 없는 80여 개의 기관은 물론 혼재된 관리체계를 유지하고 있는 중·소형 공공기관들의 정보보안 관리가 보다 체계화되기를 기대한다.

4.2. 한계 및 향후 연구계획

본 연구에서 검토되지 않은 정보보안 관리체계 설정단계에 대하여 향후 추가 검토를 진행하고, 국내·외의 정보보안 관리체계에 대해 보다 상세히 조사하며, 관리체계 점검을 위한 종합체크리스트를 구성해보고자 한다. 나아가 클라우드 서비스 및 IoT 활용 확대, AI 활용 증가에 따른 대응 등 정보보안의 업무 범위와 양은 증가할 수밖에 없는 실정이므로 "클라우드 보안 가이드라인", "IoT 보안 가이드라인" 등을 포함한 후속연구를 수행함으로써 지속적으로 변화하는 정보보안 업무환경에 적용할 수 있는 기준이 도출되도록 보완해 나가고자 한다.

[참고문헌]

[1] 산업통상자원부(2016), "정보보안 실태점검 체크리스트"

[2] 산업통상자원부(2016), "인터넷망・업무망 보안진단 체크리스트"

[3] 미래창조과학부, 한국인터넷진흥원(2017), "ISMS 인증제도 안내서"

[4] 미래창조과학부, 한국인터넷진흥원(2014), "G-ISMS 인증제도 안내서"

[5] 한국인터넷진흥원(2016), "PIMS 인증기준 세부점검항목"

[6] 미래창조과학부, 한국인터넷진흥원(2016), "정보보호준비도평가, 평가기준 및 방법"

[7] 한국인터넷진흥원(2015), "정보보호 실태조사(기업부문) 보고서"

[8] 이순형 외(2016), "미래사회와 보안문화", 인포더북스.

[9] 기획재정부(2017), "공공기관의 운영에 관한 법률" 제4조

[10] 공공기관 경영정보시스템(ALIO)

[11] 한국인터넷진흥원(2016) "국가정보보호백서"

[12] 한국인터넷진흥원(2016), "정보보호 관리체계 인증제도 현황자료"

[13] 한국정보화진흥원(2017), "국가정보화백서", '공공부문의 클라우드 보안정책 수립'

[14] 선한길(1998), 국내주요그룹의 정보보안관리 체계에 관한 사례 연구

[15] 김자희(2015), "공공기관 정보보호성숙도 모델에 관한 연구"

[16] 김재경 등(2013), "공공기관의 정보보안 관리모델 연구"

[17] 김지숙 외(2010), "민간기업과 공공기관의 정보보호 관리체계 차이비교"

[18] 김상균(2016), "공공기관 정보보안수준 향상을 위한 분임담당자 중심 업무분장 개선에 관한 연구"

[19] 김지수(2012), "조직내 정보보호최고책임자의 역할인식이 정보보호성과에 미치는 영향에 관한 연구"

[20] Dejan Kosutic(2012), "9 Steps to Cybersecurity"

[21] Lance Hayden(2014), "Oh the Humanity! Building Reliable Cultural Firewalls", RSA Conference.

[22] Corporate Governance Task Force(2004), "Information Security Governance"

NERC CIP 규정 분석에 근거한 전력시스템 신뢰도 관점에서 CPS 영향도 평가 방법에 관한 연구

강동주

고려대학교 정보보호대학원

A Proposal of CPS Impact Assessment Method for Power System Reliability based on NERC CIP Code Analysis

Dong-Joo Kang

Division of information security, Korea University.

요 약

전력망 보안은 기존의 통신 분야에 한정된 보안 이슈와는 다른 특성이 있고 이를 제대로 이해하여 보안대책을 수립할 필요가 있다. 전력망은 전력계통(power system)과 정보인프라(information infrastructure)로 구성되어 있는데, 여기서 전력계통이란 발전, 송전, 배전을 포괄하는 물리적 에너지 생산 및 유통 서비스이고, 정보인프라는 이를 모니터링하고 제어하는 기능을 수행하는 것으로, SCADA-EMS를 의미한다. 사이버위협이 증가함에 따라 전력망의 정보인프라도 이에 노출되고 있다. 이러한 맥락에서 본 논문은 북미 신뢰도 기구인 NERC CIP 모델을 참조하여 CPS 기반의 전력망 위험 평가 프레임워크를 제안한다.

I. 서론

정보인프라와 전력인프라는 상호 연계되어 있으므로, 통신망에 유입된 사이버 위협이 어떻게 전력망에 영향을 미치는 정량적으로 분석할 수 있는 모델을 수립하는 것이 CPS 기반 연구의 주요 의의라고 할 수 있다. 스마트그리드에서의 보안 문제는 이러한 2가지 사이의 상호작용을 이해하는 것이 가장 중요하다고 할 수 있다.

발전설비를 포함한 전력설비의 경우 상정사고시 신속한 대응이 필요한 관계로 비교적 간단한 패스워드를 걸거나 기기 접근성이 용이하도록 하는 경우가 많다. 그렇지 않을 경우 신속한 대응이 불가능한 경우가 생기고 그로 인한 경제적 및 사회적 손실이 막대하기 때문이다. 따라서, 사이버위협이 전력시스템에 미치는 영향을 사전에 분석하고 이를 기반으로 사전적 위험예방 조치와 사후적 위험최소화 정책을 적용할 필요가 있다. 북미신뢰도위원회(NERC)는 전력시스템을 포함한 중요 인프라를 CI(Critical Infrastructure)로 규정하고, 이를 보호하기 위한 체계를 CIP(Critical Infrastructure Protection) 규정으로 정의하였다. 본 논문에서는 NERC CIP 규정 분석을 근거로 전력시스템의 위험을 평가하기 위한 방법론적 프레임워크를 제안한다.

II. NERC CIP

신뢰도 관점에서의 사이버 보안 이슈는 전력망 운영에 직접적으로 영향을 미치는 측면에서 분석하여야 하고, NERC 규정에서도 송전급 이상의 계통운영에 영향을 미치는 건으로 한정하고 있다. 이러한 관점에서 CPS를 적용한다면, SCADA 시스템에서의 주요 3계층인 제어센터(Control Center), 주요 중앙급전 발전기 및 송전선로와 직접적으로 연계된 통신 네트워크, 배

전급 RTU 중 송전급 운영에 영향을 미치는 설비로 구분할 수 있다. 북미전기신뢰도위원회(NERC)는 사이버 보안에 대한 법률을 제정하였다. 사이버 보안을 위해 구현해야 하는 최소한의 가이드라인을 정하고 2004년 표준으로 확정하고 매년 초 갱신하고 있으며, 이후 전력설비와 전력시스템, 관련기기 등에 적용하고 있다.

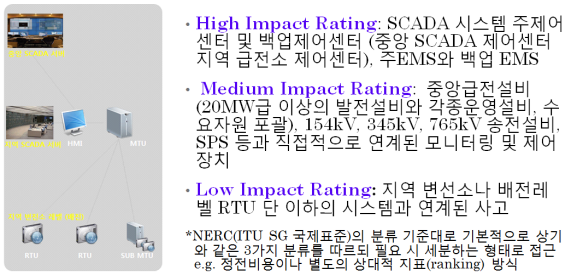


그림 1. 전력계통에 대한 영향도 평가 등급 (3단계로 구분)

SCADA 시스템에 대한 보안 위협은 상위의 전력시장 운영 및 하위의 전력계통 운영 모두에 영향을 줄 수 있으며, 반대로 전력시장과 전력계통 운영의 이상이 잘못된 정보와 데이터 부하를 전달하여 정보망에 영향을 줄 수 있다. 이러한 상호작용 평가를 위해서는 우선적으로 중요자산(critical asset)과 중요정보자산(critical information asset)에 대한 정의가 필요하고 이를 다시 해당자산의 책임당사자에게 연계하는 과정이 필요하다. 이러한 구성과 절차는 NERC 신뢰도 규정상의 구성상에서 기술하고 있는 맥락과도 일치한다.

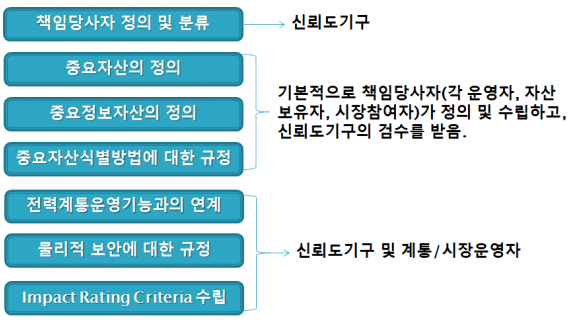


그림 2. NERC 전력시스템 신뢰도 CIP 구정 구성

중요자산 및 중요정보자산은 다시 계통운영에 있어서의 주요기능과 연계되어야 하고, 이는 다시 계통운영보조서비스(ancillary service)와 같은 주요 운영기능과 연계되어야 한다. 중요자산은 전력시스템의 물리적 설비, 중요정보자산은 해당 물리적 설비를 모니터링하고 제어하기 위한 통신인프라 및 설비로 볼 수 있다. 해당 중요자산과 중요정보자산은 다시 전력계통의 특정 운영기능과 연계되며, 이러한 기능은 전력계통의 신뢰도를 유지하는데 필수적이다.

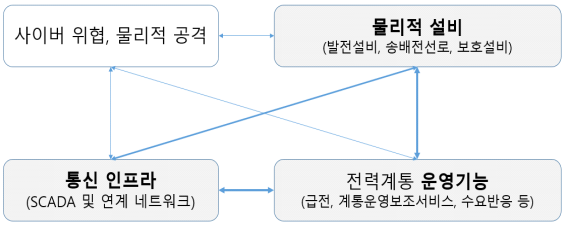


그림 3. 전력시스템의 자산-기능 간 연계개념

중요정보자산은 해당 자산이 라우팅이 되는지 여부와, 전화망으로 접속이 가능한지 여부로 구분된다. 중요자산 및 정보자산이 정해지면 이에 대한 관리와 책임을 지게 될 책임당사자를 정의하여야 하며, NERC의 규정을 기준으로 국내책임당사자와 같은 경우는 다음과 같이 대응연결이 가능하다. 단, 중요자산 및 중요정보자산 중 원자력과 직접적인 전력설비에서 벗어나는 경우는 다른 규제기구의 관리를 받는 것으로 규정하고 있기 때문에 원자력 설비나 SCADA 이외의 통신망이나 설비에 대해서는 본 연구범위에서 제외된다.

III. CPS 연구와 모델링 방법 도출

현재 대부분의 전력망 및 스마트그리드 보안 연구는 통신계층에서의 아키텍처에 대한 내용들이지만, CPS라는 키워드와 더불어 통신-전력계층 간의 상호작용을 모델링하기 위한 연구가 확대되고 있다. 미국에서는 에너지부(DOE) 주도로 전력계통 통신시스템의 보안 기술 연구가 진행되어 왔고, 현재 잠재적 위협 및 대응수단 확보차원에서 다양한연구가 진행되고 있다. 특히, EPRI나 INL(Idaho National Lab), SNL(Sandia National Lab) 등 정부출연연구소 주도로 활발한 연구 활동이 진행 중에 있다. 미 에너지성(DOE)은 국가표준기술국(NIST) 협조 하에 SCADA 시스템과 제어망에 대한 취약점을 점검하기 위해 테스트베드를 제작하여 운영하고 있

으며, 북태평양국립연구소(PNNL)와 EPRI와는 전력분야의 사이버 보안에 대한 연구 과제를 진행하고 있다. 스트베드를 가장 적극적으로 구축하여 왔다.

이러한 CPS 기반의 테스트베드는 통신망과 전력망의 상호작용을 모의하기 위한 것으로, 이종 시스템 간 연계를 특정 사이버 위협이 전력시스템에 미치는 영향을 분석하는 용도로 활용된다. 통신망의 사고가 전력망의 사고로 이어질 수 있으며, 반대로 전력망의 사고가 통신망의 이상을 초래할 수도 있다. 대표적인 사례로는 사이버공격으로 임의의 발전소가 탈락한다고 가정했을 때, 이것이 계통에 미치는 영향을 시뮬레이션 툴로 모의하는 것이다. 대상은 기본적으로 중앙급전 발전기가 된다.



그림 4. 발전소 공격 및 탈락 시뮬레이션

이외 대규모 신재생에너지 발전기의 경우도, 신뢰도에 영향을 줄 수 있는 공격 대상이 된다. 이 경우는 발전기를 탈락시킬 정도가 아니라도, 단순히 측정 데이터를 조작하는 것만으로도 시스템 균형에 영향을 줄 수 있다. 풍력발전은 그 변동성으로 전력계통에 주는 영향도 크고, 중앙에서의 통제성이 약하며, ICT와의 연계성이 크기 때문에 주요 이슈가 될 수 있다.



그림 5. 발전소 출력 데이터 조작

배전계통 수준에서 접속되는 분산전원이나 수요도 임계치를 넘을 경우 신뢰도에 위험을 줄 수 있다고 가정한다. 특히, 분산전원이나 ESS는 ICT 설비와의 연계성이 크므로 그만큼 잠재적 위협이 크다고 할 수 있다.



그림 6. 배전계통 설비에 대한 공격 시나리오

Yilin Mo 등은 기존의 보안 솔루션으로는 새롭고 복잡한 사이버-물리 시스템인 스마트그리드가 당면하게 될 새로운 보안 문제를 해결하기 힘들다고 보고, CPS 개념 기반으로 체계적인 접근법을 제안하고 있다. 우선 사이버-물리 간 상관관계에 기반하여 다음과 같이 공격유형을 분류하였다.

표 1. CPS에서의 공격과 결과 사례

결과 / 공격	사이버(Cyber)	물리(Physical)
사이버 (Cyber)	사적정보의 유출/도청	스턱스넷 (Stuxnet)
물리 (Physical)	미터 우회	물리적 파괴로 인한 불안정

상기와 같은 시나리오에 따른 영향평가 방법론을 정리하면 다음과 같다. CPS 상관관계 모델을 통해, 특정 노드가 계통에 미치는 영향평가 측면(주로 수치상의 전압이나 전력용량)을 고려할 수 있고, 중요노드추적법을 통해 네트워크상의 위치에 따른 중요성을 정량적으로 측정할 수 있다.

NERC의 규정에 따르면 중요정보자산(critical information asset)은 최소한 다음의 1가지 속성을 포함한다고 규정하고 있다. 이와 같은 규정을 고려할 경우 결국 중요정보자산은 외부로부터의 접근이 가능한 자산을 의미한다고 할 수 있다.

① 전자보안주변기기 외부와 통신하기 위한 라우팅 가능한 프로토콜 기반의 정보자산
② 제어센터(control center) 내에서 라우팅 가능한 프로토콜을 사용하는 정보자산
③ 전화망으로 접근이 가능한 정보자산

NERC 규정과 CPS 연구 사례들을 종합하여,

특정 노드에 대한 공격이 이루어질 경우 피해 정도를 상기의 이론적 모델에 근거하여 시뮬레이션 함으로써, 중요자산(critical asset) 및 중요정보자산(critical information asset)의 우선순위를 결정할 수 있다.

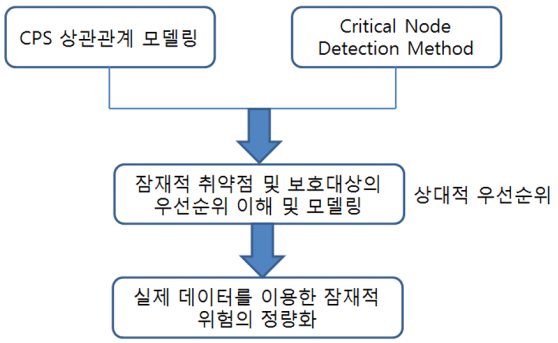


그림 7. 중요노드 선정법

IV. 결론: CPS 영향평가 절차

NERC 규정과 CPS 상호영향 평가에 대한 기존 연구사례들을 바탕으로 CPS 기반의 중요자산 및 중요자산 확인 결과를 절차도로 도식화하면 다음과 같다.

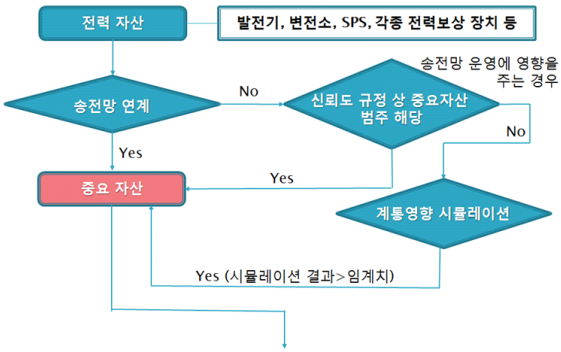


그림 8. 중요자산의 선정과정

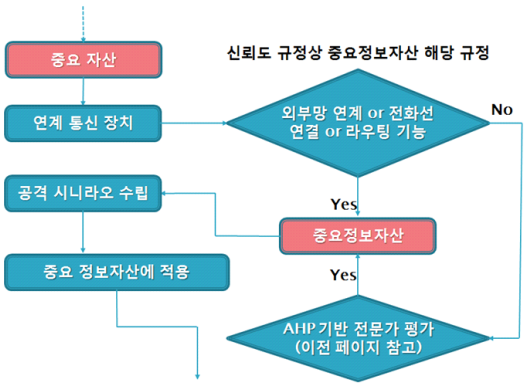


그림 9. 중요정보자산의 선정과정

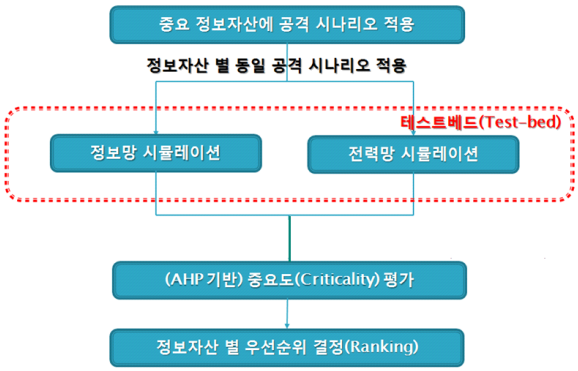


그림 10. 중요정보자산 우선순위 결정

후속 연구를 통해 테스트베드 상에서 통신망과 전력망의 연계 시뮬레이션 모델, AHP 방법론 적용 과정에서의 가중치를 비롯한 정량적 파라미터 도출에 대한 실증 연구가 필요할 것으로 판단된다.

[참고문헌]

[1] Yilin Mo, Tiffany Hyun-Jin Kim, Kenneth Brancik, Dona Dickinson, Heejo Lee, Adrian Perrig, and Bruno Sinopoli, Cyber - Physical Security of a Smart Grid Infrastructure, Page 195-209, 2011

[2] Burris, Peter, and Chris King, “A Few Good Security Metrics”, METAGroup, Inc. audio, 11 Oct. 2000. URL: http://www.metagroup.com/metaview/mv0314/mv0314.html (10 July 2001)

[3] Yi Lu, Coll, Hangzhou, China, Qiang Yang, Wenyuan Xu, and Zhiyun Lin, Cyber security assessment in PMU-based state estimation of smart electric transmission networks, Control and Decision Conference (CCDC), 2015

[4] 강동주외 4명, 전력SCADA시스템의 사이보보안 위험평가를 위한 정량적 방법론에 대한 연구, 한국정보보호학회, Vol.23, No.3, 2017.06

[5] 홍승필, 김영철, 정보보호의 이해(Introduction to Information Security), 길벗

2017 한국정보보호학회 동계학술대회

B3 양자암호

정 익 래 (고려대)

- **텔레그램의 IGE 모드에 대한 양자 안전성 검증**
  김성숙, 이지은, 김광조 (KAIST)
- **Yu와 Oh가 제안한 OAuth기반의 사용자 인증 프레임워크 정형화 검증**
  이재경, 최홍석, 한종찬, 권오제, 유일선 (순천향대학교)
- **Quantum Message Authentication Scheme using Single Qubit Measurement**
  Yeon-Ho Choi (Korea University, KIST), Min-Sung Kang, Sang Wook Han, Sung Moon (KIST)
- **공증된 conjugate coding 기반 양자 서명**
  강민성, 박병권 (KIST), 우민기 (아주대학교), 이민수, 김용수, 조영욱, 이상윤, 한상욱, 문성욱 (KIST)

텔레그램의 IGE 모드에 대한 양자안전성 검증[1)]

김성숙* 이지은* 김광조*,**

*카이스트 전산학부/ **정보보호대학원

On the Quantum Security of IGE mode in Telegram

Sungsook Kim* Jeeun Lee* Kwangjo Kim*,**

*School of Computing/ **Graduate School of Information Security, KAIST

요 약

블록암호는 일정 블록 단위로 평문을 암호화하고 있으며, 다양한 길이의 평문을 암호화를 위하여 운영 모드를 사용하는 데 널리 알려진 보안 메신저인 텔레그램은 IGE(Infinite Garble Extension)이라는 특수한 운영 모드를 사용하고 있다.

양자 컴퓨터에 대한 연구가 활발히 진행됨에 따라 현재 사용되는 블록암호들은 양자 안전성 증명에 대한 필 요성이 대두되고 있다. 특히, 블록암호의 양자 안전성은 사용하는 암호 알고리즘과 운영모드에 따라 안전성이 결정된다. 본 논문에서는 암호알고리즘이 sPRF(Standard-secure PRF)일 때 IGE 운영모드가 양자 안전성을 보장하지 않음을 증명하였다.

I. 서론

일정 블록 단위의 키와 평문 및 복호문을 가지고 있는 블록암호는 매 블록 단위로 평문(암호문)을 암호화(복호화)를 하고 있으며 평문이 블록 크기에 맞지 않을 때, 패딩 데이터를 추가하여 블록 단위로 처리할 수 있도록 다양한 운용 모드(Mode of Operation)가 있다.

현재 컴퓨터에 의한 계산적 안전성을 검증한 다양한 블록 암호들이 실제 많이 사용되고 있지만, Shor의 알고리즘[1]이 발표된 이래 수년 이후 상용화가 예상되는 양자컴퓨터를 사용한 공격에 대비하여 양자 안전성을 검증하는 것이 대단히 중요하게 대두되고 있다.

블록암호의 양자 안전성은 사용하는 블록함수와 운용 모드에 따라 그 안전성이 결정되며 본 논문에서는 텔레그램에서 사용되는 IGE(Infinite Garble Extension) 모드에 대한 양자 안전성에 대해 검증하고자 한다.

텔레그램은 대표적인 보안 메신저로, 최근 스마트폰이 보급되면서 많은 사람들이 다양한 메신저를 사용하면서 개인 정보를 보호할 수 있는 암호화된 메신저에 대한 관심이 높아지고 있으며 이러한 안전성 때문에 텔레그램의 사용자들도 급격히 증가하고 있는 추세다. 텔레그램은 자체 개발된 보안성이 있는 MT프로토(MTProto)라는 프로토콜을 사용하며, 이 프로토콜에서 IGE 모드가 활용된다.

1) 이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원(No.2017-0-00555, 양자 컴퓨터 환경에서 래티스 문제를 이용한 다자간 인증키교환 프로토콜 연구, 2013-0-00396, 생체모방 알고리즘(Bio-Inspired Algorithm)을 활용한 통신 기술 연구)과 2017년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. NRF-2015R1A2A2A01006812)

1.1 논문의 구성

본 논문의 구성으로 장Ⅱ 에서는 양자계산, IGE모드, 안전성증명에 대해 간략히 기술하고, Ⅲ장에서는 sPRF 암호 알고리즘을 IGE 모드로 사용할 때 양자 안전성을 보장하지 않음을 양자 회로를 구현하여 증명한다. 마지막으로 장Ⅳ 에서는 요약하고 추후 연구할 내용에 대해 제시한다.

Ⅱ. 배경지식

2.1 양자 계산

양자 시스템은 Hilbert공간, H의 내적 $\langle\cdot|\cdot\rangle$을 포함하고, 양자 시스템의 상태(state)는 내적이 1인 벡터 $|\psi\rangle$로 표현한다. H의 직교기저가 $B=\{|b_0\rangle,...,|b_{d-1}\rangle\}$일 때 양자 상태 $|\psi\rangle$를 측정하면 $|\langle b_i|\psi\rangle|^2$의 확률로 b_i가 나오며 이때 $|\psi\rangle$가 b_i로 붕괴 (collapse)한다고 한다. 서로 상호작용하지 않는 두 부분계 $|\psi_1\rangle\in H_1, |\psi_2\rangle\in H_2$로 이루어진 전체 계는 분리 가능한 (separable) 양자상태이며, 이는 텐서곱(Tensor Product)을 이용하여 $|\psi_1\rangle\otimes|\psi_2\rangle\in H_1\otimes H_2$로 표현한다.

2.2 Infinite Garble Extension(IGE) 모드

IGE모드는 현재 텔레그램의 MTProto에서 사용되는 모드로 1978년 Campbell[2]에 의해 처음 제안되었으며 [그림1]과 같이 동작한다.

주어진 블록암호 E에서 대칭키 암호알고리즘 Π_{IGE}=(Gen,Enc,Dec)은 다음과 같이 정의한다.

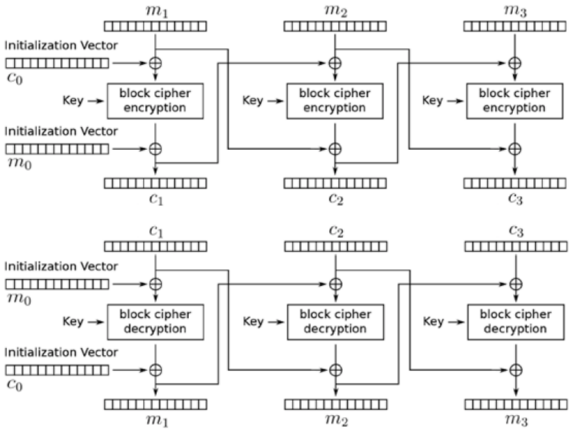
Gen : 블록암호E의 랜덤한 키 k를 생성한다.

Enc : 메시지 $M=m_1m_2\cdots m_n$에 대한 암호문 $C=c_1c_2\cdots c_n$이면

$c_i=Enc(k, m_i\oplus c_{i-1})\oplus m_{i-1}$ 이다.

이 때초 기값인 c_0, m_0은 두 번째 랜덤한 키 k_0를 이용하여 계산$c_0=Enc(k_0, m_0)$하거나임의로 선택할 수 있다.

Dec : 암호문 $C=c_1c_2\cdots c_n$과 k가 주어졌을 때, 복호화하면 $M=m_1m_2\cdots m_n$이 되고,



[그림1] IGE모드의 암호화 및 복호화

$m_i=Dec(k, c_i\oplus m_{i-1})\oplus c_{i-1}$으로 계산한다.

2.3 IND-CPA

대칭키 암호 알고리즘 Π=(Gen,Enc,Dec)는, 아래의 게임에서 다항식 시간의 공격자가 랜덤 추측(guess)보다 무시할만한(negligible) 이점(advantage)을 가질 때 IND-CPA 안전하다고 한다.

키생성 질의자가 랜덤 키 $k\leftarrow$Gen() 와 랜덤 비트 b를 생성한다.

질의 : 공격자 A가 두 개의 메시지 m_0, m_1을 선택하고, 이를 질의자에게 보낸다질의자는 메시지를 암호화 $c^*=Enc_k(m_b)$ 하고, 이를 공격자에게 보낸다.

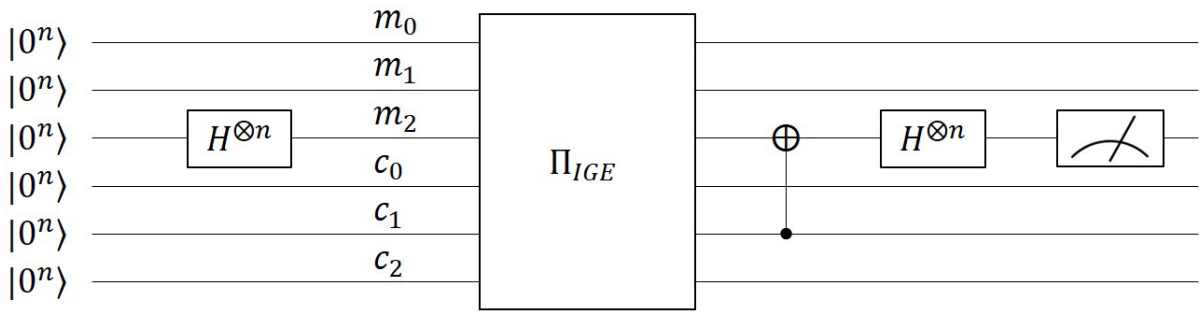
추측 : 공격자가 b'을 생성하고, $b=b'$일 때 공격자가 승리한다.

2.4 IND-qCPA[3]

IND-qCPA는 종래의 IND-CPA와 키생성, 추측 과정은 동일하나, IND-qCPA에서 질의 과정은, 양자 챌린지 질의와 양자 암호화 질의로 구분된다.

챌린지 질의 : 공격자 A가 두 개의 메시지 m_0, m_1를 질의자에게 제시하고질의자는 그 중 임의로 하나를 선택 하여 암호문$c^*=Enc_k(m_b)$을 공격자에게 알려주는 것으로 IND-CPA와 동일하다.

암호화 질의 : 공격자가 암호화 오라클에 질



[그림 2] IGE 모드를 이용하는 블록 암호에 대한 양자 공격 회로

의를 할 때 메시지가 중첩된 상태로 질의를 하며, 중첩된 상태의 암호문을 받는다.

$$\sum_m \psi_{m,c}|m,c\rangle \rightarrow \sum_{m,c} \psi_{m,c}|m,c \oplus Enc_k(m)\rangle$$

2.5 Standard-secure PRF(sPRF)[4]

종래의 질의를 하는 어떤 양자 공격자도 PRF와 truly random function을 잘 구별할 수 없을 때, PRF는 standard-secure PRF이다.

2.6 Quantum-secure PRF(qPRF)[4]

양자 질의를 하는 어떤 양자 공격자도 PRF와 truly random function을 잘 구별할 수 없을 때, PRF는 quantum-secure PRF이다.

III. sPRF를 사용한 IGE모드의 안전성

이 장에서는 블록암호가 sPRF로 증명이 되어도 IGE모드를 사용할 때 양자 공격에 안전(IND-qCPA)하지 않은 경우를 제시한다. 아래의 sPRF를 사용한 블록암호가 Simon 알고리즘[5]을 이용하면 완벽하게 키를 찾아낼 수 있다.

3.1 sPRF인 블록암호[6]

$$BC_k(x) := E_{H(k)_1}(droplastbit(x \oplus (k \| 1) \times lastbit(x))) \\ \| t_{H(k)_2}(x \oplus (k \| 1) \times lastbit(x)) \oplus lastbit(x),$$
$$E : \{0,1\}^{n-1} \times \{0,1\}^{n-1} \rightarrow \{0,1\}^{n-1} standard\ \text{secure}\ PRF$$
$$t : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}\ standard\ \text{secure}\ PRF$$
$$H : \{0,1\}^n \rightarrow \{0,1\}^n \times \{0,1\}^n\ random\ \text{oracle}$$
$$k \xleftarrow{\$} \{0,1\}^{n-1}$$

위 블록암호는 sPRF이지만 qPRF는 만족하지 않으며, $(k\|1)$주기를 가지고 있다. 자세한 설명은 [6]을 참조한다.

3.2 IGE 모드를 사용한 블록암호 공격

Π_{IGE} 가 위의 BC_k를 사용한 암호일 때, IND-qCPA의 공격자는 Π_{IGE}에 양자 질의가 가능하다. 즉, 공격자는 모든 메시지를 중첩된 상태로 BC_k에 질의를 할 수 있다.

먼저 공격자가 메시지와 암호문이 저장될 양자 레지스터 M, C를 준비한다. 메시지를 중첩된 상태로 M에 저장하고, C에는 $|0^{3n}\rangle$으로 초기화 하여 저장한다.

다음으로 공격자는 Π_{IGE}에 암호화 질의를 하고, 상응하는 결과가 C 레지스터에 저장된다. 암호화 질의 과정은 [그림 2]에 표시되어 있으며, 아래와 같은 계산과정을 거친다.

메시지 레지스터가 Hadamard 게이트 통과 후 Π_{IGE}를 적용하면 정규화된 (normalized) 상태는 다음과 같다.

$$|M,C\rangle = \sum_{m_2} |m_0\rangle|0^n\rangle|m_2\rangle|c_0\rangle \\ |BC_k(c_0) \oplus m_0\rangle|BC_k(BC_k(c_0) \oplus m_0 \oplus m_2)\rangle$$

$y := BC_k(c_0) \oplus m_0$로 두면,

$\sum_{m_2} |m_0\rangle|0^n\rangle|m_2\rangle|c_0\rangle|y\rangle|BC_k(y \oplus m_2)\rangle$이고,

CNOT 게이트를 이용하여 c_1을 m_2에 XOR 하면 아래와 같이 식(1)을 얻을 수 있다.

$$\sum_{m_2} |m_0\rangle|0^n\rangle|m_2 \oplus y\rangle|c_0\rangle|y\rangle|BC_k(y \oplus m_2)\rangle \quad 식(1)$$

BC_k가 $(k\|1)$주기를 가지므로, 식(1)에 $BC_k(y \oplus m_2) = BC_k(y \oplus m_2 \oplus (k\|1))$을 적용하여 정리하면 식(2)를 얻는다.

$$\sum_{m_2}|m_0\rangle|0^n\rangle|m_2\oplus y\rangle|c_0\rangle|y\rangle|BC_k(y\oplus m_2\oplus(k\|1))\rangle$$

$$=\sum_{m_2}|m_0\rangle|0^n\rangle|m_2\oplus y\oplus(k\|1)\rangle \quad |c_0\rangle|y\rangle|BC_k(y\oplus m_2)\rangle \qquad \text{식(2)}$$

$\gamma:=m_2\oplus y$일 때, 식(1)과 식(2)는 다음과 같이 표현된다.

$$\sum_{\gamma}|m_0\rangle|0^n\rangle|\gamma\rangle|c_0\rangle|y\rangle|BC_k(\gamma)\rangle \qquad \text{식(3)}$$

$$\sum_{\gamma}|m_0\rangle|0^n\rangle|\gamma\oplus(k\|1)\rangle|c_0\rangle|y\rangle|BC_k(\gamma)\rangle \qquad \text{식(4)}$$

따라서 공격자는 다음과 같은 상태를 얻을 수 있다.

$$\sum_{\gamma}|m_0\rangle|0^n\rangle(|\gamma\rangle+|\gamma\oplus(k\|1)\rangle)|c_0\rangle|y\rangle|BC_k(\gamma)\rangle$$

이 후, Hadamard 게이트를 통과하면,

$$\sum_{\gamma}\sum_{z}((-1)^{\gamma\odot z}+(-1)^{\{\gamma\oplus(k\|1)\}\odot z}) \quad |m_0\rangle|0^n\rangle|z\rangle|c_0\rangle|y\rangle|BC_k(\gamma)\rangle$$

$$=\sum_{\gamma}\sum_{z}(-1)^{\gamma\odot z}(1+(-1)^{(k\|1)\odot z}) \quad (\odot\text{는 내적}) \quad |m_0\rangle|0^n\rangle|z\rangle|c_0\rangle|y\rangle|BC_k(\gamma)\rangle$$

최종적으로 공격자가 측정 시 $z\odot(k\|1)=0$을 만족하는 경우 벡터 z를 얻을 수 있고, 그렇지 않으면 값이 0으로 수렴한다. n개의 독립된 벡터 z_i를 얻을 때까지 위의 과정을 반복하면 Gaussian 소거법에 의해 비밀키 k를 복구할 수 있으며 Π_{IGE}가 IND-qCPA를 만족하지 못함을 알 수 있다.

IV. 결론

양자 컴퓨터의 개발에 대비하여 양자 안전성에 대해 다양한 분야[7]에서 연구가 활발히 진행되고 있다.

본 논문에서는 텔레그램에서 사용되고 있는 IGE 모드의 IND-qCPA에 대해 알아보았다. IGE모드에 사용한 블록암호가 sPRF인 경우에는 안전하지 않음을 반례를 통해 제시하였으며, 제시된 블록암호를 사용할 경우 Simon 알고리즘을 이용하여 비밀키를 복원할 수 있음을 제시하였다.

추후 과제로는 IGE 모드가 sPRF인 블록암호를 사용할 경우 양자 안전성을 만족하진 못하지만, qPRF인 블록암호를 사용할 경우에 대한 양자 안전성 등이 요구된다.

[참고문헌]

[1] Peter W. Shor." Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer." *SIAM Journal on Computing*, 26(5):1484-1509, October 1997.

[2] Carl M. Campbell. “Design and specication of cryptographic capabilities.” *IEEE Communications Society Magazine*, 16(6):15-19, November 1978.

[3] Dan Boneh and Mark Zhandry." Secure signatures and chosen ciphertext security in a quantum computing world." In *Proceedings of the 33rd Annual International Cryptology Conference (Crypto 2013)*, pages 361-379, Santa Barbara, CA, USA, August 2013.

[4] Mark Zhandry. "How to construct quantum random functions." In *Proceedings of the 53rd Annual Symposium on Foundations of Computer Science (FOCS 2012)*, pages 679-687, New Brunswick, NJ, USA, October 2012.

[5] Peter W. Shor. "Algorithms for quantum computation: Discrete logarithms and factoring." In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science (FOCS 1994)*, pages 124-134, Santa Fe, NM, USA, November 1994.

[6] Mayuresh V. Anand, Ehsan E. Targhi, Gelo N. Tabia, and Dominique Unruh. “Post-quantum security of the CBC, CFB, OFB, CTR, and XTS modes of operation.” In *Proceedings of the 7th International Workshop on Post-Quantum Cryptography (PQCrypto 2016)*, pages 44-63, Fukuoka, Japan, February 2016.

[7] Dan Boneh and Mark Zhandry. "Quantum-secure message authentication codes." In *Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, Berlin, Heidelberg, 2013.

Yu와 Oh가 제안한 OAuth기반의 사용자 인증 프레임워크 정형화 검증

이재경* 최홍석* 한종찬* 권오제* 유일선*

*순천향대학교 정보보호학과

Formal Security Analysis on the Yu-Oh's OAuth based Authentication Framework

Jae-Kyeong Lee* Hong-Seok Choi* Jong-Chan Han* Oh-Je Kwon*

Ilsun You*

*Department of information Security Engineering, SoonChunHyang Univ.

요 약

현재 제 4차 산업혁명이 도래함에 따라 사물인터넷(Internet of Things) 기술은 이를 주도할 핵심 기술로 주목 받고 있다. 사물인터넷 기술은 사람과 사물 간 의사소통을 위해 여러 기술이 융합되어 개발된다. 이처럼 융합기반의 기술은 다양한 보안 취약점이 존재할 수 있으며, 특히 프라이버시 문제는 사물인터넷 환경에서 꼭 해결해야 할 문제 중 하나이다. 본 논문에서는 사물인터넷 환경에서의 프라이버시 문제를 개선하기 위해 2015년도에 제안된 Yu와 Oh의 사용자 인증 프레임워크를 BAN 논리를 통하여 정형화 검증을 수행함으로써 취약점을 분석하고, 그에 따른 대응방안을 제시한다.

I. 서론

현재 제 4차 산업혁명이 도래함에 따라 사물인터넷 기술은 4차 산업혁명을 주도할 주요 기술로 주목 받고 있다. 사물인터넷 기술에 대한 기대가 높아지는 만큼 보안을 위협하는 요소 또한 다양하게 존재한다[1]. 사물인터넷 기술은 여러 가지 요소의 기술들이 융합되어 개발되기 때문에 보안 취약점이 다양하게 존재 한다. 특히 프라이버시 문제는 사물인터넷 환경에서 반드시 해결해야할 문제로 간주되며 스마트 시티, 스마트 홈 등 핵심 사물인터넷 응용기술들이 성공적으로 우리생활에 정착하기 위해서 사용자 프라이버시는 중요한 보안 요소로써 보호되어야한다. 하나의 예로 보안 카메라의 유아용 CCTV 의 경우 공격자가 주소를 알게 되면 쉽게 영상과 음성을 도청 할 수 있으며 실시간 영상이 유포될 수 있다[2].

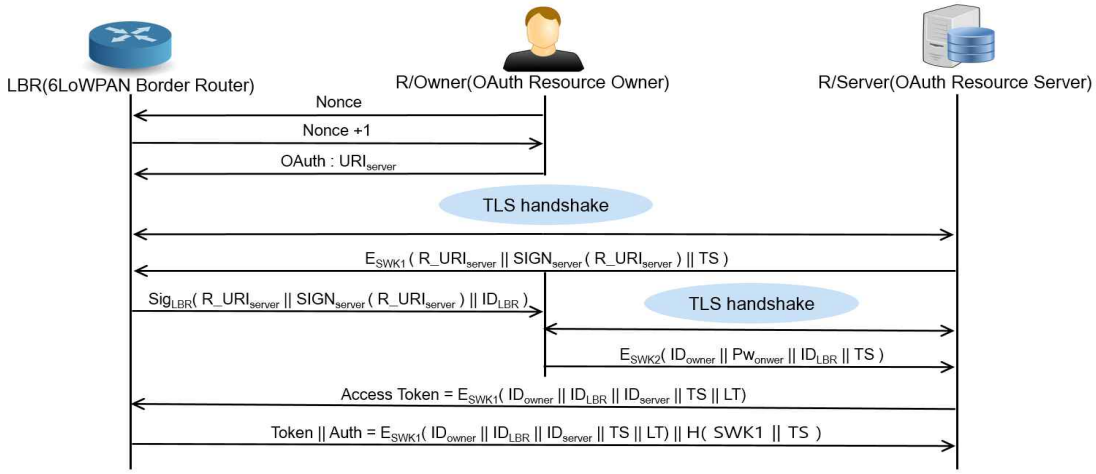
사물인터넷 환경의 프라이버시 문제를 개선하기 위하여 2015년도에 Yu와 Oh의 OAuth 기반의 사용자 인증 프레임워크가 제안되었다[3]. 제안된 프레임워크는 6LoWPAN 환경을 대상으로 스마트 환경에 사용자가 장치에 접근 시 사용자 인증을 수행하며 이 때 사용자의 정보를 남기지 않고, 제3 서비스 제공자가 제공하는 액세스 토큰(액세스 토큰)을 이용하여 인증하는 방식으로 사용자 인증 시 사용자의 정보를 남기지 않는 특징이 있다.

본 논문에서는 Yu와 Oh의 사용자 인증 프레임 워크를 BAN 논리기법[4][5]을 이용, 정형화 검증을 진행하고 취약점을 분석하며, 그에 따른 대응방안을 제시한다.

II. 관련연구

2.1 Yu와 Oh의 사용자 인증 프레임워크

해당 인증 프레임워크는 6LoWPAN 환경을 대상으로 하며 사용자(R/Owner)가 장치에 접근 시 LBR(6LoWPA Border Router)과 상호 인증을 수행한다. 이때 사용자는 LBR에 정보를 남



[그림 1] 사용자 인증 프레임워크 flow

기지 않고 서비스 제공자(R/Server)를 통해 사용자 인증을 하며 액세스 토큰을 발급하는 방식이다. OAuth 기반의 사용자 인증 프레임워크의 흐름은 [그림 1]과 같다:

① 사용자가 통신을 위해 장치를 검색하고 장치가 등록된 LBR이 응답한다.
② 서비스 제공자의 OAuth 인증을 위한 URI (Uniform Resource Identifier)주소를 LBR에 전송한다.
③ LBR과 서비스 제공자는 TLS(Transport Layer Security) handshake를 통해 보안채널을 형성한다. 이후 서비스 제공자는 사용자 인증에 사용할 R_URIserver (Redirection Server's URI)을 자신의 비밀키로 서명하고 세션키 (SWK1)로 암호화 한 메시지를 LBR에 전송한다.
④ LBR은 수신한 R_URI을 서명 값과 LBR의 식별자를 자신의 비밀 키로 서명하여 사용자에게 전송한다.
⑤ 사용자는 서비스 제공자와 TLS handshake를 통해 보안채널을 형성한다. 이후 사용자의 ID, PW, LBR 식별자 TS(Time Stamp)를 사용자와 서버 간 통신키 (SWK2)로 암호화하여 전송한다.
⑥ R/Server의 인증이 성공하면 액세스 토큰을 발급한다.

2.2 BAN 논리 기법

BAN 논리기법은 보안프로토콜 검증을 위해 사용되는 대표적인 도구이다. 아래의 [표 1]과 [표 2]는 BAN 논리에서 사용되는 기호와 규칙들을 보여준다.

[표 1] BAN 논리 기호

표현방식	의미
A believes X	A 는 B를 믿는다.
A sees X	A는 X를 포함하는 메시지를 수신했다.
A said X	A는 과거에 X를 포함하는 메시지를 보냈다.
A controls X	A는 X에 대해 전권을 가지고 있다
Fresh(X)	X는 최신의 데이터
$A \overset{K}{\leftrightarrow} B$	K는 A와 B의 공유 비밀키
$\overset{K}{\rightarrow} A$	K는 A의 공개키
$\{X\}_K$	X는 K로 암호화
$<X>_K$	X는 K와 조합
X,Y	X와 Y의 결합

[표 2] BAN 논리 추론규칙

추론규칙	표현
MM	$\frac{A believes A \overset{K}{\leftrightarrow} B,\ A sees X_K}{A believes B said X}$
NV	$\frac{A believes fresh(X),\ A believes B said X}{A believes B believes X}$
JR	$\frac{A believes B controls X,\ A believes B believes X}{A believes X}$
FR	$\frac{A believes fresh(X)}{A believes fresh(X,Y)}$
BCR	$\frac{A believes X,\ B believes Y}{A believes (X,Y)}$

MM(Message Meaning Rule), NV(Nonce Verification)
JR(Jurisdiction Rule), FR(Freshness Rule)
BC(Belief Conjunction Rule)

III. 정형화검증

3.1 BAN 논리를 통한 정형화 검증

정형화 검증은 3단계로 진행된다. 1단계는 가정(Assumption)과 목표(Goals)를 설정하는 단계이다. 2단계는 해당 프레임워크를 모델링(Idealization)하는 단계이다. 그리고 3단계 검증(Derivation)을 통하여 프레임워크를 검증한다.

본 논문에서 가정과 목표는 [그림 2]와 [그림3]과 같고, 모델링은 [그림 4]와 같으며, 마지막으로 [그림 5]와 같이 검증된다.

<Goals>

<G1> LBR believes R/Server believes R_URI_{server}

<G2> R/Owner believes R/Server believes R_URI_{server}

<G3> R/Server believes (ID_{owner} , PW_{owner})

<G4> LBR believes R/Server believes (ID_{owner} , ID_{LBR} , ID_{server})

<G5> R/Server believes LBR believes (ID_{owner} , ID_{LBR} , ID_{server})

[그림 3] Goals

<Assumptions>

<A1> LBR believes LBR $\overset{SWK1}{\longleftrightarrow}$ R/Server

<A2> LBR believes fresh(TS_1)

<A3> R/Owner believes $\overset{K}{\rightarrow} LBR$

<A4> R/Server believes R/Owner $\overset{SWK2}{\longleftrightarrow}$ R/Server

<A5> R/Server believes fresh(TS_2)

<A6> R/Server believes R/Owner controls (ID_{owner}, PW_{owner})

<A7> LBR believes fresh(TS_3)

<A8> R/Server believes LBR $\overset{SWK1}{\longleftrightarrow}$ R/Server

<A9> R/Server believes fresh(TS_4)

<H1> R/Owner believes $\overset{K}{\rightarrow} Server$

<H2> R/Owner believes fresh(R_URI_{Server})

[그림 2] Assumption

<Idealization>

(1) R/Server -> LBR : $\left\{R\_URI_{server}, \{R\_URI_{server}\}_{K_{server}^{-1}}, TS_1\right\}_{SWK1}$

(2) LBR -> R/Owner : $\left\{R\_URI_{server}, \{R\_URI_{server}\}_{K_{server}^{-1}}, ID_{LBR}\right\}_{K_{LBR}^{-1}}$

(3) R/Owner -> R/Server : $\{ID_{owner}, PW_{owner}, ID_{LBR}, TS_2\}_{SWK2}$

(4) R/Server -> LBR : $\{ID_{owner}, ID_{LBR}, ID_{server}, TS_3, LT\}_{SWK1}$

(5) LBR -> R/Server : $\{ID_{owner}, ID_{LBR}, ID_{server}, TS_4, LT\}_{SWK1}$ $H(SWK1, TS_4)$

[그림 4] Idealization

<Derivation>

<D1>LBR sees$\{R_URI_{Server}, \{R_URI_{Server}\}_{K_{server}^{-1}}, TS_1\}_{SWK1}$

<D2>LBR believes R/Server said (R_URI_{Server}, $\{R\_URI_{Server}\}_{K_{server}^{-1}}$, TS_1) by (D1), (A1), MM

<D3>LBR believes R/Server believes R_URI_{Server} by (D2), (A2), FR, NV, BCR

<D4>R/Owner sees $\{R\_URI_{Server}, \{R\_URI_{Server}\}_{K_{server}^{-1}}, ID_{LBR}\}_{K_{LBR}^{-1}}$

<D5>R/Owner sees $\{R\_URI_{Server}\}_{K_{server}^{-1}}$ by (D4), (A3), DCR

<D6>R/Owner believes R/Server said R_URI_{Server} by (D5), (H1), MM

<D7>R/Owner believes R/Server believes R_URI_{Server} by (D6), (H2), NV

<D8>R/Server sees$\{ID_{owner}, PW_{owner}, ID_{LBR}, TS_2\}_{SWK2}$

<D9>R/Server believes R/Owner said (ID_{owner}, PW_{owner}, ID_{LBR}, TS_2) by (D8), (A4), MM

<D10>R/Server believes R/Owner believes (ID_{owner}, PW_{owner}) by (D9), (A5), FR, NV, BCR

<D11>R/Server believes (ID_{owner}, PW_{owner}) by (D10), (A6), JR

<D12>LBR sees $\{ID_{owner}, ID_{LBR}, ID_{server}, TS_3, LT\}_{SWK1}$

<D13>LBR believes R/Server said (ID_{owner}, ID_{LBR}, ID_{server}, TS_3, LT) by (D12), (A1), MM

<D14>LBR believes R/Server believes(ID_{owner}, ID_{LBR}, ID_{server}) by (D13), (A7), FR, NV

<D15>R/Server sees $\{ID_{owner}, ID_{LBR}, ID_{server}, TS_4, LT\}_{SWK1}$, H(SWK1, TS_4)

<D16>R/Server believes LBR said (ID_{owner}, ID_{LBR}, ID_{server}, TS_4 , LT) by (D15), (A8), MM

<D17>R/Server believes LBR believes(ID_{owner}, ID_{LBR}, ID_{server}) by (D16), (A9), FR, NV, BCR

[그림 5] Derivation

3.2 정형화 검증 분석

사용자는 LBR이 자신의 개인키로 서명한 내용에 있는 R_URI_{SERVER}에 대하여 신뢰가 있어야한다. 사용자측은 액세스 토큰을 발급받기 위하여 서비스 제공자와 TLS handshake를 진행하는데 이때 필요한 정보가 R_URI_{SERVER}이다. 이 URI를 통하여 서비스 제공자와의 TLS handshake를 유도하는 것이다. 또한 서비스 제공자는 사용자가 가 전송한 ID_{Owner}와 PW_{Owner}를 신뢰함을 전제로 Owner를 인증한다는 액세스 토큰을 발급한다. 따라서 서비스 제공자는 사용자의 ID와 PW를 신뢰하여야 하며 서로의 토큰을 주고받는 과정에서 서비스 제공자와 LBR은 서로가 보낸 ID를 비교하여 인증을 받는 과정이므로 상호간에 액세스 토큰 속 ID를 신뢰 하여야 한다.

3.2 정형화 검증 결과

위 검증에서 <G2> R/Owner believes R/server believes R_URI_{SERVER}을 만족하기 위해선 해당 메시지 Idealization (2) 메시지에 적시성을 증명하는 메시지가 존재하여야 한다. 따라서 <H2> R/Owner believes Fresh(R_URI_{SERVER}) 가정을 추가하여 R_URI_{SERVER}에 대한 적시성 메시지를 추가하였다. 해당 (2) 메시지의 서명에서 적시성을 확인할 수 없다면 이 프로토콜 역시 재전송 공격에 취약하다. 서비스 제공자와 통신을 하였고, 이후에 다른 서비스 제공자와 통신을 할 경우 재전송 공격으로 인해 이전에 통신에 사용한 $\{R\_URI_{SERVER}\}K_{SERVER}^{-1}$ 메시지를 그대로 전송하게 되면 사용자는 이전에 통신한 서비스 제공자와 TLS handshake를 진행하는 일이 일어 날 수 있다. 즉, 이전 세션과의 통신이 재전송 공격으로 인하여 발생하는 것이다. 또한 <H1> 가정을 보면 서비스 제공자의 공개키를 신뢰한다는 가정을 추가하였는데, 이 가정 역시 없으면 LBR의 개인키로 서명한 R_URI_{SERVER}와 서비스 제공자의 개인키로 암호화한 R_URI_{SERVER}이 일치하지 않을 수 있으므로 이 역시 재전송 공격에 취약하다. 앞에서 언급한 R_URI_{SERVER}의 믿음이 도출되지 않는다.

이에 대한 대응방안으로 인증서를 이용하여 상호간 인증하는 과정이 추가되어야 한다. 또한 적시성을 확인하기 위해선 서비스 제공자의 공개키에 대한 신뢰와 R_URI_{SERVER}에 대한 적시성을 위하여 기존에 사용자와 LBR이 통신을 위하여 주고받았던 난수 값 교환을 재 교환 하는 등, 적시성의 확인이 필요하다.

IV. 결론

본 논문에서는 BAN 논리를 이용하여 Yu와 Oh의 사용자 인증 프레임워크를 정형화 검증하였으며, 재전송 공격에 취약함을 보였다. 그리고 이러한 취약점에 대한 대응 방안을 제시하였다. 향후 연구 방향으로써 자동화 검증 도구를 이용하여 다양한 공격 가능성 여부를 체계적으로 검증할 예정이다.

[참고문헌]

[1] 배상태, 김진경, “사물인터넷(사물인터넷) 발전과 보안의 패러다임 변화”, 한국과학기술기획평가원 2016.07, pp 48-49

[2] 이상홍, “사물인터넷 현황 및 주요 이슈”, 정보통신기술진흥센터, 2014.12, pp 1

[3] 유성태(Sung-Tae Yu), 오수현(Soo-Hyun Oh), “사물인터넷 환경에 적합한 OAuth 기반의 사용자 인증 프레임워크”, Journal of the Korea Academia-Industrial cooperation Society Vol. 16, No. 11 pp. 8057-8063, 2015.

[4] JAN WESSELS, “Applications of BAN logic”, CMG FINANCE B.V 2001.4 pp 3

[5] M. Burrows, M. Abadi, and R. Needham, “A Logic of authentication,” ACM Trans-actions Computer Systems, Vol. pp. 8-36 August, 1990.

[6] Jonas Olsson, “6LoWPAN demystified”, Texas Instruments, 2014.

Quantum Message Authentication Scheme using Single Qubit Measurement

Yeon-Ho Choi[1,2], Min-Sung Kang[2], Sang Wook Han[2,*], Sung Moon[2]

[1]Display and Semiconductor phyisc department, Korea University.

[2]Centor for Quantum Information, Korea Institute of Science and Technology(KIST).

요 약

본 논문에서는 양자 메시지의 무결성과 기원 인증을 제공하는 양자 메시지 인증 기법을 제안한다. 제안한 기법은 단일 큐빗 측정만으로 양자 메시지 인증 코드 상태를 생성하기 때문에 기존에 Bell 측정으로 생성하는 기법들 보다 구현이 쉽고 안정적인 동작이 가능하다. 그리고 안전성 분석을 위해 random oracle 모델에 기반하여 제안하는 기법에 적합한 quantum random oracle model을 설계하고 이것을 이용하여 quantum collision resistance를 보장함을 보인다.

Key Words: Quantum cryptography, Single qubit measurement, Quantum message authentication code state, Quantum random oracle, SWAP test

I. 서론

현대암호는 폭발적인 양자 컴퓨터의 기술 발전에 의해서 안전성에 큰 위협을 받게 되었다 [1,2]. 그러므로 양자 컴퓨터 환경에서도 안전한 양자암호가 대안으로 주목받고 있다. 양자암호는 기밀성을 제공하는 QKD 중심으로 많은 연구가 활발하게 진행되어 왔다. 하지만, 완전한 암호로서 기능을 하기 위해서는 기밀성뿐만 아니라 무결성, 인증, 부인방지라는 보안 서비스를 반드시 제공해야 하며, 이를 제공하기 위해서 양자 인증과 양자 서명이 개발되어야 한다.

현대 암호에서 인증은 크게 3가지로 메시지 인증, 개체 인증, 키 인증로 나눌 수 있으며, 이 중 메시지의 무결성과 기원 인증을 보장하는 메시지 인증은 해쉬 함수(hash function)와 같은 일방 함수(one-way function)로 구현된다 [3,4]. 이와 동일하게 양자 메시지 인증도 양자 메시지의 무결성과 기원 인증을 제공해야 하며, 그동안 원격전송(teleporation) [5, 6], CNOT 연산자 [7, 8], 푸리에(Fourier) 변환 [9], 그루버(Grover) 알고리즘 [10] 등을 이용하는 양자 메시지 인증이 제안되었다.

본 논문에서 제안하는 기법은 단일 큐빗 측정(single qubit measurement)을 이용하여 양자 메시지 인증 코드를 생성하고 이를 활용하여 인증을 수행하는 방법이다. Alice와 Bob이 임의의 얽힘 상태(arbitrary entangled state)를 공유한 상황에서 Alice가 자신의 큐빗을 측정한다면, 이들이 공유한 얽힘은 끊어지게 되고 Bob이 소유한 양자 상태는 변환하게 된다. 이렇게 변환된 Bob의 큐빗을 양자 메시지 인증 코드 상태(QMAC state)라고 한다. 기존 기법들은 Bell 측정이 수반되는 teleportation을 이용하여 QMAC state를 전송하였지만 제안하는 기법은 단일 큐빗 측정만으로 QMAC state를 전송할 수 있는 장점이 있다 [5, 6]. 그리고 제안하는 기법의 안전성은 quantum random oracle을 이용한 분석을 통해서 quantum collision resistance가 보장됨을 보인다. 이를 통해서 QMAC state가 양자 메시지의 무결성과 기원 인증을 제공함을 확인한다.

II. 본문

본 논문에서 제안하는 양자 메시지 인증 기법은 준비단계, 양자 메시지 인증 코드 생성단계, 검증단계로 구성된다. 그림 1은 우리가 제안하고자 하는 양자 메시지 인증 코드 기법의 도식이며, 각 단계에 대한 구체적인 설명은 다음과 같다.

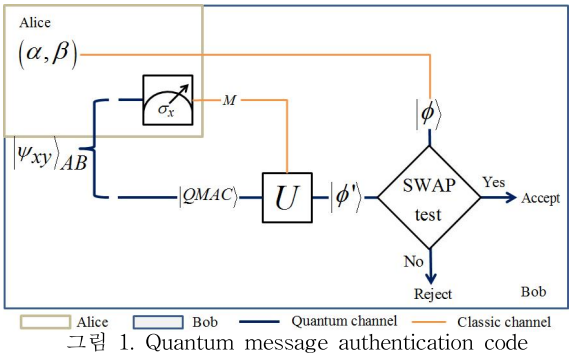


그림 1. Quantum message authentication code

2.1 준비단계

P1. 사전에 Alice와 Bob은 비밀 키 를 공유하며, 다음과 같다.

$$k = x \| y = (00,01,10,11), x, y \in \{0,1\} \tag{1}$$

P2. Alice는 Bob에게 message (α,β)를 생성한 후 공개하며, 다음과 같다.

$$\alpha = \cos\frac{\theta}{2}, \beta = \sin\frac{\theta}{2} \tag{2}$$

여기서, $\alpha| + |\beta|^2 = 1, 0 < \theta < \pi$ 이다.

P3. Alice는 k와 (α,β)를 이용하여, 임의의 얽힌 상태 $|\psi_{y}>_{B}$를 생성하고 Bob과 공유하며, 다음과 같다.

$$\begin{aligned}|\psi_{xy}\rangle &= \delta|0\rangle_A|y\rangle_B + \gamma(-1)^x|1\rangle_A|\bar{y}\rangle_B \\ &= \frac{1}{\sqrt{2}}[|+\rangle_A(\delta|y\rangle_B + \gamma(-1)^x|\bar{y}\rangle_B) \\ &\quad +|-\rangle_A(\delta|y\rangle_B - \gamma(-1)^x|\bar{y}\rangle_B)]\end{aligned} \tag{3}$$

여기서, $\delta = \alpha$, $\gamma = \beta$, $\delta|^2 + |\gamma|^2 = 1$, $y = 1 \oplus y$이다.

2.2 양자 메시지 인증 코드 생성단계

T1. Alice는 Bob과 공유한 얽힌 상태 중 자신이 소유한 큐빗을 σ_x-basis로 측정한다. Alice의 측정에 의해 Bob의 큐빗은 QMAC state $|MAC>_B$로 상태 변환하게 된다. Alice는 Bob에게 측정결과 M을 공개하며, $|QMAC>_B$는 다음과 같다.

$$|QMAC\rangle_B = \delta|y\rangle_B \pm \gamma(-1)^x|\bar{y}\rangle_B \tag{4}$$

2.3 검증단계

V1. Bob은 사전에 공유한 k와 Alice의 측정결과 M에 대응하는 unitary operator U를 $|QMAC>_B$에 적용하여 quantum message $|\phi'>$를 얻는다. 표 1은 $|QMAC>_B$에 적용되는 U를 나타낸다. $|\phi'>$는 다음과 같다.

$$|\phi'\rangle = \delta|0\rangle_B + \gamma(-1)^x|1\rangle_B \tag{5}$$

V2. Bob은 Alice가 공개한 메시지를 이용하여 quantum message state $|\phi>$를 생성하며 다음과 같다.

$$|\phi\rangle = \delta|0\rangle_B + \gamma(-1)^x|1\rangle_B \tag{6}$$

M \ $k = x \| y$	00	01	10	11
$\|+\rangle$	I	σ_x	σ_z	$i\sigma_y$
$\|-\rangle$	σ_z	$i\sigma_y$	I	σ_x

표 1. 사전에 공유한 비밀키 와 Alice의 측정결과 M에 대응되는 unitary operator $U \in \{I,\sigma_x,\sigma_y,\sigma_z\}$이며, $I,\sigma_x,\sigma_y,\sigma_z$는 pauli's operator이다.

V3. Bob은 swap test를 이용하여 양자 메시지 $|\phi>$와 $|\phi'>$의 일치여부를 검증한다. 그림 2는 SWAP test의 도식이며, swap test의 결과는 다음과 같다 [11].

$$\begin{aligned}&\frac{1}{2}|0\rangle_1(|\phi\rangle_2|\phi'\rangle_3 + |\phi'\rangle_2|\phi\rangle_3) \\ &+\frac{1}{2}|0\rangle_1(|\phi\rangle_2|\phi'\rangle_3 + |\phi'\rangle_2|\phi\rangle_3)\end{aligned} \tag{7}$$

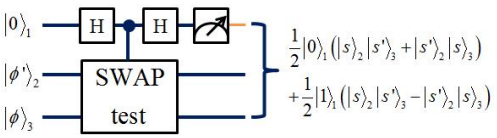


그림2. SWAP test의 도식.

$|\phi>$와 $|\phi'>$가 일치한다면, $|0>_1$만 측정되며, Bob은 Alice의 $|QMAC>_B$를 받아들인다. 그러나, 일치하지 않는다며, 확률적으로 $|0>_1$ 또는 $|1>_1$이 측정된다. $|1>_1$이 측정 될 경우, 두 양자상태 $|\phi>$와 $|\phi'>$가 일치하지 않음을 알 수 있며, 각각의 측정 확률 $\Pr_{|0>}$와 $\Pr_{|1>}$은 다음과 같다.

$$\begin{aligned}\Pr_{|0\rangle} &= \frac{1+|\langle\phi'|\phi\rangle|^2}{2} \\ &= \frac{1+\alpha^2\delta^2 + \alpha^*\delta\beta\gamma^* + \alpha\delta^*\beta^*\gamma + \beta^2\gamma^2}{2}\end{aligned} \tag{8}$$

$$\Pr_{|1\rangle} = \frac{1-|\langle\phi'|\phi\rangle|^2}{2} = \frac{1-\alpha^2\delta^2-\alpha^*\delta\beta\gamma^*-\alpha\delta^*\beta^*\gamma-\beta^2\gamma^2}{2} \quad (9)$$

여기서, 두 양자 $0>$ 이 측정될 확률은 SWAP test에서 발생할 오류 확률이다.

III. 안전성 분석

3.1 Quantum random oracle

현대암호에서 메시지 인증은 해쉬함수와 같은 일방 함수로 구현이 되며, 이러한 일방함수는 preimage resistance, 2nd preimage resistance, collision resistance 등의 3가지 조건을 만족해야한다 [3, 4]. 특히, 가장 약한 조건인 collision resistance를 반드시 만족해야 하며, 주로 random oracle을 이용하여 이를 증명한다. 제안하는 양자 메시지 인증 기법의 안전성은 random oracle의 기반인 quantum random oracle(QRO)을 이용하여 quantum collision resistance가 존재함을 보임으로써 안전성을 증명한다. 그림 3은 제안하는 양자 메시지 인증 기법의 안전성을 분석하기 위한 QRO의 도식이다.

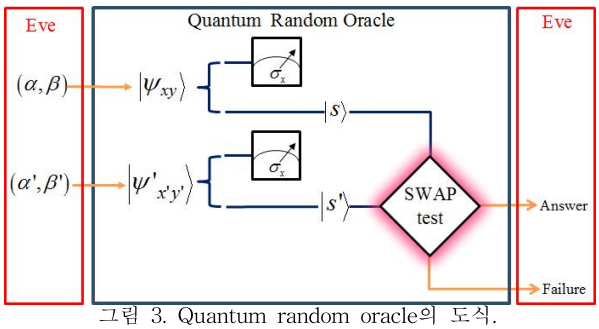


그림 3. Quantum random oracle의 도식.

Eve는 서로 다른 두 message (α,β)와 (α',β')를 QRO에 입력한다. QRO은 (α,β)와 (α',β')를 이용하여 두 entangled states $|\psi>$와 $|\psi'>$를 생성하며, 다음과 같다.

$$\begin{aligned}|\psi_{xy}\rangle &= \alpha|0\rangle|y\rangle+\beta(-1)^x|1\rangle|\bar{y}\rangle \\ &= \frac{1}{\sqrt{2}}[|+\rangle(\alpha|y\rangle+\beta(-1)^x|\bar{y}\rangle) \\ &\quad +|-\rangle(\alpha|y\rangle-\beta(-1)^x|\bar{y}\rangle)]\end{aligned} \quad (10)$$

$$\begin{aligned}|\psi'_{x'y'}\rangle &= \alpha'|0\rangle|y'\rangle+\beta'(-1)^{x'}|1\rangle|\bar{y}'\rangle \\ &= \frac{1}{\sqrt{2}}[|+\rangle(\alpha'|y'\rangle+\beta'(-1)^{x'}|\bar{y}'\rangle) \\ &\quad +|-\rangle(\alpha'|y'\rangle-\beta'(-1)^{x'}|\bar{y}'\rangle)]\end{aligned} \quad (11)$$

여기서, $\alpha=\cos\frac{\theta}{2},\beta=\sin\frac{\theta}{2},\alpha'=\cos\frac{\theta'}{2},\beta'=\sin\frac{\theta'}{2}$, $\theta\neq\theta'$, $0<\theta<\pi, 0<\theta'<\pi$ 이다. QRO은 (α,β)와 (α',β')에 의해서 생성된 $|\psi_{xy}>$와 $|\psi'_{x'y'}>$를 각각 σ_x-basis로 측정하면, $|s>$와 $|s'>$로 얻게 되며, 다음과 같다.

$$\begin{aligned}|s\rangle &= \alpha|0\rangle|y\rangle+\beta(-1)^x|1\rangle|\bar{y}\rangle \\ |s'\rangle &= \alpha|0\rangle|y\rangle+\beta(-1)^x|1\rangle|\bar{y}\rangle\end{aligned} \quad (12)$$

마지막으로 QRO은 $|s>$와 $|s'>$를 swap test를 통해서 quantum collision 발생 여부를 Eve에게 대답한다. $|s>$와 $|s'>$가 일치한다면, quantum collision이 발생하여 answer라고 대답하지만, 불일치 할 경우, quantum collision이 발생되지 않아 failure라는 대답을 할 것이다.

3.1 Quantum collision resistance

위에서 설명했듯이 두 양자상태 $|s>$와 $|s'>$가 quantum collision의 발생 유무를 swap test를 이용하여 확인한다. $|s>$와 $|s'>$가 서로 다름에도 불구하고 swap test에서 $|0>$측정될 확률이 존재하며, 이를 quantum collision으로 간주한다. 즉, quantum collision이 발생확률 Pr 은 다음 같다.

$$\Pr_c = \frac{1-\bar{\varepsilon}}{2} \quad (13)$$

여기서, $\bar{\varepsilon}$ 는 $s|s'>|^2$의 평균이며, 식 (12)를 이용하여 계산하면 다음과 같은 결과를 얻어낸다.

$$\bar{\varepsilon} = (\varepsilon_1+\varepsilon_2+\varepsilon_3+\varepsilon_4+\varepsilon_5+\varepsilon_6+\varepsilon_7+\varepsilon_8) \quad (14)$$

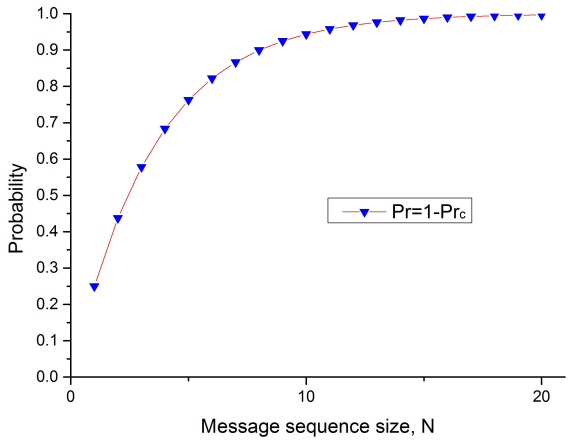
여기서 $\varepsilon_1,\varepsilon_3,\varepsilon_5,\varepsilon_7$ 는 $y=y'$ 일 때이며, 다음과 같다.

$$\begin{aligned}\varepsilon_1 &= \alpha^2\alpha'^2+\beta^2\beta'^2+2(-1)^x(-1)^{x'}\alpha^*\alpha'\beta^*\beta' \\ \varepsilon_3 &= \alpha^2\alpha'^2+\beta^2\beta'^2-2(-1)^x(-1)^{x'}\alpha^*\alpha'\beta^*\beta' \\ \varepsilon_5 &= \alpha^2\alpha'^2+\beta^2\beta'^2-2(-1)^x(-1)^{x'}\alpha^*\alpha'\beta^*\beta' \\ \varepsilon_7 &= \alpha^2\alpha'^2+\beta^2\beta'^2+2(-1)^x(-1)^{x'}\alpha^*\alpha'\beta^*\beta'\end{aligned} \quad (15)$$

$\varepsilon_2,\varepsilon_4,\varepsilon_6,\varepsilon_8$ 는 $y\neq y'$ 일 때이며, 다음과 같다.

$$\begin{aligned}\varepsilon_2 &= \alpha^2\beta'^2+\alpha'^2\beta^2+2(-1)^x(-1)^{x'}\alpha^*\alpha'\beta^*\beta' \\ \varepsilon_4 &= \alpha^2\beta'^2+\alpha'^2\beta^2-2(-1)^x(-1)^{x'}\alpha^*\alpha'\beta^*\beta' \\ \varepsilon_6 &= \alpha^2\beta'^2+\alpha'^2\beta^2-2(-1)^x(-1)^{x'}\alpha^*\alpha'\beta^*\beta' \\ \varepsilon_8 &= \alpha^2\beta'^2+\alpha'^2\beta^2+2(-1)^x(-1)^{x'}\alpha^*\alpha'\beta^*\beta'\end{aligned} \quad (16)$$

결과적으로 식 (13)의 Pr 은 3/4이다. 그러므로 제안하는 인증 기법의 안전성을 높이기 위해서는 message를 message sequence $\prod_{i=1}^{N}(\alpha_i,\beta_i)$로 사용되어야 한다. 이때, Pr_c은 $\prod_{i=1}^{N}\left(\frac{1-\bar{\varepsilon}}{2}\right)^{i}=\left(\frac{3}{4}\right)^{n}$ 이 된다. message sequence N이 17개 정도이며, Pr_c이 1% 이하가 된다. 그래프 1은 message sequence N에 따른 quantum collision이 발생되지 않을 확률 Pr를 나타낸다.



그래프 1. Message sequence size N에 따른 quantum collision 이 발생하지 않을 확률 Pr 1−Pr . message sequence size N이 클수록 quantum collision 발생 확률은 낮아진다. 다시 말해서, N>17 일 때, quantum collision의 발생 확률은 1%이하 이다.

IV.결론

본 논문에서는 단일 큐빗 측정을 이용하는 양자 메시지 인증 기법을 제안했다. 제안하는 기법은 통신 구성원들이 임의의 얽힘 상태를 공유한 상황에서 한 구성원이 자신의 큐빗에 단일 큐빗 측정을 하여 다른 구성원에게 QMAC state를 전송한다. 이는 기존에 Bell 측정을 사용하는 기법들 보다 구현할 때 효율적이며 안정적인 동작이 가능하다. 또한, 본 논문에서는 제안하는 기법에 적합한 quantum random oracle model을 설계하였으며, 이를 이용하여 제안하는 기법이 quantum collision resistance를 보장함을 보였다. 특히 message sequence size N이 17 이상일 경우 제안하는 기법에서 quantum collision 발생 확률이 1% 이하임으로 메시지의 무결성과 기원 인증이 보장됨을 알 수 있다.

사사

본 연구는 ICT R&D MSIP/IITP program(Grant No. B0101-161355), KIST research program(Grant No. 2E27231), KIST-ETRI research program(Grant No. 2V05340)의 일환으로 수행하였음.

[참고문헌]

[1] SHOR, Peter W. Algorithms for quantum computation: Discrete logarithms and factoring. In: Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on. Ieee, 1994. p. 124-134.

[2] GROVER, Lov K. A fast quantum mechanical algorithm for database search. In: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. ACM, 1996. p. 212-219.

[3] MENEZES, Alfred J.; VAN OORSCHOT, Paul C.; VANSTONE, Scott A. Handbook of applied cryptography. CRC press, 1996.

[4] STINSON, Douglas R. Cryptography: theory and practice. CRC press, 2005.of quantum messages. In: Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on. IEEE, 2002. p. 449-458.

[5] ZENG, Guihua; KEITEL, Christoph H. Arbitrated quantum-signature scheme. Physical review A, 2002, 65.4: 042312.

[6] LI, Qin; CHAN, W. H.; LONG, Dong-Yang. Arbitrated quantum signature scheme using Bell states. Physical Review A, 2009, 79.5: 054307.

[7] LI, Feng-Guang; SHI, Jian-Hong. An arbitrated quantum signature protocol based on the chained CNOT operations encryption. Quantum Information Processing, 2015, 14.6: 2171-2181.

[8] ZHANG, Long, et al. An improved arbitrated quantum signature protocol based on the key-controlled chained CNOT encryption. Quantum Information Processing, 2017, 16.3: 70.

[9] SHI, Jinjing, et al. A multiparty quantum proxy group signature scheme for the entangled-state message with quantum Fourier transform. Quantum Information Processing, 2011, 10.5: 653-670.

[10] YOON, Chun Seok, et al. Quantum signature scheme based on a quantum search algorithm. Physica Scripta, 2014, 90.1: 015103.

[11] BUHRMAN, Harry, et al. Quantum fingerprinting. Physical Review Letters, 2001, 87.16: 167902.

공증된 conjugate coding 기반 양자 서명

강민성[1], 박병권[1], 우민기[2], 이민수[1], 김용수[1],
조영욱[1], 이상윤[1], 한상욱[1], 문성욱[1]
1한국과학기술연구원 양자정보연구단, 2아주대학교 전자공학과

Certified Conjugate coding based Quantum Signature

Min-Sung Kang[1], Byung Kwon Park[1], Min Ki Woo[2], Yong-Su Kim[1],
Young-Wook Cho[1], Sang-Yun Lee[1], Sang-Wook Han[1], Sung Moon[1]
1Center for Quantum Information, Korea Institute of Science and Technology (KIST)
2Department of Electrical and Computer Engineering, University of Ajou

Abstract

본 논문은 메시지 무결성, 메시지 기원 인증, 부인방지를 제공하는 공증된 양자 서명 기법을 제안한다. 양자 컴퓨터의 발전이 급속도로 진행됨에 따라서 공개키 기반의 전자서명의 안전성이 위협 받을 가능성이 커지고 있다. 이러한 양자 컴퓨터의 위협에 대한 대안으로 다양한 양자 서명 기법들이 제안되고 있다. 그러나 기존의 양자 서명들은 사전 인증과 검증 변수가 없기 때문에 메시지 기원 인증과 부인방지 기능을 제대로 제공하지 못하는 문제점이 있다. 제안하는 공증된 양자 서명 기법은 인증기관이 인증 정보를 이용하여 서명자의 검증 변수를 관리함으로써 메시지 기원 인증과 부인방지를 제공하여 완전한 서명 기능을 수행한다. 그리고 본 연구에서는 quantum signature accecpt rate를 제안하고 이를 이용하여 제안하는 기법의 안전성을 검증한다.

I. Intro

공개 키 암호를 기반으로 하는 전자 서명은 메시지 무결성, 메시지 기원 인증, 부인방지를 제공하는 중요한 암호 기법이다 [1, 2]. 그러나 양자 컴퓨터는 소인수 분해 문제와 이산대수 문제를 효과적으로 풀 수 있다고 보고 있으며, 이로 인해 위의 문제들을 기반으로 하는 전자 서명의 안전성에 대한 이슈가 제기되고 있다 [3]. 따라서 전자 서명의 대안으로 양자 컴퓨터 환경에서도 안전한 양자 서명의 개발이 주목 받고 있는데 양자암호를 대표하는 양자 키 분배 (quantum key distribution, QKD) [4]와 달리 양자서명에 대한 연구는 활발히 연구되고 있지 못하고 있는 실정이다. 근본적인 원인은 양자 공개 키 암호 체계가 없기 때문에 기존 전자서명 기법들처럼 서명을 효율적으로 구현하는데 한계가 있기 때문이다 [5]. 이러한 한계를 극복하기 위해 중재자 양자 서명 기법들과 양자 일회용 서명 기법들이 제안되었으며, 이 기법들을 통해 전자서명의 다양한 기능들을 동일하게 제공하고자 하는 연구들이 진행되고 있다.

중재자 양자 서명 기법은 2002년 Zeng과 Keitel에 의해 처음으로 제안 되었다 [5]. 이후 다양한 중재자 양자 서명 기법 [6-8]들이 제안되었지만 구현의 어려움과 안전성 문제들로 인해 아직까지 실험적으로 연구되고 있지는 않다. 한편, Clarke등은 phase-encoded coherent states를 이용하여 양자 일회용 서명 기법 [9]을 처음으로 실험으로 구현했다 [10]. 이후 다양한 양자 일회용 서명 기법들이 이론적으로 제안되거나 실험적으로 구현되었다 [11-18]. 특히, 2015년 Wallden 등이 제안한 conjugate coding 기반 양자 일회용 서명 [13]을 제안한 이래로 QKD 플랫폼에서 양자 일회용 서명의 구현이 가능하게 되었다 [14-18].

그럼에도 불구하고 제안된 대부분의 양자 일회용 서명 기법은 메시지 기원 인증과 부인 방지를 완전하게 제공하지 않는다 [9-18]. 예를 들어 이 기법들은 양자 공개키가 누구로부터 생성되었는지 알 수 없기 때문에 서명의 생성 주체를 명확하게 하는 메시지 기원 인증 기능을 제공하지 않는다. 게다가 이 기법들은 부인 방지를 위한 어떠한 장치도 없기 때문에 서명 생성자와 수신자의 부인방지 기능을 제공하지 않는다. 이렇듯 기존 기법들이 메시지 기원 인증과 부인방지를 제공하지 못하는 근본적인 이유는 통신 구성원들이 최초 대면을 통해 인증정보를 공유하지 않았기 때문이다 [19, 20]. 또 다른 이유로는 기존 기법들은 인증된 채널을 통해 전달되는 서명에 대한 검증변수가 없기 때문이다 [19, 20]. 여기서 검증 변수는 서명 생성 정보에서 추출되고, 공인기관이 서명을 검증하는데 사용하는 변수를 일컫는다.

본 논문에서는 위의 문제를 해결하기 위한 새로운 양자 서명 기법을 제안한다. 이 기법은 conjugate coding [21]기반 양자 서명 기법에 Merkle 트리 [19, 22]를 적용하여 공증시스템을 구축한 것이며, 공증된 양자 서명(certified quantum signature, CQS)이라고 한다. 본 논문에서는 제안하는 CQS의 구체적인 절차를 설명하며, 이를 정량적으로 분석하고 검증하는 새로운 방법인 quantum signature accept rate(QSAR)을 설명한다.

II. Certified Quantum Signature Scheme

제안하는 CQS 기법에서 서명자Alice, 수신자Bob, 공증기관 CA 등이 통신구성원들이며. 준비 단계, 키 생성 단계, 서명 단계, 증명단계로 구성된다.

2.1 Preparation Phase

P1. Alice와 CA 그리고 Bob와 CA가 최초 대면을 통해 인증정보 I_{AB}와 I_{AC}를 공유한다.

2.2 Key Generation Phase

K1. Alice가 개인 키 나열 $S_M = \left(S_{m_0}, S_{m_1}, \dots, S_{m_{N-1}}\right)$ 을 생성하며, 개인 키 나열 S_M에 각각 Merkle 트리와 conjugate coding을 적용하여 검증 변수 V_A와 양자 공개 키 $|S_{m_j}\rangle = \otimes_{t=0}^{L-1}|s_j^{(t)}\rangle$를 얻는다. 여기서, $|s_j^{(t)}\rangle \in \{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$이며, $s_j^{(t)} \in \{00, 01, 10, 11\}$이다.

K2. Alice는 인증정보 I_{AB}를 이용하여 검증 변수 V_A를 CA에게 등록함과 동시에 양자 공개 키 $|S_{m_j}\rangle = \otimes_{t=0}^{L-1}|s_j^{(t)}\rangle$를 Bob에게 전송한다.

K3. Bob은 Alice로부터 전송 받은 양자 공개 키 $|S_{m_j}\rangle = \otimes_{t=0}^{L-1}|s_j^{(t)}\rangle$를 측정하여 결과 $B_{m_j} = \left(b_j^{(0)}, b_j^{(1)}, \dots, b_j^{(L-1)}\right)$를 얻는다. 여기서 $b_j^{(t)} \in \{00, 01, 10, 11\}$이다.

K1, K2단계에서 설명한 Alice의 행위 때문에 기존의 양자 서명 기법들과 달리 제안하는 기법은 통해 메시지 기원 인증을 제공할 수 있다.

2.3 Signature Phase

S1. Alice는 서명 쌍 (m_j, S_{m_j}, G_{m_j})을 Bob에게 전송한다. 여기서 m_j는 메시지이고 S_{m_j}는 개인 키이고 $G_{m_j} = \left(g_j^{(0)}, g_j^{(1)}, \dots, g_j^{(\log_2 N-1)}\right)$는 검증 변수의 생성 정보이다. 그리고 $g_j^{(p)}$ 는 다음과 같다 [19, 22].

$$g_j^{(p)} = \begin{cases} v_p^{(\lfloor j/2^p - 1 \rfloor)}, \text{if } \lfloor j/2^p \rfloor \bmod 2 = 2 \\ v_p^{(\lfloor j/2^p + 1 \rfloor)}, \text{if } \lfloor j/2^p \rfloor \bmod 2 = 0 \end{cases}, \quad (1)$$

$p = 0, 1, \dots, \log_2 N - 1$ 이며, $\lfloor \bullet \rfloor$ 는 내림함수이다. 그림 1은 Merkle 트리의 높이가 $3 = \log_2 8$이고 생성 정보 G_{m_3}의 위치를 나타낸 것이다.

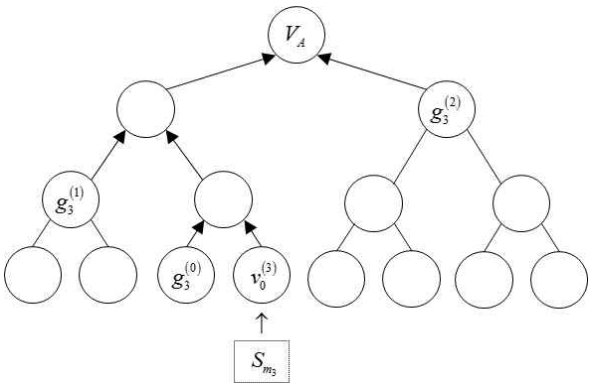


그림1. 개인 키 S_{m_j}에 대응되는 생성정보 $G_{m_3}=\left(g_3^{(0)}, g_3^{(1)}, g_3^{(2)}\right)$의 위치

그리고 S1단계에서 Alice가 생성 정보 G_{m_j}을 Bob에게 전송하는 행위는 Alice가 이후에 서명을 생성하지 않았다고 부인하는 행위를 방지할 수 있다. 이것을 서명 생성자의 부인방지라 한다.

2.4 Verification Phase

V1. Bob은 Alice로부터 전송 받은 개인 키 S_{m_j}와 생성 정보 G_{m_3}를 Merkle tree에 입력하여 검증 변수 V_B를 획득한다.

V2. Bob은 인증정보 I_{AC}를 이용하여 검증 변수 V_B를 CA에게 전송한다.

V3. CA는 Bob으로부터 전송 받은 검증 변수 V_B와 Alice가 K1 단계에서 등록한 검증 변수 V_A가 일치하는지 검증하며, 그 결과를 Bob에게 통보한다.

V4. CA로부터 검증변수가 이상이 없음을 통보 받은 Bob은 양자 공개키의 측정한 결과 B_{m_j}와 Alice의 개인 키 S_{m_j}를 QSAR를 이용하여 일치함을 검증한다. QSAR은 다음절에서 자세하게 설명한다.

V2 단계에서Bob이 검증 변수 V_B의 이상 유무를 확인하기 위해 CA에게 의뢰하는 행위는 이후에 Bob이 서명을 전송 받지 못했다고 하는 부인행위를 방지한다. 이것을 서명 수신자의 부인방지라 한다.

III. Quantum Signature Accept Rate(QSAR)

QSAR은 Hamming distance $D\left(S_{m_j}, B_{m_j}\right)$의 기댓값으로 다음과 같다.

$$QSAR=\left[\tilde{H}_0P_0+\tilde{H}_1P_1+\tilde{H}_QP_Q+\tilde{H}_2(P_2-P_Q)\right]^L \quad (2)$$

$$\sum_k P_k=1,\ k\in\{0,1,2\} \quad (3)$$

여기서, $\tilde{H}_k\equiv 3-H_k$ 이며, H_k는 $D\left(S_{m_j}, B_{m_j}\right)$의 Hamming weigh이다. 그리고 P_k는 H_k가 일어날 사건의 확률이고 k는 $s_j^{(t)}$와 $b_j^{(t)}$가 일치하는 개수를 나타낸다. 게다가 H_Q는 quantum bit error rate(QBER) [23]이 발생할 때의 Hamming distance이고 이에 대응하는 확률은 $P_Q=\frac{QBER}{2}$ 이다. 마지막으로 L은 서명의 크기를 나타낸다.

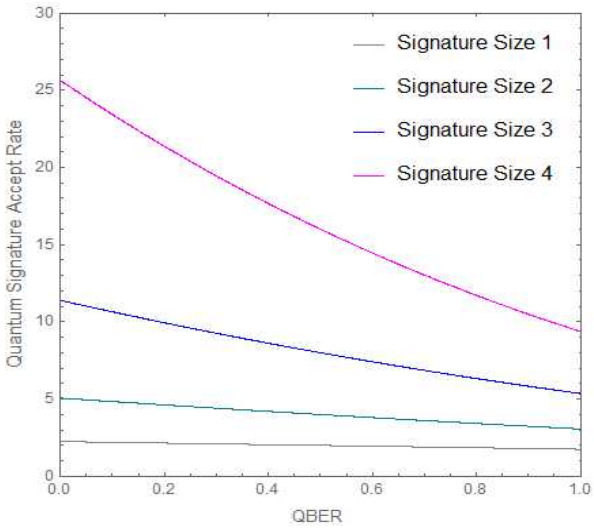


그림2. QBER의 변화에 따른 QSAR. 서명의 크기 L의 범위는 1에서부터 4이다.

제안하는 CQS에서 양자 공개키의 측정한 결과 B_{m_j}와 Alice의 개인 키 S_{m_j}가 일치하는

정도가 QSAR을 벗어나지 않는다면 메시지의 무결성이 보장된다고 볼 수 있다. 그림 2는 식 (2)의 QSAR을 그래프로 표현한 것이다. 그림 2에서 서명 크기가 증가 할수록 QSAR은 증가하고 QBER이 증가 할수록 QSAR은 감소한다.

IV. Conclusion

본 논문에서는conjugate coding 기반 양자 서명에 Merkle 트리를 적용한 CQS 기법을 제안했다. 이 기법은 CA가 검증 정보를 관리하는 공인인증 체계를 구축할 수 있는 가능성을 높였다. 그러므로 이 기법은 서명 생성자가 Alice 임을 명확하게 하는 메시지 기원 인증, Alice가 서명을 전송했다는 사실을 확정하는 서명 생성자의 부인방지, Bob이 서명을 수신했다는 사실을 확정하는 서명 수신자의 부인방지 등의 서명의 필수기능들을 제공한다. 그리고 제안하는 기법에 적합한 서명 검증의 기준인 QSAR을 제안했다. QSAR은 개인 키와 양자 공개 키 측정결과의 Hamming distance 기댓값으로 CQS에서 사용하는 서명쌍의 메시지 무결성을 검증하는 중요한 지표가 될 것으로 기대한다.

Acknowledgment

This work was supported by the ICT R&D programs of MSIP/IITP (Grant No. B0101-16-1355), the KIST research program (Grant No. 2E27231), and the KIST-ETRI research program (Grant No. 2V05340).

[References]

[1] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, Handbook of Applied Cryptography(CRC Press, Boca Raton, 1996).

[2] D. R. Stinson, Cryptography: Theory and Practice(CRC Press, Boca Raton, 2005)

[3] P. Shor, in Proceedings of the 35nd Annual Symposium on Foundations of Computer Science, (IEEE Computer Society Press, New York, 1994).

[4] C. H. Bennett and G. Brassard, in Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing, Bangalore, India, (IEEE, New York, 1984), pp. 175 - 179.

[5] G. Zeng and C. H. Keitel, Phys. Rev. A 65, 042312 (2002).

[6] H. Lee, C. Hong, H. Kim, J. Lim, and H. J. Yang, Phys. Lett. A 321, 295 (2004).

[7] Q. Li, W. H. Chan, and D.-Y. Long, Phys. Rev. A 79, 054307 (2009).

[8] X. Zou and D. Qiu, Phys. Rev. A 82, 042325 (2010).

[9] D. Gottesman and I. Chuang, arXiv:quant-ph/0105032.

[10] P. J. Clarke, R. J. Collins, V. Dunjko, E. Andersson, J. Jeffers, and G. S. Buller, Nat. Commun. 3, 1174 (2012).

[11] V. Dunjko, P. Wallden, and E. Andersson, Phys. Rev. Lett. 112, 040502 (2014).

[12] R. J. Collins, R. J. Donaldson, V. Dunjko, P. Wallden, P. J. Clarke, E. Andersson, J. Jeffers, and G. S. Buller, Phys. Rev. Lett. 113, 040502 (2014).

[13] P. Wallden, V. Dunjko, A. Kent, and E. Andersson, Phys. Rev. A 91, 042304 (2015).

[14] R. J. Donaldson, R. J. Collins, K. Kleczkowska, R. Amiri, P. Wallden, V. Dunjko, J. Jeffers, E. Andersson, and G. S. Buller, Phys. Rev. A 93, 012329 (2016).

[15] H. L. Yin, Y. Fu, and Z. B. Chen, Phys. Rev. A 93, 032316 (2016).

[16] R. Amiri, P. Wallden, A. Kent, and E. Andersson, Phys. Rev. A 93, 032325 (2016).

[17] R. J. Collins, R. Amiri, M. Fujiwara, T. Honjo, K. Shimizu, K. Tamaki, M. Takeoka, E. Andersson, G. S. Buller, and M. Sasaki, Opt. Lett. 41, 4883 (2016).

[18] H. L. Yin, Y. Fu, H. Liu, Q. J. Tang, J. Wang, L. X. You, W. J. Zhang, S. J. Chen, Z. Wang, Q. Zhang, T. Y. Chen, Z. B. Chen, and J. W. Pan, Phys. Rev. A 95, 032334 (2017).

[19] RC Merkle, “A certified digital signature.” Conference on the Theory and Application of Cryptology. Springer, New York, NY, 1989.

[20] I.T.U. Telecommunication, X.509, In: Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks, 1996.

[21] S. Wiesner, ACM Sigact News 15.178 (1983).

[22] D. J. Bernstein, J. Buchmann, E. Dahmen, eds. Post-quantum cryptography. Springer Science & Business Media, 2009.

[23] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, Rev. Mod. Phys. 74, 145 (2002).

2017 한국정보보호학회 동계학술대회

B4 블록체인(II)

신 상 욱 (부경대)

- **가상화폐 익명화 기술 연구 동향**
  이지환, 홍영기, 허준범 (고려대학교)
- **블록체인 기반 부동산 원장에 대한 법적 검토**
  김미연, 황동현, 염흥열 (순천향대학교)
- **비트코인 익명성 동향 분석**
  김승민, 이지혜, 김윤정 (서울여자대학교)
- **블록체인 기반 에너지 거래 모델 연구**
  김승리, 이정혁, 구슬예, 오현옥 (한양대학교), 김지혜 (국민대학교)

가상화폐 익명화 기술 연구 동향*

이지환, 홍영기, 허준범
고려대학교 컴퓨터학과

Survey of Cryptocurrency De-anonymization Techniques

Jihwan Lee, Younggee Hong, Junbeom Hur
Department of Computer Science and Engineering, Korea University

요 약

전통적인 금융 시스템에서는 금융 거래의 기록이 은행과 같이 오직 중앙집권적인 기관 내부에서만 처리되기 때문에 사용자의 익명성과 거래의 프라이버시가 보장된다. 그러나 최근 가파른 성장을 거듭하고 있는 비트코인을 비롯한 암호화폐 프로토콜에서는 전통적인 중앙집권적인 기관 없이 모든 거래가 공공의 블록체인(public blockchain)이라는 분산원장에 기록되기 때문에 모든 거래의 정보가 누구에게나 노출된다. 이를 감추기 위해 암호화폐 프로토콜 상에서 사용자의 익명성과 거래의 프라이버시를 보장할 수 있는 많은 기술들이 제안되어 왔다.본 논문에서 거래를 할 때 익명성과 프라이버시를 보장하는 방안들을 살펴보고 그 특징들을 비교 분석한다.

I. 서론

비트코인(Bitcoin)은 2008년 사토시 나카모토(Satoshi Nakamoto)가 최초로 고안한 암호화폐(cryptocurrency)이다. 비트코인과 같은 암호화폐는 모든 거래를 블록체인(blockchain)이라는 분산원장에 기록을 한다. 비트코인 시스템은 분산 합의 프로토콜(decentralized consensus protocol)에 기반하며 이 프로토콜은 경제적인 유인(incentive)에 따라 행동하는 채굴자(miner)들에 의해 작동한다. 2009년 이래 비트코인을 비롯한 암호화폐는 매년 폭발적인 성장을 계속해 왔고 암호화폐의 폭발적인 성장에 따라 시스템 자체의 취약성에 대한 관심이 증가했으며 그에 따른 연구가 활발히 진행되었다.

사토시 나카모토가 비트코인 시스템을 최초 제안한 백서(white paper)[1]에서 유저(user)는 가상의 비트코인 주소(pseudonymous address)를 사용하고 한 유저는 새로운 여러 개의 주소를 생성할 수 있기 때문에 시스템이 각 유저의 프라이버시와 거래의 익명성을 보장할 수 있다고 하였다. 그러나 최초 고안된 비트코인 시스템은 'taint analysis'와 여러 블록체인 분석 기법, 각 노드의 IP주소 추적 등 여러 가지 프라이버시 관련 공격 시나리오에 취약하다. 이에 따라 비트코인을 비롯한 암호화폐 시스템에서 프라이버시와 익명성을 향상시킬 수 있는 많은 기법들이 제안되었다. 이 기법들은 다음과 같이 세 카테고리로 분류할 수 있다.

- P2P 믹싱 프로토콜 (Peer-to-peer mixing protocol)
- 분산 믹싱 프로토콜 (Distributed mixing protocol)
- 알트코인 (Altcoin)

* 이 논문은 2017년도 과학기술정보통신부의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임. (No.2017-0-00395, 다크웹 스캐닝 기술 기반의 정보 수집 분석 시스템개발)

본 논문에서는 비트코인과 같은 암호화폐 시스템에서 익명성과 프라이버시를 향상시키는 여러 기술들을 살펴보고 그 특징들을 비교 분석한다.

II. P2P 믹싱 프로토콜(Peer-to-Peer Mixing Protocol)

P2P 믹싱 프로토콜에서 유저는 믿을 수 있는 써드파티(trusted third party)없이 그들의 거래를 네트워크 상에 전파한다. P2P 믹싱 프로토콜의 핵심 아이디어는 프로토콜 참여자의 집합 내에서 그들의 코인 소유권을 서로 바꿈으로서 누가 특정 코인을 보낸 것 인지를 숨기는 것이다. 그리하여 공격자가 특정 거래와 특정 유저를 연결할 수 없게끔 하는 것이 P2P 믹싱 프로토콜의 목적이다. P2P 믹싱 프로토콜은 지불을 위해 오직 하나의 거래만을 필요로 하는 장점이 있다.

2.1 CoinJoin[2]

CoinJoin은 프라이버시 보장과 코인 도난 방지를 목적으로 하는 최초의 P2P 믹싱 프로토콜이다. 유저들의 집합은 동일한 거래의 입력값과 출력값을 설정하여 같은 금액의 거래를 생성하고 소유권을 교환함으로서 외부의 공격자가 어떤 출력값이 어떤 입력 값에 대해 대응이 되는지 알 수 없게 만든다. 이는 외부 연결가능성(external linkability)문제를 해결한다. 유저는 원하는 주소에 원하는 금액만큼 입력으로 들어왔을 때만 거래를 승인함으로써 코인 도난을 방지한다. 그러나 CoinJoin은 참여자들 중 공격자가 있을 경우 내부 연결가능성(internal linkability)이 존재하는 한계를 지닌다또 한 모든 참여자들의 서명을 필요로 하기 때문에 확장성에 한계가 있으며 참여자들이 서명을 해야만 거래가 승인이 되기 때문에 Dos공격에 취약하다.

2.2 CoinShuffle[3]

앞서 언급된 내부 연결가능성 문제와 Dos공격에 대해 대처하기 위해 추가적으로 공개키 암호시스템과 서명기법 등을 추가 적용한 CoinShuffle이 제안되었다. CoinShuffle에서 참여자들 또 한 내부적으로 입력 값(input)과 출력값(output)의 연결가능성을 알 수 없으며 프로토콜의 디자인에 따라 악의적인 유저가 있다면 비난(blame)의 과정을 거친 후 그 유저를 제외 하고 프로토콜을 재개한다. 그러나 CoinShuffle은 믹싱이 한 블록 안에서 이루어지기 때문에 익명성 집합(anonymity set)이 작다는 한계를 지닌다.

2.3 ValueShuffle[4]

CoinShuffle++를 기반으로 한 프로토콜로서 지불인의 익명성(payer anonymity), 수취인의 익명성(payee anonymity) 그리고 거래 금액의 프라이버시(payment value privacy) 보장을 목적으로 한다. 이를 위해 ValueShuffle은 Confidential Transaction(CT)[5]를 통해 거래 금액의 프라이버시를, Stealth Address[6]를 통해 수취인의 익명성을, 그리고 CoinJoin 믹싱 기술을 통해 지불인의 익명성을 보장한다.

III. 분산 믹싱 프로토콜(Distributed Mixing Protocol)

분산 믹싱 프로토콜은 '믹서(mixer)' 는 '믹스(mix)'라 불리는 써드파티(third-party)가 존재 하여 일반 유저들에게 소정의 이용료(mixing fee)를 받고 믹싱서비스를 제공한다.

3.1 Mixcoin[7]

유저들은 믹스(mix)에게 표준크기의 거래를 믹싱을 원하는 금액의 코인을 함께 제공하면 믹스는 같은 금액의 다른 유저의 코인으로 바꿔 준다. 따라서 외부에서 특정한 유저와 특정한 코인을 연결할 수 없게 된다(external unlinkability). 그러나 믹싱을 수행하는 믹스는 내부적으로 어떤 유저가 어떤 거래와 연관이 있는지 연결할 수 있는 문제(internal linkability)이 있다또 한 믹스는 코인을 훔칠 수 있다. Mixcoin은 믹스가 코인을 훔쳤다는 것을 암호학적인 기술로 증명할 수 있게 해준다평판(reputation) 시스템을 사용하여, 믹스들은 경제적인 유인을 위해 행동한다는 가정 아래 만약 믹

스가 코인을 훔치거나 시스템을 악의적으로 위협한다면 네트워크 전체에 이 사실을 전파하여 다른 유저들이 그 믹스의 믹싱 서비스를 이용하지 않게 알린다는 것이다. 하지만 여전히 믹스는 내부적인 연결가능성을 알 수 있고 암호화폐 특성상 일단 코인의 도난 후에는 복구를 하지 못한다. 사후에 전체 네트워크에 해당 사실을 전파하여 그 행위를 알릴 수 있지만 사전예방이 불가능하다는 한계를 지닌다.

3.2 Blindcoin[8]

Blindcoin은 Mixcoin이 지닌 내부적 연결가능성(internal linkability)의 한계를 개선한 것으로서 은닉 서명(blind signature)을 사용한다. 유저의 입력 값과 출력 값을 은닉함으로서 내부적으로 믹스가 유저와 코인을 연결하지 못하게 한다. 그러나 여전히 믹스가 코인을 훔칠 수 있는 가능성(coin theft)은 존재하고 은닉 서명을 이용하기 위해 두 개의 별도의 거래(transaction)가 더 필요하여 비용이 더 든다는 한계를 지닌다.

3.3 TumbleBit[9]

TumbleBit에서 각 유저는 'Tumbler'라는 써드파티를 통해 빠르게 블록체인 외부에서(off-blockchain) 거래를 할 수 있다. 믹스의 기능을 수행하는 Tumbler는 거래의 수신자와 송신자가 각각 누구인지 대응시키지 못한다. TumbleBit은 유저가 믹스인 Tumbler을 신뢰하지 않아도 되는 최초의 프로토콜이다. 저자[9]는 800명의 유저가 TumbleBit을 통해서 믹싱이 가능함을 보였다. TumbleBit은 강한 익명성과 높은 확장성(scalability)을 제공하고 코인 도난이 불가능하다(coin theft resistance). TumbleBit은 기존 비트코인 시스템에 적용할 수 있다(Bitcoin-compatible). 그러나 여전히 TumbleBit은 적어도 두 개의 연속적인 비트코인 거래를 필요로 한다.

IV. 알트코인(Altcoin)

믹싱과 같이 기존 비트코인 시스템 상에서 거래 익명성과 유저 프라이버시를 향상시키는 기술 대신 완전히 새로운 알트코인(Altcoin) 프로토콜도 많이 제안되고 있다.

4.1 Zerocoin(Zerocash)[10][11]

Zerocoin은 비트코인과 달리 영지식 증명(zero-knowledge proof)을 이용하여 익명성을 보장한다. 유저는 특정 금액의 코인을 같은 금액의 다른 코인과 교환을 함으로써 익명성을 얻는다. 그러나 앞서 언급된 다른 기술들과 달리 유저는 교환을 위해 믹스에게 요구를 하지 않고 유저 스스로가 Zerocoin프로토콜을 통해 같은 양의 금액을 소유한다는 것을 증명하고 같은 금액의 코인을 교환할 수 있다. Zerocash는 Zerocoin의 연장으로서 기능성과 효율성 측면에서 향상된 버전의 영지식 증명인 SNARKs을 사용하여 거래 금액의 양, 수신자 주소 등 거래의 추가적인 정보를 숨겨 프라이버시를 향상시킨다. 그러나 SNARKs는 구현하는데 있어서 기존 프로토콜을 변경해야 하고 블록체인 프루닝(blockchain pruning)이 불가능 하다는 한계가 있다.

4.2 MimbleWimble[7]

MimbleWimble은 Confidential Transaction(CT)을 지원하는 알트코인이다. CT를 이용함으로서 프라이버시를 향상시키며 MimbleWimble은 여러 거래들을 각각이 소속된 블록과 상관없이 통합하여 처리를 하기 때문에 거래 그래프 분석(transaction graph analysis)이 불가능하며 블록체인 프루닝을 통해서 블록체인의 확장성(scalability)에 기여를 한다. 그러나 단순한 거래들의 통합은 채굴자와 같이 통합을 하는 주체에 대해서는 여전히 입력 값 출력 값 간의 연결가능성(input-output unlinkability) 문제를 노출하고 스크립트(script)를 지원하지 않는다는 한계를 지닌다.

[표 1] **익명성 및 프라이버시 향상 기술 비교**

	Type	Threat model	Bitcoin-Compatibility	Theft
CoinJoin	P2P	DoS attack Sybil attack	O	X

Coin-Shuffle	P2P	Sybil atttack	O	X
Value-Shuffle	P2P	DoS attack Sybil attack	O	X
Mixcoin	Distrib uted	internal linkability	O	O
Blindcoin	Distrib uted	coin theft	O	O
TumbleBit	Distrib uted	-	O	X
Zerocoin/ Zerocash	Altcoin	-	X	X
Mimble-Wimble	Altcoin	Dos attack	X	X

V.결론

매년 폭발적인 성장을 거듭한 비트코인을 비롯한 암호화폐의 성장세만큼 보안 및 프라이버시 이슈에 대한 연구도 활발하게 진행되고 있다. 암호화폐는 블록체인을 기반으로 하고, 블록체인은 본질적으로 모든 것을 공개하는 장부이기 때문에 유저의 익명성과 거래의 프라이버시가 위협받을 수 있다. 암호화폐가 거래의 지불 수단으로 사용되기 위해서는 익명성과 프라이버시의 보장은 필수적이다. 이러한 프라이버시와 익명성을 보장하기 위해 여러 연구가 진행되어왔고 많은 기법들이 제안되었다본 논문에서는 암호화폐 프로토콜 상에서 익명성과 프라이버시를 보장할 수 있는 여러 기법들을 비교 분석하였다.

[참고문헌]

[1] S.Nakamoto, Bitcoin: A peer-to-peer electronic cash system, 2008

[2] G.Maxwell, CoinJoin: Bitcoin privacy for the real world, https://bitcointalk.org/index.php?topic=279249, 2013

[3] T.Ruffing, P.Moreno-Sanchez, A.Kate, CoinShuffle: Practical decentralized coin mixing for Bitcoin, ESORICS, 2014

[4] T.Ruffing, P.Moreno-Sanchez, Mixing confidential transactions: Comprehensive transaction privacy for Bitcoin, Financial Cryptography, 2017

[5] G.Maxwell, Confidential transactions, https://people.xiph.org/~greg/confidential_values.txt, 2015

[6] P.Todd, Stealth addresses, https://www.mail_archive.com/bitcoin-developement@lists.sourceforge.net/msg03613.html

[7] J.Bonneau, A.Narayanan, A.Miller, J.Clark, J.A.Kroll, and E.W.Felton, Mixcoin: Anonymity for Bitcoin with Accountable Mixes, Springer Berlin Heidelberg, 2014

[8] L.Valenta and B.Rowan, Blindcoin: Blinded, accountable mixes for bitcoin, Financial Cryptography Workshops, 2015

[9] E.Heilman, L.Alshenibr, F.Baldimtsi, A.Scafuro, and S.Goldberg, TumbleBit: An untrusted bitcoin-compatible anonymous payment hub, Cryptology ePrint Archive, Report 2016/575, 2016

[10] I.Miers, C.Garman, M.Green, and A.D.Rubin, Zerocoin: Anonymous distributed e-cash from bitcoin, 2013 IEEE Symposium on Security and Privacy, 2013

[11] E.B. Sasson, A.Chiesa, C.Garman, M.Green, I.Miers, E.Tromer, and M.Virza, Zerocash: Decentralized anonymous payments from bitcoin, 2014 IEEE Symposium on Security and Privacy, 2014

[12] T.Jedusor, Mimblewimble, https://scalingbitcoin.org/papers/mimblewimble.txt, 2016

블록체인 기반 부동산 원장에 대한 법적 검토*

김미연*, 황동현*, 염흥열**

*순천향대학교 융합서비스보안학과, **순천향대학교 정보보호학과

Legal Examination on Real Estate Ledger based on Blockchain*

Mee Yeon Kim*, Dong-Hyun Hwang*, Heung Youl Youm**

*Department of Convergence Service Security, **Department of Information Security Engineering, SoonChunHyang University.

요 약

최근 각광받는 블록체인 기술은 다양한 산업분야에 적용이 가능할 것으로 예상되고 있다. 그중 부동산 산업에 블록체인을 도입했을 시, 그 기대와 장점은 부각되고 있지만 실질적인 국내연구는 미미한 수준이다. 하지만 해외에서는 블록체인을 부동산 분야에 적용하기 위한 다양한 노력을 하고 있으며 실질적인 도입을 위한 플랫폼의 개발 및 시험도입을 성공적으로 마친 상태이다. 이에 본 논문에서는 블록체인 기반의 부동산 원장기술의 국내 도입을 위해 해외 사례를 기반으로 개인정보보호 측면에서의 준거성을 확보할 수 있게 법적 측면에서 검토하였다.

I. 서론

현재 가장 주목받는 기술인 블록체인은 다양한 산업에 활용 가능할 것으로 예상되고 있다. 그 중 부동산도 활용 가능성이 기대되는 산업분야로 주목받고 있지만 기대와 장점만 부각되고 실제 활용 연구나 사례는 미미한 수준이다.[1] 또한 블록체인이 갖는 투명성이라는 특징 때문에 개인정보보호 유출에 대한 우려가 제기되고 있다. 한 번 기록되면 삭제되기 힘든 특징 때문에 개인정보가 블록체인에 등록되면 유출, 악용 등의 위험에 취약해진다. 그래서 이런 점을 보완하고자 본 논문은 기존에 제시된 블록체인 기반 부동산 원장의 개인정보 취급에 대해 조사하고, 국내 개인정보보호법 적용 시의 위험성을 검토하고자 한다.

* 본 연구는 미래창조과학부 및 한국인터넷진흥원의 “고용계약형 정보보호 석사과정 지원사업”의 연구결과로 수행되었음

II. 부동산 산업에서의 블록체인 활용

2.1 블록체인 도입 시 장점

블록체인을 부동산 산업에 도입할 경우 여러 장점들을 얻을 수 있다. 가장 손꼽히는 장점 3가지 중 첫 번째는 투명성 제고이다. 자산의 거래내역을 살펴보는 등 거래 관련 데이터에 쉽게 접근가능하고, 가격 비교도 수월하다. 두 번째 장점으로는 사기 위험의 축소이다. 블록체인으로 거래내역을 추적할 수 있기 때문에 판매자, 구매자, 그간의 구매자에 대한 신원확인이 가능해지며 사기의 위험을 줄일 수 있다. 또한 블록체인에는 전자 소유 인정 문서가 같이 등록되며, 한 번 등록되면 복제가 불가능하여 위조도 힘들게 된다. 세 번째 장점은 부동산 거래 프로세스의 가속화 및 단축이다. 중개자의 검증이 간소화되거나 사라지게 되고 거래는 smart contract를 이용하여 자동화 된다. 결제는 코인을 활용하여 진행할 수 있어 복잡한 금융 거래를 단축할 수 있다.[2][3]

2.2 해외 사례

2.2.1 ATLANT 모델

ATLANT는 이더리움 블록체인에 기반한 부동산 플랫폼 서비스를 제공하는 ICO (Initial Coin Offering)기업이다. 크게 두 가지 서비스를 제공하는 데 하나는 재산 소유권을 토큰화하여 부동산 재산을 토큰으로 판매하는 것이며, 다른 하나는 P2P 렌탈 서비스로, 수수료를 줄이는 반면 거짓 리뷰의 가능성을 최소화 하는 서비스를 제공한다고 주장한다.[4] 여기서 P2P 렌탈은 전세계약, 월세계약과 같은 임대차 부동산 계약보다는 숙박 서비스로 해석되어 본 논문에서는 다루지 않는다.

ATLANT 플랫폼에서는 팔고자 하는 자산은 초기 자산 토큰 입찰(Property token offering, PTO)을 통해 플랫폼에 등록한다. PTO는 smart contract를 생성하여 자산을 토큰화하는 것으로, 토큰이 된 자산을 팔거나 토큰을 통해 건물을 건설하기 위한 펀드 자금을 모을 수 있다.

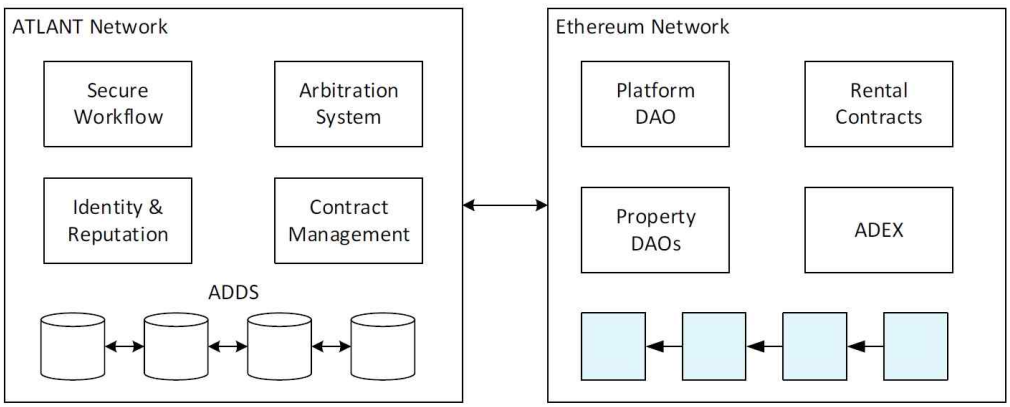
플랫폼 노드들은 자기 자산을 팔고자 하는 소유자를 찾고, 소유자의 자산을 새로 플랫폼에 등록할 것인지 말 것인지를 투표로 결정한다. 이후, 등록 시 자산에 대한 로펌과 관리 회사를 투표로 결정한다. 로펌은 법문서를 검증하고, 자산에 적용되는 문서에 디지털 서명을 한다. 플랫폼에 자산을 등록하기 위해 몇 가지 요건을 충족시켜야 하는데 법문서를 해당 사법권에 전송해서 승인받아야 하고, 가격이 일정 이상이여야 하며, 유치권이 없어야 한다. 해당 조건이 충족되면 smart contract를 생성하여 자산을 토큰화하며, 토큰과 ETH나 ERC20을 교환하는 EVM 거래로 발행된다. 이때 자산과 관련된 법적 문서(Ricardian contract, RC)는 ATLANT의 데이터 저장소에 보관하고 해시만 이더리움 블록체인에 기록한다. 그러면 PTO가 생성되어 토큰들이 생성된다. 생성된 자산 토큰은 자유롭게 거래할 수 있다.

ATLANT 플랫폼은 독립된 P2P 네트워크에 부동산 재산을 디지털화 시키는데 특화된 프로토콜을 사용한다. 그리고 자산관리, 중재 등을 수행하며, ATLANT의 프로토콜이 아틀란트 독립 네트워크와 이더리움 기반 smart contract를 연결시킨다. 이더리움 네트워크에선 smart contract를 이용하여 자산 토큰화, 거래 등을 수행한다.[5]

2.2.2 Velox.RE 모델

Velox.RE는 Chicago's cook county Recorder of Deeds와 협업하여 비트코인 블록체인 기반하는 부동산 소유권 이전 파일럿 프로그램을 개발하였다. 부동산을 토큰화하여 재산의 권리를 이전하고 내역을 공공 기록에 등록한다. 이때 거래는 실제 법을 준수한다고 말한다.[6][7]

프로그램은 colored coin이라는 특수 비트코인을 생성하여 부동산을 디지털화하고 전송한다. 이는 재산에 대한 일련의 권리를 디지털 문서화하고 그것을 양도인에게서 양수인에게 이전한다. 우선 양도인은 velox.RE의 소프트웨어



<그림 1> ATLANT 플랫폼의 기술적 개요

로 colored coin을 생성한다. 양도인과 양수인은 온라인으로 공증받을 내용을 협의하고 velox.RE 플랫폼에서 이전될 재산 및 합의서를 확인한다. 이전 사항이 모두 확정되고 상호 확인 후 양도인과 양수인은 디지털 문서에 전자서명을 한다. 이는 소유권 이전을 승인한다는 것에 대한 증거가 된다. 그리고 공증인이 이를 공증한다. 양도인은 colored coin을 양수인에게 전송하여 소유권을 이전한다. colored coin 거래 기록이 비트코인 네트워크에 전송 및 블록에 등록된다. 또한 Velox.RE 시스템은 거래내역을 종이 문서화하여 공문서로 보관될 수 있도록 만든다.[7]

[표 1] 각 단계에서의 메타데이터 필드

메타데이터 필드	내용
colored coin 메타데이터	양도인 이름, 법적 재산 기술서, 재산 인덱스 번호, 주소, 권리 등
colored coin 거래	양도 선언, 양수인 이름, 권리 문서 종류 등
블록체인의 종이 문서 버전	거래 아이디, 거래 해시, 블록 높이, 타임스탬프, QR코드 링크 등

III. 국내 도입 시 법적 위험성 검토

3.1 개인정보보호법 측면에서의 검토

본 장에서는 각 모델의 개인정보를 다루는 방식을 국내 개인정보보호법 측면에서 검토하였다. ATLANT 모델의 경우, 법적 문서인 RC를 ATLANT의 데이터 저장소에 보관하고 RC의 해시값만 smart contract에 입력한다.[5]

Velox.RE의 경우, 개인정보가 포함된 메타데이터는 colored coin의 데이터 저장소에 별도로 저장한다. 그 전체 데이터를 해시한 값을 비트코인 거래의 op_return 값에 입력하여 참조를 하되 개인정보에 대한 내용을 알 수 없게 하였다.[7]

여기서 블록체인에 기록되는 해시값은 개인정보일까? 국내 개인정보보호법에 따르면 “개인정보”란 살아 있는 개인에 관한 정보로서 성명, 주민등록번호 및 영상 등을 통하여 개인을 알아볼 수 있는 정보(해당 정보만으로는 특정 개인을 알아볼 수 없더라도 다른 정보와 쉽게 결합하여 알아볼 수 있는 것을 포함한다)를 말한다.[8] 이때 ‘쉽게 결합하여’란 결합 대상의 정보에 대해 ‘입수 가능성’이 존재해야 하고, ‘결합가능성’이 높다는 것으로 풀이한다. ‘입수 가능성’과 ‘결합가능성’은 표2에 설명된 바와 같다.[9]

그렇다면 해시를 통해 개인정보를 쉽게 입수할 수 있고, 결합가능한지 따져보아야 할 것이다. ATLANT, Velox.RE 두 모델 전부 별도로 데이터 저장소를 두고 개인정보가 담긴 법적 계약서를 저장했다. 또한 블록체인에 참조하는 식별자는 해시를 통해 일방향 암호화를 적용하여 해시값만으로는 개인정보를 알 수 없을 뿐더러, 각각의 개인정보 필드에 대한 해시가 아닌 전체 데이터 문서의 해시로 개인정보와의 직접적인 연결성을 갖지 않는다. 이 때 데이터 저장소에 대해 접근통제와 같은 적절한 보호조치를 취할 경우에만 해시를 통해 개인정보를 입수할 가능성이 없어지므로 해시값이 블록체인에 등재되어도 개인정보가 아니여서 문제가 되지 않는다. 제시된 2개의 모델 경우, 데이터 저장소에 대한 설명이 없어 적절한 보호조치를 취하고 있는지 정확히 알 수 없었다.

[표 2] 각 모델의 입수가능성 및 결합가능성 충족여부

		입수가능성	결합가능성
설명		두 종 이상의 정보를 결합하기 위해서는 결합에 필요한 정보에 합법적으로 접근, 입수할 수 있어야 함	현재의 기술 수준을 고려하여 비용이나 노력이 비합리적으로 수반되지 않아야 함
모델	ATLANT	자산과 관련된 법적 문서를 ATLANT의 데이터 저장소에 보관	Smart contract에 자산과 관련된 법적 문서 전체를 해시한 값을 입력
	Velox. RE	Colored Coin 내의 별도의 데이터 저장소에 법적 문서 보관	비트코인 거래의 op_return 데이터 필드에 문서 전체를 해시한 값 입력
충족여부		알 수 없음	불가능

3.2 개선사항

본 장에서는 확실한 준거성을 보장하기 위해 모델 도입 시 위배될 수 있는 몇몇의 유의사항을 식별하고, 개선사항을 제시하고자 한다.

우선 블록체인 기반 부동산 플랫폼 서비스를 운영할 시, 제시된 모델과 같이 블록체인에는 해시값을 등록하고, 개인정보가 표기된 법적 문서를 비롯한 개인정보는 블록체인과 별도의 저장소를 두어야 한다. 또한 개인정보를 보관하는 저장소는 개인정보보법 제 29조에 따라 안전성 확보에 필요한 기술적, 관리적 및 물리적 조치를 하여야 한다.[10] 구체적으로는 고시 제3조~13조(안전조치 기준 적용, 내부 관리계획의 수립•시행, 접근 권한의 관리, 접근통제, 개인정보의 암호화, 접속기록의 보관 및 점검, 관리용 단말기의 안전조치, 물리적 안전조치, 재해•재난 대비 안전조치, 개인정보의 파기 등)에 따라 저장소를 운영해야 한다.[11] 또한 저장소에 보관된 계약서를 재계약, 변경, 제공 등의 목적으로 열람을 요청할 때는 제17조, 제18조에 따라 정보주체의 동의를 다시 받아야 한다.[12]

둘째로 매물을 플랫폼에 등록할 때, 민감정보에 대한 별도의 동의를 받아야 한다. 매물 및 내부 사진을 같이 등록할 때, 경우에 따라 사진을 통해 개인의 사생활을 침해받을 수 있다. 이 경우 내부 사진은 민감정보로 볼 수 있기 때문에, 제 23조에 따라 매물 및 내부 사진을 같이 올리는 경우 별도의 동의를 받아야 한다.[13]

IV. 결론

본 논문에서는 블록체인 기반 부동산 원장에 대하여 조사하고 그를 국내 도입할 경우, 확인해야할 법적 준수 요건을 살펴보았다. 우선 블록체인을 부동산 원장에 도입하면 투명성 재고, 사기의 위험 축소, 부동산 거래 프로세스의 가속화 및 단축 등 3가지 장점을 조사했다. 그 뒤 현재 해외에서 실험 중에 있는 블록체인 기반 부동산 플랫폼 모델, ATLANTA와 Velox.RE 두 가지 예시를 살펴본 후, 국내 도입할 경우 개인정보보호법의 준거성에 위반될 수 있는 법적 요건을 각 모델에 따라 검토하였다. 향후 이 요건에 따른 모델을 연구하여 국내에 구현 시, 법적으로 참고할 수 있기를 바란다.

[참고문헌]

[1] Moody's Investors Service, Robust, cost-effective applications key to unlocking blockchain's potential credit benefits, 2016.7.

[2] Wilfrid Donkers, Jan-Willem Santing, Blockchain: the next game changer in real estate?, Deloitte, 2016.03.03.

[3] Don Oparah, 3 Ways That The Blockchain Will Change The Real Estate Market, TechCrunch, 2016.02.06.

[4] ATLANT's webpage, https://atlant.io/

[5] Julian Svirsky, Neil Mohinani, Denis Donin, ATLANT Platform White Paper v0.97, ATLANT, 2017.07.29.

[6] Kyle Torpey, Chicago's Cook County to Test Bitcoin Blockchain-Based Property Title Transfer, Bitcoin Magazine, 2016.10.06.

[7] Ragnar Lifthrasir, Permissionless Real Estate Title Transfers on the Bitcoin Blockchain in the USA! - Cook County Blockchain Pilot Program Report, Linkedin, 2017.06.28.

[8] 개인정보보호법 제1장 제2조 제1호

[9] 행정자치부, 개인정보보호 법령 및 지침•고시 해설, 2016.12.

[10] 개인정보보호법 제4장 제29조

[11] 개인정보보호법 시행령 제5장 제30조 제3호 고시 3조~13조

[12] 개인정보보호법 제3장 제1절 제17조, 제18조

[13] 개인정보보호법 제3장 제2절 제23조

비트코인 익명성 동향 분석*)

김승민, 이지혜, 김윤정
서울여자대학교 정보보호학과

Analysis of Bitcoin Anonymity Trend

Seung-Min Kim, Jihye Lee, Yoonjeong Kim
Department of Information Security, Seoul Women's University.

요 약

2008년 익명의 프로그래머인 사토시 나카모토(Satoshi Nakamoto)에 의해 비트코인(Bitcoin)이라는 신뢰성 있는 제 3자가 없이, 피어 투 피어 방식을 사용한 전자 화폐가 제안되었다. 이러한 피어 투 피어 전자화폐인 비트코인에서 익명성 문제는 중요한 주제다. 비트코인의 각 거래들은 블록 단위로 묶여 있으며 각 블록들은 사슬처럼 연결되어 있다. 이를 블록체인(Blockchain)이라 하며, 모든 비트코인 거래는 블록체인이라는 거래 장부를 통해 투명하게 공개되기 때문에 비트코인 네트워크상의 모든 사용자들이 모든 거래정보를 갖고 있다고 할 수 있다. 이러한 이유로 신분 추정이 가능한 주소를 한 번이라도 거치게 된다면, 비트코인 소유자 정보와 거래 기록까지 추정하는 것이 가능하다. 우리는 이 논문에서 기존의 비트코인 익명성에 관한 연구와 현황을 분석한다.

I. 서론

비트코인(Bitcoin)이란 2008년 사토시 나카모토(Satoshi Nakamoto)라는 익명의 프로그래머 혹은 집단에 의해 처음 제안된 피어 투 피어(peer-to-peer) 방식을 사용한 전자 화폐이다 [1]. 비트코인이 모든 거래를 공개적으로 드러내지만, 비트코인 사용자들 각각의 공개키는 익명으로 유지함으로써 사용자들의 개인 정보보호를 유지할 수 있다. 일반 비트코인 사용자들은 누군가가 다른 사용자와 거래를 하고 있지만 거래를 연결하는 정보는 알 수 없다. 따라서 사용자 정보인 개인키 소유주가 밝혀질 경우 링크를 통해 동일한 소유자의 다른 거래가 노출될 수 있는 위험이 있다.

본 논문에서는 기존의 비트코인 익명성 주제를 확장한 연구를 소개한다 [2]. 이 논문의 Ⅱ장에서는 비트코인과 비트코인 익명성을 설명하고, Ⅲ 장에서는 기존의 비트코인 익명성 연구를 다루며, Ⅳ 장에서는 익명성에 대한 기타 현황을 기술한다.

II. 비트코인 익명성

2.1 비트코인

비트코인은 피어 투 피어 파일 공유와 공개키 암호화를 결합한 새로운 형태의 전자 화폐로, 중앙서버 또는 제어 지점이 없이 분산화 된 시스템 구조로 이루어져 있다 [3]. 비트코인의 각 거래들은 블록 단위로 묶여 있다 [4]. 채굴자들은 이전 블록까지의 거래 전부가 포함된 새 블록을 형성한다. 이를 블록체인(block chain)이라 하는데, 모든 비트코인 거래에는 블록체인에 포함되기 위해 유효한 서명이 필요하다. 각 암호키 소유자들은 그 전까지의 거래 내역에 다음 소유자의 공개키를 덧붙인 뒤에 자신의 비밀키로 암호화하는 디지털 서명을 하고 넘긴다. 돈을 받는 사람은 서명 소유자들의 체인과, 서명들을 검증할 수 있다 [1].

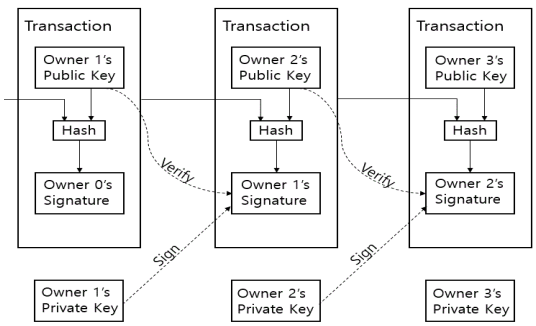


그림 1. Bitcoin electronic coin: a chain of a digital signature [1]

*) 이 논문은 미래창조과학부 및 정보통신기술진흥센터의 대학ICT연구센터육성 지원(IITP-2017-2014-0-00636)과 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원(No. NRF-2016R1A2B4011984)을 받아 수행된 연구임.

2.2 비트코인 익명성

비트코인에서 익명성은 중요한 주제이다 [5]. 비트코인은 익명성을 갖고 있으며 동시에, 투명성을 갖고 있기 때문에, 비트코인이 완전한 익명성을 갖고 있다고 할 수 없다 [6]. 비트코인 네트워크 참가자들 모두는 임의의 암호화된 공개키-비밀키 키쌍이 저장된 지갑을 갖는다. 비트코인 주소에는 소유자가 누구인지에 대한 정보가 포함되어 있지 않아 익명성이 보장될 수 있다. 그러나 모든 비트코인 거래는 블록체인이라는 거래 장부를 통해 투명하게 공개되기 때문에 신분 추정이 가능한 주소를 한 번이라도 거치게 된다면, 비트코인 소유자 정보와 거래 기록까지 추정하는 것이 가능하다. 공격자는 사용자와 공개키 사이를 일대 다로 매핑을 구성하고 사용자와 함께 시스템 외부의 정보를 연결하려고 시도한다. 현재의 비트코인 구성은, 사용자의 매핑을 사용자의 노드에만 있는 그들의 공개키에 저장하고 각각의 사용자들이 필요한 만큼의 공개키를 생성할 수 있도록 함으로써 익명성 유출을 막기 위해 노력한다.

익명의 온라인 마켓들은 구매자와 판매자들의 신상을 확인할 수 있는 방법을 아주 어렵게 하여 주로 포르노물, 무기, 또는 마약과 같은 “black market”으로 이용되어 왔다 [7]. 2011년 2월에 운영을 시작한 온라인 마켓 실크로드는 온라인 환경에서 거래를 수행하기 위해 판매자와 구매자들을 위한 사회기반시설을 제공한다. 실크로드에서 비트코인을 이용하여 물건을 구매할 수 있으며, 구매자는 미리 자신의 비트코인을 생성한 다음 실크로드 에스크로에 물건값을 지불한다. 그러면 에스크로에서 판매자한테 구매자가 지불한 돈이 지불된 후 구매 완료가 된다. 비트코인을 이용하고 에스크로를 통한 거래를 하기 때문에 실크로드에서 구매자와 판매자 정보의 익명성이 보장된다.

국내에서 2016년 10월 서울지방경찰청이 비트코인으로 마약을 매매한 혐의로 80명을 검거, 5명을 구속해 첫 비트코인 추적 수사 성과로 주목받았다. 이는 범죄 용의자가 이용 중인 비트코인 주소만 알고 있으면 거래 내역 파악은 쉽다는 의미다 [8]. 비트코인 주소 자체는 익명이지만 실제로 돈이 입출금되는 환전 계좌는 실명이기 때문에 완벽한 익명성은 불가능하다.

III. 기존의 비트코인 익명성 연구

코인조인 [9], 제로코인 [10], 믹스코인 [11], 단델리온 [12, 13, 14], On-blockchain/Off-blockchain [15] 등 기존의 비트코인 익명성에 관한 연구들이 있다. 여기서는 제로코인과 단델리온에 대해 자세히 알아보고 기존 연구들 각각의 내용은 표 1에 정리되어 있다.

3.1 제로코인

2013년 Ian Miers 등이 제안한 제로코인(Zerocoin)은 비트코인 트랜잭션에 진정한 암호화 익명성을 추가하는 암호 해독 기법이다. 이는 신뢰할 수 있는 당사자를 추가하지 않고 개별 비트코인 거래 간의 연결을 끊는 암호화 기술을 사용하는 분산 전자 현금 시스템이다 [10].

제로코인을 비트코인으로 전환하려면 코인 주인은 영지식 증명(zero-knowledge proof)을 통해 증명을 해야 한다. 제로코인은 코인 C와 코인을 교환하기 위해 사용되는 임의의 번호 S가 연결되어 있지 않기 때문에 익명성이 보장된다.

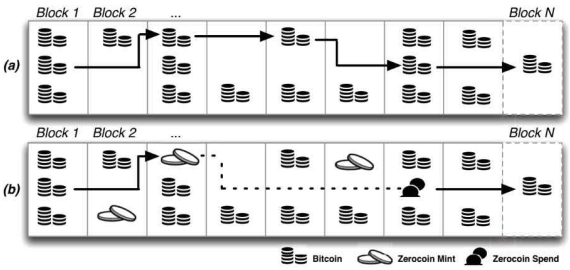


그림 2. (a)Bitcoin blockchain, (b)Zerocoin blockchain [10]

3.1.1 영지식 증명(Zero-Knowledge Proof)

영지식 증명(Zero-Knowledge Proof)이란, 어떤 내용을 알고 있을 때 그 내용을 직접 보여주지 않고 그것을 알고 있음을 증명하는 방법이다. 증명하고자 하는 자가 그 지식을 알고 있다는 것을 제외하고는 어떤 정보도 확인자에게 노출되지 않아야 한다.

3.2 단델리온

2017년 2월, 비트코인 네트워크에 대한 새로운 개인정보보호백서 '단델리온(Dandelion)'이 게시되어 비트코인 커뮤니티에서 많은 관심을 받았다 [12]. 이 프로젝트는 사람들이 인터넷 프로토콜 주소 (IP)를 스누핑하고 찾아내는 것을 막는 것을 목표로 하는 새로운 유형의 트랜잭션 브로드 캐스팅 방법이다 [13].

단델리온은 상대방으로부터 숨겨진 그래프를 통해 다른 사용자의 메시지를 혼합하여 최적의 익명성을 얻는다. 즉, 비트코인 P2P 스택을 재설계하여 사용자를 트랜잭션에 연결하려는 분산된 상대(예 : 봇넷(botnet))에 익명성을 제공

한다 [14].

그림 3은 모든 네트워킹 프로토콜의 Recall과 Precision에 대한 경계를 나타낸 그래프이다. Precision과 Recall은 탐지확률로, Recall은 단순히 추정치의 완전성을 포착하는 일반적인 익명성 척도이고, Precision은 정확성의 척도이다. 이 그래프에서 'DANDELION'은 하한선에 가까운 (Recall, Precision) 영역을 달성하면서 강력한 익명성을 지니고 있다. 점 p는 네트워크에 있는 노드의 수에 대한 결탁 노드의 비율이다. 그래프의 두 선 사이에 있는 경우에만 (Recall, Precision)지점을 실행할 수 있음을 보여준다. 또한 모든 네트워킹 프로토콜은 적어도 (a)의 한 지점에서 음영 영역과 교차하는 악의적인 상대에게 달성 가능한 (Recall, Precision)영역을 생성한다.

그래프에서 'DANDELION'이라고 표시된 지점은 Dandelion의 달성 가능한 정밀 Recall 영역의 Paraeto frontier, 즉, 경계점이다. 이를 달성 할 수 있는 영역과 비교한다. 'Diffusion'이라고 표시된 점은 2015년 Bitcoin 서버 네트워크의 스냅 샷에서 확산 프로세스를 시뮬레이션하고 차선의 소스 클래스를 사용하여 얻은 것이다. 이 때문에 달성할 수 있는 영역은 플롯된 점을 포함해야 하지만, 더 클 수 있다.

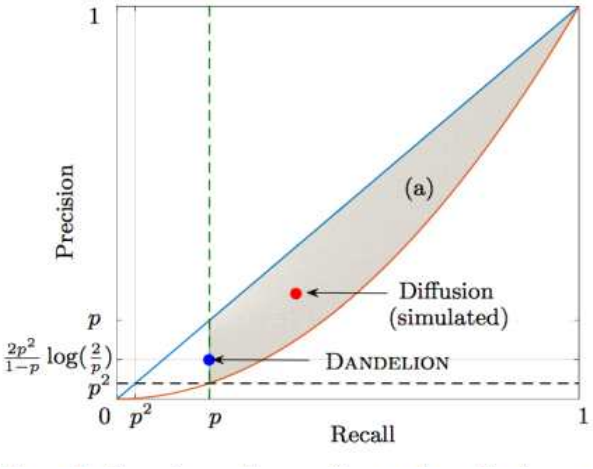


그림 3. 모든 네트워킹 프로토콜의 Precision과 Recall의 경계 [14]

솔루션	특징	장점	단점
코인조인(CoinJoin) [9]	별개의 트랜잭션을 결합하여 이전 거래로부터 누가 비트코인을 받았는지 모르도록 한 방식	-	-
제로코인(Zerocoin) [10]	신뢰할 수 있는 당사자를 추가하지 않고 개별 비트코인 거래 간의 연결을 끊는 암호화 기술	다른 모든 암호 해독 기능에 통합될 수 있음	비트코인보다 약 50배 많은 계산 능력 필요 제 3자에 대한 검증 비용이 많이 듦
믹스코인(Mixcoin) [11]	혼합 서비스와 혼합을 원하는 사용자와 관련되어 있음 효과적인 익명성을 위해, 청크 크기가 표준화되어야 함	일반 비트코인 사용자가 쉽게 구현할 수 있음	청크 크기를 너무 높게 설정하면 블록체인에서 더 많은 거래가 필요함
단델리온(Dandelion) [12,13,14]	비트코인 네트워크에 대해 제시된 새로운 개인정보보호백서 비트코인 P2P 스택을 재설계하여 사용자를 트랜잭션에 연결하려는 분산된 상대(예 : 봇넷(botnet))에 익명성을 제공	더 강력한 공격자 모델과 보다 강력한 전파 구현이 포함됨 모든 국가의 정부가 비트코인 사용자에 대해 엄격한 AML 및 KYC 규정을 시행 할 때 사용됨	노드가 부적절하게 내용을 잘못 전달하거나 오작동 하는 경우, 부적절한 영향을 줄 수 있다.
On-blockchain and Off-blockchain[15]	Soft fork를 통해 비트코인과 호환되는 새로운 익명성 스키마	트랜잭션 속도에 대한 실용성, 이중 지불 또는 DoS, Sybil attack에 대한 보안, 연결해제 가능한 트랜잭션에 대한 익명성 간의 새로운 절충안을 제공	-

표 1 . 비트코인 익명성 솔루션의 특징 및 장단점 분석

IV. 비트코인 익명성 현황

비트코인은 기본적으로 익명성을 보장하지만, 그 자체만으로는 익명성이 완전히 보장되지 않는다. 비트코인 시스템은 비트코인을 사용하는 컴퓨터들을 연결하는 거래 데이터베이스인 자체 블록체인 상에 거래 내역을 기록한다. 이 블록체인은 실질적으로 비트코인 거래의 장부 역할을 한다 [16].

비트코인을 현금이나 공식화폐로 환전을 할 때 익명성은 없어진다. 대부분의 가상화폐 거래소는 본인인증을 거치도록 되어 있기 때문에 본인인증을 거치면 아무리 비트코인 거래 시 익명성을 보장받았다고 해도, 공식화폐로 환전할 때 그 주체가 누구인지 알 수 있게 되는 셈이다. 현재 국내 가상화폐 거래소는 모두 본인인증을 해야만 환전이 가능하며 해외 대부분 거래소들도 본인인증을 하고 있다. 일부 거래소의 경우 메일 인증만 받는 곳도 있다.

비트코인의 불안한 익명성 때문에 최근 이를 강화한 새로운 가상화폐들도 생겨나고 있다. 대표적인 예로 모네로와 Shadow Cash가 있다. 모네로는 익명성에 초점을 둔 가상화폐로 CryptoNote 프로토콜(완전한 익명성을 위한 암호화 기법 [17])을 채택하여 사용자의 익명성을 보호한다 [18]. Shadow Cash는 PoSv3 (Proof-of-stake) 로 알려진 공개 소스 프로토콜을 기반으로 하여 백그라운드에서 모든 고급 암호화를 처리하여 사용자 친화적 인 간단한 인터페이스를 통해 익명성 및 개인 정보를 제공하는 것을 목표로 한다 [19].

V. 결론

본 논문에서는 비트코인 익명성에 대해 정리 분석하였다. 비트코인은 이중적인 익명성을 갖는 가상화폐이다. 비트코인은 정부, 은행 등 기관의 개입이 없기 때문에 추적당할 우려가 없다고 여겨지며, 익명성을 제공하는 동시에 블록체인을 통한 사용자의 거래 내역을 확인할 수 있다. 이러한 점은 각종 범죄에 악용된다. 비트코인 개발자는 블록체인과 비트코인 네트워크에서 제공하는 익명성 수준을 높이려는 노력을 기울여야 하는 동시에 사용자 정보유출이 발생하면 신뢰가 있는 수사 기관에서 추적이 가능하도록 만들어야 한다. 또한 블록체인 기술과 관련된 비트코인 사용자의 개인 정보 보호 의식을 강화해야한다.

[참고문헌]

[1] S.Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System”, 2008.

[2] 김승민, 이지혜, 김윤정, “비트코인 익명성 강화에 관한 연구”, 한국정보보호학회 하계학술대회, 순천향대학교, 2017년 6월.

[3] M.Swan, “Blockchain: Blueprint for a new economy”, O′Reilly Media, Inc., 2015.

[4] A.M.Antonopoulos, “Mastering Bitcoin: unlocking digital cryptocurrencies”, O′Reilly Media, Inc., 2014.

[5] F.Reid and M.Harrigan, “An analysis of anonymity in the bitcoin system, Security and privacy in social networks”, Springer New York, 197-223. 2013.

[6] “Bitcoin Anonymity vs Transparency”, http://www.getnews.co.kr/view.php?ud=2734BTH, 2017년 10월 확인.

[7] N.Christin, “Traveling the Silk Road: A measurement analysis of a large anonymous online marketplace”, Proceedings of the 22nd international conference on World Wide Web. ACM, 2013.

[8] 비트코인은 추적 불가능? 돈으로 바꾸면 잡는다, 전자신문, 2016년 12월 6일 기사, http://www.etnews.com/20161206000251

[9] D.A.Wijaya, J.Liu, and R.Steinfeld, “Anonymity in Bitcoin”, 2016.

[10] I.Miers, C.Garmanet, M.Green and A.D.Rubina, “Zerocoin: Anonymous distributed e-cash from bitcoin”, Security and Privacy (SP), 2013 IEEE Symposium on, IEEE, 2013.

[11] J.Bonneau, A.Narayanan, A.Miller, J.Clark, J.A.Kroll and E.W.Felten, “Mixcoin: Anonymity for Bitcoin with accountable mixes”, International Conference on Financial Cryptography and Data Security, Springer Berlin Heidelberg, 2014.

[12] https://news.bitcoin.com/ dandelion-bitcoinanonymize-transaction-broadcasts/, 2017년 10월 확인.

[13] http://www.coindesk.com/bitcoin-developers-publish-bip-dandelion-privacy-project/, 2017년 10월 확인.

[14] Bojja Venkatakrishnan, Shaileshh, Giulia Fanti and Pramod Viswanath. "Dandelion: Redesigning the Bitcoin Network for Anonymity." Proceedings of the ACM on Measurement and Analysis of Computing Systems 1.1 (2017): 22.

[15] E.Heilman, F.Baldimtsi and S.Goldberg. "Blindly signed contracts: Anonymous on-blockchain and off-blockchain Bitcoin transactions." International Conference on

Financial Cryptography and Data Security. Springer Berlin Heidelberg, 2016.

[16] 류재철, 블록체인과 개인정보보호걸림돌, DT광장, 디지털 타임스 2017년 7월 30일 기사, http://www.dt.co.kr/contents.html?article_no=2017073102102351049001

[17] https://cryptonote.org/inside.php

[18]https://namu.wiki/w/%EB%AA%A8%EB%84%A4%EB%A1%9C

[19]https://en.wikipedia.org/wiki/Shadow_(software)

[20] 이유미, 완벽한 익명성은 없다, 이데일리 2017년 6월 19일 기사, http://www.edaily.co.kr/news/NewsRead.edy?SCD=JE41&newsid=03198006615963360&DCD=A00504

[21] 비트코인은 추적 불가능한 지하경제용 화폐 아니다, Korea-Benchmark, 2017sys 5월 23일 기사, http://www.kbench.com/?q=node/178720

블록체인 기반 에너지 거래 모델 연구

김승리[1] † 이정혁[1] 구슬예[1] 오현옥[1] 김지혜[2]††

[1]한양대학교 정보시스템학과

[2]국민대학교 전자공학과

Energy Trading Models based on Blockchain

Seung Ri Kim[1] Jeong Hyuk Lee[1] Seul Ye Gu[1] Hyun Ok Oh[1] Ji Hye Kim[2]

[1]Dept. of Information System, Hanyang University

[2]Dept. of Electronics Engineering, Kookmin University

요 약

현재 전력 발전 방법은 기존의 대규모의 중앙 발전 후 배전 방식에서 프로슈머(prosumer)방식으로 변화하고 있다. 이에 따라 소규모 단위로 에너지의 공급과 수요를 관리하는 지역전력망인 마이크로그리드(microgrid)방식이 국내외 다수의 기업에서 연구되고 있다. 마이크로그리드에 P2P(Peer to Peer) 네트워크 환경을 도입하게 되면 서버구축 비용을 절감할 수 있고 보안성을 증대시킬 수 있다. 최근 P2P 네트워크 환경에서 분산 거래 시스템을 구축하는 방법 중 하나로 블록체인(blockchain) 기술이 주목 받고 있다. 블록체인 기술은 해시를 통해 연결된 분산 환경 내의 공유원장으로써, 신뢰할 수 있는 중앙기관 없이 안전하게 거래를 수행할 수 있는 특징이 있다. 본 논문에서는 에너지 거래방식을 거래주체별로 분류하고 각 방식에 맞는 블록체인 기술 적용 방안에 대해 제안한다.

I. 서론

최근 태양열 에너지를 활용한 전력발전 및 신재생에너지를 활용한 소규모 및 중규모 형태의 전력발전 방법이 대두됨에 따라, 중앙에서 모든 전기를 발전하고 공급하는 기존의 중앙관리형 전력공급시스템은 한계에 부딪히게 되었다. 발전소에서 생산된 전기 에너지를 소비자에게 전달하는 현재의 단방향 전력공급방식은 직접 에너지를 생산하고 공급까지 가능한 프로슈머에는 적합하지 않다. 이와 같은 한계를 극복하기 위해 기존의 대형 발전 및 대량의 전력공급방식을 사용하는 것이 아닌 소규모 지역 내에서 에너지를 생산하고 공급과 수요를 관리하는 마이크로그리드가 각광받고 있으며 최근 국내외 기업에서 활발히 연구되고 있다 [8].

이와 같이 소규모의 전력에너지를 공급하고 구매할 수 있는 환경에서는 P2P 시스템을 사용하여 거래를 관리하는 것이 중앙에 데이터베이스를 구축하여 거래데이터를 관리하는 것보다 비용 및 관리 측면에서 효율적이다. 실제로, 다수의 국외기업에서 P2P 시스템을 사용하여 전력 거래를 수행하는 프로젝트를 진행하고 있다.

특히, 몇몇 기업에서는 블록체인 기술을 에너지 거래에 적용하는 프로젝트를 수행하고 있다. 블록체인은 네트워크에 참여하는 모든 참여자가 서로 합의하여 공유원장을 가지는 기술로 공유원장의 거래 데이터가 들어간 각 블록은 해시를 통해 연결되어 있기 때문에 조작할 수 없는 특징이 있다. 즉, 블록체인 기술을 통해 신뢰할 수 있는 중앙기관 없이 P2P 네트워크에서 모든 참여자가 신뢰성 있는 거래를 수행할 수 있다. 또한 블록체인 기술은 블록을 합의하여 연결하는 형태에 따라 그 특성이 달라진다. 본 논문에서는 에너지 거래를 발전주체 및 거래주체에 따라 분류하고 각 거래 유형별 적용

† 제 1저자
†† 교신저자

가능한 블록체인 기반 에너지 거래모델을 제안한다.

II. 배경지식

2.1 블록체인

블록체인은 거래 기록을 관리하는 중앙 서버 없이, P2P 네트워크상의 모든 참여자들이 공동으로 거래 기록을 관리하는 분산 원장(ledger) 관리 기술이다 [1]. 블록은 일정 기간 동안 수행된 거래 기록을 담고 있으며, 블록 헤더는 이전 블록에 대한 해시 값을 포함하고 블록들이 연결되어 체인을 형성한다. 이는 특정 블록을 조작하기 위해서는 해당 블록에 연결된 모든 이전 블록을 조작해야함을 의미한다. 또한, 신규 블록을 생성하는 과정을 채굴(mining), 채굴을 수행하는 노드를 채굴자(miner)라고 한다. 블록체인의 채굴 방식은 다음과 같이 분류된다.

2.1.1 작업 증명 방식(Proof of Work)

채굴자가 신규 블록을 생성하기 위해서는 특정 범위의 해시 값을 출력하는 해시 입력을 찾아야 한다. 해시 함수는 입력에서 출력을 구하는 계산은 쉬우나, 역으로 출력에서 입력을 구하는 것은 매우 어렵다는 특징을 가지고 있다. 채굴자는 조건을 만족하는 해시 값이 나올 때까지 해시의 입력에 임의로 생성된 토큰 값(nonce)을 계속 변경하며 대입한다. 채굴자가 가진 컴퓨팅 파워가 높을수록 토큰 값을 찾을 확률이 높아진다.

2.1.2 지분 증명 방식(Proof of Stake)

작업 증명 방식과 달리 해시 입력이 이전 블록 데이터와 현재 거래정보, 현재 시간으로 결정되어 있고, 보유한 지분(stake)이 많을수록 찾아야 하는 해시 출력의 범위가 넓어진다 [2]. 지분은 참여자가 소유한 코인으로 보유한 코인이 많을수록 신규 블록을 생성할 확률이 높아진다.

2.1.3 위임된 지분 증명 방식(Delegated Proof of Stake)

지분 증명 방식을 사용하여 신규 블록을 생성하나, 신규 블록에 사전에 선출된 대표자(delegate)들의 서명이 있을 경우에만 블록이 유효하다고 판명한다 [3]. 대표자는 특정 규칙에 의해 선출되고 변경된다.

2.1.4 활동 증명 방식(Proof of Activity)

작업 증명 방식과 지분 증명 방식을 결합한 것으로 보유한 코인 양에 비례하는 확률로 선발된 N명의 지분 보유자(stakeholder)들이 신규 블록 생성에 참여한다 [4]. 네트워크에 실제 참여하고 있는 온라인 상태의 지분 보유자에게만 보상을 하는 인센티브 구조이다.

2.2 에너지 거래에서 블록체인 활용 사례

최근 전력 수송 비용 절감과 거래 절차 단순화, 보안 강화 등을 위하여 에너지 거래에 블록체인을 활용하는 다양한 프로젝트가 진행되고 있다. 특히, 중개자 없이 생산자와 소비자가 직접 에너지를 거래하는 플랫폼을 구축하려는 시도가 지속되고 있다.

2.2.1 TranActiveGrid

2016년 4월 미국 브루클린 지역에서 시행된 프로젝트인 TransActiveGrid는 건물에서 잉여 생산된 태양열 에너지를 블록체인을 사용하여 인접한 가정에 직접 판매하는 방법을 연구하였다 [5]. 위 프로젝트에는 스마트 계약(smart contract)을 통해 에너지 거래가 가능한 블록체인인 이더리움(Ethereum)이 사용되었다 [6].

2.2.2 Power Ledger

2016년 8월 호주의 블록체인 신생 업체인 Power Ledger와 다양한 에너지 회사가 진행했던 프로젝트이다 [7]. 신재생 에너지의 소유자가 잉여 에너지를 소비자에게 직접 판매하고 전력의 생산 지점에서 소비 지점까지 추적할 수 있는 시스템을 구축하였다. 또한 전력 계량기가 블록체인과 직접 통신할 수 있게 하여 기존에 발생하였던 많은 하드웨어 신뢰 문제를 극복하였다.

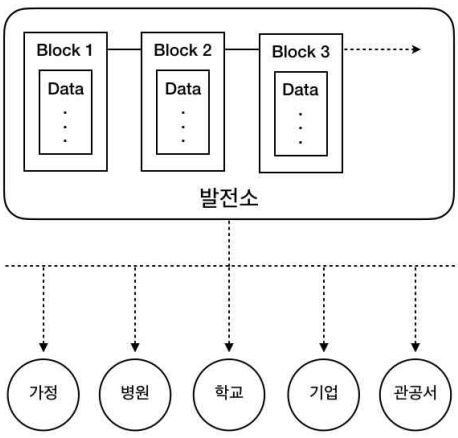
III. 블록체인 기반 에너지 거래 모델

블록체인에서 거래를 합의하기 위한 증명 방법에 따라 다양한 에너지 거래 모델 설계가 가능하다. 전기 에너지를 생산하는 가구에는 각 가정에서 생산하는 전력의 수치를 측정하고 이러한 정보를 저장하는 스마트 미터기가 설치되어 있다. 스마트 미터기가 설치된 가구들이 하나의 블록체인 네트워크로 연결되어 있을 때, 각 가구들은 자신이 생산한 전기 에너지를 거래할 수 있다. 또한, 거래에 참여하는 집단의 규모에 따라 다양한 형태의 블록체인 기술을 적용할 수 있다.

3장에서는 블록체인을 활용한 세 가지 에너지 거래 모델을 제안한다.

3.1 중앙 집중 형태의 에너지 거래

현재의 대부분의 전력공급방식은 원자력, 화력 발전소 등에서 생산된 전력이 송전소와 변압소를 거쳐 가정이나 학교, 기업 등으로 배전되는 형태이다. 이러한 형태에서는 전력 생산 및 공급 데이터를 저장하고 있는 발전소 데이터베이스(database)를 블록체인 구조로 적용하는 것을 고려할 수 있다. [그림 1]과 같이 발전소 내 데이터 구조를 블록체인과 같이 분산된 데이터로 변경할 경우 기존의 데이터베이스 구축보다 비용 측면에서 저렴하며 저장된 데이터를 조작하는 것이 어렵기 때문에 공격자에 의해 데이터가 위·변조되는 것을 방지할 수 있다.

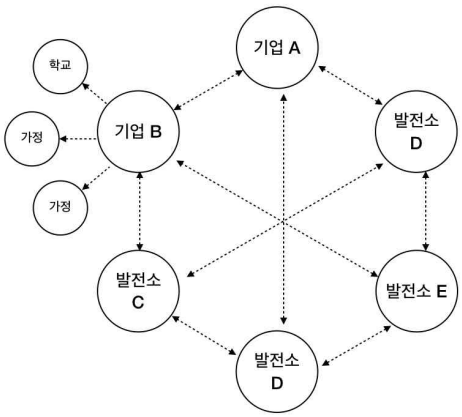


[그림 1]

3.2 부분적 분산 형태의 에너지 거래

전기 에너지를 생산 및 거래에 참여하는 대상이 상대적으로 적고 제한되어 있을 때는 사설 블록체인(private blockchain)을 구축하는 것이 효과적이다. 아래 [그림 2]는 일종의 B2B(Business-to-Business)네트워크로 에너지를 생산하고 판매할 수 있는 발전소나 기업이 거래의 주체이다. 이후 소비자에게 전력을 공급하고 판매하는 것은 기존의 중앙 관리 전력시스템과 동일하다. 사전에 지정된 기업 및 전력 발전소만이 거래 네트워크에 참여할 수 있고 이러한 형태의 네트워크에서는 위임된 지분 증명 방식을 사용할 수 있다 [3]. 이것은 블록을 생성할 수 있는 결정권을 지닌 대표자들을 미리 선출 한 후 블록을 생성하는 것으로 일반 노드들은 초기 블록을 생성하고 최종적으로 대표자의 유효한 서명이 있어야 블록이 연결된다. 이것은 기존의 작업 증명 방식과 지분 증명 방식이 가진 한계를 보완한다.

먼저, 미리 선출된 대표자의 서명에 의해 블록이 연결되므로 블록의 생성속도가 빠르다.



[그림 2]

또한 기존의 지분 증명 방식에서 발생하는 뇌물공격(bribe attack)과 같은 취약점 공격에 상대적으로 강하다 [3].

3.3 완전 분산 형태의 에너지 거래

마이크로그리드는 일종의 스마트 그리드(smart grid)로 전기 에너지를 생산 및 거래를

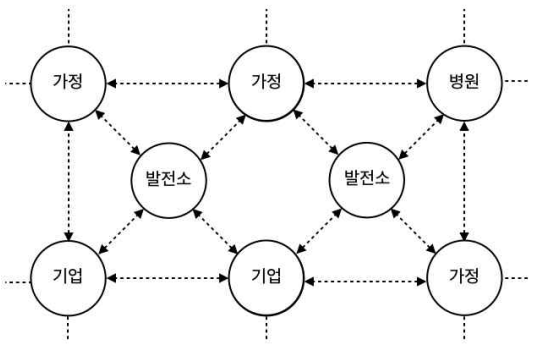
할 수 있는 지역 전력망을 의미한다 [8]. 기존의 중앙관리형 전력 공급 시스템은 발전소에서 전기 에너지를 생산하여 소비자에게 배전하는 구조였으나, 전기 에너지를 직접 생산하고 공급하는 프로슈머의 등장으로 전력의 자급자족이 가능하게 되었다.

이와 같이 다수의 가구가 참여한 에너지 거래 네트워크에서는 적합한 거래 합의 증명 방식으로는 작업 증명 방식과 지분 증명 방식, 활동 증명 방식이 있다.

먼저, 작업 증명 방식을 사용할 경우 블록헤더의 해시 값을 찾기 위한 작업을 통해 이 거래가 유효함을 증명할 수 있고, 이 작업은 모든 노드, 즉 에너지 거래에 참여하는 모든 가구들이 참여할 수 있으므로 거래에 대한 신뢰성이 높다 [1]. 그러나 거래가 확인되어 블록이 연결되기까지 많은 시간이 소요되고 작업 증명이 컴퓨팅 파워에 좌우되므로 고가의 장비와 많은 전력을 필요로 한다.

반면에 지분 증명 방식은 초기 블록 생성 시에만 작업 증명 방식이 사용되고 이후에 블록을 연결하기 위해서는 소유한 지분을 사용한다 [2]. 지분 증명 작업 방식을 사용할 경우, 작업 증명 방식에 비해 블록 연결 속도가 빠르며 상대적으로 적은 전력을 필요로 한다.

또 다른 활용으로는 활동 증명 방식이 있다 [4]. 이것은 작업 증명 방식과 지분 증명 방식을 결합한 것으로 거래에 참여하는 온라인 상태의 노드에게만 보상을 하는 방식이다. 따라서 실제 에너지 거래에 참여하는 가구만이 거래 합의 증명에 따른 보상을 받을 수 있다.



[그림 3]

IV. 결론

본 논문에서는 에너지 거래에서 사용할 수 있는 블록체인의 종류와 활용방안에 대해 알아보았고, 이를 이용하여 실제 구현 가능한 세 가지 에너지 거래 모델을 제안하였다. 특히 마이크로그리드에서 블록체인을 활용할 경우 다양한 장점이 존재한다.

첫째, 중앙관리형 서버에 비해 안전하다. 거래 내역은 각각의 노드에 저장되므로 이 값을 위·변조하는 것이 거의 불가능하다. 둘째, 거래 내역이 투명하게 관리되기 때문에 네트워크에 참여하고 있는 누구나 블록체인에서 발생한 모든 거래내역을 검증하는 것이 가능하다. 셋째, 인적요소 및 중개인을 거치지 않으므로 구축 및 운영 비용이 절감된다.

향후에는 영지식(zero-knowledge)증명을 통해 익명성을 보장하는 제로캐시(zerocash)등을 적용하여, 거래 내역의 무결성을 제공하면서 거래 정보는 보호할 수 있는 방안에 대한 연구도 고려할 수 있을 것이다 [9].

V. Acknowledge

본 연구는 교육부(2016R1D1A1B03934545)의 재원으로 한국연구재단(National Research Foundation of Korea)의 지원과 정보통신기술진흥센터(2017-0-00661)의 재원으로 미래창조과학부의 지원을 받아 수행된 연구임.

[참고문헌]

[1] NAKAMOTO, Satoshi. Bitcoin: A peer-to-peer electronic cash system. 2008.

[2] KING, Sunny; NADAL, Scott. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. *self-published paper,* 2012

[3] BitFury Group, Proof of Stake versus Proof of Work(White Paper) v1. 2015

[4] BENTOV, Iddo, et al. Proof of Activity: Extending Bitcoin's Proof of Work via Proof of Stake [Extended Abstract] y.

ACM SIGMETRICS Performance Evaluation Review, 2014

[5] http://transactivegrid.net

[6] WOOD, Gavin. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 2014

[7] https://web.powerledger.io

[8] LASSETER, Robert H.; PAIGI, Paolo. Microgrid: A conceptual solution. IEEE, 2014

[9] SASSON, Eli Ben, et al. Zerocash: Decentralized anonymous payments from bitcoin. *IEEE Symposium on*. IEEE, 2014

2017 한국정보보호학회 동계학술대회

B5 고교생논문

박 영 호 (세종사이버대)

- **챗봇을 활용한 계정탈취의 위험성**
  정희태, 박민혁 (목포대학교 정보보호영재교육원), 김민수 (목포대학교)
- **RFID Skimming 해킹 기법에 관한 고찰**
  김건우, 김태현, 성유원, 여상준, 이진우, 이창엽, 이달원 (대구대학교 정보보호영재교육원)
- **미공개 된 타원곡선 암호시스템에서의 파라미터 탐지 방법**
  조현주, 최태균 (하나고등학교)
- **Mac OS X의 디지털 증거 수집 방식 결정 방안 제시**
  이준수 (경기과학고등학교), 신희재 (세종대학교), 조재민 (서울과학기술대학교), 강성철 (서울과학기술대학교), 박진우 (경기대학교)

챗봇을 활용한 계정탈취의 위험성

정희태†, 박민혁†, 김민수‡ 1)

† 목포대학교 정보보호영재교육원
‡ 목포대학교 정보보호학과

† Risk of Account Theft using Chatbots

Hee-Tae Jeong, Min-Hyuk Park, Minsoo Kim

† Institute of Information Security Education for the Gifted at MNU
‡ Department of Information Security Engineering, Mokpo National University

요 약

현대는 스마트폰 기술과 SNS 환경이 사람들간의 소통하는 창구가 되고 있다. 한 사용자의 개인정보 탈취는 그 사용자의 이름을 이용한 사회공학적 공격이 연속될 수 있다. 본 논문에서는 페이스북에 챗봇을 적용하여 악의적으로 사용자의 접근 토큰을 탈취할 수 있음을 보인다. 이를 통하여 챗봇 기술에 경각심을 갖게 하고 개인정보보호에 대한 인식을 높이고자 한다.

I. 서론

현대는 스마트폰과 다양한 IT 기술이 일상생활에 접목하면서 SNS(Social Networking Service)라는 새로운 소통의 장소가 생겨나기 시작했다. SNS의 대표적인 사이트로는 인스타그램(Instargram), 트위터(Twiter), 페이스북(Facebook) 등이 존재한다. 이러한 SNS는 다수와 소통하려는 좋은 목적이었지만, 이를 악용하는 사례가 나타났다.

본 논문에서는 챗봇(chatbot)을 이용하여 페이스북 계정을 탈취할 수 있는 방법을 다루려고 한다. 페이스북에서 계정탈취 방법은 계속 지능화하고 있다. 인공지능 챗봇은 다양한 SNS에서 활용되고 있다. 본 논문에서는 페이스북에 챗봇을 구현하고, 채팅과정에서 토큰의 탈취 가능성을 분석한다.

II. 관련연구

2.1 챗봇

챗봇이란 쉽게 말해 채팅하는 로봇(chatting robot)이다[1]. 이것은 정해진 응답 규칙에 따라 사용자 질문에 응답할 수 있도록 만들어진 시스템이다. 전자게시판이나 통신망에서 여러 사용자가 다양한 주제를 가지고 실시간 모니터링 화면을 통해 대화를 나누는 채팅과 자동으로 사람이 하던 일을 수행하는 기계인 로봇의 합성어이다. 챗봇은 인공지능을 기반으로 사람과 자동으로 대화를 나누는 소프트웨어에 해당된다.

챗봇은 기본적으로 챗봇 API를 제공하는 플랫폼(platform)을 통해서 챗봇 서버와 서로 통신을 주고받을 수 있는 '요청과 응답(Request-Response)' 구조를 따른다[2]. 사용자가 메신저 대화창에 특정한 메시지를 입력하면, 메신저 사업자의 챗봇은 해당 메시지에 적합한 응답을 해달라고 챗봇 서버에 응답을 요청하는 식이다. 사용자가 보낸 메시지 규칙에 따라 챗봇 서버의 인공지능에서는 응답 메시지를 선택한다. 메시지 규칙은 텍스트 전달 방식 외에도 영상, 이미지, 웹주소(URL) 등을 결합하여 보낼 수 있다.

2.2 사회공학기법

사회공학은 공격 대상에게 어떠한 수단으로 접근하는가에 따라 인간 기반(Human Based) 공격과 컴퓨터 기반(Computer Based) 공격으로 나눌 수 있다[3]. 인간 기반 공격은 공격자가 공격 대상에게 직접 접근하거나 전화 등을 이용한 접근 방법이고, 컴퓨터 기반 공격은 악성코드, 컴퓨터 프로그램, 웹 사이트 등을 이용한 접근 방법이다. 일반적으로 사회공학 기법은 인간 기반의 수단을 이용하는 공격 형태를 지칭한다. 그 중에서 피싱(Phishing)은 사회공학적 방

1) 교신저자

법 및 기술적 은닉기법을 이용해서 민감한 개인 정보, 금융계정 정보를 절도하는 신종 금융사기 수법이다[4]. 챗봇은 이러한 사회공학적 공격에 이용될 수 있다.

III. SNS 계정정보 탈취

챗봇은 SNS에서 사회공학적 공격의 수단이 될 수 있다. 여기에서는 페이스북에서 챗봇이 이용되는 사례를 설명한다.

3.1 페이스북의 접근토큰 탈취



[그림 1] 페이스북 접근 토큰 탈취

페이스북에서는 사용자 계정의 접근토큰(access token)을 탈취하기 위해 “방문자 추적기”를 이용된 적이 있다(그림 1). 게시글을 올렸을 때, 자신과 연결된 지인들에게 홍보를 하게 된다. 지인들은 이 게시글을 보고 호기심을 갖고 따라하게 되며, 결국 이를 주도한 업체는 접근토큰을 수집할 수 있다[5].

이렇게 수집된 접근토큰은 개인 프라이버시를 침해하게 된다. 해당 계정의 대화 내용을 살펴볼 수 있을 뿐만 아니라, 계정 주인의 의사에 반하여 상업적으로 이용될 수 있다. 불법 도박이나 성인 사이트 광고 페이지에 대하여 ‘좋아요’ 행위를 수행하도록 지시할 수 있다. 만약, 계정의 활동 내용에 금융정보나 비밀번호 같은 중요한 기록이 있다면 더욱 큰 피해를 입을 수 있다.

게시글로 피싱사이트로 유도하는 방식은 다음과 같다. 인지도가 많은 페이지들의 게시글을 대상으로 댓글을 달아 호기심을 자극한 뒤 피싱 사이트 링크로 접속을 유도한다(그림 2). 첨부된 링크에 접속을 하게 될 경우 피싱 사이트로 접속을 하게 된다. 피싱 사이트의 서비스를 이용하기 위해서 다시 로그인을 하도록 요구한다(그림 3). 이때 일반 사용자들은 아무런 의심 없이 해당 서비스를 이용하게 되고 결국 계정의 접근토큰을 넘겨주게 된다.



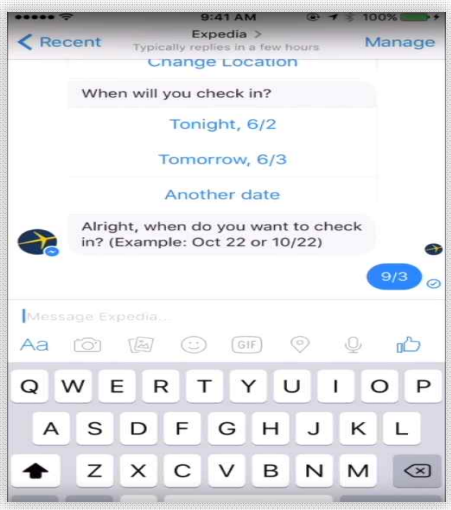
[그림 2] 접속 유도 화면



[그림 3] 피싱 사이트에서 계정 요구

3.2 채팅 봇의 위험성

챗봇을 활용한 서비스 중에는 호텔 예약 서비스를 제공하는 챗봇인 ‘익스피디아(expedia)’, 상품 주문 서비스 챗봇인 화장품 유통기업 ’세포라(Sephora)’ 등이 있다. 이런 챗봇들의 공통점은 고객에게 서비스를 제공하는 점이다. 이러한 서비스를 이용하기 위해서는 챗봇이 제공하는 기능에 사용자가 접근(access)해야 한다는 것이다. 챗봇은 사용자와의 대화를 통해 관심을 유도한 뒤 어느 정도의 관계형성이 되면 서비스 제공을 미끼로 던진다. 이러한 미끼는 피싱사이트로 사용자를 유도하여 계정 탈취를 수행할 수 있다. 이렇게 챗봇을 기반으로 사회공학적 공격 기술을 ASE(Automated Social Engineering)이라 한다[6].



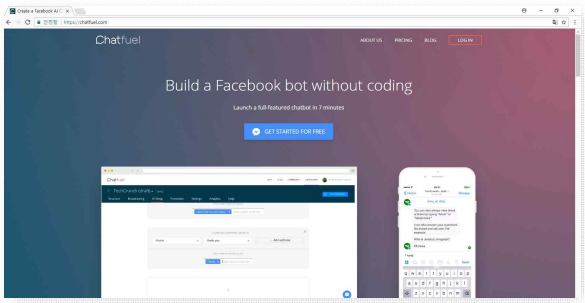
[그림 4] 익스피디아 챗봇

IV. 챗봇의 악용과 대응 방법

현재도 페이스북 계정을 탈취하기 위해 여러 수단을 사용하는 것을 목격할 수 있다. 챗봇 활용 기술이 많이 이용되고 있는 시점에서 페이스북에 챗봇 서비스를 적용한다면, 이를 통해 페이스북 접근토큰을 탈취할 수 있을 것이다.

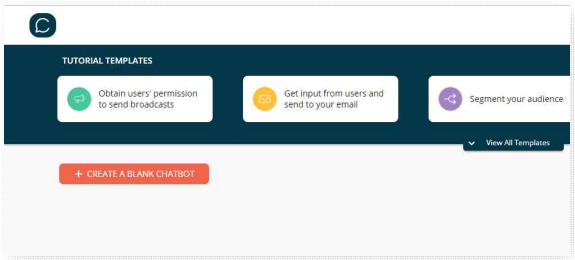
4.1 페이스북에 챗봇 연동

페이스북에 챗봇을 설치하는 방법은 이미 알려져 있다. (그림 5)와 같이 페이스북 챗봇을 쉽게 설치하여 운영할 수 있다[7]. 이러한 챗봇의 대화를 통하여 호기심을 유도하는 사회공학적 기법을 적용해볼 수 있을 것으로 생각된다.



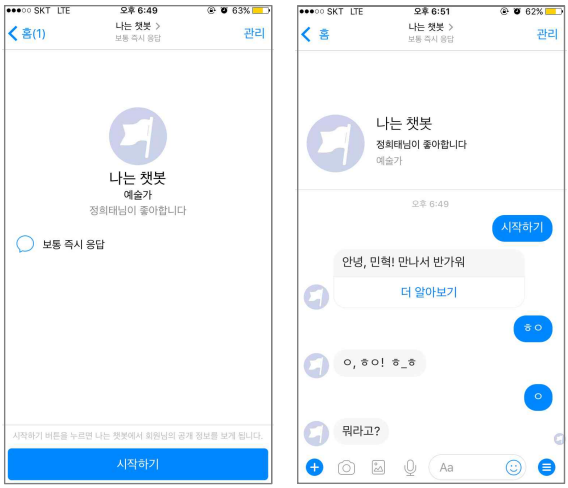
[그림 5] 페이스북 봇 제작 사이트

본 논문에서는 페이스북 챗봇을 구축해보았다. (그림 6)의 왼쪽 아래 “Create a Blank Chatbot”을 선택하여 챗봇을 생성한다. 그리고 “connect to facebook” 선택하여 자신의 페이스북에 연결을 시도한다. 이후에 페이스북 페이지를 생성하여 채팅을 시작할 수 있다.



[그림 6] 새로운 챗봇 생성 화면

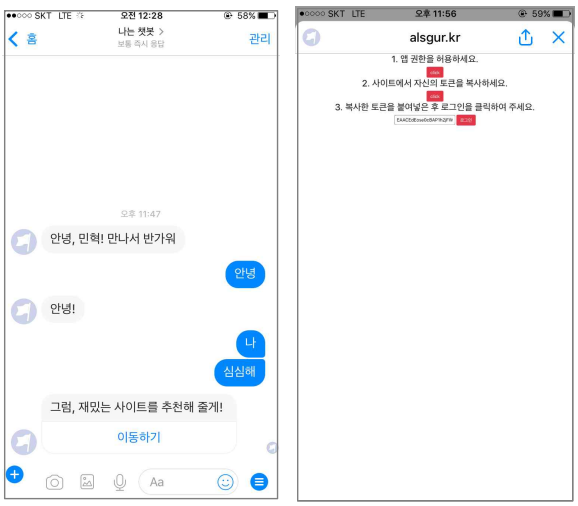
(그림 7)은 페이스북에 챗봇을 연동한 결과를 보여준다. “채팅 시작하기” 버튼을 통하여 채팅을 할 수 있고 챗봇과 대화를 할 수 있다. 챗봇은 등록되어 있는 메시지로 대화를 수행한다. (그림 7)의 오른쪽 화면처럼 “ㅇ, ㅎㅇ! ㅎ_ㅎ”와 같이 챗봇에 등록되어 있지 않는 메시지를 입력하면, 챗봇은 “뭐라고”와 같은 알려지지 않는 내용에 대한 대응 메시지가 나타난다. (그림 7)의 오른쪽 화면에서처럼 “더 알아보기” 연결 링크가 생성되어 있다. 이 연결을 특정 URL로 지정한다면, 임의의 스크립트를 수행하는 사이트로 유도할 수 있다.



[그림 7] 페이스북에 생성한 챗봇

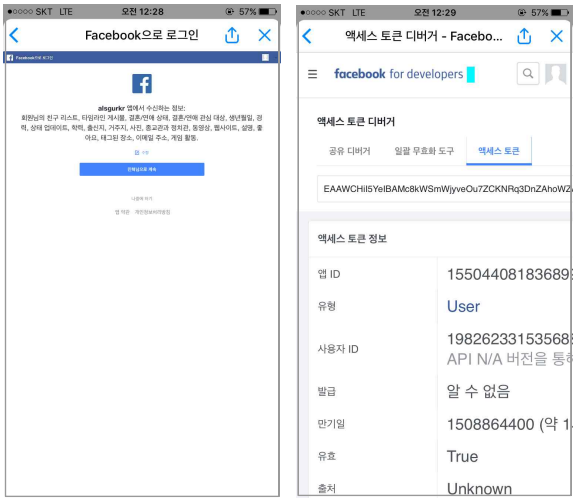
4.2 챗봇을 활용한 페이스북 접근 토큰 탈취

본 논문에서는 챗봇을 이용하여 피싱 사이트로 유도하는 과정을 구현하였다. (그림 8)의 왼쪽 화면에 “이동하기” 버튼을 생성하고, 이 버튼을 클릭하면 오른쪽 화면으로 이동하도록 하였다. 챗봇에 대화 목록에 “심심해”와 같은 특정 메시지가 나타나면 “그럼, 재밌는 사이트를 추천해 줄게!”와 같은 메시지와 함께 사이트 링크가 나타나도록 하였다. 이처럼 사용자와의 대화 속에 친근하게 관련 링크를 보여준다면, 사용자가 쉽게 관련 링크에 접속하게 된다.



[그림 8] 챗봇을 이용한 피싱 사이트 유도

위와 같이 피싱 사이트로 접속할 경우, (그림 9)와 같이 페이스북 계정을 요구한다. 페이스북 계정에서 방문자를 추적하는 서비스를 활용할 목적이다. (그림 8)의 우측 화면처럼 페이스북 계정의 접근토큰을 요구할 경우, 개인정보보호에 대한 인식이 얕은 사용자는 이러한 요구를 허락하기 쉬울 것이다.



[그림 9] 챗봇을 이용한 피싱사이트 유도

4.3 페이스북 접근 토큰의 악용

페이스북 접근 토큰으로 페이스북의 친구목록을 추적하여 이용할 수 있다. 페이스북 친구들에게 특정 게시글을 보여 주거나 특정 페이지에 "좋아요" 요청을 할 수 있다. 만약 친구들에게 특정 사이트로 유도하는 게시글을 보여주는 경우, 피싱 사이트로 유도하여 추가적으로 친구들의 계정을 탈취할 수 있다

본 논문에서 접근 토큰을 얻기 위해 지능적으로 SNS 사용자의 심리를 이용해 미끼를 던지는 방식을 사용하였다. 이러한 방법으로 사용자 페이스북의 접근토큰을 탈취하는 것은 친구목록에 있는 사람들에게까지 피해를 주게 된다. 또한 친구들의 접근 권한을 탈취할 가능성이 있으며, 이는 계속해서 피싱 사이트로의 유도 게시글이 파급될 수 있게 된다.

Ⅴ. 결론

챗봇과 같이 새로운 플랫폼이 등장하면서 다양한 서비스 구조가 형성되고 있다. AI 챗봇 서비스는 그 자체로 일반 사용자의 흥미를 유발하고 쉽게 신뢰할 수 있게 된다. 본 논문에서는 페이스북에 챗봇을 적용하여 접근토큰이 탈취됨을 보임으로서, 사회공학적 공격의 형태로 악용될 수 있다는 것을 보였다. 이렇게 SNS와 챗봇을 이용한 정보 탈취는 점점 고도화될 것이다. 따라서 새로운 IT 기술의 흐름을 파악하고 개인정보보호에 대한 인식을 높여야겠다.

[참고문헌]

[1] Wikipedia, https://en.wikipedia.org/wiki/Chatbot
[2] David Niki, "How to build an FAQ Chatbot with API.AI using Node.js and PHP", https://medium.com/
[3] 사회공학 기법 (정보 보안 개론, 2013. 6. 28., 한빛아카데미(주))
[4] SANS Institute, "사회공학공격", Ouch!, 2014년 2017년 4월
[5] "페북 액세스 토큰 주의하세요", 연합뉴스,
[6] Claudio Marchiondelli, "The dark side of the chatbot", May 2016
[7] "Build a Facebook bot without coding", https://chatfuel.com

RFID Skimming 해킹 기법에 관한 고찰

김건우*, 김태현*, 성유원*,
여상준*, 이진우*, 이창엽*,
이달원*
*대구대학교 정보보호영재교육원

A Consideration on RFID Skimming Hacking Technique

Geon-U Kim*, Tae-Hyeon Kim*, Yu-Won Seong*,
Sang-Jun Yeo*, Jin-U Lee*, Chang-Yeop Lee*
Dal-Won Lee*
*Information Protection Gifted Education Center, Daegu University.

요 약

우리가 현제 사용하고 있는 카드는 자신의 신분을 나타내는 용도로 쓰이고 있으나 RFID Skimming 해킹 기법에 의해 카드가 복제될 수 있는 문제점이 있다. 이에 우리는 RFID Skimming 해킹 기법에 대해 알아보고 우리 국내에서는 어떻게 대응할 수 있는지에 대해 알아보았다.

I. 서론

[1)]RFID 시스템은 물품 등의 사물에 태그를 부착하여 먼 거리에서도 각 사물의 정보를 수집, 저장 및 추적함으로써 사물의 원격처리 및 관리, 정보교환 등 다양한 기능을 제공한다.

그리고 물체를 사이에 두고도 정보를 수신할 수 있다는 점에서 사용하기에 쉽다. RFID 시스템은 따로 전력이 필요하지 않아, 교통카드와 버스정류장, 고속도로와 같이 실생활에서 유용하게 쓰이고 있다. 이러한 RFID 기술은 반도체 기술의 지속적인 발전에 의한 컴퓨팅 능력의 발전과 네트워크의 기반으로 현재는 상용화 되는 사물 네트워크([2)]IoT) 쪽의 핵심 기술로써 부각이 되고 있다. 그래서 시중에 널리 상용화 된 RFID가 보안상에 문제가 생긴다면 피해가 클 것이다.

그래서 우리는 RFID의 취약점 중에서 가장 취약하다고 생각하는 RFID Skimming 해킹 기법에 대한 고찰을 시작하게 되었다.

II. 사례

2.1 전자 소매치기[1]

현재 대부분 신용카드는 RFID 기술로 만들어져 있다. 신용카드 결제 단말기와 접촉하면 결제가 끝난다는 점을 악용한 사례이다.

지하철이나 공연장 등과 같이 사람들이 많이 있는 곳에서 신용카드 결제 단말기 장치에 자신이 원하는 금액을 입력시켜두고 신용카드를 가지고 있는 사람에게 접근하여 신용카드가 있는 위치에 신용카드 결제 장치를 갖다 대어 손쉽게 돈을 탈취시키는 것이다.

물론 한 번에 많은 양의 돈을 결제할 수 없도록 돼 있기에 피해 금액이 적은 편이다. 하지만 적은 양의 금액인 경우 서명이 필요하지가 않기에 안전하다고 볼 수 없다.

1) RFID(Radio Frequency IDentification)
2) IoT(Internet of Things)

2.2 [3]ATM 카드복제 장치[2]

신용카드를 가지고 있는 사람들이 자연스럽게 카드를 넣게 되는 곳은 ATM기기이다. 사람들은 자연스럽게 ATM기기에 신용카드를 넣게 되는데 범죄자가 이러한 점을 노려서 신용카드 단말기인 ATM기기에 사용되는 카드 판독기처럼 생긴 카드 복제 장치인 스키머(Skimmer)를 달아 놓은 것이다.

특히나 스키머는 평범한 카드 결제 단말기와 비슷한 형태이기에 당하는 사람은 이를 모를 수 있기에 그 심각성은 크다.

III. 해킹 원리 및 대응방안

3.1 해킹의 기본 원리



그림 [3.1-1] RFID의 원리[3]

기본원리는 RFID의 원리[3]와 같다.

① RFID 리더에서 안테나로 신호를 전송한다.
② 안테나가 신호를 전파의 형태로 생성하여 전파한다.
③ RFID 카드의 IC칩에서 공진 주파수에 맞는 전파를 자체 안테나로 수신하다.
④ RFID 카드의 IC칩은 에너지를 공급받고 정보를 다시 전파의 형태로 전파한다.
⑤ 리더에 달린 안테나에서 전파를 수신하여 리더로 신호를 전송한다.
⑥ 리더는 신호를 변환하고 변환된 신호를 자료수집 장치와 호스트 시스템으로 전달한다.

이렇듯이 RFID 시스템은 크게 태그, 리더기(안테나, 리더), 호스트로 나누어져 있다. 여기에서 공격자는 RFID 카드의 주파수 대역에게 맞는 리더기를 준비하여 별다른 보안 없이 RFID의 정보를 읽어오는 것이다. 여권에도 RFID가 있지만 이마저도 인터넷 검색을 통해서도 손쉽게 여권의 정보를 얻어 낼 수 있다.

3.2 대응 방안

RFID Skimming 해킹 기법에 대응할 방법이 있다. 바로 사용자가 카드를 꺼내어 쓰기 전까지 통신을 못 하게 하면 되는 것이다. 이러한 방법을 적용할 때 3.1의 ③ 번 “RFID 카드의 IC칩에서 공진 주파수에 맞는 전파를 자체 안테나로 수신하다.”에서 막으면 된다는 것이다.

전파를 막는 방법으로는 주파수의 진폭보다 더 촘촘한 철망을 사용하거나 일정 두께 이상의 철제 소재로 감싸면 전파수신을 막을 수 있다.

또는 RFID 방식의 신용카드 대신 [4]IC 카드를 이용하여 암호화된 통신을 이용하거나 접촉식 IC 카드를 이용하여 무선통신 기능을 사용하지 않는 신용카드를 이용함으로써 이러한 문제점들을 해결할 수 있다.

3.3 대응 사례

3.3.1 쿠킹호일을 통한 대응 방안[4]



그림 [3.3.1-1] 쿠킹호일을 통한 대응 방안[4]

RFID 통신을 간단하게 쿠킹호일로 감싸는 방법으로 전파의 수신을 막을 방법이 있다. 만약 집에 알루미늄 테이프가 있다면 지갑 안쪽에 알루미늄 테이프를 붙임으로써 간단한 차폐 지갑을 완성 시킬 수 있다.

3.3.2 차폐 필름 이용하기

3) ATM(Automated teller machine)

4) IC 카드를 스마트카드라고 표현하기도 한다.



그림 [3.3.2-1] 차폐 필름이 있는 카드[5]

RFID의 통신을 막을 방법 중 주파수 해당 전파의 주파수 진폭보다 작은 철망으로도 전파를 막아낼 수 있다. 이러한 원리를 이용하여 개발된 차폐 필름이 있다.

RFID의 원리의 특성상 전파만 막아도 카드를 읽을 수 없기에 물리적으로 막을 수 있는 간단한 방법이다.

또한, 향후 연구과제로 보자기 천 형태의 차폐 필름을 만들어 보고 싶다. 이러한 제품이 만들어지게 된다면 제품 태그가 많이 모여 있는 창고에서 간단히 태그 부분을 덮어버리면 함부로 태그의 정보를 읽어오기 힘들기에 보안성을 높이는데 좋은 효과를 낳을 것으로 기대가 된다.

3.3.3 IC 카드[6] 사용하기

IC 카드는 단순한 정보의 저장은 물론이고 간단한 연산까지 가능한 작은 집적회로가 탑재된 카드를 말하는 것이다.

접촉식의 경우 8개의 접점을 통한 정보의 송수신이 이루어지며, 비접촉식의 경우 전자기 유도 원리에 기초를 둔 [5]RF 통신 방식으로 데이터를 송수신한다.

기존 [6]MS 카드보다, 내장된 초소형 마이크로칩은 암호화 [대칭키 + 공개키 암호 적용] 및 보안이 기존 MS 보다 강력하여, 해킹과 위변조, 카드 복제에 상당한 어려움을 겪으므로, 효과적인 대안이 될 것이라 예상된다.

3.3.4 육안으로 스키머 구분하기[7]

카드 스키밍은 카드가 단말기에 정상적으로 삽입되는 공간에 부착하거나 그 위에 부착하여 정보를 읽어 낸다. 즉 카드가 삽입되는 공간을 잘 살펴보길 바란다. 또한 상점에서의 실제 지불기계 단말기 위 가짜 오버레이 부품을 설치하여 스키밍을 진행하는, 이 방법을 오버레이 스키밍(Overlay Skimming)이라고 한다.



그림 [3.3.4-1] 스키밍 장치 [6]

카드 삽입구 확인: 오버레이 스키밍의 경우 부품은 부피를 많이 차지하므로 단말기의 카드 삽입구간이 특히 크게 보이는 경우에는 의심해 볼 필요가 있다.

백라이트 확인: 스키밍 장치가 키패드 백라이트를 가려 꺼져 있을 경우, 주의가 필요하다

카드인식률 확인: 스키밍 장치를 설치한 경우 카드 인식이 잘 안 되어, 여러 번 결제를 시도하는 경우도 있다.

동작 표시등 확인: 비접촉식 방식의 카드의 경우 읽을시 에 초록색 LED가 작동해야 하나, 키보드 백라이트처럼 동작하지 않을 경우, 이 또한 의심해야 할 징후이니 주의가 필요하다.



그림 [3.3.4-1] 스키밍 장치 [6]

카드 삽입 깊이 확인: 스키머 두께 때문에 리더기에 카드가 너무 깊숙이 들어가는 경우가 발생한다.

5) RF(Radio frequency): 무선 주파수의 약칭
6) MS 카드(Magnetic Stripe Card): 자기장을 띄는 검은색 띠에 정보를 기록한 카드이다.

IV. 결론

RFID Skimming 해킹 기법을 통해 손쉽게 사용자의 돈을 빼가거나 개인정보를 탈취하는 등 다양한 문제가 발생할 수 있는, 무시할 수 없는 취약점이다.

본 논문에서는 RFID Skimming 해킹 기법의 취약점으로 인해 발생할 수 있는 사례를 다루고, 해당 취약점을 막기 위해 어떤 대책 방안이 있고, 어떻게 대처할 수 있는지 알아보았다.

마지막으로, 취약점을 막기 위해 다양한 보호 기법, 대책 방안과 같은 기술적 대처 방법, 대책 방안들도 중요하지만, 사용자가 이러한 취약점이 있다는 걸 숙지하고, 사용 시 좀 더 주의를 기울여 사용하게 된다면, 보다 효과적으로 해당 취약점으로 인한 피해를 줄일 수 있을 것이다.

[참고문헌]

[1] 전자소매치기 관련 뉴스
http://www.yonhapnews.co.kr/bulletin/2016/02/23/0200000000AKR20160223127200797.HTML

[2] ATM 카드복제 장치 관련 뉴스
http://news.mk.co.kr/newsRead.php?year=2015&no=628391

[3] RFID 기본원리
http://koc.chunjae.co.kr/Dic/dicDetail.do?idx=10078

[4] 쿠킹호일을 통한 대응방안
http://wezard4u.tistory.com/3646

[5] 차폐 필름이 있는 카드
http://skimblack.tistory.com/category/?page=1

[6] IC 카드
https://ko.wikipedia.org/wiki/%EC%8A%A4%EB%A7%88%ED%8A%B8%EC%B9%B4%EB%93%9C

[7] 육안으로 스키머(Skimmer) 구분하기
https://mic.com/articles/166615/heres-what-a-credit-or-atm-card-skimmer-looks-like-and-the-one-trick-to-use-to-protect-yourself#.fMMuQ71gX

[8] RFID 보안 및 주요 응용 서비스
http://www.boannews.com/media/view.asp?idx=13487

[9] 스마트 카드와 암호 | 공주대학교 응용수학과 정보보호연구실 서창호

https://www.google.co.kr/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwjllYfwm_3WAhXINpQKHfNPDZMQFgglMAA&url=http%3A%2F%2Fwww.nor.co.kr%2Fxe%2F%3Fmodule%3Dfile%26act%3DprocFileDownload%26file_srl%3D538%26sid%3Da7b88d6c5b8f02988e2905c0872600df&usg=AOvVaw3vBh5il316HGj4u2FxRfnG

[10] 이제 신용카드에도 '스마트' 바람이 분다.
http://blogsabo.ahnlab.com/1418

[11] 유비쿼터스 컴퓨팅의 핵심 RFID HANDBOOK Klaus Finkenzeller (지은이) | 이근호 | 한호연 || 강병권 | 조영빈 (옮긴이)

미공개 된 타원곡선 암호시스템에서의 파라미터 탐지 방법

조현주, 최태균

하나고등학교

A Parameter Detection Algorithm from Unveiled Elliptic Curve Cryptosystem

Hyunjoo Cho, Taegyun Choi

Hana Academy Seoul

요 약

소수체 상의 타원곡선 암호시스템에서 타원곡선 위의 점이 일정 개수 주어졌을 때 미공개 된 파라미터를 탐지하는 방법을 제안한다.

I. 서론

1976년 디피와 헬만은 두 사용자가 안전하지 않은 네트워크 상에서 비밀키를 교환할 수 있는 디피헬만 키 교환 프로토콜을 제안했다.[1] 디피-헬만 키 교환 개념이 도입된 후 국방, 외교 분야 외에도 사회 전반적 분야에 걸쳐 공개키 암호가 사용되었다. 대표적인 공개키 암호로는 RSA와 ECC 등이 있다. RSA 공개키 암호 알고리즘은 1978년 R. Rivest, A. Shamir, L. Adleman에 의해 제시되었고 ECC 암호시스템은 N. Koblitz와 V. Miller[2]에 의해 1985년 각각 제안되었다.

ECC는 다른 공개키 암호시스템과 비교했을 때, 같은 수준의 보안성을 구축하는 데에 요구되는 키의 크기가 비교적 작고 RSA 또는 DSA보다 약 10배 정도 빠른 수행 속도[3]를 보여준다. 따라서 저경량, 초고속이라는 ECC의 장점은 전송량과 계산량이 비교적 열악한 무선 환경에서 특히 돋보이는데 이러한 장점에 의해 사물인터넷(IoT)에서도 타원곡선 암호에 대한 연구가 더욱 활발하게 진행되고 있다. 최근 사물인터넷[4]은 만물을 네트워크로 연결하여 유비쿼터스 시대를 향한 새로운 패러다임을 제시하고 있는데 현재의 암호시스템은 사용하는 키의 크기가 크다는 점 등 제약이 존재한다. 이에 비해, 타원곡선 암호시스템의 경우 사용하는 비트가 적기 때문에 IoT에서의 쓰임에 적합하다.[5] 이 외에도 ECC는 무인자동차 표준화와 비트코인과 같이 실생활에서 그 쓰임이 더욱 증가하고 있다.

현대 암호시스템은 케르크호프스(Kerckhoffs)의 원리[6]에 따라 암호 시스템의 비밀성은 키의 비밀성에만 의존해야 함을 원칙으로 하고 있다. 다시 말해 암호 시스템이 알려지더라도 보안성이 유지되어야 한다는 것이다. 국방, 외교에서는 좀 더 안전성 확보위해 시스템 파라미터를 공개하지 않고 암호 체계를 사용하는 경우가 있다. 키 이외의 모든 정보를 반드시 공개해야 한다는 것은 아니지만 이를 보안을 위한 수단으로써 사용하는 것은 한계가 있다.

본 논문은 ECC에서 파라미터를 감추는 것이 보안에 기여하는지 여부를 확인해보고자 한다.

파라미터를 미공개하여도 얼마 정도의 정보, 즉 타원곡선 상의 점이 일정 개수 주어졌을 때 파라미터를 역추적하는 방법을 분석해보았다. 2절에서는 소수체 상에서 타원곡선 암호시스템의 미공개 된 파라미터 탐지 방법을 직접 구현해 본다. 3절에서는 2절에서의 알고리즘을 예시를 통해 확인해본다.

II.타원곡선 암호 시스템의 미공개 된 파라미터 탐지 방법

대부분의 타원곡선 암호는 소수체 위에서 정의되고 소수체 위의 타원곡선은

$$E(F_p)=\{(x,y)|y^2=x^3+ax+b\}\cup\{O\}$$

로 정의되며 O는 무한원점이다. 이 때, 우리가 구해야하는 파라미터는 타원곡선 방정식을 결정하는 상수 $a\,(\mathrm{mod}p), b\,(\mathrm{mod}p)$(이하 a,b)와 소수 p(단, $p>3$)이다.

본 논문에서는 F_p상에서 타원곡선 암호시스템에 대해 타원곡선 암호를 사용하는 통신에서 타원곡선 위의 점을 알았다고 가정했을 때 타원곡선의 미공개 된 파라미터를 찾는 방법에 대해서 탐구했다. 이제 점의 개수에 따라서, 타원곡선의 파라미터인 a,b,p를 구할 수 있는지의 방법에 대해 알아보자.

1) 점 1개를 알았을 때

타원곡선 상의 한 점을 (x_1,y_1)이라고 하자. 이 점을 타원곡선의 방정식에 대입하면

$$y_1^2=x_1^3+ax_1+b\,(\mathrm{mod}p)$$

라는 식을 만족한다. 하지만 a,b,p 파라미터에 대한 한 가지의 관계식만 주어졌을 뿐, 더 이상의 정보를 얻을 수 없으므로 점 1개가 주어졌을 때 타원곡선의 파라미터를 구할 수 없다.

2) 서로 다른 점 2개가 주어질 때

서로 다른 두 점을 $(x_1,y_1),(x_2,y_2)$라고 하자. 각 점을 타원곡선 방정식에 대입하면,

$$\begin{cases}y_1^2=x_1^3+ax_1+b\,(\mathrm{mod}p)\\ y_2^2=x_2^3+ax_2+b\,(\mathrm{mod}p)\end{cases}$$

라는 연립방정식을 얻을 수 있으며, 위 식을 만족하는 정수 c_i가 존재한다. 위 식을 정수 c_i에 대해 정리하면

$$\begin{cases}y_1^2-x_1^3-ax_1-b=pc_1\\ y_2^2-x_2^3-ax_2-b=pc_2\end{cases}$$

위 두 식에서 b를 소거하여 연립하면

$$a(x_2-x_1)+(y_1^2-y_2^2-x_1^3+x_2^3)=p(c_1-c_2)$$

라는 결과를 얻는다. 이 때, a를 제외한 좌변은 상수이고, 우변은 p의 배수라는 사실을 얻을 수 있지만, p의 값이 정해지지 않았기 때문에 a,p를 구하기 위해서는 정보가 부족하다. 따라서 일반적으로 서로 다른 두 점이 주어질 때는 소수 p를 구할 수 없으므로 타원곡선의 파라미터를 구할 수 없다. 하지만, 특수한 경우에서는 타원곡선 위의 두 점의 덧셈공식을 통해 타원곡선 위의 다른 점을 구할 수 있다. 서로 다른 두 점 $(x_1,y_1),(x_2,y_2)$에 대해, 두 점을 통해 얻을 수 있는 타원곡선 위 다른 점의 좌표 (x_3,y_3)는

$$x_3=(\frac{y_2-y_1}{x_2-x_1})^2-x_1-x_2\,(\mathrm{mod}p),$$

$$y_3=(\frac{y_2-y_1}{x_2-x_1})(x_1-x_3)-y_1$$

이다. 본 연구에서는 p의 값이 정해지지 않았으므로, $\frac{y_2-y_1}{x_2-x_1}$의 값이 정수로 나타나지 않는다면 분수의 모듈러 값을 구할 수 없다. 즉, $\frac{y_2-y_1}{x_2-x_1}$이 정수이면 타원곡선의 덧셈정리를 활용해 또 다른 점 (x_3,y_3)을 구해 세 점이 주어진 상황에서 타원곡선의 파라미터를 구하는 방법을 사용할 수 있고, 자세한 방법에 대해서는 바로 다음에 다룰 것이다. 결론적으로, 서로 다른 2개의 점이 주어졌을 때는 서로 다른 두 점 $(x_1,y_1),(x_2,y_2)$에 대해 $\frac{y_2-y_1}{x_2-x_1}$이 정수인 특수한 경우를 제외하면 타원곡선의 파라미터를 구할 수 없다.

3) 점 3개가 주어질 때

서로 다른 세 점을 $(x_1,y_1),(x_2,y_2),(x_3,y_3)$라고 하자. 각 점을 타원곡선 방정식에 대입하면,

$$\begin{cases}y_1^2=x_1^3+ax_1+b\,(\mathrm{mod}p)\\ y_2^2=x_2^3+ax_2+b\,(\mathrm{mod}p)\\ y_3^2=x_3^3+ax_3+b\,(\mathrm{mod}p)\end{cases}$$

라는 연립방정식을 얻을 수 있으며, 이를 만족하는 정수 c_i가 존재하여 다음의 식으로 표현할 수 있다.

$$\begin{cases} y_1^2 - x_1^3 - ax_1 - b = pc_1 \cdots ① \\ y_2^2 - x_2^3 - ax_2 - b = pc_2 \cdots ② \\ y_3^2 - x_3^3 - ax_3 - b = pc_3 \cdots ③ \end{cases}$$

위의 식을 두 개씩 연립하면,

$$\begin{cases} ①-②: a(x_2-x_1)+(y_1^2-x_1^3-y_2^2+x_2^3)=p(c_1-c_2)\cdots④ \\ ①-③: a(x_3-x_1)+(y_1^2-x_1^3-y_3^2+x_3^3)=p(c_1-c_3)\cdots⑤ \end{cases}$$

라는 결과를 얻는다. 식④, ⑤를 연립하여 a를 소거하면 다음과 같은 식을 얻는다.

$$(y_1^2-x_1^3-y_2^2+x_2^3)(x_3-x_1)-(y_1^2-x_1^3-y_3^2+x_3^3)(x_2-x_1)$$
$$=p\{(c_1-c_2)(x_3-x_1)-(c_1-c_3)(x_2-x_1)\}$$

이 때, 우변이 p의 배수이므로 좌변의 값도 p를 소인수로 가짐을 알 수 있다. 따라서 좌변의 값에 대한 소인수를 구한 후 각각의 소인수를 구한 후 p의 값을 도출할 수 있다. 하지만, 우변의 수가 커질 경우 그 소인수가 아주 커질 수 있기에 이 자체로부터 정수론의 소인수분해의 문제가 발생할 수 있다. 그러므로 p의 값을 구하는 것이 어려워질 수 있다.

4) 서로 다른 점 4개가 주어졌을 때

서로 다른 점 4개가 주어졌을 때는 서로 다른 점 3개가 주어졌을 때보다 더 확실한 p의 값을 도출해낼 수 있다. 서로 다른 점 4개를 3개씩 총 4쌍으로 묶어 각각의 쌍마다 3)의 과정을 반복하면, p를 소인수로 가지는 서로 다른 수를 최대 4개까지 구할 수 있다. 4개의 정수 모두 p를 소인수로 가지기에, 4개의 정수의 gcd를 구해 그 gcd의 소인수 중 p의 값을 구하면 보다 정확하게 p의 값을 구할 수 있다. 서로 다른 점이 4개보다 많을 때에는 p를 소인수를 가지는 더 많은 정수를 구할 수 있으므로, 이들의 gcd를 구하면 p의 값에 정확하게 접근할 수 있다는 사실을 확인할 수 있다.

III. 예시

$y^2 = x^2 + 18x + 37 \pmod p$의 타원곡선 방정식에서 점$(4,1), (10,23), (2,34), (6,19)$가 타원곡선 위의 점임을 전제로 점의 개수에 따라 파라미터를 직접 찾아보자.

1) 점 1개 $(4,1)$가 주어졌을 때

타원곡선의 방정신에 점을 대입하면,

$$4a+b+63 = pc_1, \text{ (단 } c_1\text{은 정수)}$$

라는 관계식 1개만 얻을 수 있으며, 파라미터 a, b, p의 구체적인 값을 구할 수 없다.

2) 점 2개 $(4,1),(10,23)$가 주어졌을 때

점을 타원곡선에 대입해 정리하면, 정수 c_i에 대해

$$\begin{cases} 4a+b+63 = pc_1 \\ 10a+b+471 = pc_2 \end{cases}$$

이고, 두 식을 연립해 b를 소거하면

$$6a+408 = p(c_2-c_1)$$

이다. p는 3보다 큰 소수이므로

$$a+68 = p\frac{(c_2-c_1)}{6}$$

이며, $\frac{c_2-c_1}{6}$은 정수라는 것을 구할 수 있다. 그러나 이 역시 a, b의 p에 대한 관계식만 구할 수 있을 뿐 p의 값은 구할 수 없다. 또한, $(4,1)$과 $(10,23)$을 합했을 때

$$\frac{y_2-y_1}{x_2-x_1} = \frac{11}{3}$$

정수의 점이 나오지 않기에 새로운 점 (x_3, y_3)를 구할 수 없다. 그러므로 이 경우, 점 2개가 주어졌을 때 파라미터를 구할 수 없다.

3) 점 3개 $(4,1), (10,23), (2,34)$가 주어졌을 때

이 또한 타원곡선의 식에 대입한 후

$$\begin{cases} 4a+b+63 = pc_1 \\ 10a+b+471 = pc_2 \\ 2a+b-1148 = pc_3 \end{cases}$$

두 식씩 연립해 b를 소거하면,

$$\begin{cases} 6a+408 = p(c_2-c_1) \\ 2a+1211 = p(c_1-c_3) \end{cases}$$

이다. 이 식 또한 연립하여 a를 소거하면

$$3225 = p(4c_1-c_2-3c_3)$$

라는 식이 도출된다. 즉, 좌변에 있는 3225가 p를 소인수로 가짐을 의미하고, 3225를 소인수분해하면 $3225 = 3\times5^2\times43$이다. 이 때, $p>3$이므로 p는 5 또는 43이다. 하지만 $p=5$를 대입할 경우 $(10,23)$에 대해 $9 \neq 37 \pmod 5$이므로 $p=43$임을 알 수 있다. 주어진 점을 대입하여

모듈러 연산을 진행하면 $a=18(\mathrm{mod}43)$, $b=37(\mathrm{mod}43)$로 파라미터를 모두 구할 수 있다.

하지만 43은 비교적 작은 소수이기 때문에 세 점과 소인수분해를 통한 간단한 대입만으로도 바로 구할 수 있었지만, 암호학에서 쓰이는 수처럼 큰 수가 사용될 때는 인수분해가 어렵기 때문에 다른 점까지 고려해 gcd를 구해야 한다.

4) 점 4개 (4,1), (10,23), (6,19), 이 주어졌을 때

앞서 했던 것처럼,

$$\begin{cases} 4a+b+63=pc_1 \cdots ⑥ \\ 10a+b+471=pc_2 \cdots ⑦ \\ 2a+b-1148=pc_3 \cdots ⑧ \\ 6a+b-145=pc_4 \cdots ⑨ \end{cases}$$

위의 식을 3개씩 연립하여 3)의 과정과 같이 소거하는 작업을 통해 p를 소인수로 가지는 정수를 찾은 후 gcd를 구하여 p를 찾는다. 각각 ⑥, ⑦, ⑧를 연립하면 $p=3225$, ⑥, ⑦, ⑨을 연립하면 $p=1032$, ⑦, ⑧, ⑨를 연립하면 $p=387$을 알 수 있었다.

$$\gcd(3225,1032,387)=129$$

$$129=3\times 43$$

이고 $p>3$인 소수이므로 바로 $p=43$임을 알 수 있다. 3)의 과정과 같이 점을 대입하여 a,b의 값을 구하면 3)의 결과와 같이 $a=18(\mathrm{mod}43)$, $b=37(\mathrm{mod}43)$의 값을 얻었다. 따라서 타원곡선 상의 서로 다른 점을 많이 얻을수록 p의 값을 보다 정확히 구할 수 있다.

IV.결론

본 연구에서는 소수체 상에서 정의된 타원곡선 암호에서의 미공개 된 파라미터 탐지 방법에 대한 연구를 진행했다. 타원곡선 위의 한 점을 알 때에는 타원곡선의 파라미터를 구할 수 없었다. 두 점을 알 경우, 특수한 상황, 즉 타원곡선 상의 덧셈으로 새로운 정수로 이루어진 타원곡선 상의 점을 찾아낼 수 있을 때에만 타원곡선의 파라미터를 탐지할 수 있었다. 세 점 이상의 점이 주어졌을 경우, 타원곡선의 파라미터를 구할 수 있었으며, 더 많은 점을 알아낼 수 있었을 때 최대공약수를 활용하여 정확하게 타원곡선 파라미터를 찾아낼 수 있었다.

따라서 케르크호프스의 원리에서 '키를 제외한 모든 시스템의 다른 모든 내용이 알려지더라도 암호체계는 안전해야한다.'라는 의미를 확인하게 되었으며 암호체계에서 단순히 파라미터를 숨겨서 안전성을 확보하는 것이 아니라, 키의 안전성을 기반으로 하여 암호의 안전성을 확보하는 것이 중요하다.

[참고문헌]

[1] Diffie, W., and M. Hellman. "New directions in cryptography." IEEE Transactions on Information Theory, vol. 22, no. 6, 1976, pp. 644 - 654.

[2] Ronald Rivest, Adi Shamir, Leonard Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.", Communications of the ACM, 1978.

[3] Miller, Victor S. "Use of Elliptic Curves in Cryptography." Lecture Notes in Computer Science Advances in Cryptology — CRYPTO '85 Proceedings, 1985, pp. 417 - 426.

[4] 강병석, 추현승, "최근 사물인터넷 및 드론 기술 동향에 관한 연구", 한국통신학회, 2017.

[5] 최준혁, 손동현, 권태경, "자원 제약적인 사물인터넷 장비를 위한 타원곡선 암호와 하드웨어 가속 기반의 보안 방안", 한국통신학회, 2015.

[6] Auguste Kerckhoffs, 'La cryptographie militaire', Journal des sciences militaires, vol. IX, pp. 5 - 38, Jan. 1883, pp. 161 - 191, Feb. 1883.

Mac OS X의 디지털 증거 수집 방식 결정 방안 제시*

이준수,[1†] 신희재,[2] 조재민,[3] 강성철,[3] 박진우[4]

[1]경기과학고등학교, [2]세종대학교, [3]서울과학기술대학교, [4]경기대학교

Determination of Digital Evidence Collection Method in Mac OS X

Jun-Su Lee,[1†] Hee-Jae Shin,[2] Jae-Min Cho,[3] Seong-Cheol Kang,[3] Jin-Woo Park[4]

[1]Gyeonggi Science High School, [2]Sejong University, [3]Seoul National University of Science and Technology, [4]Kyonggi University

요 약

압수 수색 절차 중 디지털 증거를 수집하는 과정에서 안티 포렌식 혹은 침해사고의 정황이 발견되었을 경우 원칙인 선별압수가 아닌 대상 시스템의 디스크를 이미징하는 방식을 따라야 할 필요성이 있다. 하지만 Windows 이외의 타 OS의 경우 증거 수집 방식을 결정하는 데에 있어서 충분한 논의가 이루어지지 않고 있다. 이에, 본 논문에서는 조사 대상 시스템의 운영체제가 Mac OS X일 때 대상 시스템의 디스크의 이미징 여부를 결정하는 데에 중요한 영향을 미치는 안티 포렌식 혹은 침해사고의 정황을 파악하는 데에 어떤 요소를 참고할 수 있는지에 대하여 다루었다. 본 연구의 결과는 수사관이 참고할 수 있는 구체적인 정보를 제시하는데 활용되어 실무 수사의 효율성을 높일 수 있을 것으로 기대된다.

I. 서론

디지털 증거를 수집하는 방식은 크게 선별압수와 물리 이미징/복제(이하 '이미징'이라 한다)로 나누어진다. 선별압수의 경우에는 논리적인 단계에서 파일의 복제를 통해 증거를 수집하는 반면 이미징의 경우 대상 이미지 자체를 물리적으로 복제하여 dd 등의 포맷 등으로 수집한 뒤 추후 분석이 이루어진다 [1]. 압수영장을 발급받은 뒤 디지털 증거를 수집할 때 선별압수 절차를 따르는 것이 원칙이나 여의치 못한 사정이 발생할 경우 수사관은 대상 시스템의 디스크를 이미징하여 증거를 수집한 뒤 분석을 시행할 수 있다 [2].

하지만 최근의 판례로 인해 압수영장을 발급받아 디지털 증거를 수집하는 과정의 양상은 이전과 크게 달라졌다. 이전까지는 이미징을 통한 증거 수집의 시행 빈도가 잦았으나 2015년도 '종근당 압수 및 수색 사건'에서 피의자의 인권 보호 등을 이유로 피의자의 하드디스크 전체를 이미징하여 분석한 뒤 얻어낸 증거의 법적 효력을 무력화하는 판례가 등장한 이후로 선별압수 우선의 원칙이 확실하게 정립되었다고 해도 과언이 아니다 [3].

그럼에도 불구하고 수사관이 선별압수의 원칙을 따르는 대신 저장 매체 자체를 확보하거나 저장 매체를 이미징하여 수집할 수 있는 몇 가지 예외적인 사항이 존재한다. 이들에는 용의자가 안티 포렌식(Anti-Forensic) 도구를 사용했거나 사건이 침해사고에 연루되었을 가능성이 있을 때가 속한다. 본 경우들에서는 시스템의 상태를 최대한 그대로 유지한 뒤 분석하는 것이 우선되기 때문이다. 이 이유로 효율적인 수사를 위해 적합한 증거 수집 방식을 채택하는 것이 중요하다고 할 수 있다.

* 본 논문은 한국정보기술연구원 BoB 프로젝트의 지원으로 작성되었습니다.

† 주저자.

Windows의 경우 증거수집 도중 이미징의 필요성을 파악하기 위한 논의가 활발히 이루어지고 있다 [4]. 반면 낮은 점유율과 낮은 수사 빈도의 문제로 인해 Mac OS X(이하 'OS X'라 한다)와 같은 타 OS의 경우는 그렇지 않은 경향을 보이고 있다. 하지만 타 OS에서도 디스크 이미징을 시행해야 하는 상황을 배제할 수 없으므로 충분한 논의가 이루어져야한다고 볼 수 있다.

이에, 본 논문에서는 OS X의 증거 수집 과정에서 수사관이 선별압수 방식을 취하는 중 대상 시스템의 디스크를 이미징 해야 할 필요성을 인지하고 대상 시스템에 대한 수사 방식을 결정하는 방식에 대해 논의한다.

II. 시스템 이미징 결정 여부

대상 저장매체를 이미징 하는 경우는 크게 안티 포렌식 도구의 사용 여부와 침해사고 연루의 여부로 나뉜다. 안티 포렌식의 흔적이 발견되었을 경우에는 추후 증거를 복원 혹은 추출하기 위한 작업인 파일 카빙(File Carving, 이하 '카빙'이라고 한다)을 위해 이미징을 시행한다. 침해사고가 연루되었을 가능성이 있는 경우에는 시스템에 저장된 사건과 관련된 정보인 아티팩트(Artifact)를 보존하고 분석하기 위하여 이미징이 실시된다.

2-1. 안티 포렌식

용의자가 범행의 흔적을 숨기기 위해 증거를 은닉, 암호화, 조작하는 등의 동작을 취하는 것을 안티 포렌식이라고 일컫는다. 안티 포렌식이 이루어지면 추후 대상 시스템에 대한 분석과 카빙을 통하여 은닉 혹은 변형된 증거를 얻어야할 필요성이 있다. 때문에 증거 수집 시 대상 시스템의 디스크를 이미징하거나 원본 디스크를 압수하는 절차가 이루어진다. 안티 포렌식의 정황에는 주요 시스템 파일의 와이핑, 안티 포렌식 툴의 동작 기록, 시스템 포맷, 파티션 분리 혹은 은닉 등이 있다.

2-2. 아티팩트

범행이 침해사고와 연루되었을 때 시스템에서 남긴 정보를 토대로 하여 정확한 정황을 파악해야 할 필요성이 있다. 이 때 시스템과 관련된 정보를 담고 있는 파일 혹은 경로를 통틀어 아티팩트라고 한다. 침해사고의 정황은 프로그램 실행 기록 혹은 백그라운드 프로세스, 네트워크 접속 기록 등의 침해의 흔적과 사건의 경황, 침해를 당한 것으로 생각되는 피해자의 주장 등으로 파악할 수 있다. 만약 침해사고가 일어난 것으로 파악될 경우 추후 보존된 아티팩트를 통해 분석 절차가 이루어져야하기 때문에 시스템 디스크의 이미징을 시행한다.

III. OS X의 증거 수집 방식 결정

지금까지 정리한 시스템 디스크 이미징 결정 여부가 OS X에서 수사 방식을 결정하는 기존의 방식에 어떻게 적용될 수 있는지 구체적인 예시를 제시한다.

기존 압수수색 영장이 발급된 이후 디지털 증거의 수집 절차는 다음과 같다. ①디지털 증거 압수의 과정 중 피압수자와 변호인 등 제 3자의 참관 여부를 확인한다. ②포렌식 툴을 이용하여 영장 발부 사유와 관련된 데이터를 선별한다. ③선별한 압수 파일을 체계화하여 소유자, 소지자, 보관자 및 이에 준하는 당사자들에게 교부한다.

위의 과정 중 본 논문에서는 ①과 ②의 과정 사이에서 포렌식 도구를 통해 해당하는 경우가 발견되었을 때 안티 포렌식 동작 혹은 침해사고의 가능성을 인지한 이후 증거 수집 방식을 결정할 수 있도록 하는 내용을 제시한다.

다음에서는 OS X에서 안티 포렌식과 침해사고의 여부를 파악하기 위해서 시스템의 어떤 부분을 참조할 수 있는지에 대해서 다룬다.

3-1. 안티 포렌식 감지

안티 포렌식의 여부는 안티 포렌식 도구를 사용한 흔적과 주요 증거 파일이 수집하기 어려운 형태로 저장되었을 때로 나뉠 수 있다.

안티 포렌식 도구를 사용한 정황은 시스템 로그와 같은 시스템 로그와 같은 시스템 파일들이 임의로 삭제되었을 때와 안티 포렌식 도구를 사용한 기록이 대상 시스템에 남아있는지

를 통해 알 수 있다. OS X에서 대표적으로 사용되는 안티 포렌식 툴에는 BitDefender 등의 안티바이러스(시스템 기록 삭제), FileVault 등의 디스크 암호화 프로그램, 루트킷(프로세스, 파일 은닉) 등이 있다. 설치된 프로그램들의 목록 혹은 작동한 프로그램의 목록을 확인함으로써 안티 포렌식 도구의 사용 정황을 파악할 수 있다. 앞서 언급한 사항을 확인하기 위한 경로는 다음과 같다.

프로그램 설치 기록
/Library/Receipts/InstallHistory.plist
설치된 프로그램 목록
/Applications/*

주요 증거 파일이 수집하기 어려운 형태로 저장되었을 때의 경우에는 파티션 분할/은닉을 통한 파일 및 시스템 은닉, 암호화된 파일, 계층을 난해하게 조작한 경우 등이 속한다. 파티션 분할 혹은 은닉이 의심될 경우 드라이브 정보를 확인하거나 Disk Utility를 이용하여 의심되는 파티션의 존재 여부를 확인할 수 있다. 암호화된 파일의 경우 정상적인 열람이 불가한 확장자명 혹은 메타데이터를 가졌을 때 그 여부를 의심할 수 있다. 또는 RSA, base64 등의 전형적인 암호화 알고리즘의 결과 암호문이 보이는 경향성과 파일 내용의 분포 경향성을 계산하여 그 값이 일정할 때를 생각할 수 있다. Bootcamp, VMWare 등의 가상 운영체제 혹은 zip, rar 등 전형적인 압축 형식의 파일 내부에 증거 파일이 있을 것으로 파악될 경우 또한 파일을 은닉하기 위한 시도로 의심할 수 있다. Bootcamp의 경우 해당 프로그램이 설치된 파티션의 정보를, VMWare Fusion의 경우 설치 경로의 파일을 확인할 수 있다.

VMWare Fusion에서 가상 머신 파일들이 저장되는 경로
~/Documents/Virtual Machines.localized

3-2. 침해 사고 감지

침해사고는 대개 사건의 정황 혹은 피의자의 주장 등을 통해 파악될 수 있다. 이 외에서 대상 시스템에서 구체적으로 사용할 수 있는 기록에는 로그 파일, 실행한 프로그램 목록, 네트워크 접속 정보 등이 있다.

설치된 프로그램의 목록을 확인할 때 IRC, 악의적 스크립트 등 침해사고에 사용되었을 가능성이 큰 프로그램들을 DB화 한 이후 해시값(Hash Value)를 대조하는 방식이 주어질 수 있다. 실행한 프로그램 목록과 시작 프로그램, 백그라운드 프로세스 정보를 담고 있는 파일 및 경로도 마찬가지의 방법이 적용될 수 있다. 네트워크 접속 정보의 경우 해외, VPN 등의 IP 기록이 남아있을 때 의심할 수 있다.

다음 목록은 지금까지 다룬 침해사고를 파악하는 데에 있어서 사용될 수 있는 정보들의 위치를 나타낸다.

시스템 로그
/var/log/*
시스템 시작시 작동하는 프로세스 목록
/Library/StartupItems/*
/System/Library/StartupItems/*
주기적으로 실행되는 프로세스
/usr/lib/cron/jobs/*
/etc/crontab
시스템 설정 확인
/Library/Preferences/*

IV. 결론

4-1. 연구의 의의 및 한계

본 연구에서는 OS X에서 디지털 증거를 수집하는 과정에서 대상 시스템 디스크의 이미징 여부를 결정하는 방식에 대해 다뤘다. 이미징이 필요한 경우를 안티 포렌식과 침해사고가 연루된 경우로 나누어 OS X에 대해 각각의 경우에

대해 실제 수사에서 참조할 수 있는 정보 및 그 경로를 듦으로써 대상 시스템의 디스크를 이미징 하는 데에 있어 판단할 수 있는 기준을 제시했다는 데에 의미가 있다. 또한 수사 방식을 체계적이고 신속하게 결정함으로써 OS X를 대상으로 한 수사의 효율을 높일 수 있을 것으로 기대된다. 하지만 본 연구는 기존 시스템 상의 정보를 참조하는 데에 의존하므로 충분한 논의 혹은 체계화를 통해 관련 정보의 꾸준한 갱신이 이루어지지 않을 경우 추후 적용이 어렵다는 한계가 있다. 이 한계를 극복하기 위해서 대상 정보들의 데이터베이스화와 OS X 대상 현장 기반 분석 도구의 제작과 관련된 연구의 활성화가 해결책으로 제시될 수 있을 것이다.

[참고문헌]

[1] 원용기. 디지털증거에 대한 계층적 접근 방안 연구. Diss. 서울대학교 융합과학기술대학원, 2016.

[2] 형사소송법 제106조 제3항.

[3] 대법원 2015.7.16, 자, 2011모1839, 전원합의체 결정.

[4] 고려대학교 디지털포렌식연구센터. 디지털 증거 처리 가이드라인.

2017 한국정보보호학회 동계학술대회

C1 정보보호모델

권 헌 영 (고려대)

- **Network Address Mutation을 위한 시스템 아키텍쳐 설계**
  안성범(과학기술연합대학원대학교), 우사무엘, 박경민, 문대성 (한국전자통신연구원)
- **차량 내부통신 보안 안전성 평가기술 기반 연구**
  이선우, 이동훈 (고려대학교)
- **기업 서버의 침해 사고 대응을 위한 침해사고 준비도 모델 (Incident Response Readiness Model for Business Servers)**
  신현우, 박성미, 유운선, 황지수 김동현, 한창희 (KITRI BoB), 김종현 (더존 포렌식 센터)
- **보안성이 강화된 SaaS 개발 생명주기 연구**
  정윤송, 이동훈 (고려대학교)

Network Address Mutation을 위한 시스템 아키텍쳐 설계

안성범*, 우사무엘**, 박경민**, 문대성**

*과학기술연합대학원대학교 정보보호공학과

**한국전자통신연구원

Design System Architecture for Network Address Mutation

Seong-Beom Ahn*, Samuel Woo*, Kyung-Min Park*, Dae-sung Moon*

*Division of information security engineering, University of Science and Technology

**Electronics and Telecommunications Research Institute

요 약

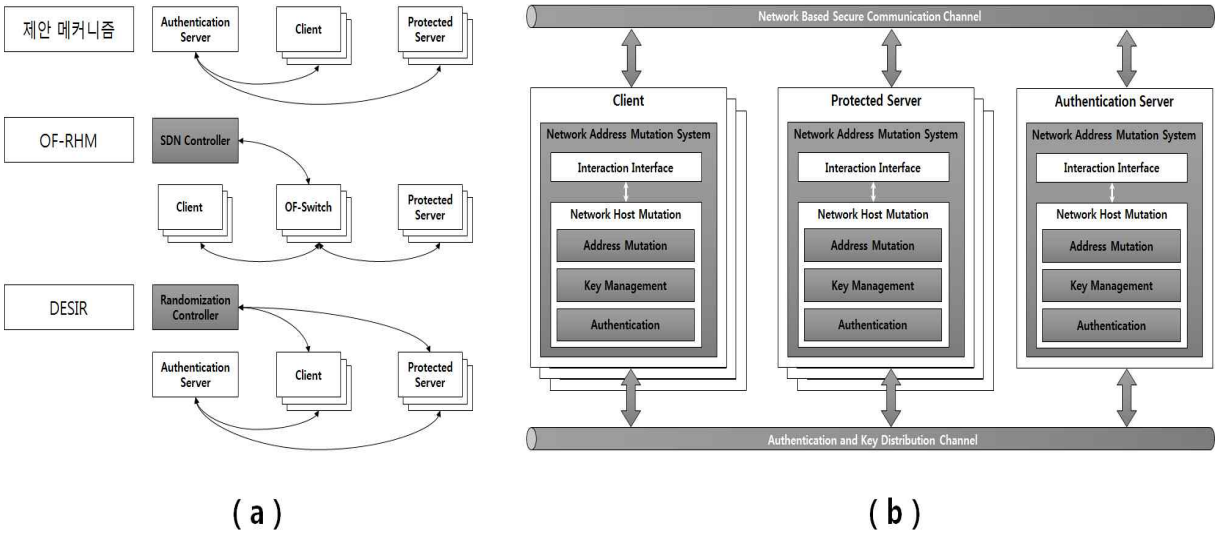
이동 표적 방어(Moving Target Defense)는 공격 대상의 주요 속성을 지속적으로 이동하여 공격에 대한 불확실성을 증가시키는 기술을 말한다. 이동 표적 방어 기술의 여러 범주 중 네트워크 주소 변이(Network Address Mutation)에 대한 연구가 활발히 진행 중에 있다. 네트워크 주소 변이는 보호 대상의 네트워크 주소를 주기적으로 변경하여 정찰 공격에 대한 선제적 방어를 한다. 본 논문에서는 분산 환경 기반 네트워크 주소 변이 시스템 아키텍쳐를 제안한다. 그리고 기존의 대표적인 연구를 선정하고 세 가지 관점으로 안전성과 성능을 분석했다. 그 결과, 우리가 제안한 시스템 아키텍쳐는 기존 연구들에서 발생할 수 있었던 문제점들을 해결하고 성능이 개선 될 것이라고 분석되었다.

I. 서론

사이버 킬체인의 첫 단계는 정찰 공격이다[1]. 공격자는 표적 시스템에 접근하기 위해서 정찰 공격을 한다. 예를 들어, 공격자는 표적 시스템의 취약점을 발견하기 위해 초기 단계에 네트워크 주소 스캐닝을 한다. 스캐닝 이후, 공격자는 표적 시스템이 사용하는 서비스 프로토콜의 알려진 취약점을 악용하여 공격을 한다[2]. 정적 네트워크 구성은 단번의 정찰 공격에 구성 정보가 쉽게 노출되는 위험이 있다. 이와 달리, 주기적으로 변화하는 동적 네트워크 구성은 공격자의 정찰 공격에 대한 비용(시간, 노력 등)을 증가 시킬 수 있다. 이처럼 사이버 공격에 대한 선제 방어 행위를 하는 이동 표적 방어(Moving Target Defense)에 대한 연구가 각광받고 있다.

일반적으로 공격자가 특정 표적에 대하여 공격에 성공할 때 까지 취약점 분석과 침입 시도를 할 수 있는 시간적 이점이 있기 때문에 방어자 입장에서는 상대적으로 불리한 비대칭(Asymmetric) 상황이 발생한다. 이동 표적 방어는 공격 대상의 주요 속성을 지속적으로 이동(변경)하여 공격과 방어 사이의 비대칭 상황을 극복하는 기술을 말한다. 이동 표적 방어는 공격 대상이 되는 공격 표면(Attack Surface)을 주기적으로 변화시킴으로써 노출된 정보를 무력화 시키고 공격에 대한 비용을 증가시킨다[3].

이동 표적 방어 기술의 범주는 다양하지만 네트워크 주소 변이(Network Address Mutation) 기술에 대한 연구가 가장 활발히 진행 중에 있다. 특히 SDN(Software Defined Network) 컨트롤러를 기반으로 네트워크 주소 변이를 수행하는 기술이 주를 이루고 있다[4][5]. SDN 기반 네트워크 주소 변이는 외부에 가상 주소만



[그림 1] (a)제안 기법 및 관련연구 시스템 구성도, (b)제안 기법 세부 기능 블록

노출시키고 실제 주소를 감출 수 있다. 하지만 공격자가 보호대상 서브넷 내부에 존재하는 경우, 해당 서브넷에 소속된 모든 보호대상의 네트워크 주소는 노출되는 심각한 문제점을 내포하고 있다. 본 논문에서는 네트워크 주소 변이에 대한 기존 연구를 분석하였다. 이와 더불어 분산 환경 기반 네트워크 주소 변이 시스템 아키텍처를 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 네트워크 주소 이동에 관련된 연구를 서술하고 3장에서는 익명 IP주소 생성 알고리즘의 상세한 동작 방식과 제안된 알고리즘의 안전성 분석을 다루며, 4장에서는 결론을 내린다.

II. 관련연구

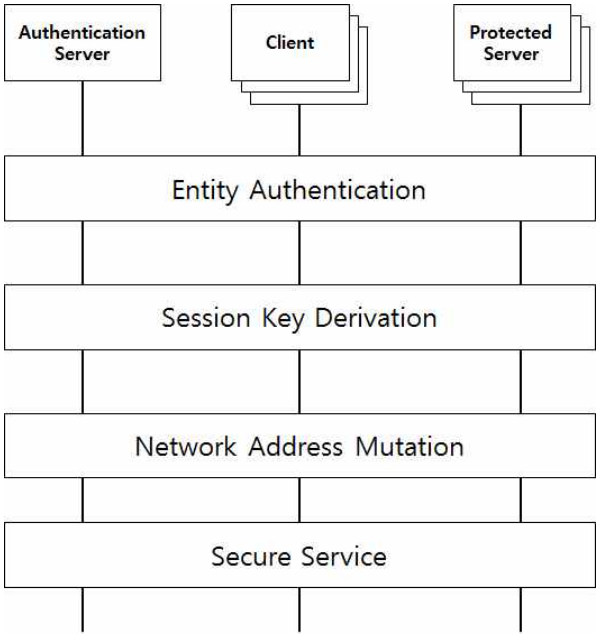
가상 IP주소 이동과 관련된 연구는 OF-RHM(OpenFlow Random Host Mutation)이 있다[6]. OF-RHM 연구는 SDN 컨트롤러를 기반으로 보호하려는 호스트의 가상 IP주소를 변경 한다. 정당한 사용자가 특정 호스트에 접근하기 위해서 OF(OpenFlow) 스위치는 해당 호스트의 가상 IP주소와 실제 IP주소를 맵핑시킴으로써 접속 할 수 있게 한다. OF 스위치 내부 호스트들의 실제 IP주소는 변경되지 않기 때문에 내부 공격자는 공격을 수행 할 수 있다.

기존에 활발히 연구 중인 SDN기반 가상 IP 주소 변이와 달리, DESIR(Decoy-Enhanced Seamless IP Randomization)연구는 실제 IP주소를 변이 한다[7]. DESIR는 Randomization 컨트롤러가 보호대상에게 할당한 IP주소를 선정하고, 보호대상 서버에 적용시킨다. 변경된 보호대상 서버의 실제 IP주소는 인증서버에 전송되고, 정당한 사용자는 인증서버를 통해서 보호대상 서버에 접속하여 IP주소가 변경되더라도 서비스 연결을 보장할 수 있도록 한다.

OF-RHM은 공격자에 의해 OpenFlow 스위치 하부의 모든 호스트들의 실제 IP주소가 노출될 수 있는 치명적인 문제점이 있다. DESIR는 SDN기반 시스템의 문제점을 해결하기 위해 서브넷 단위가 아닌 개체 단위의 실제 IP주소 이동을 적용하였다. 하지만 Randomization 컨트롤러가 단일 장애점(Single Point of Failure)으로 작용한다. 따라서 Randomization 컨트롤러 장애 시 주소 이동 서비스 전체가 마비되는 심각한 문제점이 발생한다. 또한 주소 이동이 될 때마다 서비스에 참여하는 개체 간 통신 오버헤드가 발생한다. 본 논문에서는 주소 생성 방식을 각 개체들로 분산시켜 보안성을 향상시키고 통신을 최소화 하여 성능을 개선시킬 수 있는 시스템 아키텍처를 제안 하였다.

III. 제안 기법

네트워크 주소 변이를 위한 기법은 분산 관



[그림 2] 제안 기법 주소 변이 프로세스

리 기법과 중앙 관리 기법으로 구분 할 수 있다. 중앙 관리 기법은 주소 생성 모듈을 통해 개체 단위의 주소 변경을 통해 유연한 주소 변경이 가능하지만 관련연구에서 언급한 문제점들이 발생할 수 있다. 따라서 본 장에서는 분산 환경 기반 네트워크 주소 변이 시스템 아키텍쳐를 제안하고 기존 연구와의 안전성/효율성을 분석한다.

3.1 분산 환경 기반 네트워크 주소 변이 시스템 구성

분산 환경에서 네트워크 주소 변이를 제공하는 시스템 구성도 및 세부 기능 블록은 [그림 1]과 같다. 제안 기법에서는 네트워크 주소 변이를 사용하는 모든 개체가 보호대상의 IP를 생성할 수 있기 때문에 DESIR에 존재했던 Randomization 컨트롤러를 제거할 수 있다. 이를 통해 부가적인 통신 발생을 최소화 할 수 있다. 또한 SDN 컨트롤러를 기반으로 가상 IP를 생성하지 않기 때문에 OF-RHM 구성 중 SDN 컨트롤러와 OF스위치를 제거하여 내부 공격에 대한 안전성을 강화할 수 있다.

3.2 네트워크 주소 변이 수행 절차

네트워크 주소 변이 기술은 [그림 2]와 같이 4단계로 구성된다.

1. 개체인증(보호대상 서버와 클라이언트들은 인증서버로부터 개체 인증)
2. 세션 키 교환(정당한 개체 간에 비밀값을 공유)
3. 주소변이(네트워크 주소를 생성하고 보호대상의 주소를 변경하는 단계, 주소 충돌 회피를 위한 동기화필요)
4. 보안통신(네트워크 주소 변이를 유지하며 공격자의 분석을 어렵게 하는 행위)

3.3 안전성 분석

본 절에서는 제안 기법과 기존 기법의 안전성과 성능을 분석한다. 우리는 기존 연구들 중 가장 대표적인 두 가지 연구를 선정하고 다음 세 가지 관점으로 안전성과 성능을 분석했다.

1. 내부 공격자로부터의 안전성
2. 통신 오버헤드를 발생하지 않는 통신 효율성
3. 단일장애로부터의 안전성

	내부 공격 안전성	통신 효율성	단일 장애 안전성
OF-RHM	X	O	X
DESIR	O	X	X
Ours	O	O	O

[표 1] 제안 알고리즘 안전성 분석

OF-RHM은 내부 공격자에 의해 해당 서브넷 하위의 모든 호스트의 모든 실제 IP주소가 노출될 수 있다. DESIR는 주소 변경이 발생할 때 마다 주소 생성 모듈과 보호 대상 사이에서 통신이 발생하기 때문에 성능 과부하에 대한 문제점이 발생한다. 마지막으로, OF-RHM과 DESIR는 중앙 집중 방식의 주소 생성으로 인해 단일 장애점 문제가 발생한다. 우리가 제안한 기법은 분산 환경 기반 네트워크 주소 변이를 하기 때문에 기존 연구에서 발생 할 수 있는 문제점들을 해결하여 보다 더 안전한 것으로 분석된다.

IV. 결론

본 논문은 네트워크 주소 변이를 위한 분산

기반 시스템 아키텍쳐를 제안하였다. 기존 연구들은 중앙 집중 주소 생성 기법을 사용하였다. 하지만 중앙 집중 주소 생성 기법은 공격자에 의해 주소 생성 모듈이 노출되면 전체 시스템의 정보가 노출되는 단일 장애점을 갖고 있기 때문에 보안에 취약하다. 본 논문에 제안된 시스템 아키텍쳐는 기존의 단일 장애점을 극복하고, 각 개체 간 통신량을 최소화 하여 성능 및 보안성이 더 향상될 것으로 기대된다.

[참고문헌]

[1] Lockheed Martin, “Cyber Kill Chain”, http://www.lockheedmartin.com/us/what-we-do/aerospace-defense/cyber/cyber-kill-chain.html.

[2] Elias Bou-Harb, Mourad Debbabi and Chadi Assi, “Cyber Scanning: A Comprehensive Survey”, IEEE Communications Surveys & Tutorials, 16(3), pp. 1496-1519, November, 2013.

[3] Gui-lin Cai, Bao-sheng Wang, Wei Hu and Tian-zuo Wang, “Moving target defense: state of the art and characteristics”, Frontiers of Information Technology & Electronic Engineering 17(11), pp. 1122-1153, November, 2016.

[4] Panos Kampanakis, Harry Perros and Tsegereda Beyene, “SDN-based solutions for Moving Target Defense network protection”, Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014, June, 2014.

[5] Douglas C. MacFarland and Craig A. Shue, “The SDN Shuffle: Creating a Moving-Target Defense using Host-based Software-Defined Networking”, MTD '15 Proceedings of the Second ACM Workshop on Moving Target Defense, pp. 37-41, October, 2015.

[6] Jafar Haadi Jafarian, Ehab Al-Shaer and Qi Duan, “OpenFlow Random Host Mutation: Transparent Moving Target Defense using Software Defined Networking”, Proceedings of the first workshop on Hot topics in software defined networks, pp. 127-132, August, 2012.

[7] Jianhua Sun and Kun Sun, “DESIR: Decoy-enhanced seamless IP randomization”, IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications, April, 2016.

차량 내부통신 보안 안전성 평가기술 기반 연구1)

이선우* 이동훈*

*고려대학교 정보보호대학원

Research on Security-Evaluation Technology for In-vehicle Network

Sun-Woo Lee* Dong Hoon Lee*

*Graduate School of Information Security, Korea University.

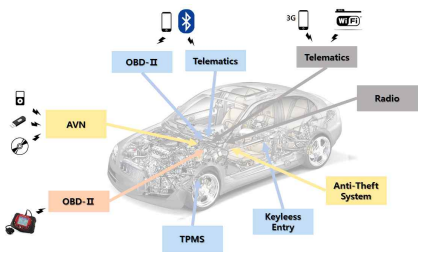
요 약

최근 차량과 ICT(Information Communication Technology)의 융합으로 인하여 차량 시스템이 무선 네트워크를 통해 외부기기와 연결되고 있다. 이러한 차량과 외부 기기의 연결은 사용자들에게 많은 편의와 엔터테인먼트를 제공하지만, 차량에 대한 해킹 공격의 위험 또한 증가하는 결과를 초래한다. 차량은 사용자의 생명뿐만 아니라 보행자 등 주변 환경과 밀접하게 연관되어있으므로, 차량에 공격이 발생할 경우, 인명 및 재산 피해가 심각하다. 이를 방어하기 위해 차량의 내부통신에 대한 보안 시스템 개발이 현재 많이 이루어지고 있으나, 내부통신에 대한 보안 안전 기준은 수립되어있지 않은 실정이다. 따라서 본 논문에서는 차량 내부통신 보안 안전성 평가방법을 서술한다.

I. 서론

최근 ICT(Information Communication Technology) 기술이 발전함에 따라 차량과 ICT 기술이 융합되어, 차량과 외부 기기가 무선 네트워크를 통해 연결되고 있다. 현재 차량에는 [그림 1]과 같이 다수의 외부 네트워크와 연결되어있고, 이러한 연결점들을 통해 사용자들에게 많은 편의서비스 및 엔터테인먼트 기능을 제공한다. 그러나 외부 네트워크와의 연결점 증가는 이를 통한 해킹 공격의 위험 또한 증가하는 결과를 초래하였다. 실제 많은 연구팀 및 기관에서 실제 차량을 이용하여 해킹 실험을 수행함으로써, 차량 내부통신에 대한 보안상의 문제점을 지적하였다[1-5].

차량은 사용자의 생명뿐만 아니라 보행자 등 주변 환경과 밀접하게 연관되어있다. 따라서 차량에 해킹 공격이 발생할 경우, 인명 및 재산 피해가 심각하다. 이를 방어하기 위해 차량의 보안 안전성 확보가 중요하다. 현재 차량 내부통신 보안 시스템에 대한 연구 및 개발이 이루어지고 있으나[6-8], 그러나 차량 내부통신에 대한 보안 안전 기준이 수립되어있지 않아 개발된 보안 시스템에 대한 평가가 불가능하다.



[그림 1] 차량과 외부 네트워크의 연결점

본 논문에서는 차량의 보안 안전성을 확보하기 위한 차량 내부통신 보안 안전성 평가방법

1) 본 연구는 국토교통부 및 국토교통과학기술진흥원의 연구비지원(17TLRP-B117133-02)으로 수행된 연구임.

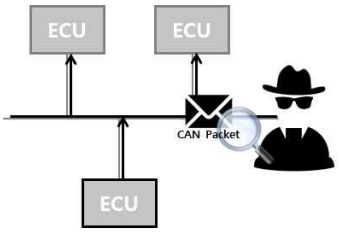
을 서술한다.

II. 배경지식

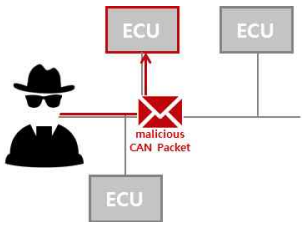
차량은 다수의 ECU(Electronic Control Unit)로 구성되어있으며, 각 ECU들은 내부통신을 통해 차량을 제어한다. 차량 내부통신은 대표적으로 CAN(Controller Area Network)이 사용된다. CAN은 전기적 노이즈에 매우 강하고 빠른 통신 속도로 인해 실시간 통신이 가능하다는 장점이 있다.

그러나 CAN 프로토콜은 암호화 및 인증 기능을 제공하지 않아 기밀성, 가용성 및 무결성을 보장하지 않는다. 따라서 공격자는 [그림 2]와 같이 통신 내용을 도청함으로써 사생활 침해 또는 공격을 위한 사전 정보를 수집할 수 있다. 또한 [그림 3]과 같이 악의적인 메시지를 전송함으로써 차량을 강제 제어할 수 있다. 마지막으로 [그림 4]와 같이 공격자가 악의적으로 메시지를 다량으로 전송함으로써 정상 ECU의 메시지 전송을 방해해 차량의 정상적인 운행을 불가능하게 할 수 있다.

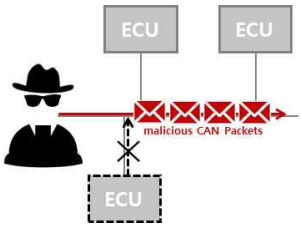
이러한 공격이 일어날 경우, 운전자의 생명뿐만 아니라 보행자 및 주변 차량 또한 안전에 위해를 가할 수 있다. 따라서 이러한 공격들을 방어하기 위해 차량 내부통신에 대한 보안 시스템의 연구 및 개발이 이루어지고 있다.



[그림 2] 기밀성 부재로 인한 도청 공격



[그림 3] 무결성 부재로 인한 악성 메시지 삽입 공격



[그림 4] 가용성 부재로 인한 메시지 전송 방해 공격

III. 차량 내부통신 보안 안전성 평가 방법

본 절에서는 앞서 분석한 CAN 프로토콜의 취약점을 토대로, 차량 내부통신 보안 안전성 평가를 위한 방법을 서술한다. 차량에 대한 해킹 공격은 사용자의 안전과 밀접하게 연관되어 있으므로, 차량 안전 표준인 ASIL을 참고하였다.

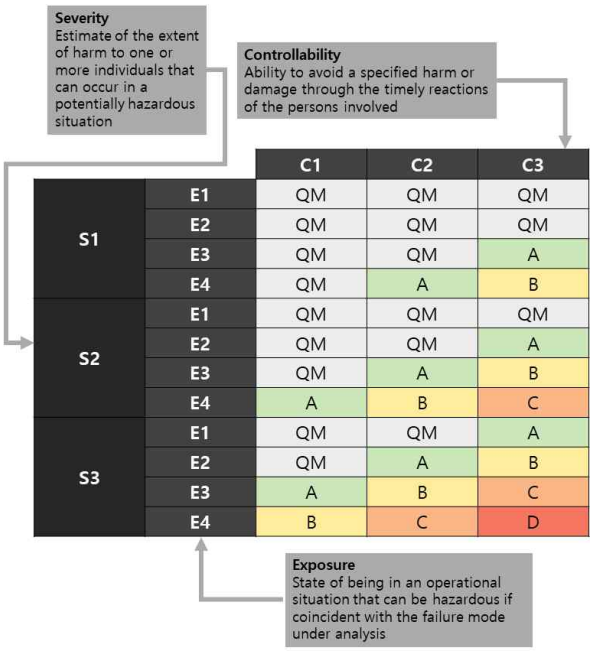
3.1 ASIL(Automotive Safety Integrity Level)

ASIL은 차량 시스템의 안전 등급을 평가하는 기준으로, IEC61508[2]의 SIL(Safety Integrity Level)을 차량의 특성에 맞게 조정한 것이며 차량 기능 안전 표준 ISO26262내 표준으로 지정되어있다.

ASIL은 [그림 4]와 같이 Severity, Controllability 및 Exposure의 3가지의 구성요소로 이루어져 있다. Severity는 위험 사건이 발생했을 경우, 위해의 정도를 나타내며 S0 < S1 < S2 < S3의 4가지 등급으로 구분되어있

2) 전자장치에 대한 기능안전 표준

다. Controllability는 운전자에 의한 재난상황에 대한 통제 가능성을 나타내며 C0 < C1 < C2 < C3의 4가지 등급으로 구분되어있다. 마지막으로 Exposure는 위험 상황이 발생하는 기간이나 상황 빈도를 나타내며 E0 < E1 < E2 < E3 < E4의 5가지 등급으로 구분되어있다. ASIL 등급은 각 항목별로 평가된 등급을 토대로 최종적인 등급이 산정되며, 총 5가지 QM[3] (Quality Management) < A < B < C < D등급으로 분류된다. ASIL D등급은 가장 위험성이 높은 등급을 의미한다. 차량 시스템 개발자는 각 등급의 기준에 맞는 시스템 개발 프로세스를 따라야 하며, 시스템의 생명주기를 따라 관리해야 한다.



[그림 5] ASIL 다이어그램

3.2 차량 내부통신 보안 안전성 평가 기반 연구

차량 안전 표준인 ASIL을 기초한 보안 관점에서의 차량 내부통신 보안 안전성을 평가하기 위한 방법은 다음과 같다.

차량에 대한 해킹 공격은 무결성 > 가용성 > 기밀성 순으로 안전에 위해를 가할 수 있다. 따라서 ASIL 항목 중 Severity에 이를 대응시킨다. 또한 ASIL의 Exposure는 CAN 프로토콜의 취약점에 대한 노출 가능성으로 볼 수 있으나, 공격자가 차량 내부통신에 접근이 가능하면 공격을 수행할 수 있으므로 Exposure 등급은 모두 동일하다고 가정한다. 마지막으로 차량 내부통신에 보안 시스템이 적용된 경우, 해당 시스템의 대응 능력에 따라 공격으로 인한 위험을 통제할 수 있다. 따라서 차량 내부통신 보안 시스템과 ASIL의 Controllability를 대응시킬 수 있으며, P1 < P2 < P3 등급으로 분류된다. P3 등급은 보안 시스템이 적용되어 있지 않은 경우를 뜻한다. 각 항목별 대응 결과는 [표 1]과 같으며, 최종적으로 차량 내부통신 보안 안전성 평가를 위한 다이어그램은 [그림 6]과 같다. 차량 내부통신 보안 시스템의 능력 등급 및 해킹 공격으로 인한 위험도 등급을 평가한 후, 이를 토대로 차량 내부통신의 안전성은 A < B < C < D등급으로 평가된다. D등급의 경우, 차량 내부통신에 대한 보안 시스템이 제대로 적용되어 있지 않아 차량 해킹 공격에 위험하다는 것을 의미한다.

[표 1] ASIL 평가 항목과 CAN 프로토콜의 안전성 평가 항목 분류표

ASIL 평가 항목	CAN 프로토콜의 안전성 평가 항목	비고
Severity	기밀성 < 가용성 < 무결성	무결성의 잠재적 심각도가 가장 높음
Controllability	차량 내부통신 보안 시스템의 대응 능력	세 등급으로 분류
Exposure	내부통신에 대한 접근 용이성	동일하다고 가정

3) QM(Quality Management): ISO에 따른 안전 요구가 불필요한 안전한 상태를 의미함



[그림 6] 차량 내부통신 보안 안전성 평가를 위한 다이어그램

IV. 결론

최근 차량에 대한 해킹 공격의 위험이 증가하고 있다. 차량은 사용자의 안전뿐만 아니라 보행자 등 주변 환경과 밀접한 연관이 있기 때문에 차량 내부통신에 대한 보안은 중요하다. 따라서 본 논문에서는 차량에 대한 보안 안전성을 확보하기 위해 차량 내부통신을 평가하는 방법을 서술하였다.

차량 내부통신 보안 안전성 평가를 위해 기 분석한 CAN 프로토콜의 취약점을 토대로, ISO26262의 ASIL을 참고하여 다이어그램을 제시하였다. 본 연구의 내용은 현재 차량뿐만이 아니라 향후 자율주행차량까지 내부통신 보안 안전성 평가기술로 활용될 것으로 기대된다.

[참고문헌]

[1] Koscher, Karl, et al. "Experimental security analysis of a modern automobile." Security and Privacy (SP), 2010 IEEE Symposium on. IEEE, 2010.

[2] Checkoway, Stephen, et al. "Comprehensive Experimental Analyses of Automotive Attack Surfaces." USENIX Security Symposium. 2011.

[3] Petit, Jonathan, and Steven E. Shladover. "Potential cyber attacks on automated vehicles." Intelligent Transportation Systems, IEEE Transactions on 16.2 (2015): 546-556.

[4] Miller, Charlie, and Chris Valasek. "Adventures in automotive networks and control units." DEF CON 21 (2013): 260-264.

[5] Miller, Charlie, and Chris Valasek. "Remote exploitation of an unaltered passenger vehicle." Black Hat USA 2015 (2015).

[6] Woo, Samuel, Hyo Jin Jo, and Dong Hoon Lee. "A practical wireless attack on the connected car and security protocol for in-vehicle CAN." Intelligent Transportation Systems, IEEE Transactions on 16.2 (2015): 993-1006.

[7] Radu, Andreea-Ina, and Flavio D. Garcia. "LeiA: a lightweight authentication protocol for CAN." European Symposium on Research in Computer Security. Springer International Publishing, 2016.

[8] Kurachi, R., et al. "CaCAN-Centralized authentication system in CAN." Proc. escar 2014 Europe Conference, Hamburg, Germany. 2014.

기업 서버의 침해사고 대응을 위한 침해사고 준비도 모델

신현우*, 박성미*, 유운선*, 황지수*, 김동현*, 한창희*, 김종현**

*KITRI BOB

**더존 포렌식 센터

Incident Response Readiness Model for Business Servers

Shin Hyunwoo*, Park Sungmi*, Yoo Woonseon*, Hwang Jisoo*, Kim Donghyun*, Han Changhee*, Kim Jonghyun**

*KITRI BOB

**DOUZON Digital Forensic Center

요 약

전 세계적으로 침해사고 대응에 대한 관심이 증가하는 가운데, 2016 정보보호실태조사에 의하면 국내 기업의 침해사고에 대한 인식은 현저히 낮은 편이다. 침해사고를 탐지하고 전문가의 도움을 받아도 침해사고 준비가 되어 있지 않아 피해가 심각한 상황이다. 따라서 본 논문은 침해사고에 대한 적당한 대응을 하기 위해 필요한 침해사고 준비도를 제안하고자 한다. 먼저, 기존 침해사고 대응 절차를 비교분석하고 그 문제점을 도출하였다. 이에 따라 문제점을 극복할 방안으로 침해사고 준비도의 관리적·기술적 구성요소와 기업 서버에 맞춘 준비도 도입 방안을 제시하였다. 또한, 중소기업에는 부담이 될 수 있는 비용의 문제를 해결하고자 오픈소스 툴 등을 사용한 디지털 증거 수집 가이드라인을 제안하였다.

I. 서론

Breach Level Index에서 공개된 자료에 따르면 1분 당 7,297 건의 정보 침해사고가 전 세계에서 발생하고 있다[1]. 우리나라의 경우 기업이 침해에 노출된 위험성이 높을 뿐만 아니라 그 경각심도 매우 낮다. 2016년 한국인터넷진흥원(KISA)에서 시행한 기업부문 정보호호실태조사에 의하면, 조사대상인 사업체들 중 17.1%만이 침해사고 대응활동을 시행하고 있었으며, 침해사고를 경험한 기업체가 관계기관에 문의 또는 신고한 경우는 고작 9.2% 인 것으로 나타났다. 특히, 미신고 사업체의 37%가 침해정도가 심각했음에도 불구하고 신고하지 않은 것으로 드러났다. 정보침해에 대한 부실한 대응과 낮은 인식은 이미 침해가 발생한 경우 피해회복이 불가능하거나 혹은 침해사실이 밝혀짐으로써 받게 될 더 큰 불이익을 감내하기 어렵다는 데 있다. 그 동안 국내외 공공기관들이 기업들에게 제시하고 있는 침해사고 대응 모델들은 침해 예방 또는 분석에 중점을 두고 준비조치를 소홀히 다룸으로써 성공적인 침해사고 대응에 한계가 있다는 비판을 받아 왔다. 이미 전문가들은 빈번하고도 끊임없이 발생하는 침해사고는 침해를 대비한 방어(prevention)만으로 해결할 수 있는 문제가 아니며 그 이전 단계에서 침해사고의 가능성을 인지하고(acceptance) 준비조치를 취하는 것을 권장하고 있다[2]. 2016년 12월 금융보안원이 금융회사 디지털 증거 확보와 수집·분석 정책을 권장하는 등 침해사고 준비도 가이드를 공개한 것이 그 대표적인 예이다.

침해사고 준비도는 2001년 John Tan이라는 학자가 범죄사건에 소요되는 2시간의 포렌식 활동 당 40시간의 포렌식 조사 시간이 든다

는 사실을 확인하고 자신의 논문에 처음 소개했던 절차인 포렌식 준비도(Forensic Readiness)에서 파생된 개념이다[3]. 그 동안의 선행연구들을 토대로 살펴보면[4] 침해사고 준비도는 침해사고 전 디지털 증거를 효과적으로 수집 및 분석하기 위한 환경 극대화와 잠재적 증거를 신속히 수집할 수 있는 계획 수립을 통해 침해사고의 재발 및 피해확산 방지를 목적으로 하고 있다. 침해사고 준비도는 분석에 중점을 둔 포렌식 준비도에 비해서는 특히 침해사고에 대한 선제적 대응에 중점을 둔 절차라고 할 수 있다.

본 논문은 침해사고 준비도 중심의 정보침해사고 대응 정책을 제시하기 위해 제2절에서 현 국내외 침해 사고 모델의 현황과 한계점을 분석한다. 3절에서는 한계를 극복하기 위한 대안으로 Linux서버를 중심으로 설계된 침해사고 준비도 모델을 제안하고, 실무상 적용가능성에 대해 설명한다. Linux는 저렴한 비용으로 운용할 수 있어 국내의 많은 중소기업들의 서버 OS로 사용되고 있는데 그 중요도에 비해 국내외 침해사고 대응 절차에서 구체적으로 다뤄지지 않아 본 논문의 침해사고 준비도 모델 대상으로 선정하였다.

II. 국내외 침해사고 대응 모델의 특징과 한계

모든 산업 분야에서 정보 시스템에 대한 침해사고는 국가적으로도 큰 손실로 이어진다는 것은 이미 우리는 수많은 사례를 통해 경험하고 있다. 이러한 피해를 줄이기 위해 세계 여러 나라의 공공기관들은 침해사고 대응

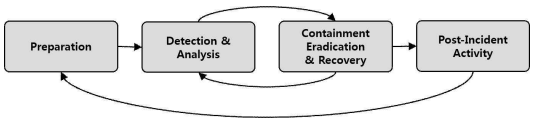


Figure 1 NISA Incident Response(IR) Life Cycle [5]

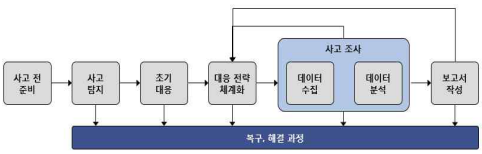


Figure 2 KISA IR Procedure Model [6]

가이드라인 및 모델을 제시하고 있고, 우리나라는 '정보통신망 이용촉진 및 정보보호 등에 관한 법률' 등 여러 가지 법률적인 지원책을 마련하고 있다.

우선 미국의 경우를 살펴보면 정부의 기술표준 및 민간부문의 적합성 평가 활동을 조정하는 정부기관인 국립표준기술연구원(NIST)은 침해사고 대응 표준 절차로서 NIST 800-61 Computer Security Incident Handling Guide를 제시하고 있다. NIST의 가이드에 나타나 있는 대응 절차는 1단계 - 준비(Preparation), 2단계 - 탐지와 분석(Detection & Analysis), 3단계 - 격리, 제거 및 복구 (Containment, Eradication & Recovery), 4단계 - 사후조치(Post-Incident Activity), 총 4 단계로 구성되어 있으며, 순환하는 구조로 표현되어 있다. 순환은 2단계와 3 단계에서 한 차례 이루어지는데 예를 들어 악성코드를 제거하면서 다시 탐지와 분석 단계로 돌아와 악성코드에 감염된 다른 호스트는 없는지를 확인하는 작업 방식이다. 침해사고 라이프사이클(Life Cycle)이라고 불리는 NIST 가이드라인의 이 절차는 각 단계의 상호작용을 중점을 두어 유동적인 침해사고 대응을 권장하는 것이 특징적이다.

한국의 경우 KISA에서 7단계 대응 절차로 구성된 '침해사고 분석 절차 가이드라인(2010년)'을 제시한 바가 있다. 이 가이드라인은 침해사고 조사를 중심적으로 사고의 복구와 해결을 위해 취해야 할 절차를 단계별로 설명하고 있다. 한국과 미국의 모델을 비교하면 침해사고 라이프 사이클의 2 단계에 해당되는 '탐지와 분석'이 KISA의 모델에서는 사고탐지-초기대응-대응전략체계화-사고조사(데이터수집-분석) 등 5단계의 절차로 세분화되어 있음을 알 수 있다. 그리고 NIST의 3단계인 '격리, 제거 및 복구'가 KISA모델에서는 사고 대응의 7단계에 별도로 규정하지 않고 앞의 5단계의 과정 전체를 '복구, 해결 과정'으로 표현하고 있다는 점이

특징적이다. 그러나 큰 틀에서 보면 미국 NIST의 침해사고 라이프사이클에 비하여 단계가 세분화되어 있다는 점을 제외하면 큰 차이를 보이고 있지는 않는다.

양국의 침해사고 대응 모델을 살펴본 결과, NIST의 침해사고 라이프사이클은 준비, 탐지, 분석, 사후조치라는 큰 틀에서 각 기관에 해석의 자유를 주고 순환해야한다는 점을 강조하고, 특히, 4단계에서 1단계로 다시 순환하는 것은 사고 후 발견한 개선사항을 준비 단계에 적용시킨다는 의미로 이해할 수도 있다. 그러나 사실상 탐지와 분석의 기반이 될 수 있는 준비 단계를 독자적으로 다루고 있지 않고 있다는 점에서 침해사고 준비도에 대한 설명과 지침으로서의 역할을 기대하기는 어렵다. 또한 한국의 KISA 모델의 경우 가이드라인의 대부분이 5단계로 구성된 분석단계에 치중 되어 있고, 첫 단계인 사전준비단계는 복구, 해결과정에 포함되지 않는 사고대응 전 준비 과정으로서 간단한 요약만이 서술되어 있다는 점을 고려할 때, KISA 모델의 경우에도 침해사고 준비도에 대한 중요성을 인식하지 못하고 있음을 알 수 있다.

이상과 같이 기존의 침해대응모델들은 주로 침해사고 대응 또는 분석에 관점을 두고 있다는 점에서 침해사고에 대한 충분한 대비모델이라고 보기는 어렵다. 따라서 본 논문은 이러한 침해사고에 대한 대응시점을 침해의 이전단계로 옮긴 침해사고 준비도를 제시함으로써 선제적인 대응책을 제시하고자 한다.

III. 기업 서버를 위한 침해사고 준비도

본 연구는 중기업에서 준비 가능한 침해사고 준비도를 구성하는 요소를 제시하고, 이를 바탕으로 기업의 중요자산 중 서버를 대상으로 침해사고 준비도가 도입된 침해사고 대응 방안을 구체적으로 제안하고자 한다.

침해사고 준비도가 도입된 침해사고 대응 절차는 사고 탐지를 기점으로 침해사고 준비와 침해사고 대응 절차로 구분할 수 있다. 침해사고 준비는 침해사고 대응의 수월한 진행을 보조하는 데에 있다는 점을 감안하면 침해사고의 원인분석을 위한 관리적·기술적 환경 구축이 필수적으로 선행되어야 한다.

구체적인 구성요소를 정리해보자면 <Table 1>과 같다.

Table 1 Components of IR Readiness

침해사고 준비도	구성요소
관리적 환경요소	- 인적 관리 (담당자 선정, 직원 교육 훈련 등) -법적 요구사항 안내서 (정보보호법 컴플라이언스 등) - 디지털 증거 수집, 처리 및 보존 가이드라인 - 초기 대응 절차 안내서 - 비상 연락망 체계 구축 - 정기 감사를 통한 지속적인 침해 탐지 계획 수립
기술적 환경요소	- 디지털 포렌식 툴 설치 및 구매 - 디지털 아티팩트 수집 자동화 - 서버 환경 설정 - IDS 등 관제 시스템 구축

침해사고 준비도를 구성하는 관리적·기술적 환경의 효과적인 구축은 침해대응능력 뿐만 아니라 향후 디지털 증거의 활용능력을 향상시킬 수 있는 이점이 있다. 이 모델을 서버에 한정하여 적용할 경우 아티팩트(Artifact) 수집은 기술적 환경요소에서 가장 필수적 구성요소이다. 기존의 침해대응 기법은 침해사고가 발생한 이후 디지털 증거를 수집하는 것에 집중하고 있으며, 공격자가 악의적으로 로그를 지웠을 경우 분석에 필요한 정보를 온전히 확보할 수 없는 문제를 안고 있다. 따라서 블랙박스 개념을 차용한 디지털 아티팩트 수집 자동화 등의 방법을 활용하여 침해관련 디지털 증거를 지속적으로 수집한다면 차후 분석할 수 있는 정보량을 극대화할 수 있다.

이 때 지속적으로 수집할 아티팩트로는 시스템 정보, 로그 파일, 휘발성 정보, 네트워크 패킷 등이 있으며, <Figure 3>처럼 가독성이 좋고 어플리케이션에서 처리하기도 용이한 yaml형식 파일로 관리하는 것을 권장한다. 사용자가 파일에 추가된 아티팩트를 입력하는 것만으로 관리할 수 있기 때문이다.

```yaml
---
name: LinuxSyncookieState
doc: Whether the system uses syncookies.
sources:
- type: FILE
  attributes: {paths: ['/proc/sys/net/ipv4/tcp_syncookies']}
labels: [Network, System]
supported_os: [Linux]
urls: ['https://www.kernel.org/doc/Documentation/networking/ip-sysctl.txt']
---
```

Figure 3 yaml example

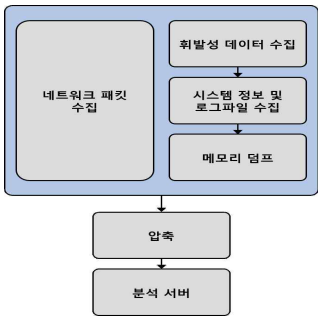


Figure 4 Data Collection Process

수집된 아티팩트들은 <Figure 4>에 나타냈듯이 압축 후 분석 서버로 전송되어 저장된다. 분석 서버에서는 수집된 데이터의 제각기 다른 형식을 같은 형식으로 변환하는 정규화(Normalization)하여 날짜와 시간 기준으로 분석할 수 있는 타임라인 기능을 제공한다. 추후 고급 분석을 위해 CSV와 같이 다른 포맷으로의 추출 또한 지원한다. 이러한 침입흔적을 바탕으로 수집된 아티팩트들은 궁극적으로 성공적인 분석으로 유도함으로써 신속한 피해 회복을 보장하고 조사로 인한 비용발생을 감소시킬 수 있다.

그러나 사전적 정보수집과 같은 방안은 비용의 문제로 중소기업 이하의 사업 환경에 부합하지 않아 보일 수 있다. 이러한 문제는 Fast IR 또는 GRR과 같은 침해사고 커뮤니티에서도 널리 사용되는 오픈소스툴을 도입함으로써 해결이 가능하다. 또한, 도구뿐만 아니라 기업의 특성에 맞춘 가이드도 함께 제공함으로써 기술적·관리적 접근 방안을 모색해야 한다. 침해사고 준비도 도입을 통해 침해사고 대응에 적합한 관리적·기술적 환경을 구축한다면, 침해사고로 인해 발생하는 비용과 시간적 피해를 줄일 수 있을 것이다.

IV. 결론

우리나라의 경우, 해마다 심각한 개인정보유출 사건들이 발생하고 있는 상황임에도 불구하고 선진외국에 비하여 침해사고에 대한 경각심이 높지 않다. 그 동안 KISA와 NIST에서 제공한 침해사고 대응 절차 모델이 침해사고가 발생한 이후, 방어와 분석에 관한 것이었다면 이제는 실효성을 고려해 방어와 분석이 성공적으로 이루어질 수 있도록 준비단계의 체계화를 논의해야 할 시점이 되었다. 이러한 관점에서 본 논문은 성공적인 침해사고 대응의 기반이 되는 침해사고 준비도 구축 필요성과 침해사고 준비도를 구성하는 요소, 특히 서버를 대상으로 적용될 수 있는 아티팩트 수집 방안을 제시하여 모델을 구체화하고자 시도하였다. 그러나 침해사고 준비도 구축은 직접적인 피해이전 단계라는 점에서 전문 보안 대응팀을 구축하고 있지 않는 중소기업의 경우에는 이에 대한 이해도가 낮아서 자칫 호응도가 떨어질 우려가 있다. 이러한 점을 감안하여 우리나라에 제시되는 침해사고 준비도 모델과 사용설명서는 국내 중소기업을 대상으로 침해사고 준비도의 필요성과 더불어 절차에 대해 자세한 소개를 통하여 적극적인 도입하도록 유도할 필요가 있다.

[참고문헌]

[1] “Data Records Compromised in First Half of 2017”, Breach Level Index, Available at: www.breachlevelindex.com

[2] J. Hart, “Evolving from Breach Prevention to Breach Acceptance to Securing the Breach”, RSA Conference Europe, 2013.

[3] J. Tan, “Forensic Readiness”, @stake, 2001.

[4] S. Baek, J. Lim, “A Study on the Forensic Readiness as an Effective Measure for Personal Information Protection (kor)”, Internet and Information Security, vol.3, no.2, 2012, pp.34-64.

[5] National Institute of Standards and Technology, Computer Security Incident Handling Guide, 2012, p. 21.

[6] Korea Internet & Security Agency, Incident Analysis Procedure Guide, 2010, p. 12.

보안성이 강화된 SaaS 개발 생명주기 연구*

정윤송† 이동훈‡

고려대학교 정보보호대학원

Research on SaaS Development Lifecycle with Improved Security*

Yun Song Jeong† and Dong Hoon Lee‡

Graduate School of Information Security, Korea University.

요 약

SaaS는 클라우드 플랫폼에서 서비스 형태로 어플리케이션을 제공하는 모델이다. 기존의 SaaS 개발 생명주기를 준수하여 소프트웨어를 개발하는 경우 보안성이 취약하여 데이터 프라이버시 혹은 시스템이 훼손될 가능성이 존재한다. 본 논문에서는 보안성 측면에서 SaaS 개발 생명주기가 가지는 문제점을 분석한 후에 소프트웨어 수준에서 보안성을 향상시킬 수 있는 방안을 연구한다.

I. 서론

NIST에서는 클라우드 컴퓨팅이 제공하는 서비스 모델을 IaaS, PaaS, SaaS로 구분한다[1]. 서비스형 소프트웨어(Software as a Servce, SaaS)는 클라우드 환경에서 동작하는 응용프로그램을 서비스 형태로 제공하는 모델이다. 이 모델에서는 어플리케이션을 서비스 사용자의 로컬 저장소에 설치하지 않고 원격 클라우드 저장소에 저장한다. SaaS는 서비스 제공자가 직접 데이터 센터를 구축하지 않고도 서비스를 제공한다는 장점이 있다.

Gartner에서는 기업 데이터 트래픽이 현재 10%에서 2021년에 27%가 보안 경계선(Perimeter Security)을 우회할 것이라 예상한다[2]. 클라우드 컴퓨팅이라는 새로운 환경은 IT 인프라를 일부 혹은 전부 아웃소싱하기 때문에 보안 문제가 제기될 수밖에 없다. 아웃소싱된 데이터에 대한 프라이버시 문제뿐만 아니라 가상화 환경에서 발생하는 취약점도 존재한다. 이때 함수암호를 이용해서 클라우드에 저장되는 데이터를 암호화하면 사용자의 프라이버시 침해에 대한 우려 없이 데이터를 아웃소싱할 수 있다.

SaaS에서는 중앙 서버에서 소프트웨어를 관리하기 때문에 서비스 사용자가 직접 패치를 설치할 필요가 없다. 일반 소프트웨어에 비해 SaaS는 원격 클라우드 저장소의 어플리케이션에 대한 새로운 버전 릴리스가 용이하다. 따라서 SaaS를 개발할 때에는 변화하는 요구사항을 반영하여 점진적으로 개발하는 과정을 반복할 수 있는 애자일(agile) 모델이 적합하다. 본 논문에서는 기존 SaaS 개발 생명주기에 애자일 모델과 소프트웨어 보안성 터치포인트(software

* 이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2017-0-01863, 함수암호 기법 설계·분석 및 구현기술 연구)

† 주저자, song91415@naver.com

‡ 교신저자, donghlee@korea.ac.kr

security touchpoints)를 적용하여 보안성이 강화된 SaaS 소프트웨어 개발 생명주기를 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로서 애자일 개발방법론, 소프트웨어 보안성 터치포인트 및 OWASP의 클라우드 환경에서의 10가지 위험을 소개한다. 3장에서는 기존의 SaaS 개발 생명주기를 적용했을 때 발생할 수 있는 문제점을 분석하고 새로운 소프트웨어 개발 생명주기를 제안한다. 4장에서 결론을 맺고 향후 연구방향을 제시한다.

II. 관련연구

2.1 애자일 개발 방법론

애자일 운동은 2001년 소프트웨어 실무자와 컨설턴트가 출판한 소프트웨어 개발 선언서를 통해서 제안되었다. 애자일 개발 방법론은 점증적(짧은 주기의 소프트웨어 릴리스), 협동적(작업과정에서 사용자와 개발자의 긴밀한 소통), 직설적(용이하고 문서화된 학습 및 수정)이고 적응적(변화에 대응)인 특징을 가진다[3]. 각 반복 주기마다 사용자의 피드백을 받아 다음 생명주기에 적용한다.

2.2 소프트웨어 보안성 터치포인트

소프트웨어 보안성 터치포인트는 이미 개발된 개발 생명주기의 각 단계마다 실무적으로 검증된 소프트웨어 공학의 모범 사례 집합을 적용하여 보안성을 쉽게 강화시킬 수 있다. 터치포인트는 [표 1]과 같이 높은 유효성 순으로 번호가 매겨진다[4].

[표 1] 소프트웨어 보안성 터치포인트

번호	터치포인트
1	코드 리뷰(Code Review)
2	구조적 위험 분석 (Architectural Risk Analysis)
3	모의 침투 테스트(Penetrating Test)
4	위험 기반 보안 테스팅 (Risk-based Security Testing)
5	악용 사례(Abuse Case)
6	보안 요구사항(Security Requirements)
7	보안 운영(Security Operations)

2.3 OWASP의 클라우드 환경에서의 10가지 위험

OWASP에서는 클라우드 컴퓨팅과 SaaS 환경의 위험을 다음의 10가지로 꼽았다[5].

1) 책임성과 데이터 소유권: 클라우드에 아웃소싱된 데이터에 대한 책임 소재가 불명확하다.

2) 사용자 식별자 연합: 서로 다른 클라우드 제공자는 SAML를 이용한 통합된 인증(Single Sign On)으로 사용자를 식별한다. 그 결과, 클라우드 업체의 백 엔드에서 쉽게 식별자를 연합할 수 있다.

3) 규제 준수: 데이터 프라이버시에 대한 법은 국제적으로 서로 상이하다.

4) 사업 연속성과 회복성: 사업 연속성은 클라우드 제공자에게 책임이 있다.

5) 사용자 프라이버시 및 데이터 이차 사용: 서비스 사용자의 데이터 이차 사용에 대한 동의하에 데이터를 활용해야 한다. 데이터 활용을 위해서는 개인정보 비식별화와 데이터 암호화가 필수적이다.

6) 서비스 및 데이터 통합: 사용자와 클라우드 데이터 센터 사이에서 전송되는 데이터가 보호되어야한다.

7) 다중 임대 및 물리적 보안 : 여러 클라이언트가 클라우드 자원을 공유하므로 논리적 분리가 필수적이다.

8) 사후 분석 및 포렌식 지원: 사후에 로그 파일을 열람할 때, 국가별 관할권이 상이하다.

9) 기반구조 보안: 기반구조와 구성이 안전하게 설정되어야 한다.

10) 비생산적 환경 노출: 보안을 적용하지 않은 비생산적인 환경은 취약점이 될 수 있다.

III. 제안하는 보안성이 강화된 개발 생명주기

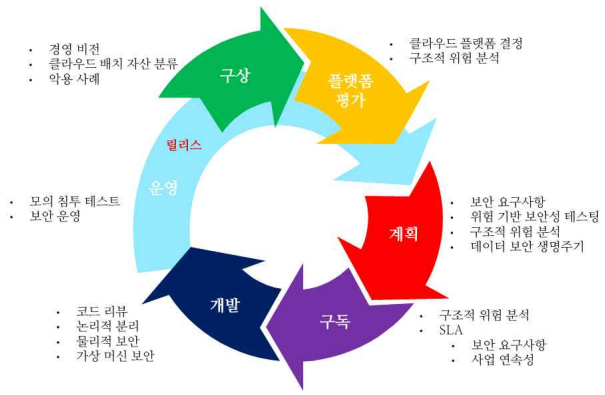
3.1 기존의 서비스형 소프트웨어 개발 생명주기의 문제점(SaaS Development Lifecycle, SaaSDLC)

2011년 H. Kommalapati 등은 클라우드 제공자를 생명주기에 포함한 SaaSDLC를 제안하였

다. 클라우드 서비스들이 기본적으로 만족해야 할 원칙은 발견성, 접근성, 경제적 타당성, 확장성, 지원 가능성이다. 이러한 속성을 만족하기 위해서 SaaSDLC는 기존 소프트웨어 개발 생명주기에 플랫폼 평가, 구독, 운영 단계를 추가하였다. 기존 SaaSDLC는 보안을 위해서 개발 단계에서 보안 응용프로그램(SSO, 권한 부여 정책 등)을 추가해야 한다고 기술되어 있다[6]. 보안성은 기능성(functionality)와는 전혀 다른 속성이므로 개발 생명주기 전 단계에 걸쳐 보안성을 고려해야 한다. 개발 단계에서 보안기능을 가진 응용프로그램을 추가하는 것만으로는 보안성이 취약할 가능성이 있다.

3.2 보안성이 강화된 SaaS 개발 생명주기

기존의 SaaSDLC에 OWASP의 클라우드 환경에서 10가지 위험, 클라우드 보안 연합(Cloud Security Alliance, CSA) 보안 가이드 및 Cloud SSDLC[7]를 참고하여 보안성이 강화된 SaaSDLC를 제안한다. [그림 1]과 같이 계획, 구독, 개발, 운영단계는 반복적으로 수행한다.



[그림 1] 보안성이 강화된 SaaS 개발 생명주기

3.2.1 구상 단계(Envisioning Phase)

구상 단계에서는 다양한 시장 조사를 통해서 경영 기회를 포착하고 전체적인 서비스 비전을 계획한다. 이 과정에서 클라우드 플랫폼의 특징을 파악하여 경영 비전과 범위에 적합한 클라우드 플랫폼 목록을 작성한다.

이 때, 정보의 민감도를 고려하여 클라우드 플랫폼의 형태를 결정해야 하므로, 클라우드 환경에 배치할 자원을 분류하는 과정이 필수적이다[7]. 이 과정에 악용 사례를 적용하면, 공격자 입장에서 시스템 행동을 기술하여 어떤 자산을 무엇으로부터 얼마나 오랫동안 보호해야 할지 식별할 수 있다. 즉, 클라우드 환경에 배치할 자원을 파악하여 보호해야 할 대상과 기간을 책정한다. Exploiting Software[8]의 공격 패턴을 참고하여 악용 사례를 도출할 수 있다.

3.2.2. 플랫폼 평가 단계(Platform Evaluation Phase)

플랫폼 평가 단계에서는 서비스를 배치할 가장 적합한 클라우드 플랫폼을 결정한다. 구상 단계의 플랫폼 목록에서 성능, 확장성, 신뢰성(Reliability), 재해 복구, 보안, 프라이버시를 모두 고려하여 비교한다. 클라우드 사업 연속성과 빠른 복구를 지원하는 플랫폼을 선택해야 한다. 기능적 및 비(非)기능적 측면에서 개념증명(Proof of Concept, POC)를 해야 하는데, 이때 비기능적 측면은 보안을 염두에 두어야 한다.

테스트 전 모든 단계에서는 구조적 위험 분석을 시행한다. 플랫폼 평가 과정에서 가정을 문서로 명시하고, 악용 사례에서 발견한 공격들을 식별하여 구조상의 결함(flaw)을 찾아낸다. 구조적 위험 분석은 공격 저항 분석, 모호성 분석과 약점 분석의 세 단계로 이루어진다. 매 단계 위험을 추적하여 구조 수준에서 위험을 관리할 수 있다.

3.2.3 계획 단계(Planning Phase)

계획 단계에서는 해당 반복 주기에서의 요구사항을 도출해야 한다. 이해 당사자, 즉 클라우드 제공자, 서비스 사용자와 서비스 제공자의 요구사항을 분석한다. 이때 보안이 요구사항 수준에 반드시 명시되어야 한다. 보안 요구사항은 암호화 등의 기능적인 보안과 악용 사례나 공격 패턴의 속성을 모두 포함한다.

도출된 요구 사항을 반영한 프로토타입을 설계하여 다음 두 가지 방식의 위험 기반 보안성 테스팅을 수행한다: (1)일반적인 기능성 테스팅 기법을 이용한 보안성 테스트 (2)앞선 단계에서 도출한 악용 사례, 공격 패턴, 위험 분석 결과를 기반으로 한 보안성 테스트. OWASP의 클라우드 환경에서의 10가지 위험을 중심으로 보안성 테스팅을 수행한다.

의료 정보, 범죄 기록과 같은 서비스 사용자의 민감한 데이터와 같은 경우에는 CSA에서 정의한 데이터 보안 생명주기[9]에 따라 생성부터 폐기단계까지 데이터를 관리해야 한다. 데이터 보안 생명주기는 생성, 저장, 사용, 공유, 아카이브, 폐기 여섯 단계로 이루어진다. 각 단계에서 고려해야 할 사항은 [그림 2]와 같다.



[그림 2] 데이터 보안 생명주기

3.3.4 구독 단계(Subscribing Phase)

서비스 제공자는 클라우드 플랫폼 제공자와 서비스 수준 협약(Service Level Agreement, SLA)를 맺는다. SLA를 통해서 클라우드 제공자가 서비스 사용자의 데이터를 유출시켰을 때 책임을 물을 수 있다[7]. 이 때, 각 이해당사자는 서로 다른 보안 요구사항을 가지고 있으므로 합의가 필요하다.

OWASP가 선정한 클라우드 보안성 위험 중 첫 번째는 책임성과 데이터 소유권이다. 아웃소싱된 사용자의 데이터에 대한 책임소재에 관한 문제이다. SLA을 맺을 때는 앞서 도출된 모든 보안 요구사항과 사업 연속성이 명시되어야 클라우드 제공자에게 법적 책임을 물을 수 있다.

3.3.5 개발 단계(Developing Phase)

개발 단계는 개발 환경 구축, 구현, 테스팅의 하위 단계로 이루어진다. 기존의 생명주기에서 어플리케이션 보안을 구현하는 것만으로는 전체 소프트웨어의 안전성을 보장할 수 없다.

구현이 이루어지고 난 이후에는 코드 리뷰를 반드시 수행한다. 정적 분석 도구를 이용한 코드 스캐닝은 흔한 취약점을 찾아낼 수 있다. 단, 코드 리뷰는 구조 분석과 함께 이루어져야만 한다. 코드 리뷰 단독으로는 50% 정도의 보안성 문제만을 발견할 수 있다.

다중 임대, 즉 여러 서비스 제공자가 같은 클라우드 플랫폼을 이용할 때에는 물리적 분리가 되지 않으므로 논리적 분리가 필수이다. 개별 키 관리와 관리자 접근의 감사가 필요하다. 클라우드 가상 머신 관리자의 모니터링과 침입 감지 및 안티 바이러스, 취약점 스캐닝 방지 등의 가상 머신 보안 시스템을 구축해야 한다[7].

3.3.6 운영 단계(Operation Phase)

운영 단계에서는 실제 서비스를 배치하고 유지한다. 재해복구 테스팅을 수행하고 지속적인 업데이트와 보안성 패치를 통해서 사용자 경험을 개선해야 한다.

동작 중인 시스템에서 모의 침투 테스트를 통해 실제 환경에서 현장의 소프트웨어 이해를 할 수 있다. 모의 침투 테스트는 소프트웨어 개발 생명주기의 이전 단계의 구조적 위험 분석이 끝난 이후에 의미가 있다. 터치포인트의 보안 운영을 적용하여 네트워크 보안 전문가를 투입시켜서 개발팀이 놓친 네트워크 보안을 강화한다.

애자일 모델에서는 짧은 반복 주기마다 소프트웨어를 점진적으로 릴리스하고 피드백을 다음 반복 주기에 반영한다. 운영 단계에서는 서비스 사용자와 서비스 제공자도 함께 참여하여 피드백을 받고 계획 단계부터 다시 반복하도록 한다.

IV. 결론

본 연구에서는 보안성을 고려하지 않은 기존 SaaSDLC의 문제점을 제기하고, 이를 해결하기 위해 소프트웨어 수준에서 개발 생명주기의 각 세부 단계에 터치포인트를 적용하여 보안성을 향상시키는 방안에 대하여 연구하였다. 향후 단순히 소프트웨어 수준이 아니라 IaaS, PaaS 모델에서 높은 보안성을 달성할 수 있는 방안에 대한 연구가 필요하다. IaaS나 PaaS 모델에서 높은 보안성을 달성하기 위해서는 시스템 레벨의 보안 분석을 수행하는 등 추가적인 방안이 필요하다.

[참고문헌]

[1] P. Mell and T. Grance, The NIST Definition of Cloud Computing, NIST Special Publication 800-145, Sept. 2011

[2] Gartner, 5 Steps to Closing SaaS Security Gaps, Aug. 2016. https://www.gartner.com/smarterwithgartner/five-steps-to-closing-saas-security-gaps-2/

[3] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, Agile Software Development Mehtods: Review and Analysis, Sep. 2017

[4] G. McGraw, Software Security : Building Security in, Jan. 2006

[5] OWASP, OWASP Cloud 10 Project https://www.owasp.org/index.php/Category:OWASP_Cloud_%E2%80%90_10_Project

[6] H. Kommalapati and W.H. Zack, The SaaS Development Lifecycle, http://www.infoq.com/articles/SaaS-Lifecycle.

[7] T. Kao, C. MAo, C. Chang and K. Chang, “Cloud SSDLC:Cloud Security Governance Deployment Framework in Secure System Development Life Cycle”, Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on, pp.1143-1148, Jun. 2012

[8] G. Hoguland and G. McGraw, Exploiting Software:How to Break Code, 1st Ed. Addison-Wesley Professional, Feb. 2004

[9] Cloud Security Alliance, Security Guidance for Critical Areas of Focus in Cloud Computing V2.1, Dec. 2009

2017 한국정보보호학회 동계학술대회

C2 IoT보안(I)

오 성 준 (고려대)

- **사물 인터넷 환경의 통신 보안에 관한 연구**
  정진혁 (고려대학교)
- **랜섬웨어의 동향과 사물인터넷(IoT) 보안대책**
  김동휘 (고려대학교)
- **IoT 악성코드 사례 분석 및 미래 위협 제시**
  이지오, 이세빈, 염흥열 (순천향대학교)
- **도메인 기반 블록체인을 이용한 IoT 접근제어 연구**
  김미선, 박경우, 서재현 (목포대학교)

사물 인터넷 환경의 통신 보안에 관한 연구

정진혁

고려대학교 컴퓨터 정보통신 대학원 소프트웨어보안학과

A Study on the Communication Security of IoT

JIN-HYUK JUNG

Division of information security, Korea University.

요 약

사물인터넷(IoT)에 연결된 디바이스의 보안은 시간이 지날수록 중요한 사항이 되고 있고 디바이스간의 통신 구간의 보안, 특히 암호화가 중요해지고 있다. 이런 측면에서 IoT 환경에서 통신 구간 인증 및 암호화 용도로 주로 사용되는 DTLS (Datagram Transport Layer Security)는 중요성이 증가하고 있지만 IoT 환경의 특수성으로 고려해야할 것들이 있다. 즉, 암호화나 인증 과정은 많은 CPU 연산이 필요하고 디바이스 환경은 이를 수행하기에 충분치 않다는 점이다. 본 논문에서는 이런 환경에서 인증 및 암호화를 해결하기 위해 연구된 기존 제안들을 정리하고 이를 통해 고려할 점들과 향후 연구방향에 대해 제안하고자 한다.

I. 서론

사물 인터넷은 디바이스가 인터넷에 연결되어 data가 전송/수집/분석되는 과정을 의미한다. 이 논문에서는 데이타가 전송되는 구간의 보안 기술에 대해 논의할 것이다.

인터넷에 연결되는 디바이스가 많아 질수록 데이타는 무한히 증가할 것이고 이에 따른 보안의 중요성도 높아질 수 밖에 없다. 많은 사람들에게 중요한 정보와 판단의 자료를 제공하는 디바이스로부터 나오는 Raw 데이타는 위변조 되어서는 안되고 또한 불법적인 접근이 발생해도 안된다.

이런 관점에서 DTLS는 사물 인터넷에 사용되는 디바이스의 통신 보안 기술로 많이 언급되면서 사용되고 있다. DTLS가 무엇이고 사물 인터넷 환경에 기여하는 내용을 살펴볼 것이다. 또한 최적화된 기술을 이용하여 사물 인터넷에 적용할 수 있는 기존의 방법들을 조사하고 설명한 후 적용 시 고려 사항을 제안하고자 한다.

II.관련 연구

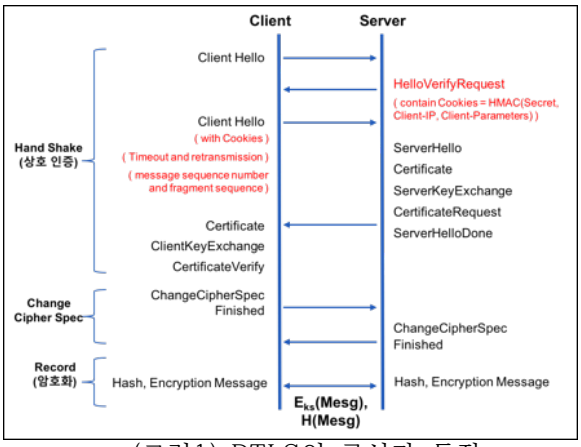
2.1 DTLS의 동작 개요

DTLS (Datagram Transport Layer Security) 란 TCP 프로토콜에 보안성을 제공하는 TLS 프로토콜을 UDP 프로토콜에 적용 가능하게 해주는 보안 프로토콜이다. DTLS는 통신을 하는 양단간의 송수신되는 데이타의 도청, 간섭, 메시지 변조 등 네트워크 상의 공격을 막을 수 있도록 도움을 준다. 특히, UDP 상에서 보안기능을 수행할 수 있기 때문에 사물 인터넷 통신 환경에서 보안성을 보증할 수 있는 수단이 되고 있다. DTLS의 구성과 TLS와의 차이점을 설명한다.

2.2 DTLS 구성 [1]

DTLS는 (그림1)에서와 같이 동작한다. 동작과정을 TLS와 동일하게 4가지 프로토콜로 분류할 수 있다.

HandShake / Change Cipher Spec / Record / Alert 프로토콜로 구성되어 있다.



(그림1) DTLS의 구성과 동작

2.3 TLS와의 차이점 [1]

TLS와의 큰 차이점은 UDP 상에서 동작하는 것으로 다음과 같은 내용이 TLS와 다르다. UDP의 비연결/비신뢰성으로 인해 발생한다.

먼저 Packet Loss가 발생할 수 있다. 그래서 DTLS에서는 Timeout 동작을 통해 packet retransmission을 할 수 있다. 또한 Sequence number를 handshake 안에 넣어두어 Packet Reordering에 활용한다. Retransmission이 발생하면 당연히 packet 안의 Sequence는 유지된다.

둘째로 Fragmentation Sequence number를 사용한다. UDP 상에서는 fragmentation이 발생할 경우가 많기 때문이다.

마지막으로 DOS Protection을 하기 위해 Stateless Cookies 값을 이용한다. 이를 사용하면 중복 레코드를 거부하며 재전송을 방지한다.

2.4 CoAP와 DTLS

국제 인터넷 표준화 기구(IETF) 에서는 제약을 가진 노드들을 위한 웹 프로토콜로 CoAP (Constrained Application Protocol)을 정의 하였으며, oneM2M, 유럽통신표준협회 (ETSI), 오픈 모바일 얼라이언스(OMA) 등 다른 표준단체에서 CoAP을 채택한 표준을 개발하고 있다. [2]

CoRE 워킹그룹은 2010년 결성되어 메모리가 작고, 처리 성능도 낮으며, 전력도 충분하지 않은 node 및 저전력에 트래픽 전송율도 낮은 네트워크 환경에서 동작할 수 있는 CoAP 프로토콜을 만들고 있다.

여기서 Node란 데이터 공간(RAM)이 10kbps 이하이고, 코드 공간(Flash)이 100kbps 이하인 Node로 정의되어 있으며 이를 기준으로 표준화에 반영한다.

CoAP은 사물 인터넷용 웹 프로토콜로서 기존의 HTTP 웹서비스와 잘 연동될 수 있다. 그래서 모든 자원을 URI로 명시하고 GET, PUT, POST, DELETE의 기본 메소드를 이용하여, 해당 자원에 대한 행위를 정의한다. 따라서 CoAP을 수행하는 Node들은 기존 웹서비스 구조와 쉽게 결합할 수 있으며 웹 기반 사물 인터넷 서비스를 구성할 수 있다.

프로토콜 경량화를 위해 UDP를 사용하는 것뿐만 아니라 프로토콜 레이아웃도 단순화 하여 전문의 Header는 Binary 형태로 사이즈를 줄였다. 추가적인 특징으로 RESTFul 방식을 지원하며, 암호화를 위해 DTLS를 사용한다. (TCP 기반이 아니므로 SSL/TLS는 사용 불가하다)

2.5 문제점

이상을 요약하면 사물 인터넷 환경에서 기기에서 발생하는 정보를 CoAP 프로토콜을 이용하여 수집, 교환하는데 CoAP는 UDP 를 기반으로 동작한다. UDP기반의 보안 프로토콜인 DTLS는 이 경우 통신 구간을 암호화 용도로 사용된다.

하지만 DTLS 프로토콜의 상호 인증과 암호화로 인한 통신의 부하, 성능 저하는 필연적이며 사물 인터넷 환경의 저사양 장비들에게 맞는 최적화 방안이 필요하다.

III. 기존 연구

위와 같은 문제로 기존에 다음과 같은 연구가 있었다.

3.1 경량화 암호 알고리즘 적용 [3,4]

DTLS를 사용하게 되면 RecordProtocol에서는 암호 알고리즘으로 국제 표준 알고리즘인 AES를 주로 사용하게 된다. 하지만 사물 인터넷 환경의 디바이스는 저성능, 저전력이라는 특성이 있고 이로 인해 연산 능력이 약하여 AES를 제

대로 사용하기 어렵다. 그래서 표 1과 같이 AES 대신에 경량 암호화 알고리즘을 사용하는 방법이 있을 수 있다. 처리 속도가 AES 보다 빠르다.

	LEA	HIGHT	PRINCE
알고리즘 구분	128 비트 블록	64 비트 블록	64 비트 블록
키 길이	128/192/256 bits	128 bits	128 bits

(표 1) 경량 암호화 알고리즘

3.2 해쉬 트리 기반을 이용한 경량화된 DTLS 메시지 인증 [5]

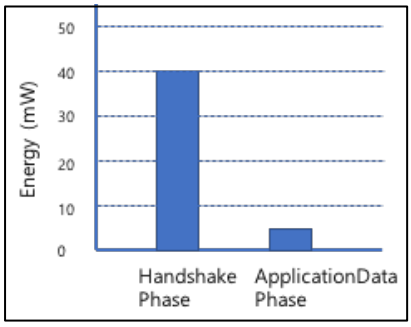
핸드쉐이크 과정이 종료되고 안전한 채널이 수립된 이후 DTLS에서는 메시지의 무결성을 제공하기 위해 HMAC을 사용한다. 발췌 논문에서는 이를 Hash Tree 기법을 이용하여 데이타 위변조와 인증자를 이용한 공격 (인증자 위조, replay attack)에 안전하며 통신 과정의 비용을 줄이는 방법을 제안하였다.

이 제안 방법은 검증을 위한 HMAC값을 Payload에 맞게 계속 보내지 않아도 되므로 통신 패킷량이 많을수록 기존의 DTLS 동작 방식보다 효율성이 좋아진다.

3.3 핸드쉐이크 과정의 개선

DTLS의 핸드쉐이크 과정은 공개키 연산을 수행하기 때문에 수행 성능이 저하되거나 소모 전력이 증가한다. 다만, 이런 방식은 사전에 안전한 방법으로 클라이언트와 Secrete key를 분배하는 가정이 있어야 한다.

발췌 논문[6]의 그림 2와 같은 측정 결과값을 보면 핸드쉐이크 계층의 에너지 소비량은 레코드 계층 대비 8배 이상을 소모하는 것을 확인했다.

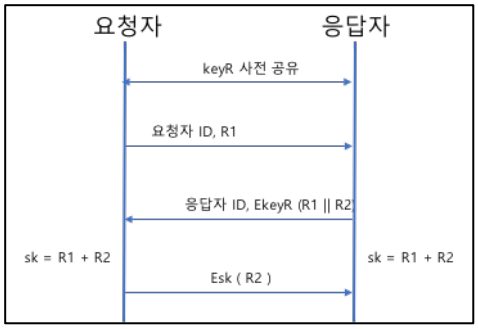


(그림 2) DTLS내 계층별 에너지 소비량 비교

1) 대칭키 기반 인증 및 세션키 동의 시스템 [7]

대칭키 기반 암호화 모듈만 사용할 수 있는 자원 제한적인 장치에서 장치 간 인증 및 세 션키 공유 방안을 제공하는 방식이다.

요청자와 응답자 사이에는 비밀키 값 keyR이 장치 설정 단계에서 사전에 안전하게 저장되어 있다고 가정한다. 상호 인증 절차는 다음과 같다.

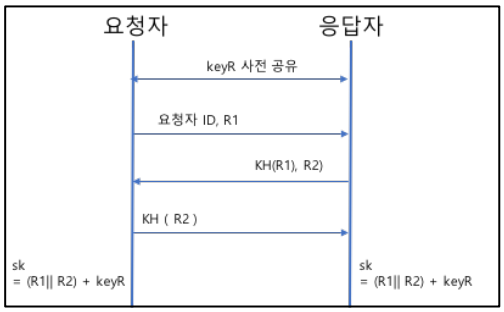


(그림 3) 대칭키 기반 상호 인증 절차

2) 해시 기반 인증 및 세션키 동의 시스템 [7]

해쉬 함수만 사용할 수 있는 자원 제한적인 장치에서 장치 간 인증 및 세 션키 공유 방안을 제공하는 방식이다.

요청자와 응답자 사이에는 비밀키 값 keyR과 해쉬 함수인 h(x)가 이 장치 설정 단계에서 사전에 안전하게 저장되어 있다고 가정한다. 상호 인증 절차는 다음과 같다.



(그림 4) 해쉬 기반 상호 인증 절차

위에서 제안한 대칭키/해쉬 기반의 인증 시스템의 경우 핸드쉐이크 과정에서 공개키 연산을 대칭키 연산이나 해쉬 함수로 대체하였다.

3.4 보안 적용 시 고려사항

위 내용에서 사물 인터넷 UDP 환경에 사용하는 DTLS의 제약과 자체 개선 방법 그리고 DTLS가 아닌 다른 방식을 통한 키 공유 방법에 대해 언급하였다. 이 논문에서는 사물 인터넷 환경의 디바이스 간 통신에 보안을 적용 시 효율성 높이기 위해 다음을 고려해야 한다고 제안한다.

1) 경량 암호화 알고리즘의 적용 : 경량 암호화 알고리즘은 저사양, 저전력을 위해 만들어진 것으로 대칭키를 사용하는 암호화 통신에서 적극적으로 사용되어야 한다.

2) 디바이스 사양과 사용자 환경에 맞는 암호화 통신 방식 선택

3) SW 보안 Patch : 디바이스에 설치되는 소프트웨어가 보안적인 침해를 당하는 경우를 대비해야 한다. 이를 위한 관리 정책과 기술적 기반을 마련해야 한다.

IV. 결론

본 논문에서는 사물 인터넷 환경의 제약 사항과 이에 따른 통신 상의 부하를 줄일 수 있는 여러 가지 기존의 방법들을 정리하였다. 그리고 이를 기반으로 세가지 고려사항을 제안하였다.

기존의 방법들 모두 사물 인터넷 환경에서 사용할 수 있는 좋은 대안들이지만 이런 대안들도 제안한 세 번째 고려 사항인 보안 Patch에 대한 뒷받침이 필요하다. 저사양, 저전력의 디바이스를 고려해 프로토콜을 개선한다 할 지라도 SW적인 결함을 수정할 수 있는 기술적, 정책적 기반이 없으면 보안적인 문제가 발생할 수 있다.

사물 인터넷 환경에서 공격에 의한 데이터의 위변조는 심각한 장애나 오류를 초래할 수 있는 문제이므로 디바이스들의 생명주기 전단계, 디바이스 사양 및 특성을 고려한 보안 기술 개발이 앞으로는 더 필요하다고 본다.

[참고문헌]

[1] RFC 6347 : Datagram Transport Layer Security Version 1.2

[2] RFC 7925 : The Constrained Application Protocol (CoAP)

[3] 송경진, 김동호 “저전력 IoT 상에서의 경량 DTLS 프로토콜 적용 방안” 2016년 춘계학술발표대회 논문집 제23권 제1호(2016. 4)

[4] 박기태, 한효준, 이재훈 “OpenSSL 상에서 LEA 설계 및 구현” 한국통신학회논문지 '14-12 Vol.39B No.12

[5] 이부형, 이성범, 문지연, 이종혁 “해시 트리 기반의 경량화된 DTLS 메시지 인증” 한국통신학회논문지 ’15-10 Vol.40 No.10

[6] 권혁진, 강남희 “사물인터넷에서 경량화 장치 간 DTLS 세션 설정 시 에너지 소비량 분석“ 한국통신학회논문지 '15-08 Vol.40 No.08

[7] 박지예, 신새미, 강남희 “사물 인터넷 환경에서 경량화 장치 간 상호 인증 및 세션키 합의 기술” 한국통신학회논문지 '13-09 Vol.38B No.09

랜섬웨어의 동향과 사물인터넷(IoT) 보안대책

김동휘

고려대학교 컴퓨터정보통신 대학원

소프트웨어 보안학과

Trends of the Ransomware and the Security Measures of Internet of Things

Dong-Hwi Kim

Dept. of Software Security, Korea University.

요 약

최근 여러 사이버 위협이 지속되는 가운데 랜섬웨어에 의한 피해사례가 늘고 있다. 서비스형 랜섬웨어(RaaS), 암호화 화폐의 등장으로 누구나 쉽게 공격을 할 수 있는 환경이 조성됐으며, 사물인터넷(IoT)시대가 도래하게 될 경우 모든 기기가 목표가 될 수 있어 이에 대한 대응책 마련이 시급하다 .본 논문에서는 새로운 국면을 맞은 랜섬웨어 공격 동향을 살펴보고 사물인터넷(IoT) 관점에서 대응방안을 모색해 본다.

I. 서론

랜섬웨어는 비즈니스에 가장 강력한 위협 중 하나다. 디지털화된 데이터, 중요한 응용프로그램 및 기타 시스템 등이 많이 사용되고 중요한 자산이 됨으로써 이러한 자산에 대한 접근이 막힌다면 기업에 큰 재앙이 될 것이다.

직원 교육 및 감염 예방 능력 향상을 위한 노력에도 불구하고, 랜섬웨어 공격은 여전히 지속되고 있다. 이러한 시점에서 기업은 현재의 위협에 따라 즉각적으로 대응함과 동시에 앞으로 일어나 미래에 어떻게 기업을 보호할 것인가도 고려해야 한다.

트렌드마이크로의 연구 보고서인 <랜섬웨어 - 과거, 현재 그리고 미래>에 따르면 랜섬웨어는 2000년대 중반 그 샘플이 처음 출현하면서 사이버 범죄 집단이 피해자를 강탈하기 위해 가장 많이 쓰는 도구로 급부상한다.

4차 산업혁명시대를 맞아 사물인터넷(IoT)가 상현실(VR)·증강현실(AR) 등 신기술이 부상하면서 보안 위협이 커지고 있다. 특히 최근 국내외에서 각종 해킹 사건이 발생하면서 보안 리스크에 대한 우려가 증폭될 전망이다. 미국 리서치업체 프로스트앤설리번은 “IoT 시장은 오는 2020년까지 아시아·태평양 지역에서만 그 규모가 793억달러(약 90조 8000억원)로 커질 것 “이라며 ”IoT가 랜섬웨어 공격을 받으면 재앙이 일어날 수 있다“고 경고했다.

본 논문에서는 랜섬웨어로 인한 사물인터넷(IoT)의 피해를 줄이고자 보안대책 방법론을 제시하고자 한다. 랜섬웨어의 동향을 알아보고 사물인터넷(IoT)의 각 구성요소 측면에서 대응 방안을 알아보고자 한다.

II. 관련연구

2.1 랜섬웨어 개념 및 역사

랜섬웨어란 몸값을 뜻하는 Ransom 과 Software의 합성어로 시스템을 잠그거나 데이

터를 암호화해 사용할 수 없도록 만든 뒤, 금전을 요구하는 악성 프로그램을 말한다.

초기 랜섬웨어는 2005년과 2006년 사이 러시아를 중심으로 퍼졌다. 그 후 2012년까지 다른 유럽 국가에서도 피해자들이 나타나기 시작했다.

2013년에는 악명 높은 CryptoLocker를 비롯한 '크립토 랜섬웨어' 샘플이 나왔다. 이런 타입의 감염은 암호화 시킨 데이터로의 접근을 차단할 뿐만 아니라 피해자가 몸값을 지불하지 않는 겨우 특정 기간이 지나면 암호화된 파일을 삭제할 수 있는 능력이 있어 점점 더 사용자에게 큰 위협이 되었다.

2015년 29개의 랜섬웨어군이 발견된 것과 비교해 2016년에는 247개 제품군으로 확대되며 752%라는 엄청난 증가율을 보여줌으로써 2016년은 "랜섬웨어의 해"라는 별명까지 얻게 되었다. 해마다 공격자들은 해커들과 함께 랜섬웨어 감염으로 약 10억 달러에 달하는 이익을 챙겼다.

2017년에는 4월 일어났던 WannaCry(이하 워너크라이)와 같은 새로운 랜섬웨어와 현재까지도 유럽의 기업, 정부 기관 및 유틸리티 제공업체를 대상으로 공격하고 있는 Petya(이하 페트야) 등이 차지하고 있다. 또한 The Verge에 따르면 2017년 6월 말 페트야의 변종으로 여겨졌던 새로운 랜섬웨어가 워너크라이 감염에 사용된 것과 동일한 EternalBlue 익스플로잇을 이용하는 사용자에게 영향을 미치고 있다고 보고되었다. 이 새로운 변종은 현재 'NotPetya'라고 불리고 있다.

2.2 대상의 변화

랜섬웨어의 대상이 변화하고 있다. 최근 호주 호텔이 랜섬웨어 공격으로 객실 문이 잠기는 사태, 우크라이나 지역 발전소의 대규모 정전 사태 등이 발생했다.

기존에는 PC와 보안이 취약한 서버를 대상으로 했다면 최근에는 모바일 기기를 감염시키는 사례가 나오고 있으며 미래에는 모든 기기에 인터넷이 연결되는 사물인터넷(IoT)을 목표로 하는 랜섬웨어가 활개를 칠 것으로 예상된다.

2.3 시장의 변화

랜섬웨어 시장이 커지면서 개발자간의 가격경쟁도 치열하게 진행되고 있다. 서비스형 랜섬웨어(RaaS)로 비용을 줄이고 수익을 높이는 방법이 제안되는가 하면, 단돈 45만원의 저렴한 가격으로 이용할 수 있는 랜섬웨어도 등장했다. 공격에 성공한 것 중 지금까지 알려진 가장 저렴한 랜섬웨어는 '필라델피아' 랜섬웨어로 45만원이면 블랙마켓에서 살 수 있다. 이 랜섬웨어는 2월 춘천 시내버스 정류장 버스정보 안내기가 감염된 바 있다.

2.4 환경의 변화

최근 사이버상에서 거래되는 전자화폐의 일종인 가상화폐의 시가총액이 140조원을 넘으며 급속도로 성장하고 있다. 거래 당사자를 추적, 확인할 수 없는 가상화폐는 일반 사용자들보다 해커와 범죄자들에게 유용하며 특히 금전을 요구하는 랜섬웨어의 주요 결재수단으로 이용되고 있다.

III. 랜섬웨어의 미래

4차 산업혁명 시대가 도달함에 따라 랜섬웨어의 공격 대상 범위가 넓어지고 있다.

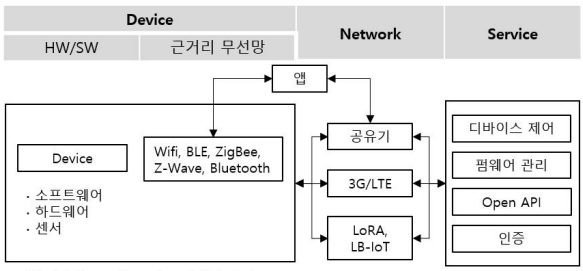
스마트TV 해킹 사고가 실제로 발생했다는 사실이 알려지면서 스마트 가전기기를 이용한 랜섬웨어 공격도 현실로 다가올 것이라는 경고도 나온다. 위키리크스가 공개한 CIA 사이버정보센터 문서에서는 CIA가 스마트 TV를 악성코드에 감염시켜 전원이 꺼진 상태에서도 방에서 들리는 소리를 수집해 CIA 서버로 전송한 사실이 기록돼 있었다.

만일 의료기기, 헬스케어 장비가 랜섬웨어에 감염된다면 수술이나 치료를 받는 도중의 환자 생명이 위험해질 수 있다. 주행 중인 스마트카에 랜섬웨어 공격을 진행해 자동차를 갑자기 멈추게 하거나 오작동을 일으켜 사고를 발생시킬 수도 있다.

보안대책이 시급하게 마련되지 않는다면 사물인터넷(IoT)은 RoT(Ransomware of Thing)로 변화하게 될 것이며 랜섬웨어는 인터넷에 연결

된 모든 것을 위협하게 될 것이다.

IV. 보안대책



- 사물인터넷(IoT) 구조 및 보안진단 범위

※ 출처 : LG CNS 2017.07

4.1 사물인터넷(IoT) 기기

초경량·저전력·저성능의 특성을 지니고 있는 기기가 인터넷에 연결될 때 기존 PC 환경에서 사용되던 암호화 알고리즘은 복잡하고 무거워서 사물인터넷 기기에서 사실상 구현하기가 어렵다. 따라서 사물인터넷(IoT) 기기에 최적화된 보안 기술과 엔드포인트 솔루션이 필요하다.

4.2 사물인터넷(IoT) 네트워크

사물인터넷에서 주로 사용되는 통신 기술로는 지그비(ZigBee), 와이파이(Wi-Fi), RFID(Radio Frequency IDentification), 블루투스 등이 있다. 사물인터넷의 경우 경량 장치들 간 및 경량 장치와 플랫폼 간의 정보 공유 시 환경을 고려한 경량화 보안 프로토콜의 사용이 고려되어야 한다. 데이터 전송과 사용자의 접근 권한을 안전하게 관리하는 검증된 보안 프로토콜을 고려해야 한다.

4.3 사물인터넷(IoT) 서비스/시스템

제조사와 서비스 제공자는 보안 취약점이 발견되면 이를 분석하고 보안패치를 신속히 배포하여 사후 조치 방안을 마련해야 한다. 또한 지속적인 패치 및 업데이트를 배포할 수 있는 여건을 갖추어야 한다. 하지만 사물인터넷(IoT) 기기들은 비용절감 및 효율성에 치중된 탓에 업데이트가 허술한 경우가 많은데 앞으로 출시되는 기기에 대해서 최소한의 보안 시스템을 의무화 시키는 정책이 필요할 것으로 보인다.

V. 결론

지금까지 랜섬웨어에 대한 보안의 중요성과 그 방안에 대해 논하였다. 최근 변종 및 진화된 다수의 랜섬웨어 출현과 함께 사람들이 그에 대한 경각심을 가지고 있다. 따라서 보안 기술이 발달하고 국가간 공조도 강화되면서 랜섬웨어 악성코드를 배포하는 것이 쉽지 않아졌다. 하지만 모든 사물이 인터넷에 연결되는 미래의 랜섬웨어는 계속적으로 변화를 시도하여 공격 영역을 넓힐 것이다.

논문에서 제시한 측면 외에도 다양한 방면으로 감염이 될 확률이 있기 때문에 사물인터넷(IoT) 관련 기술은 설계부터 폐기까지 보안 기술을 고려해야 한다. 이를 위해서는 사물인터넷(IoT) 보안에 대한 다양한 연구가 지속적으로 필요하다.

[참고문헌]

[1] Trend Micro, “Ransomware: Past, Present, and Future”, 2017

[2] 배상태 외, KISTEP InI 14호, “사물인터넷(IoT) 발전과 보안의 패러다임 변화”, 2016.7

[3] 한국인터넷진흥원, “IoT 공통보안원칙”, 2016.3

[4] 데일리안, http://www.dailian.co.kr/news/view/664662/?sc=naver, 2017.10

[5] DATANET, http://www.datanet.co.kr/news/articleView.html?idxno=110173, 2017.04

[6] DATANET, http://www.datanet.co.kr/news/articleView.html?idxno=110156, 2017.04

IoT 악성코드 사례 분석 및 미래 위협 제시

이지오, 이세빈, 염흥열

*순천향대학교 정보보호학과

Anaylsis of IoT targeted Malwares and their Future Threats

Ji-o Lee, Se-bin Lee, Heung Youl Youm

*Department of information security engineering, Soonchunhyang University.

요 약

사물인터넷(IoT)는 4차 산업혁명시대를 맞이하며 다양한 산업 간의 융·복합을 통해 다양한 시장에서 각광받고 있다. 사물이 인터넷에 연결됨에 따라 사용자들에게 다양한 편리함과 유용성을 제공하고 있다. 하지만 사용자들이 사용 하고 있는 IoT기기들은 사용자의 민감한 정보까지 소유하고 있어 보안위협이 발생할 경우 사용자의 사생활 유출 및 다양한 정보가 유출될 수 있어 매우 위험하다. IoT기기를 겨냥한 악성코드 공격이 발생하면서 IoT기기에 대한 보안의 중요성은 증가하고 있다. 본 논문에서는 IoT 기기를 공격한 보안 위협 동향과 대표적인 사례인 미라이 악성코드를 상세 분석하여 기술한다. 이후 나타날 수 있는 미래 보안 위협을 제시하며, 공통적인 특징을 가진 악성코드 위협을 최소화 시킬 수 있는 방안을 제시한다.

I. 서론

최근 인터넷의 빠른 발전을 통해 많은 인터넷 서비스 사용자가 어디에서든 인터넷을 통해 수많은 웹페이지에 접속하고 정보를 수집한다. 즉, 현대 사회에서 인터넷이라는 매개체는 인터넷 서비스 사용자에게 편리함을 제공하고 있다. 때문에 4차 산업혁명시대를 맞이하며 인터넷이라는 개념은 더 넓게 퍼져나가고 있다. 특히 사물 또는 물리적, 논리적 개체와 개체 통신이 네트워크로 구성되어 기존의 인터넷과 통신망으로, 지능형 인터페이스를 가지고 능동적으로 상호작용하는 기술인 IoT(Internet of Things)가 빠른 속도로 개발되고 있다. [1]

사물인터넷(IoT)이 전 세계적으로 확산 세에 접어들었다. 현실적으로 가능한 사례들이 발표되면서 IoT가 다양한 산업 간의 융·복합을 통해 전자상거래, 스마트 홈, 헬스케어, 지능형 차량 서비스, 스마트 그리드 등 다양한 시장에서 각광받고 있다. [1] 사물인터넷의 발전은 긍정적인 모습만을 보이는 것이 아니다. IoT 산업의 빠른 발전을 보안 기술은 따라가지 못하고 있으며, 이는 해커들에게 좋은 공격대상이 되었고 DDoS (Distributed Denial of Service)공격, 정보 유출, 사생활 유출 등의 문제를 발생시켰다.

IoT는 다양한 지역과 산업 전반에서 막대한 기회를 제공할 것임이 확실하다. 하지만 현재 IoT 산업에서 가장 큰 문제는 공격 위협이다. 근래까지도 IoT에 대한 기술 발전과 보안 기술 사이에 여전히 근본적인 격차가 존재한다. 특히 취약한 IoT기기를 겨냥한 악성코드 침투 공격은 해커들에게 애용되고 있다. 지난 2016년 10월 21일 DNS 서비스 제공업체 다인(Dyn)이 대규모 DDoS(Distributed Denial of Service) 공격을 받아 총 76개의 사이트 일제히 마비되거나 서비스가 지연되는 사건이 대표적이다.[2] 해당 악성코드는 취약한 IoT기기를 스스로 스캔하여, 악성코드에 감염시켜 거대한 봇넷을 구축하였다. 기존 PC를 통한 DDoS와는 규모가 다르다는 점에서 큰 이슈가 되었고, 이후 2개월만

본 논문은 과학기술정보통신부 및 정보통신기술진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음. [2015-0-00264 , IoT 환경에서 프라이버시 보호 국제 표준화]

* 순천향대학교 정보보호학과 (myjiolee11@gmail.com, tkaqls12@gmail.com, hyyoum@sch.ac.kr)

에 다양한 변종 악성코드를 유포하며 IoT 보안 위협이 심각함을 인지하게 하였다.

4차 산업혁명시대를 맞이하며 IoT 산업은 새로운 IT 기반의 융합서비스를 창출하고 있지만, 해커들이 침투할 수 있는 쉬운 환경이 만들어지고 있다. 지능적인 기기들로 편리함은 제공하고 있지만 그로 인해 개인의 사생활과 개인정보가 유출되는 위험한 사회가 될 수 있다는 것이다. 이를 대비하기 위해서 IoT 환경에서의 프라이버시 보호를 위한 표준화 제작이 중요하며, 이후 IoT 기기를 겨냥한 보안 위협에 대한 공격을 연구하여 기기에 보완해야한다.

본 논문에서는 IoT 기기를 공격한 보안 위협 동향과 대표적인 사례인 미라이 악성코드를 상세 분석하여 기술한다. 이후 나타날 수 있는 미래 보안 위협을 제시하며, 공통적인 특징을 가진 악성코드 위협을 최소화 시킬 수 있는 방안을 이야기한다.

II. 본론

2-1) IoT란

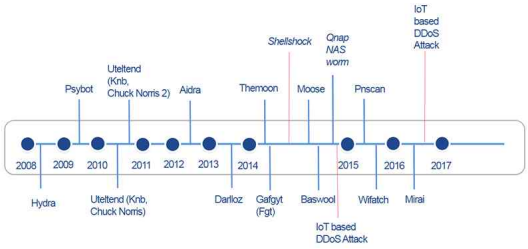
정보기술을 바탕으로 우리 생활 주변의 전자기기를 네트워크로 연결하여 사람과 사물 간 정보를 교류하고 상호 소통하도록 하는 서비스를 IoT라고 한다[3].

IoT는 스마트 디바이스, 클라우드 컴퓨팅, 빅데이터 등 IT 기술을 이용하여 다양한 산업과 융합을 통해 미래 산업을 창출하는 것에 의의가 있다. 예를 들면 자동차가 스스로 주차가 되는 서비스, 외부에서도 자신의 집을 통제할 수 있는 서비스, 상점에 들어서면 소비자가 관심이 있을 만한 상품의 쿠폰을 스마트폰으로 전송하는 서비스 등이 있다.

2-2) IoT와 악성코드

IoT 기술은 다양한 시장에서 각광받으며 빠른 발전을 하고 있지만, 절대 안전한 환경이 아니다. 오히려 빠른 발전으로 인해 보안적인 기술은 많이 미비하며 해커들에게는 공격하기 좋은 환경이 되고 있다. 해커들은 IoT 기기를 겨냥하여 악성코드를 침투시켜 적은 비용으로 큰 효과를 얻고 있다. 가장 대표적인 동작은 기기를 봇으로 만들어 분산 서비스 공격을 통해 서비스를 일시적으로 마비시키는 동작이다. 해당 악성코드들은 꾸준히 나타나며, IoT 환경의 위험성을 부각시키고 있다.

[그림 1]은 IoT 기기의 기반환경인 임베디드 리눅스를 겨냥한 악성코드의 타임라인이다.



[그림 1] -임베디드 리눅스 악성코드 타임라인[4]

2-3) 미라이(Mirai) 악성코드와 동작

지난 2016년 9월 21일 대규모의 DDoS 공격 사례가 발견되었다. 해당 DDoS 공격 규모는 평균 600Gbps이며, 기존의 DDoS 공격보다 규모가 크다. 해당 사례의 특이점은 DDoS 공격의 주체는 PC가 아닌 IoT 기기이다. 악성코드를 유포하여 IoT기기가 감염이 되면, 해당 IoT기기는 주변에 있는 다른 IoT 기기를 스캔하여 해당 악성코드가 겨냥한 취약점이 있다면 다른 IoT기기 또한 웜처럼 감염 시켰다[6].

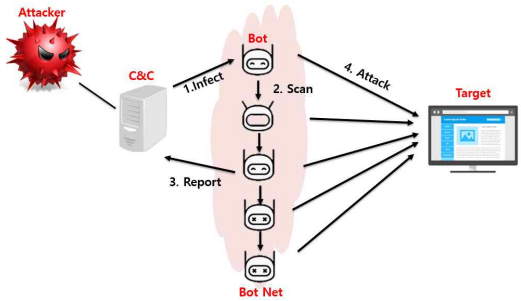
해당 악성코드는 Mirai라고 명명되었으며, IoT 기기들을 감염시켜 거대한 봇넷을 구축하였다. 2016년 10월 초 소스코드가 공개된 후 변형이 증가하고 있으며 UDP Flood, Syn Flood, ACK Flood 등의 다양한 DDoS 공격 기능을 가지고 있다.

미라이 악성코드는 기존 DDoS 악성코드와는 [그림 2]와 같은 차이점을 가진다.

	DDoS	Mirai
대상	PC	IoT 기기
전파	사용자 설치	Bot 스캔
감염	다양함	IoT 기기 취약점
대응	보안 패치, IP 차단	보다 어려움
기타사항	자가 파괴	접근 차단

[그림 2] - 기존 DDoS와의 차이점

미라이 악성코드의 동작 과정은 [그림 3]과 같다. 공격자가 한 IoT 기기를 감염시키고, 취약한 아이디와 패스워드를 가진 다른 기기를 찾아 전사적 공격을 하여, 원처럼 다른 기기들을 감염시킨다. 이후 C&C를 통해 봇넷을 관리하고 대상을 공격하게 한다.



[그림 3] - 미라이 악성코드 동작 과정

현재 미라이 악성코드의 소스는 공개되어 있는 상태이며, 다양한 변종 악성코드들이 발생하여 IoT 기기들을 위협하고 있다. 심지어 윈도우 미라이 악성코드도 나타나, 주변의 IoT 기기들을 감염시키게 한다.

미라이 악성코드 공개 소스는 Mirai Bot, C&C, Loader로 구성되어 있다. 대부분의 미라이 변종 악성코드들도 비슷한 형태로 이루어져 있다.

Mirai Bot은 스캔을 하여 다른 기기들을 감염시킨다. 주요 소스들의 행위는 [그림 4]와 같다.

Mirai Bot	요약
main.c	소켓 통신 및 Main 함수, 재부팅 방지
checksum.c	TCP, UDP Checksum
attack.c	DDoS 공격 수행
rand.c	랜덤 변수 및 페이로드 생성
resolv.c	도메인 -> IP 주소로 변경
Killer.c	메모리 스크래핑으로 원격 차단
scanner.c	다른 IoT 기기 장비 검색
Table.c	C&C 주소 및 난독화 함수 포함

[그림 4] - Mirai Bot 행위

미라이 악성코드가 봇넷을 구축하는 과정은 다음과 같다.

1) 다양한 랜덤 IP 대역을 생성하여 해당 IP들을 스캔시도를 한다.

2) 특정 23포트(Telnet)을 겨냥하여 생성한 IP 대역들에 접속한다.

3) 접속한 IP에서 해당 악성코드가 가지고 있는 IoT가 가장 많이 사용하는 62가지의 ID/PW로 각 기기들의 접속을 한다.

4) 접속 가능한 IoT 기기를 찾아 Loader를 통해 감염시키고, 감염 기기에 대한 정보를 C&C에 보고한다. [IP 주소, 포트, 접속가능유무, 접속ID/PW 등]

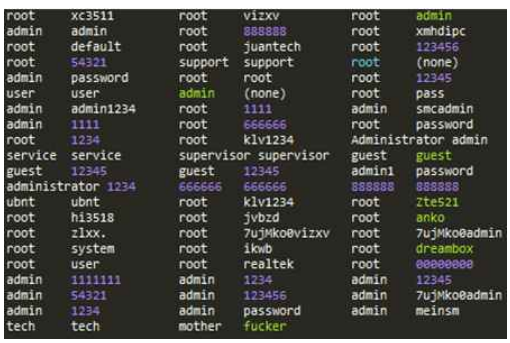
5) UDP Flood, Syn Flood, ACK Flood 등의 다양한 DDoS 공격을 가능하게 한다.

2-4) 미라이(Mirai) 악성코드 특이점

미라이 악성코드는 강력한 악성코드가 될 수 있는 몇 가지의 특징점을 가진다.

메모리 스크래핑(Memory Srcaping)이라는 기술을 이용하여 메모리에 계속 상주하면서 다른 장치 원격 연결을 제한하고, 자신만의 봇넷을 구축한다는 점이다[5].

두 번째는 전사적 공격을 통해 IoT 기기들이 보편적으로 많이 사용하는 ID/PW로 해당 기기에 접속한다는 점이다. 해당 부분은 보안 의식이 미비한 사용자들이 ID/PW를 변경하지 않아 생기는 경우이다. 해당 취약점은 간단하지만 효과적인 방법으로 미라이 악성코드를 빠르게 확산시킨다.



[그림 5] 사용되는 62가지 취약 ID/PW

세 번째는 웜의 특징을 가진다는 점이다. 미라이 악성코드는 Busybox의 wget명령어를 이용하여 취약한 ID/PW로 접속한 IoT기기에 Mirai 악성코드를 다운로드 받아 실행함으로써 악성코드를 전파시킨다.

마지막으로 임베디드 장비는 동작도중 멈춰 서비스가 중지되는 것을 방지하기 위해 자동 재부팅 기능(Watchdog)이 존재한다. 미라이 악성코드는 메모리의 상주하며 감염 시스템이 재부팅 하는 것을 방지하기 위해 Watchdog 기능을 무력화 시킨다.

또한 윈도우 미라이 악성코드는 기존의 미라이 악성코드보다 6개의 포트[22,135,445,1433,3306,3389]를 더 스캔하여, 감염 대상의 폭을 넓혔다. 이전의 이슈가 되었던 SMB 프로토콜도 윈도우 미라이 악성코드에서 먼저 사용되었다는 점이 있다.

2-5) 지속되는 IoT 보안 위협

미라이 악성코드는 IoT 기기를 통하여 거대한 봇넷을 구축하여 DDoS 공격을 수행한다는 점에서 큰 이슈가 되었다.

이후에도 IoT 위협은 지속적으로 발생하고 있다. 가장 대표적으로 발생한 보안 위협은 2017년 9월에 나타난 '아이오트룹(IOTroop)'이다. 아이오트룹은 9월 말에 발견되어 한달새 100만개 이상의 조직을 감염시켰다하며, 이는 미라이보다 더 빠르게 전파된다고 한다. 해당 악성코드는 다양한 무선 IP 카메라들을 대상으로 한다. 해당 악성코드는 취약한 기본 ID/PW 뿐만 아니라 다양한 취약점을 이용하여 감염시키기에 미라이 악성코드보다 빠른 전파로 봇넷을 구성하고 있다고 한다. 아이오트룹은 기술적인 측면에서 미라이와 유사점이 있지만 미라이보다 더 정교한 공격으로 평가받는다[7].

III. 결론

해당 사례들을 미루어보아 IoT 기기는 PC보다 더 많은 수량이 보급이 될 것이며, 해당 기기들을 빠르게 봇넷으로 만들어 공격자가 관리하여 공격을 수행하는 것이 중요하다. 때문에 다양한 기기들의 취약점을 지속적으로 노릴 것이기에, 디바이스에 대한 보안이 필수적으로 이루어져야한다. 또한 DDoS 공격만이 봇넷의 목적이 아닐 수 있다. 해커의 의도에 따라 각 기기에 포함되어 있는 개인정보가 유출 될 위험도 있다. 때문에 IoT 환경에서의 프라이버시 관련 표준이 중요시 될 것이다.

즉, 기술적인 보안뿐만이 IoT를 위협하는 악성코드의 대비할 수 있는 방법이 아니다. 현재까지는 IoT 기기를 주체로 하여 분산 서비스 공격을 하는 공격 사례들이 많지만, 이후에는 IoT 기기 내에 포함되어 있는 정보들이 더 위험한 공격대상이 될 수 있으니, 해당 개인정보를 보호하기 위한 수단이 제정되어야 한다.

[참고문헌]

[1] KISA, 2016년 Mirai 악성코드 동향, Dec, 2016

[2] Ahnlab, IoT 환경 위협하는 '리눅스 악성코드 Top 5, Dec, 2016

[3] 위키피디아, "사물인터넷", https://ko.wikipedia.org/wiki/%EC%82%AC%EB%AC%BC_%EC%9D%B8%ED%84%B0%EB%84%B7

[4] 임철수, IoT 서비스 활용사례 분석 및 산업 활성화 이슈, Dec, 2015

[5] 하우리, "IoT 기기 겨냥한 디도스 공격용 멀웨어 소스코드 공개", http://hauri.co.kr/information/news_view.html?intSeq=9030&page=1, Oct, 2016

[6] AhnLab, "안랩, POS 내 금융정보 노리는 악성코드 주의 당부", http://blog.ahnlab.com/ahnlab/tag/8343, Mar, 2016

[7] 알약, "급격히 증가하고 있는 새로운 IoT 봇넷", http://blog.alyac.co.kr/1374, Oct, 2017

도메인 기반 블록체인을 이용한 IoT 접근제어 연구[1)]

김미선*, 박경우**, 서재현*
**목포대학교 정보보호학과
**목포대학교 융합소프트웨어학과

A Study on IoT Access Control using Domain base Blockchain

Mi-Sun Kim*, Kyung-Woo Park**, Jae-Hyun Seo*
*Dept of Information Security, Mokpo National University
**Dept of Software Convergence Engineering, Mokpo National University

요 약

가상화폐를 통해 알려진 블록체인 기술은 분산형 상호인증을 기반으로 하며, 이는 다양한 장치들이 상호 통신을 통해 데이터를 처리하는 IoT 기술과 밀접한 연관성을 갖고 있어 IoT 환경의 다양한 응용을 위해 연구되어 지고 있다.

본 논문은 중앙 서버의 통제없이 도메인 내의 IoT 장치간의 작업 증명을 통해 각 장치의 권한을 증명하고, 이를 수행할 수 있는 도메인 기반 블록체인을 구성하는 IoT 접근제어 방법을 제시한다.

I. 서론

블록체인은 누구나 네트워크에 참여할 수 있는 퍼블릭 블록체인(Public Blockchain)의 개념으로 고안되었으나, 불특정 다수가 참여하는 네트워크에서 운영됨에 따라 관리, 기술적 문제를 안고 있다[1].

이에 다양한 응용분야에서는 한정된 참여자만으로 분산 네트워크를 구성하는 프라이빗 블록체인(Private Blockchain) 기술이 부각되고 있다.

블록체인 기술이 IoT를 위한 보안 이슈로 대두됨에 따라, IoT 환경에 블록체인을 통합하는 기술이 연구되고 있다[2]. 그러나, IoT 환경 내의 장치는 대용량의 블록체인의 저장을 위한 공간 확보 및 처리에는 어려움이 있을 수 있어 본 연구에서는 프라이빗 블록체인을 기반으로 멀티 블록체인을 구성하여 작업증명을 통한 접근제어를 수행하고자 한다.

기존 연구[3]를 통하여 CapSG라는 게이트웨이를 통하여 중앙 집중적인 Capability 토큰의 발행 및 관리를 수행하는 IoT 접근제어 시스템을 구현하였으며, Capabiltiy 토큰 기반의 접근제어의 효율성을 증명하였다. 그러나 토큰 발행에 있어서 중앙집중적인 관리가 이루어지고 있어, IoT 장치 및 사용자의 수가 증가함에 따라 이를 처리하기 위한 게이트웨이의 부하가 많아지고 게이트웨이가 발행하는 토큰에 대한 신뢰성에 대한 문제가 발생할 수 있다는 약점을 가지고 있다.

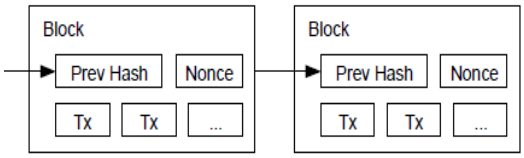
본 논문은 도메인 내의 IoT 장치간의 작업 증명을 통해 각 장치의 권한을 증명하고, 이를 수행할 수 있는 도메인 기반 블록체인을 구성하는 IoT 접근제어 방법을 제시한다.

II. 본론

1) 본 논문은 한국전력공사지원 스마트 에너지 캠퍼스 사업으로 지원된 연구임.

2.1 프라이빗 블록체인과 작업증명

블록체인은 가상화폐의 하나인 비트코인의 기반 기술로 사용되는 일종의 분산 공개 디지털 장부를 의미한다. 비트코인에서는 블록에 대해 다수의 사용자로부터 작업증명을 수행하고 이를 통해 거래내용에 대한 인증을 받는다. 블록의 위변조 여부를 판단하기 위해 트랜잭션이 모인 블록의 해시를 만들게 되며, 이 해시는 이전 블록의 해시를 포함하고 그림 1처럼 논리적으로 연결되어 있어 블록체인을 이룬다.



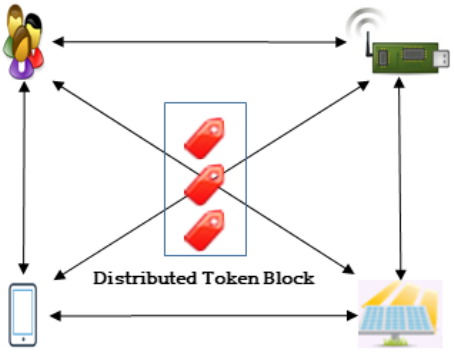
(그림 1) 블록체인 개념도[4]

비트코인에서 작업증명은 다수의 의사 결정을 통해 작업에 대한 인증을 수행하는데 하나의 장치(CPU)가 하나의 작업에 대해 한 번의 의사결정을 수행할 수 있도록 한다[4]. 이로 인해 다수의 장치를 가지거나, 다수의 의견을 모을 수 있는 누군가로 인한 위변조를 피할 수 없다는 문제를 갖기 때문에 충분히 많은 참여자를 포함한 분산 인프라에 적합하다.

그러나, 비트코인이 채택한 퍼블릭 블록체인 기술은 불특정 다수가 참여하는 네트워크를 유지하고 관리하는데 많은 자원을 투입해야 한다는 문제, 중요 내부 정보가 투명하게 공개되는 문제, 느린처리 속도와 거래자의 익명성 등 여러 가지 단점을 가지고 있다[1]. 이에 대한 대안으로 프라이빗 블록체인 기술이 부각되고 있으며, 이는 한정된 참여자만으로 분산 네트워크를 구성하는 허가형 분산원장의 형태로 구현된다.

2.2 블록체인을 이용한 분산형 IoT 접근제어

도메인 기반 블록체인을 이용하는 접근제어 방법을 연구하기 위하여 기존 연구[5]를 통하여 그림 3과 같이 블록체인을 이용한 분산 토큰 블록을 구성하여 다수의 참여자의 인증을 통하여 토큰의 생성, 위임, 폐기 과정을 수행할 수 있는 블록체인 기반 분산형 IoT 접근제어 시스템을 구성하였다.



(그림 2) 블록체인 기반 분산형 IoT 접근제어 시스템

구성된 시스템에서 각 노드는 일반 노드와 풀 노드로 구분할 수 있다. 일반 노드는 트랜잭션을 생성하고 전송하는 기능을 가지며, 풀 노드는 일반 노드의 기능과 함께 전체 블록체인을 유지하고 블록을 생성, 확인하는 노드들을 의미한다. 풀 노드는 토큰의 작업증명 과정에 동일한 참여권을 갖을 수 있으며, 블록을 저장할 수 있는 저장소를 유지하여야 한다.

또한, 본 시스템은 프라이빗 블록체인 구성을 위하여 블록체인 관리자를 둘 수 있으며, 블록체인 관리자는 일반 노드와 풀 노드의 생성, 도메인 생성 그리고 도메인간 블록체인을 구성할 수 있는 정책을 생성하는 기능을 포함한다.

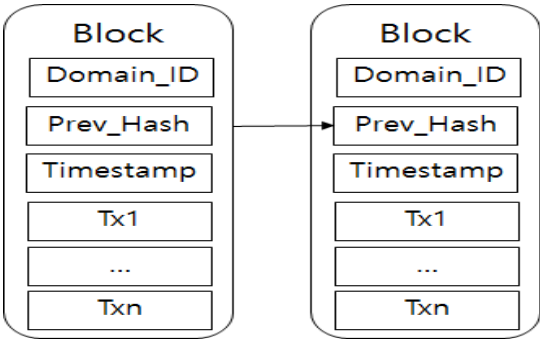
기존의 블록체인이 가상화폐에 대한 트랜잭션 작업에 국한되어 있다면, 본 시스템에서 블록체인은 다양한 리소스에 대한 접근 권한을 포함하고 있기 때문에 단일 블록체인으로 모든 정보를 유지하는데에는 한계가 있을 수 있다. 또한, 대용량의 블록체인의 저장을 위한 공간 확보 및 처리에는 어려움이 있을 수 있어 본 연구에서는 도메인을 통한 멀티 블록체인으로 구성하고자 한다.

2.3 도메인 기반 블록체인 구성

그림 3과 같이 IoT 환경내의 리소스에 대해 도메인별 분류를 하고 도메인별로 블록체인을 구성한다.

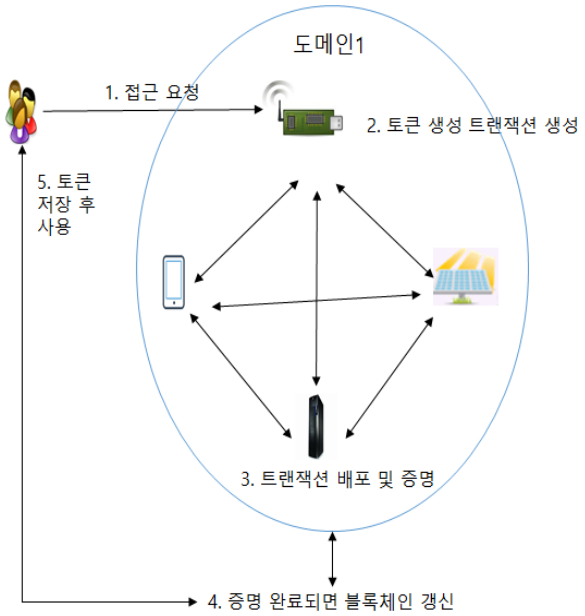
이 방법에서 블록들은 도메인 식별자(Domain _ID)를 갖으며, 도메인 내 토큰에 대

한 트랜잭션들로 구성될 수 있다. 이 방법은 어떤 리소스들을 하나의 도메인으로 관리할 것인가의 문제를 포함하고 있어, 프라이빗 블록체인과 분산형의 결합된 방법으로 처리될 수 있다.



(그림 3) 도메인 기반 블록체인

도메인은 IoT 환경내의 영역내의 장치들로 구분하며, 각 도메인 내의 블록에 대한 토큰의 생성, 위임, 폐기에 대한 트랜잭션이 도메인 블록에 포함된다. 해당 도메인 블록에 대한 작업증명은 도메인 내의 장치들로 국한한다. 도메인 기반 블록체인의 작업증명 과정은 그림 4와 같이 진행된다.



(그림 4) 도메인 기반 블록체인 작업증명 처리

각 블록체인은 도메인 ID로 식별되며, 트랜잭션 내역은 도메인 내의 장치들에게만 공개된다.

도메인과 도메인 사이의 트랜잭션 처리는 블록체인 관리자를 통하여 정의된 정책에 의하여 새로운 블록체인을 구성하여 유지, 증명될 수 있다.

III. 결론

본 논문은 중앙 서버의 통제없이 도메인 내의 IoT 장치간의 작업 증명을 통해 각 장치의 권한을 증명하고, 이를 수행할 수 있는 도메인 기반 블록체인을 구성하는 IoT 접근제어 방법을 제시한다.

프라이빗 블록체인 개념을 도입하여 도메인 내의 구성원들에 의해 작업증명이 이루어짐에 따라 트랜잭션의 내용이 익명의 모든 개체에게 노출됨을 제어할 수 있으며, 도메인별로 블록체인을 구성함에 따라 저장공간을 최소화할 수 있다는 장점을 갖는다.

추후 상세 아키텍쳐 설계 및 시스템 구현에 대한 연구를 진행하여 시스템의 보안성을 평가할 것이다.

[참고문헌]

[1] 백종찬, 한승환, 안상욱, 김영진 and ChrisHong, 블록체인 기술의 발전과정과 이해,(주)피넥터, 2016.

[2] A. Ouaddah, A. Abou Elkalam and A. Ait Ouahman, “FairAccess: a new Blockchain-based access control framework for the Internet of Things”, Security and Communication Networks, pp. 5943-5964, 2017.

[3] Jin-Bo Kim, Deresa Jang, Mi-Sun Kim and Jae-Hyun Seo, "The Access Control platform of the IoT Service using the CapSG",KIPS Tr.Software and Data Eng., Vol 4, No 9, pp. 337-346, 2015.

[4] Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, bitcoin.org, 2009.

[5] 김미선, 박경우, 서재현, “블록체인을 이용한 토큰 기반 IoT 접근제어 연구”, 한국정보처리학회 추계학술대회 논문집, 2017.

2017 한국정보보호학회 동계학술대회

C3 IoT보안(II)

이 병 천 (중부대)

- **Leakage Attack과 태그 가장 공격에 안전한 PUF 기반 RFID 인증 프로토콜**
  권태연 (고려대학교), 변진욱 (평택대학교), 정익래 (고려대학교)
- **NB-IoT 적용사례 분석을 통한 안전성 향상 방안 연구**
  우시재, 곽진 (아주대학교)
- **무선 LAN 취약점에 따른 IoT 보안**
  김지원, 최효경, 김지수, 이세은, 최나래, 최은정 (서울여자대학교)
- **IoT 보안 최신 동향 분석**
  김지원, 최효경, 안혜준, 최은정 (서울여자대학교)

Memory Leakage Attack과 태그 가장 공격에 안전한 PUF 기반 RFID 인증 프로토콜†

권태연*, 변진욱**, 정익래*

*고려대학교 정보보호대학원 정보보호학과
**평택대학교 정보통신학과

PUF-Based RFID Authentication Secure against Memory Leakage Attack and Tag Impersonation Attack

Taeyean Kwon*, Jin Wook Byun*, Ik Rae Jeong*

*Department of Information Security, Graduate School of Information Security, Korea University
**Department of Information and Communication, Pyeongtaek University

요 약

PUF(Physically Unclonable Function)는 디바이스에 하드웨어적으로 구현되어 디바이스가 갖는 고유한 성질을 노이즈로 활용하여 같은 입력 값에도 랜덤한 값을 출력한다. 이러한 PUF는 경량 장비에서도 쉽게 구현될 수 있기 때문에 태그 RFID 프로토콜에 암호 프리미티브로서의 활용도가 매우 높다. 본 논문에서는 이러한 PUF를 이용하여 태그 RFID 인증 프로토콜을 안전하게 설계한 기존 연구를 분석하고, 이를 더 안전하면서도 효율적으로 개선시킨 방법을 제안한다. 제안하는 프로토콜은 태그와 리더간의 상호인증과 forward secrecy를 제공하면서 memory leakage attack과 태그 가장 공격에 안전하도록 설계되었으며, 기존 프로토콜에 존재했던 비효율적인 면을 개선하여 프로토콜의 성능을 향상시킨 프로토콜이다.

I. 서론

RFID(Radio-Frequency Identification)는 리더의 라디오 주파수 대역을 이용해 무선으로 데이터를 주고받을 수 있는 방식으로 태그의 ID를 식별하는 시스템을 말한다. RFID는 급속도로 대중화되어 많은 곳에서 사용이 되고 있는 반면에 사용자의 위치추적과 같은 개인의 프라이버시 등의 문제들이 발생하게 되었다. 이를 위해 기존의 연구들은 안전한 태그 RFID 통신을 위한 프로토콜을 제안해 왔지만, 키 관리나 물리적 공격으로 인한 비휘발성 메모리 노출과 같은 공격에 안전하며 효율적인 프로토콜을 제안하는 연구는 아직 진행 중에 있다.

본 논문은 태그를 임의로 복제할 수 없도록 같은 입력 값에도 랜덤한 값을 출력하는 PUF 기반 RFID 프로토콜을 제안한다. PUF 사용 시 동일한 암호 키 도출을 위해 Fuzzy Extractor를 적용한다. 또한 태그의 프라이버시를 보장하면서 서버의 검색 시 소모되는 연산량을 최소화하기 위해 Diffie-Hellman 기반의 일회용 키와 배타적 연산 그리고 일방향 해쉬함수를 적절히 활용한다. 제안되는 프로토콜은 2013년도에 Moriyama 등이 제시한 PUF기반 인증 프로토콜을 [1] 개선한 것으로, 이 프로토콜이 제공하는 보안 성질을 동일하게 제공하면서 기존의 프로토콜보다 서버 측 성능을 향상시켰다.

II. 관련 연구

Moriyama 등은 memory leakage attack에 안전한 PUF 기반 인증 프로토콜을 제안하였다 [1]. 제안된 논문은 암호학적 키를 얻기 위해 Fuzzy Extractor를 사용하고 PRF 함수를 이용

† 이 논문은 2017년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. 2017R1D1A1B03032424)

하여 전송하는 메시지의 무결성을 제공한다. 이 프로토콜은 매 세션마다 비휘발성 메모리에 저장되어 있는 비밀 값들을 안전하게 갱신시키는 키 업데이트 메커니즘을 추가적으로 제안하여 태그 가장 공격과 태그 트레이싱 공격에 안전하도록 설계되었으며, Forward secrecy를 제공한다. 하지만 이 프로토콜은 현재 인증을 수행하는 태그의 식별자를 모르기 때문에, 리더에 등록되어 있는 모든 태그에 대해 인증 값을 생성하여 비교해야만 하는 비효율적인 상황이 발생한다. 본 논문에서는 Moriyama 등과 같이 태그 가장 공격과 태그 트레이싱 공격에 안전하고, Forward secrecy를 제공하면서도 리더의 성능을 향상시키는 방법을 제안한다.

III. 암호 프리미티브

3.1 PUF (Physically Unclonable Function)

PUF는 물리적으로 복제가 불가능한 함수를 말하며, 디바이스 내에 하드웨어로 구현되어 있다. 디바이스 내에 물리적인 특징을 반영하여 같은 입력에 대해서도 매번 랜덤한 값을 출력하기 때문에 구현되어 있는 기기의 특성에 많은 영향을 받는다. 따라서 같은 입력에 대해서도 매번 랜덤한 값을 출력하며, 이러한 특징으로 인해 공격자가 시스템에 물리적인 공격을 수행한다 할지라도 출력 값을 제외한 어떠한 정보도 얻을 수 없다.

PUF가 지니는 성질은 Robustness, 예측 불가능성(unpredictable), 물리적 복제 불가능성(physically unclonable)으로 크게 세 가지를 말할 수 있다. Robustness는 같은 입력에 대해서 매번 랜덤한 값을 출력하여도, 그 값은 일정한 HD(hamming distance) 값 내에 존재하는 것을 의미한다. 예측 불가능성은 PUF의 (입력, 출력) 쌍을 많이 알고 있다고 할지라도 새로운 입력에 대한 출력을 예상할 수 없음을 의미한다. 마지막으로 물리적 복제 불가능성은 구분이 불가능한 두 개의 PUF를 생성하는 것은 어렵다는 것을 의미한다.

$(d,n,l,h,\varepsilon_{puf})-$secure PUF는 다음 세 가지 사항을 만족해야 한다.

① 임의의 입력 $y_1,y_2\in\{0,1\}^k$에 대해, 동일한 PUF와 동일한 입력 값에 대한 출력 값은 threshold 값인 d 내의 값을 갖는다.

② n개의 PUF와 l개의 서로 다른 입력 값들에 대한 최소 엔트로피는 h 이상을 만족해야한다.

③ 공격자가 PUF에 물리적인 공격을 수행하더라도, 출력 값 외에 얻을 수 있는 비밀 정보는 없다.

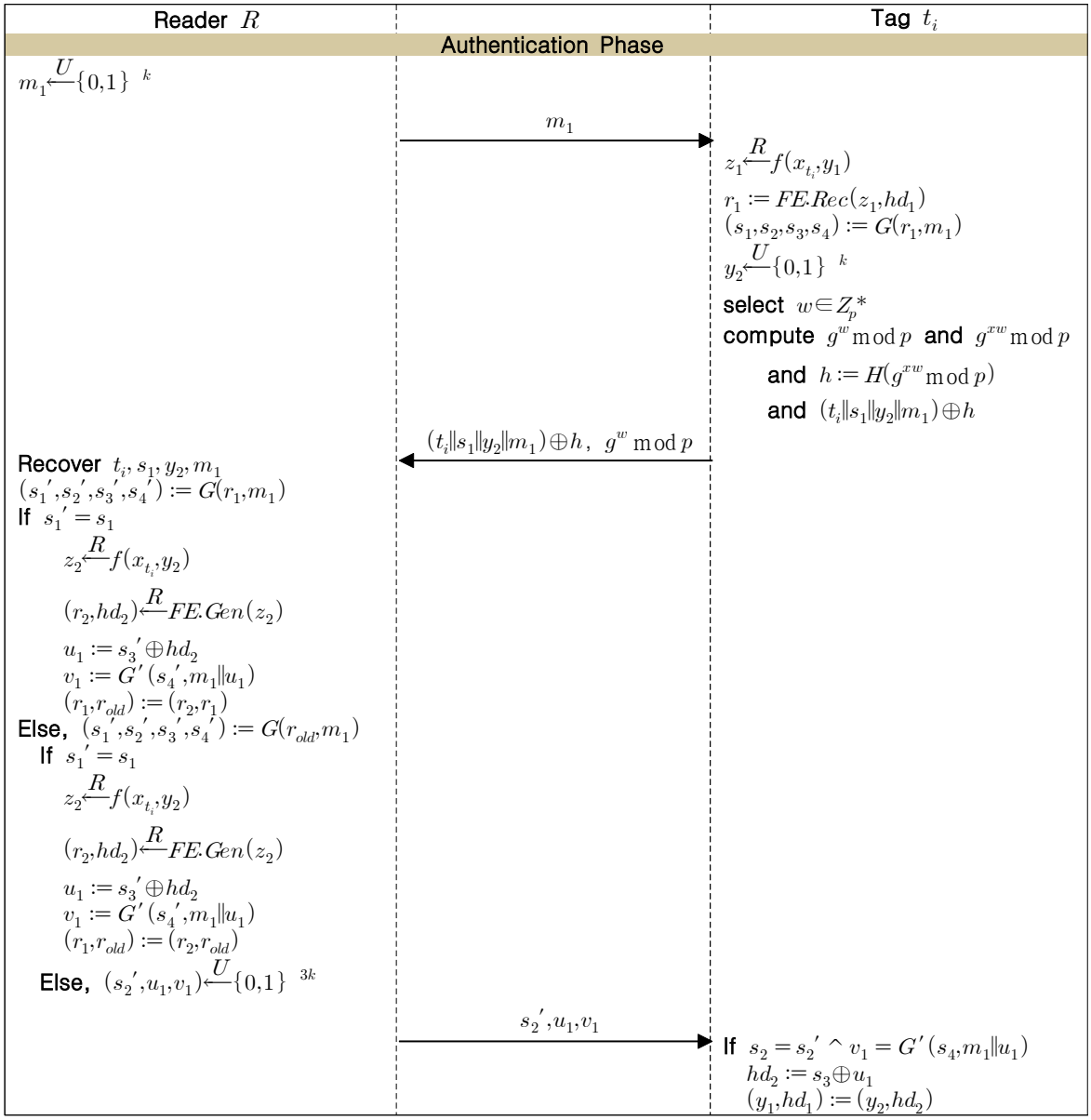
3.2 Fuzzy Extractor

Fuzzy Extractor(:=FE)는 충분히 랜덤한 키를 생성 혹은 복구하는 연산자로, 키를 생성하는 $FE.Gen$ 알고리즘과 키를 복구하는 $FE.Rec$ 알고리즘으로 구성되어 있다. $FE.Gen$ 알고리즘은 입력 z에 대해 키 r과 helper data인 hd를 출력한다. $FE.Rec$ 알고리즘은 $z'(z$와 유효한 HD 내에 존재하는 값)과 hd를 입력으로 r을 복구한다. $(r,hd):=FE.Gen(z)$가 주어졌을 때, HD의 threshold 값인 d와 최소 엔트로피 값인 h에 대해 $HD(z,z')\le d$와 $\overline{H}_\infty(z')\ge h$를 만족하면 $r'=FE.Rec(z',hd)$은 r과 같은 값으로 복구된다. 본 논문에서는 위를 만족하는 FE를 $(d,h,\varepsilon_{fe})-$secure FE라 정의한다.

IV. 제안하는 프로토콜

Moriyama 등이 제시한 PUF기반 RFID 인증 프로토콜은 키 업데이트 메커니즘을 사용하여 매 세션마다 비휘발성 메모리에 저장되어 있는 비밀 값들을 안전하게 갱신하는 것으로 태그 가장 공격과 태그 트레이싱 공격에 안전하도록 하였으며, Forward secrecy를 제공하였다.

하지만 Moriyama 등의 프로토콜은 태그에서 리더로 인증 값들을 전송할 때, 리더는 어느 태그에서 인증 값들을 생성하여 보내는지 알 수 없다. 따라서 인증 값들을 검증할 때마다 등록되어 있는 모든 태그들에 대해 인증 값들을 생



[표 2] 제안하는 PUF기반 RFID 인증 프로토콜

성 및 비교해보고, 일치하는 값에 해당하는 태그를 인증에 참여하고 있는 태그라고 판단한다. 또한 현재 인증에 참여하고 있는 태그가 이전 세션에서 키 업데이트를 실패했을 경우를 감안하여, 이전 세션에 사용된 키를 함께 저장하고 그 키를 이용한 인증 값으로 다시 한 번 검증을 수행한다.

본 논문에서는 효율성을 증진시킨 PUF 기반 RFID 인증 프로토콜을 [표 2]에 제안하였다.

4.1 제안하는 프로토콜 설명

먼저 리더에서 m_1을 비콘 메시지로 브로드캐스팅하면, 인증을 수행하고자 하는 태그는 비휘발성 메모리에 저장되어 있는 비밀 값 y_1을 PUF의 입력으로 하여 출력한 z_1을 $FE.Rec$ 알고리즘의 입력으로 암호키 r_1을 계산한다. 암호키 r_1를 기반으로 리더의 비콘 메시지 m_1을 PRF의 입력으로 하여 얻은 (s_1, s_2, s_3, s_4)는 인증에 사용될 값으로 사용된다.

태그가 리더에게 보내는 값은 해쉬된 세션키로 암호화 한 값 $(t_i \| s_1 \| y_2 \| m_1) \oplus h$, 해당 세션에서 사용할 태그의 공개키 $g^w \bmod p$이다. (단, t_i : 태그식별자, s_1 : 인증 값, y_2 : 다음 세션키

를 생성하기 위한 비밀 값, m_1 : 비콘 메시지)

리더는 태그가 보낸 값을 받아 t_i, s_1, y_2, m_1을 복구하고, 태그식별자가 t_i인 태그의 비밀 값들을 이용하여 생성한 인증 값이 태그가 보낸 인증 값과 일치하는지 검증한다. 검증이 성공하면 키 업데이트를 수행하고, 실패할 시 이전 세션에 사용된 비밀 값을 이용하여 검증을 한 번 더 수행한다. 만약, 이전 세션에 대한 비밀 값으로도 검증에 성공하지 못한다면, 공격으로 간주하여 키 업데이트는 수행하지 않는다.

리더에서 검증이 성공하고 키 업데이트가 수행되면 태그에게 s_2', u_1, 그리고 v_1를 전송한다. (단, s_2' :새로운 인증 값, u_1 : 다음 세션에서 사용될 키를 복구하는 데에 필요한 hd_2를 s_3'로 암호화한 값, v_1 : u_1의 무결성을 확인할 수 있는 값) 태그는 이 새로운 인증 값 s_2'과 v_1이 자신이 계산한 값과 일치한다면, 키 업데이트를 수행한다.

리더에서의 키 업데이트
$(r_1, r_{old}) := (r_2, r_1)$ 혹은 $(r_1, r_{old}) := (r_2, r_{old})$
태그에서의 키 업데이트
$(y_1, hd_1) := (y_2, hd_2)$

[표 3] 리더와 태그에서의 키 업데이트

본 논문에서 제안하는 프로토콜은 전송되는 데이터의 암호화를 위해 공개키 기반의 세션키를 해쉬하여 XOR하는 One-Time Pad 방법을 사용하고 있다. 따라서 Moriyama 등에서 제안하는 프로토콜보다 강한 안전성을 제공하며, 태그식별자를 전송함으로써 리더에 등록되어 있는 모든 태그에 대한 인증 값들을 생성할 필요가 없어졌다.

Moriyama 등의 프로토콜
$n\log_2 n \times COMP + 2 \times PUF + FE.Rec + FE.Gen + (n+1) \times PRF6 + 4 \times PRF1$
제안하는 프로토콜
$\log_2 n \times COMP + 2 \times PUF + FE.Rec + FE.Gen + 2 \times PRF4 + PRF1 + 3 \times \exp + 2 \times hash$

[표 4] Moriyama 등과 제안하는 프로토콜의 연산량

[표 4]는 Moriyama 등과 제안하는 프로토콜에서 요구되는 연산량을 정리한 것으로, 증가하는 연산량은 Hashed DH로 인한 $3 \times \exp + 2 \times hash$에 비해 $(n-1)\log_2 n \times COMP + (6n+1) \times PRF1$만큼 연산량이 감소하는 것을 알 수 있다. (단, $COMP$는 DB에서 태그식별자를 검색 및 비교하는 연산을 의미하고, PRF#는 PRF 함수의 출력이 #개인 것을 의미한다.)

이는 공개키를 사용하여 데이터를 안전하게 전송하는 것은 물론이거니와 그럼에도 불구하고 프로토콜 상의 연산량을 줄이고 성능을 향상시킨다는 측면에서 효율적임을 보인다.

V. 결론 및 향후연구

본 논문에서는 Moriyama 등이 제안했던 PUF 기반 RFID 인증 프로토콜을 효율적으로 개선하되 memory leakage와 태그 가장 공격의 안전성을 그대로 유지했다. 안전성 모델 및 안전성 증명에 대한 연구가 현재 진행 중이고 더 나아가 제안된 프로토콜의 효율성을 입증하기 위해 실험연구도 함께 진행 중이다.

[참고문헌]

[1] D.Moriyama, S.Matsuo, and M.Yung, PUF-Based RFID Authentication Secure and Private under Memory Leakage, IACR Cryptology ePrint Archive, p. 712, 2013.

[2] G.Hanaoka and K.Kurosawa, Between Hashed DH and Computational DH: Compact Encryption from Weaker Assumption, IEICE transactions on fundamentals of electronics, communications and computer sciences, Vol. 93(11), pp. 1994-2006, November, 2010.

[3] 변진욱, PUF 기반 RFID 인증 프로토콜의 효율적 설계에 관한 연구, 정보보호학회논문지, Vol. 24(5), pp. 987-999, 2014.

[4] A.Arbit, Y.Livne, Y.Oren, and A.Wool, Implementing public-key cryptography on passive RFID tags is practical, International Journal of Information Security, Vol. 14(1), pp. 85-99, 2015.

NB-IoT 적용사례 분석을 통한 안전성 향상 방안 연구

우시재*, 곽 진**

*아주대학교 컴퓨터공학과 정보보호응용및보증연구실

**아주대학교 사이버보안학과

A Study on the Improvement of Security through Analysis of NB-IoT Use Case

Si-Jae Woo*, Jin Kwak**

*ISAA Lab., Department of Computer Engineering, Ajou University.

**Department of Cyber Security, Ajou University.

요 약

IoT 기술의 연구가 활발하게 이루어지면서 저전력 광범위를 지원하는 LPWA 네트워크 기술이 연구되고 있다. LPWA 네트워크 기술이 발전하면서 이에 해당하는 기술 중 하나인 NB-IoT 기술에 대한 연구도 활발하게 진행되고 있으며, 다양한 산업군에서도 NB-IoT 기술을 활용하고 있다. NB-IoT에 대한 연구와 산업화가 진행되면서 이에 대한 안전성 문제도 제기되고 있다. 본 논문에서는 NB-IoT 기술이 적용되고 있는 사례들을 비교 분석하여 보안 위협을 도출하고 이에 대한 보안 요구사항을 도출하여 NB-IoT 기술의 안전성 향상을 위한 방안에 대해 제안한다.

I. 서론

센서 데이터를 전송하기에 적합하고 광범위 저전력을 지원하는 NB-IoT는 다양한 산업군에서 산업 인프라를 구성하는데 사용되고 있으며, 이를 이용한 다양한 적용사례들이 등장하고 있다[1]. 기술의 발전과 산업화가 진행됨에 따라, NB-IoT 기술에 대한 안전성 문제도 제기되고 있다[2].

본 논문에서는 LPWA의 대표 기술인 NB-IoT의 적용사례에 대한 보안 위협 분석을 진행하고 그에 대한 보안 요구사항을 도출한다.

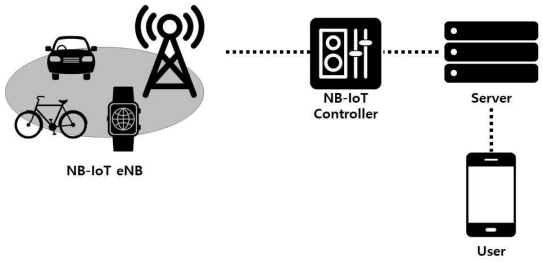
본 논문의 구성은 다음과 같다. 2 장에서 관련 연구를 분석하고, 3 장에서는 NB-IoT 기술 적용사례에 대한 보안 위협을 분석한다. 4 장에서는 분석된 보안 위협을 기반으로 보안 요구사항을 도출하고 5 장에서 결론을 맺는다.

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 대학ICT연구센터육성 지원사업의 연구결과로 수행되었음(IITP-2017-2015-0-00403)

II. 관련 연구

2.1 NB-IoT

LTE Cat NB1이라고도 불리는 NB-IoT는 저전력 광범위 기술 LoRaWAN과 함께 LPWA의 대표적인 기술로 주목받고 있다. LoRaWAN이 비면허 대역을 사용하는 반면, NB-IoT는 면허 대역을 사용하고 언제 어디서나 가상으로 동작한다. 기존 무선 표준은 전력 절감 기능을 지원하지 않기 때문에 NB-IoT는 고가의 디바이스보다 저가의 디바이스에 적합하다. NB-IoT는 기존 모바일 네트워크 상에서 보다 간편하고 효율적으로 연결할 수 있으며, 적은 양의 데이터를 안전하고 안정적으로 처리할 수 있다[2]. 특히 신호가 잘 전달되지 않는 건물이나 지하 등에서도 우수한 성능을 지원하기 때문에 다양한 산업군에서 활용되고 있다[1]. [그림 1]은 NB-IoT를 활용한 산업화 환경의 대표적인 예시를 보여준다.



[그림 1] NB-IoT를 활용한 산업화 환경

2.2 Smart Parking

주차 문제를 해결하기 위해 HUAWEI는 Smart Parking을 개발하여 시민들에게 실시간으로 주차 정보를 제공하여 보다 나은 주차 관리를 가능하게 지원한다. Smart Parking은 IoT와 모바일 지불 기술을 활용하여 목적지 근처에 주차장소를 사전에 찾아 예약할 수 있게 지원한다. NB-IoT 기술을 활용하여 IoT 네트워크 환경을 구축하였고, 이를 통해 주차장 내에 주차 가능 공간을 신속하게 사용자들에게 제공한다[1].

2.3 Pet Tracking

Pet Tracking은 사용자가 애완동물의 활동 및 위치를 추적할 수 있게 지원해주는 응용 프로그램이다. NB-IoT 칩셋이 내장된 목걸이를 애완동물에게 부착하여 사용자에게 정보를 제공한다. 해당 NB-IoT 칩셋은 GPS 및 위치 기반 서비스를 활용하는 정보를 수집하고 전송하며, 사용자의 선호도에 따라 일정 주기 혹은 실시간으로 정보 수집 및 전송을 수행한다[1].

2.4 Smart Bicycle

네덜란드에 있는 자전거 대여 회사는 자전거 프레임에 M2M SIM 카드를 내장하여, 대여 중 도난당한 자전거를 회수하는데 큰 도움을 받고 있다. 자전가가 반납되지 않을 경우, 자전거 내에 부착되어 있는 SIM 카드를 통해 도난 자전거를 쉽고 빠르게 찾을 수 있다[1].

2.5 STRIDE 모델링

STRIDE 모델링은 Microsoft 사에서 보안 위협을 분류할 수 있게 개발한 기법이다. STRIDE 모델링은 S(Spoofing Identity-ID 스푸핑), T(Tampering with data-데이터 위·변조), R(Repudiation-부인), I(Information Disclosure-정보 노출), D(Denial of Service-DoS), E(Elevation of Privilege-권한 상승)로 구성되어 있다[3].

2.6 OWASP

OWASP(Open Web Application Security Project)는 국제 웹 보안 표준 기구이며 정보 노출, 악성 코드, 취약점 진단 등 다양한 정보보호를 연구하는 단체이다. OWASP IoT TOP 10은 IoT에서 가장 발생 빈도가 높고, 보안상 영향을 많이 줄 수 있는 상위 10개를 정리하여 발표한 문서이다. OWASP IoT TOP 10 중 상위 5 가지 항목으로는 IWI(Insecure Web Interface-안전하지 않은 웹 인터페이스), IA(Insufficient Authentication/Authorization-미흡한 인증 및 인가 검증), INS(Insecure Network Services-안전하지 않은 네트워크 서비스), LTE(Lack of Transport Encryption-미흡한 전송 암호화), PC(Privacy Concerns-개인정보보호)가 있다[4].

III. 보안 위협 분석

3.1 STRIDE 모델링 기반 보안 위협

NB-IoT를 적용한 사례를 분석하고, 이에 대한 보안 위협을 STRIDE 모델링 기반으로 분석하였다. 적용 사례 모두 센서 디바이스를 기반으로 사용자에게 필요한 정보를 제공해준다는 공통점을 갖고 있었으며, 이를 통해 공통적인 보안 위협을 분석하였다.

[표 1] STRIDE 기반 NB-IoT 보안 위협

위 협	내 용
S	- 공격자는 정상적인 네트워크로 위장하여 센서 디바이스에 접근할 수 있다.

T	- 공격자는 센서 디바이스에 접근하여 센서에서 발생하는 정보를 위·변조할 수 있다.
R	- 공격자는 센서 디바이스의 위·변조 사실을 부인할 수 있다.
I	- 공격자는 센서 디바이스의 정보를 탈취하여 개인정보와 같은 민감 정보를 유출시킬 수 있다.
D	- 공격자는 다수의 센서 디바이스 또는 네트워크 서비스를 사용하지 못하도록 설정하여 서비스 제공 업체를 위협할 수 있다.
E	- 공격자는 센서 디바이스에 접근하여 권한을 상승하고, 이를 악용할 수 있다.

3.2 OWASP 기반 보안 위협

OWASP IoT TOP 10 중 상위 5 가지 항목을 기반으로 NB-IoT 적용사례에 대한 보안위협 분석을 진행한다.

[표 2] OWASP 기반 NB-IoT 보안 위협

위 협	내 용
IWI	- 공격자는 취약한 자격 증명을 사용하거나 일반 텍스트 자격 증명을 캡쳐하여 계정을 탈취하고 웹 인터페이스에 접근할 수 있다.
IA	- 인증 권한 부여가 충분하지 않으면 데이터 손실, 정상 사용자의 접근 거부 등이 발생할 수 있으며 디바이스 또는 사용자 계정이 손상될 수 있다.
INS	- 안전하지 않은 네트워크 서비스는 버퍼 오버 플로우 공격이나 서비스 거부 공격에 취약하여 사용자가 디바이스에 접근할 수 없게 만든다. 안전하지 않은 네트워크 서비스를 사용하게 될 경우, 다른 서비스 이용자를 대상으로 진행되는 서비스 거부 공격에 악용될 수 있다.
LTE	- 암호화 없이 전송하게 되면 로컬 네트워크나 인터넷을 통해 데이터가 유출될 수 있다.
PC	- 개인정보가 포함된 데이터를 안전하게 보호하지 않아서 발생되는 문제이다.

IV. 보안 요구사항

3장에서 분석된 보안 위협에 대응하는 NB-IoT 보안 요구사항을 도출한다. NB-IoT는 저전력 광역 IoT 네트워크를 지향하기 때문에 보안 요구사항이 서비스 가용성에 영향을 미쳐서는 안된다. 그러므로 기존 IT 환경에서 사용되던 대응방안보다 경량화 된 대응방안을 중심으로 도출하였다.

4.1 STRIDE 모델링 기반 보안 요구사항

❑ Spoofing Identity

- 디바이스 접근 시 패스워드, 생체 인식 등의 인증 기술을 제공해야 하며, 경량암호 알고리즘을 사용하여 전력 소비와 구현 면적을 최소화하고 고속의 암·복호화 구현을 제공해야 한다.

❑ Tampering with data

- 위·변조에 대응이 필요한 값에 해시함수를 적용하는 등 해당 정보에 대한 무결성을 제공해야한다.

❑ Repudiation

- 센서 디바이스들의 정보를 수집하는 디바이스에 감사로그를 저장하여 악의적 행위에 대한 부인을 방지할 수 있어야 한다.

❑ Information Disclosure

- 개인정보 등과 같이 유출에 민감한 값을 경량 암호 알고리즘을 이용하여 암호화하거나 비식별화 조치를 진행하여 데이터에 대한 기밀성을 제공해야 한다.

❑ Denial of Service
- 해당 시스템에 처리 가능한 요청 수를 제한하는 Throttling을 적용하여 서비스 가용성이 침해되지 않도록 서비스를 제공해야 한다.

❑ Elevation of Privilege
- 각 사용자에게 필요한 최소한의 권한을 제공하고, 원격 업데이트를 통해 권한 상승에 이용될 수 있는 보안 위협에 대응할 수 있어야 한다.

4.2 OWASP 기반 보안 요구사항

❑ Insecure Web Interface
- 강한 암호 복구 알고리즘을 사용해야 하고, 유효한 계정에 대한 정보를 공격자에게 제공되지 않도록 관리해야 한다.
- 3~5회의 로그인 시도 실패 후 계정 잠금 등을 통해 접근 제어 관리해야 한다.
- 내부 및 외부 네트워크 트래픽에 자격 증명이 노출되지 않도록 관리해야 한다.

❑ Insufficient Authentication/Authorization
- 권한을 다양하게 나누어 제공하고, 접근 제어에 대한 세분화 작업을 진행한다.
- 두 가지 이상의 인증 요소를 구현해야한다.
- 민감한 기능에 대한 재인증 기능을 제공해야 한다.

❑ Insecure Network Services
- 필요한 포트만을 노출하고 있는지 확인한다.
- 서비스가 버퍼 오버 플로우 및 퍼징 공격에 취약하지 않도록 관리한다.
- 비정상적인 서비스 요청 트래픽은 서비스 게이트웨이 계층에서 탐지되고 차단되어야 한다.

❑ Lack of Transport Encryption
- 업계 표준 암호화 기술을 사용하여 전송 중 데이터를 보호해야 한다.
- 안전한 암호화 키 교환을 보장해야 한다.
- 수신된 데이터에 대한 무결성 확인을 보장해야 한다.

❑ Privacy Concerns
- 수집된 모든 데이터가 확인되지 않거나 익명으로 처리되도록 보장해야 한다.
- 인가된 개인만이 수집된 개인정보에 접근할 수 있도록 관리해야 한다.
- 수집된 데이터에 대한 저장 기간을 설정해야 한다.

V. 결론

본 논문에서는 LPWA 네트워크의 대표 기술 중 하나인 NB-IoT를 분석하고, NB-IoT를 적용한 사례인 Smart Parking, Pet Tracking, Smart Bicycle을 대상으로 STRIDE 모델링과 OWASP 기반의 보안 위협 분석을 진행하였다.

3 가지 적용사례 모두 NB-IoT를 적용한 사례이기 때문에 보안 위협의 결과 유형이 비슷하게 도출되었다. 그러므로 3 가지 사례에 대한 대응 방안을 NB-IoT 전체에 대한 보안 요구사항으로 도출하였다. NB-IoT는 저전력 광역을 지원하는 네트워크 기술이기 때문에 기존 IT 환경에서 사용되던 대응방안보다 경량화 된 대응방안을 기반으로 보안 요구사항을 도출하였다.

향후 본 논문에서 분석한 보안 위협 이외에 추가적인 보안 위협을 도출하고, 그에 대한 보안 요구사항에 대한 연구가 필요하다.

[참고문헌]

[1] HUAWEI, “NB-IOT : Enabling New Business Opportunities”, HUAWEI White Paper, December 2016.

[2] Rashmi Sharan Sinha, Yiqiao Wei. Seung-HoonHwang, “A survey on LPWA technology: LoRa and NB-IoT”, The Korean Institute of Communications and Information Sciences ICT Express 3, pp.14-21, March, 2017.

[3] Bruce Potter, “Microsoft SDL Threat Modelling Tool”, Network Security, pp.15-18, January, 2009.

[4] Daniel Miessler, Craig Smith, “OWASP Internet of Things”, OWASP Internet of Things Project, OWASP, May, 2014.

[5] J. Schlienz, D. Raddino, “Narrowband Internet of Things Whitepaper”, ROHDE & SCHWARZ, August, 2016.

무선 LAN 취약점에 따른 IoT 보안*

김지원, 최효경, 김지수, 이세은, 최나래, 최은정
서울여자대학교 정보보호학과

IoT Security According to Wireless LAN Vulnerability

Ji-Won Kim**, Hyo-Kyung Choi**, Ji-Su Kim, Se-Eun Lee, Na-Rae Choi, Eun-Jung Choi**
*Division of information security, Seoul Women's University.

요 약

최근 사물인터넷(이하 IoT) 기기 사용이 급증하면서 IoT 환경의 보안 위험성이 높아지고, 그에 따른 피해가 급증하고 있다. IoT 기기 서비스는 유무선 네트워크로 연결된 망에서 통신을 하므로 통신 과정에서 해킹 등의 공격이 발생할 수 있는데 특히 무선 네트워크의 쉬운 접근성으로 인해 무선 LAN 기반 IoT 환경에서의 취약점 대응이 점점 중요해지고 있다. 따라서 본 논문에서는 IoT와 무선 LAN 개념을 토대로 무선 LAN 환경에서의 IoT 기기 취약점 분석 실습을 통해 취약점 대응방안을 모색해 본다.

Key Words : IoT, IoT Device Vulnerability, Wireless LAN, Wireless Network

I. 서론

4차 산업혁명 시대라는 말이 등장하면서 정보통신 기술은 단순한 학문에서 벗어나 시대의 흐름을 주도하는 개념으로 자리 잡았다. 이에 따라 최근의 ICT(Information and Communication Technologies) 환경은 비약적으로 발전하면서 국가의 혁신적인 성장의 지표가 되었고, 그 중 일상생활에서 사물과 인터넷을 연결해 다양한 서비스를 제공하는 IoT는 기술과 효용성을 인정받아 4차 산업혁명 시대의 키워드로 떠올랐다.

IoT는 제조, 에너지, 금융 등 산업 전반에 걸쳐 다양하게 활용되고 있으며 초연결사회로 나아가기 위한 수단으로도 주목받고 있다. [1] 또한 일반인에게도 보편적으로 IoT 기기가 보급되면서 그 파급력과 활용사례의 범위가 점차 넓어지고 있다.

IoT 서비스는 네트워크로 연결되어 통신을 하므로 각 객체 간의 상호 정보를 자유롭게 교환할 수 있는 환경 구축이 가능한 반면, 이러한 특징으로 인해 통신 과정에서 발생하는 여러 위협 요인이 존재한다. 이로 인해 정보 유출, 데이터 위변조, 서비스 거부, 사생활 침해 등의 보안 문제가 대두되고 있다.[2] 특히 IoT가 주로 통신하는 무선 LAN 환경은 유선 네트워크와는 달리 접근성이 뛰어나 쉽게 해킹이 가능하고 다양한 공격의 우회로로 사용되기도 한다.

따라서 본 논문에서는 IoT와 무선 LAN에 대한 개념을 바탕으로 무선 LAN 기반 IoT 환경에서의 취약점 분석을 실습한다. 본 연구를 통해 IoT 기기 사용 시 위협이 되는 무선 LAN 취약점에 대한 심각성을 적극 파악하여 이에

* 본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 SW중심대학지원사업의 연구결과로 수행되었음(2016-0-00022)

대한 대응방안과 기대효과를 제시한다.

II. 사물인터넷(IoT)과 WLAN

2.1 IoT와 WLAN

IoT(IoT: Internet of Things)는 사람, 사물, 공간 등 모든 것들이 인터넷으로 서로 연결되어 그에 대한 정보가 생성, 수집되고 공유 활용되는 것(정보통신정책 연구원), 기존의 유선통신 및 모바일기반 인터넷보다 진화된 다음 단계의 인터넷(한국인터넷진흥원) 등으로 다양하게 정의된다.

IoT 기기의 네트워크는 무선 디바이스 네트워크를 중심으로 발전해왔으며 대표적인 무선 디바이스 네트워크 기술은 무선 근거리 통신망(Wireless Local Area Network) 전송 규격을 위한 IEEE 802.11 표준을 기반으로 한다. 이를 통해 여러 IoT 기기간의 정보 교환, 연동이 이루어진다.[3]

IEEE 802.11 기반의 무선 LAN 서비스 구조는 크게 Ad-hoc 모드와 Infrastructure모드로 구분된다. Ad-hoc 모드는 각각의 단말이 일대일 통신을 수행하는 구조이고, Infrastructure 모드는 각각의 무선 단말이 AP를 통해서 네트워크 환경으로 연결되는 구조이다. IoT 기기는 Infrastructure모드로 연결되는데 이는 AP / 게이트웨이에 항상 접속해있기 때문에 사용자의 스마트폰을 통한 IoT 기기 제어가 가능함을 의미한다.[4] 따라서, 다이렉트 접속 시 Wi-Fi 사용이 중단되는 단점을 보완할 수 있다. 하지만 IoT 기기가 항상 AP 에 접속이 되어 있기 때문에 전류와 자원 소모의 오버헤드가 발생한다는 문제가 발생한다. 지속적인 상태 확인이 필요 없고 일회성 제어가 필요한 IoT 기기의 경우에도 지속적인 접속 유지를 해야 하는 단점이 있다.[5]

IoT 기기가 Wi-Fi에 접속하는 과정은 우선 스마트폰에서 AP를 스캔하여 접속할 AP를 정하고 인증을 거친 뒤 AP와 연결된다. 그 후 IoT 기기를 검색, 연결하여 AP와 IoT 기기 사이의 연결을 완료한다.

2.2 무선 LAN 위협 요인

무선 LAN 기술이 등장하면서 기존 유선의 통신에서 벗어나 고속의 편리한 통신이 가능해졌지만 보안의 위협 요인도 대두되고 있다.

무선 LAN 서비스 활용 형태에 따라 구성요소의 차이는 있지만 대부분 공통적인 보안 위협 요소 및 보안 취약점을 가지고 있다.[4] 주요 요인은 다음과 같다.

비인가 사용자 접근(Unauthorized Access)은 모든 단말에 데이터를 전송하는 브로드캐스팅 망인 무선 LAN 시스템에서 발생하는 중대한 위협 요인이다. AP 수신 영역 내에 있는 모든 단말은 해당 AP로 접속을 시도할 수 있고, 사용자 인증 등에 관한 보안 기능을 수행하지 못하는 LAN 시스템이라면 비인가 사용자의 접근이 가능하기 때문이다.[4]

중간자 공격(Man In The Middle attack, MITM)은 공격자가 통신장비처럼 클라이언트와 웹 서버와의 경로 상에 들어가게 되면 모든 내용을 감청 및 위변조가 가능하게 되는 공격을 의미한다. [6] 예를 들어, TCP/IP 프로토콜 취약점을 공격할 경우, 중간자 공격으로 제 3자가 IoT 기기의 네트워크로 침투하여 트래픽을 가로채 결국 기기에 접근 할 수 있는 것이다.[7]

2.3 IoT 기기 해킹 사례

무선 LAN 위협 요인으로 발생 가능한 공격 중 분산 서비스 거부(Distributed Denial of Service, DDoS) 공격이 있다. DDoS 공격은 일반 사용자들의 PC에 악성코드를 심어 서버에 대량 트래픽을 송신하는 공격방법으로, 공격자가 공격자 스스로의 외부노출을 피하면서도 감염자의 pc에 대량의 데이터를 보낼 수 있다.[8]

예로, 작년 10월 미 동부 지역 마비 사태를 일으키며 많은 사람들에게 IoT 보안에 대한 경각심을 불러일으킨 '미라이 봇넷' 공격이 있다. 다수의 IoT 기기를 악성 코드 '미라이'에 감염시켜 DDoS 공격용 봇(Bot) 역할을 하게 만들고, 미국 주요 인터넷 호스팅 서비스 업체인 '딘

(Dyn)' 서버를 공격함으로써, '딘'을 통해 DNS 서비스를 받고 있던 사이트에 연쇄적인 장애를 발생시켰다. [9]

III. Air-Monitor 실습

3.1 시나리오

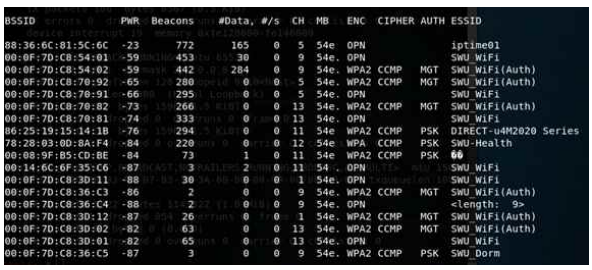
Wi-Fi에서 인증과정을 수행할 때 handshake를 이용하여 패스워드를 공유하는 protocol을 사용한다면, 도청 후 복호화 하여 crack할 수 있다. 이에 공격자는 crack하고자 Wi-Fi에서 어떤 암호화 protocol로 인증을 하는지 알아내, 해당 protocol에서 사용하는 암호화 알고리즘을 파악하여 protocol 정보에 따른 취약점 공격 또는 알고리즘에 따른 데이터 복호화를 시도한다.

만약 crack하고자 하는 IoT 기기와 공격자가 동일한 무선 망 안에서 동작한다면, 공격자는 해당 LAN내에서 도청하며, 다른 외부 망에서 동작한다면 default gateway를 통해 도청을 한다. 또한 인증 방식이 없다면 공격자가 패킷을 위조할 경우 LAN입장에서 연결 대상이 공격자인지 일반 사용자인지 판단할 수 없다.

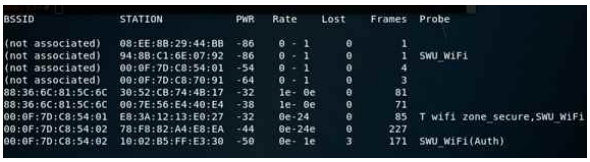
따라서 이후 절에서는 IoT 기기가 연결되는 해당 Wi-Fi에 보편적인 암호화 인증방식이 사용된다는 가정을 하고 그 암호화 방식을 알아내는 실습을 진행한다.

3.2 실습

Aircrack-ng suit이 자동으로 설치되어 있는 칼리리눅스 환경에서 진행한다. VM ware에서 작동할 경우 USB 무선랜카드를 사용한다.



airmon-ng를 통해 Wi-Fi 어댑터를 모니터 모드를 실행하여 공중에 떠다니는 무선 트래픽을 훔쳐본다. 이 전에 wlan을 up(활성화) 시킨 후 airmon-ng를 통해 가동시킨다. airodump-ng를 통해 연결 가능한 네트워크 목록을 확인한다. 목록 중 상위 부분은 연결 가능한 AP이며 아래 부분은 Client 목록이다.



특정 Wi-Fi의 암호화 방식을 알고 싶다면 ENC 부분을 확인한다. 저자의 실습 장소에서 연결 가능한 네트워크들은 대부분 WPA2 암호화 방식을 사용하고 있다. (OPN은 암호화 인증을 사용하지 않는다는 의미이다.)

3.3 분석

리눅스 운영체제의 Air-Crack 서비스를 사용하면 공중에 떠다니는 트래픽을 캐치할 수 있고, 캐치 된 정보 안에는 BSSID, ENC 등 Wi-Fi에서 주요한 정보들이 담겨져 있다. (Air-Crack의 원리는 모든 패킷 간의 통신에 대한 주파수 분석으로 서비스에서 캐치한 트래픽 정보를 운영체제 모니터에 출력해주는 것이다.)

이 중 IP와 MAC 정보를 얻게 되면 주소를 우회하여 IoT 기기에 접근할 수 있고, 암호화 시스템 정보를 얻게 되면 IoT 기기에서 사용하는 AP에 대해 어떤 방식으로 공격하거나 도청한 데이터를 복호화 해야 할지 파악할 수 있다.

위의 실습의 경우 Wi-Fi 암호화 기술은 WPA2였다. WPA2의 경우 자체 취약점이 많이 드러나지 않았기에 실제 WPA2 암호 키를 회수하고 크랙하는 것 보다는 도청으로 가로채는 것이 보편적이다. 따라서 위의 Wi-Fi를 사용하는 IoT 기기를 공격할 때 가장 먼저 공격자는 Spoofing 후 부차적으로 암호화 protocol의 취약점 공격을 시도할 것임을 유추할 수 있다.

만약 암호화 인증이 없는 AP의 경우 우회를 통해 같은 망, 같은 LAN 안에 있을 시 사용자와 IoT 기기간의 통신을 패킷만으로 관찰할 수 있다. 하지만 위와 같이 암호화 인증을 사용하는 경우 데이터 원문 확인을 위해선 이를 복호

화 하는 과정까지 진행해야 한다.

결론적으로 위의 실습에서는 IoT기기나 사용자와 통신하지 않는 공격자가 암호화 protocol을 공기 중 트래픽만으로 알아낼 수 있음을 확인했다. 이를 통해 공격자는 데이터 복호화를 진행할 수 있다. 그렇게 되면 무선 LAN의 고질적인 문제점인 비인가 사용자 접근은 해결되지 않는다. 무선 LAN의 비인가 사용자 접근이 개인이나 가족 단위의 IoT 기기에서 일어날 경우, 프라이버시 침해 및 개인정보 유출 등 심각한 보안 문제로 이어지게 된다. 이 상황에서는 결국 무선 LAN 상에서의 모든 통신이 도청으로부터 자유로울 수 없다.

IV. 대응방안 및 기대효과

IoT 기기 해킹에 대한 가장 적극적인 대응방안은 정부차원의 제도적 지원이지만, 실현이 가능한 기술적 해결책도 필요한 실정이다.

AP에서는 인증을 시도할 때 따로 구축된 인증 서버를 통해 서비스한다. 따라서 대부분의 무선망이 이미 구축되어 있는 서버를 통해 통신하고, 이는 무선망을 사용하는 IoT 기기 또한 동일하다. 이러한 서비스에서의 문제점을 앞의 절에서 확인하였다. 만약 인증 서버와 공유기가 결합되어 자체적으로 인증을 수행하고, 특정 거리 이내에서만 통신이 가능한 홈 IoT 기기 전용 공유기가 개발된다면, 가정, 사무실 등에서 사용되는 단거리 IoT 기기에 한해 취약점 발생 가능성이 현저히 낮아지게 된다. 이에 따라 본 논문에서는 분석 취약점에 따른 해결방안으로 홈 IoT 기기 전용 공유기 개발을 제안한다.

포함되어야 하는 기능은 다음과 같다. 먼저, AP가 인증을 자체적으로 수행하도록 한다. 공유기 내에 서버와 클라이언트가 동시에 구축되면 인증을 시도할 때 멀리 있는 서버를 통해 서비스를 하지 않아도 되고, 나아가 통신 가능 거리를 제한할 경우 Air-Crack 공격으로 패킷이 노출되는 위험을 줄일 수 있다.

다음으로 새로운 암호화 기법이 들어간 공유기를 개발하거나, Air-Crack 공격을 통해 파악이 가능한 Wi-Fi 정보에 암호화 기법이 나타나지 않도록 하는 서비스를 구축한다. 또한 네트워크 공개키 암호화가 없이도 인증이 가능한 자체 IoT 암호화 서비스를 구현하게 된다면 도청을 근본적으로 막을 수 있다.

위와 같은 기능이 포함된 공유기 개발을 해결책으로 제시함에 따라 실질적으로 서비스가 구현된다면 IoT 기기의 Wi-Fi 인증과정을 사용자가 원하는 거리에서만 진행하여 무선 LAN에서 Wi-Fi 정보가 탈취되는 가능성을 낮출 수 있다.

이 해결책에 따른 기대효과는 다음과 같다. 첫째, 개인이나 가정에서 사용되는 단거리, 단일 IoT 기기의 개인 정보 유출을 예방하여 보안 위협으로부터 벗어난다. 둘째, 본질적인 IoT 기기의 해킹 위험을 낮출 수 있다.

[참고문헌]

[1] IoT(Internet of Things) 산업 동향, 보안연구부-2016-053, 금융보안원, (2016.10.20.)
[2] 장봉임, IoT 보안 기술 연구, 보안공학연구논문지, (2014.10.04.)
[3] 최영규, 사물인터넷(IoT)네트워크 환경에서 OAuth기반 사용자 인증 기법, p.13
[4] 강유성, 무선 LAN 보안 취약점과 단계적 해결 방안, 한국전자통신연구원, p.3
[5] 이상곤, WiFi 블루투스의 비콘을 이용한 비접속 무선 IoT 네트워크 제어 시스템, p.13-14
[6] 조성민;이훈, 중간자공격을 이용한 보안서버 무효화공격에 대한 대응법, 한국정보통신학회(2016), p95
[7] 길민권, 사이버 공격의 새로운 영토, 사물인터넷(IoT), 데일리시큐, 2016.08.16, http://www.dailysecu.com/news/articleView.html?idxno=15311
[8] 홍성혁, DDoS 공격에 대한 분석 및 대응방안, 한국디지털정책학회(2014), p425
[9] 김미희, IoT 보안 위협에 따른 대응 방안, it daily, 2017.06.01, http://www.itdaily.kr/news/articleView.html?idxno=83499

IoT 보안 최신 동향 분석*

김지원, 최효경, 안혜준, 최은정
서울여자대학교 정보보호학과

Trend Analysis of IoT Security

Ji-Won Kim**, Hyo-Kyung Choi**, Hye-Jun An**, Eun-Jung Choi**
*Division of information security, Seoul Women's University.

요 약

ICT 분야의 융합 기술의 중요성이 대두되면서 최근 사물과 인터넷을 연결시킨 IoT 기술 환경은 급성장하고 그 가치가 높아지고 있다. 하지만 기술의 발전은 다시 보안 위협의 증가로 이어진다. 따라서 본 논문에서는 IoT 최신 보안 동향을 파악하여 IoT 분야에서의 보안 중요성을 재기하고 보안 위협에 따른 대책을 적극 제시한다.

Key Words: IoT, IoT Security, Trend Analysis

I. 서론

4차 산업혁명의 시대로, 융합 기술의 중요성이 대두되고 있다. 사물인터넷(IoT)도 융합 기술 중의 하나로 그 가치가 높아지고 있다. 하지만 그와 비례하게 보안 위협 또한 증가하고 있다. 날이 갈수록 발전하는 기술과 함께 현재의 IoT 보안 기술과 동향은 어느 시점에 있는지 파악해 그 기술을 이해할 필요성이 존재한다. 따라서 본 논문에서는 IoT 보안에 대한 개념과 IoT 보안 위협 동향, IoT 보안 기술에 대한 동향을 분석하고, 그에 대한 대책을 제시한다.

II. 사물인터넷(Internet of Things)

IoT는 인터넷을 기반으로 모든 사물을 연결하여 사람과 사물, 사물과 사물 간의 정보를 상호 소통하는 지능형 기술 및 서비스를 말한다. 한국인터넷진흥원(2012)은 IoT 기술은 초연결사회의 기반 기술로서 사물 간 인터넷 혹은 개체 간 인터넷으로 고유 식별이 가능한 사물이 만들어낸 정보를 인터넷을 통해 공유하는 환경이라고 정의하였고, ITU에서는 IoT는 '객체의 제약'을 해결하는 것이 핵심이라고 말했다.[1]

IoT 시장은 급격하게 발전하고 있고, 표1에서 보는 것처럼 2025년까지 글로벌 시장에 IoT 설치를 기반으로 한 기기 시장이 약 750억 까지 증가할 것으로 예상한다. 앞으로도 IoT 시장은 다양하고 거대해질 것이라고 다수의 전문가들이 예측한다.

* 본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 SW중심대학지원사업의 연구결과로 수행되었음(2016-0-00022)

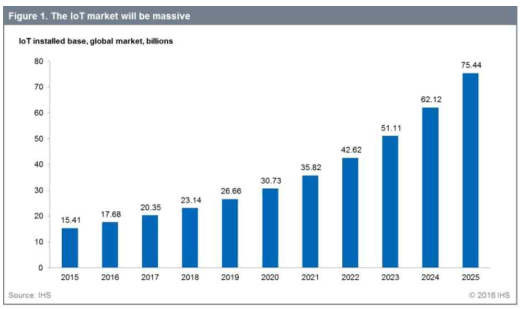


표1. IoT시장 성장 예상 측도(출처:IHS)

IoT의 네트워크는 무선 디바이스 네트워크를 중심으로 발전해왔으며 무선 디바이스 네트워크 기술은 다양한 기기와의 정보 교환, 연동을 위해 무선 근거리 개인 통신망(WPAN) 전송 규격을 위한 IEEE 802.15.4 표준과 이를 기반으로 산업에 적용하려는 다양한 기술(Wi-Fi, ZigBee 등)이 있다.[4] 그 중에서도 Wi-Fi가 스마트 폰이 IoT 기기와 접속하는데 대표적으로 많이 사용되는 기술이다.

III. IoT 보안 동향

3.1

인간을 둘러싼 사물들이 서로 연결되어 커뮤니케이션하는 환경이 되면서 개인 정보는 많아지고, 유출의 위험성은 더욱 커졌다. 이러면서 IoT 보안이라는 용어가 나왔는데 IoT보안은 IoT의 성장과 동시에 보안을 강화해야 한다는 의미에서 나타난 용어이다. 이제는 기업에서 최초 제품 기획 단계부터 보안을 최우선으로 고려하는 전환 인식이 필요하고, 사용자도 개인 보안에 위협을 받지 않도록 네트워크에 연결되는 기기를 선택할 때 기능뿐 아니라 보안 능력을 우선 요소로 놓아야 하는 시대가 온 것이다.[2]

3.2 IoT 보안 위협 동향

IoT로 인해 모든 사물이 해킹 대상이 될 수 있기 때문에 IoT 보안 위협이 대두가 되고 있다. IoT 적용에 따른 보안위협을 크게 센서/디바이스, 네트워크, 플랫폼/서비스로 나눠서 살펴보자.

먼저 센서/디바이스 분야이다. 디바이스들이 다양화되고, 종류가 늘어나면서 저 사양 기기 사용이 늘어나고, 현재 보안 기술로는 저 사양 기기에 백신, 암호화, 인증 등 보안을 적용하기 곤란한 경우가 많고, 디바이스 수가 많아지면서 보안패치 적용 곤란, 통신 내용 모니터링 곤란에 따른 보안 취약성이 증가하게 된다.[7]

다음으로 네트워크 분야이다. ZigBee, WiFi, Bluetooth 등 이종무선 네트워크 간 상호연동이 되면서, 일정한 보안수준을 유지하기 어렵고, 디바이스간 통신이 지원되면서 디바이스 인증이 제한적으로 지원된다. 또한 클라우드 가상화 서비스를 통한 좀비 PC 대량 생산, 냉장고, 청소 로봇, 의료기기 등 대규모 디바이스에 악성코드를 감염시켜 트래픽 폭증 공격이 가능해졌다.[7]

마지막으로 플랫폼/서비스 분야이다. 공개 플랫폼을 통한 기기-서비스 간 허위 데이터 전송/오작동 등의 공격이 일어날 수 있고, IoT 디바이스가 수집한 단편 정보의 중앙 집중 및 조합으로 사용자 신원정보가 유출될 수 있는 위험성이 있다.[7]

3.3 디바이스 보안

IoT 보안 위협들이 발생함에 따라 이를 보안하기 위한 여러 가지 분야의 보안 기술 또한 존재하며 개발 중이다. IoT 보안 기술은 IoT 디바이스와 센싱된 데이터에 대한 통신/네트워킹 분야, 센싱 데이터 저장/가공/활용 분야, 서비스 분야로 나눠볼 수 있다.

첫 번째로 IoT 디바이스 보안이다. 악의적 역할을 수행하는 디바이스로 인해 오동작 혹은 의도하지 않은 기능이 수행되는 것을 방지하기 위해 디바이스에 대한 인증 및 식별이 반드시 필요하다. IoT 디바이스에 대한 인증 및 식별을 위해 사용될 수 있는 기술로서는 인증서를 사용하는 방법, ID/패스워드를 이용하는 방법, 권

한을 가지고 토큰을 사용하는 방법, SIM 같은 식별 장치를 사용하는 방법, 유일한 식별자를 사용하는 방법 등 다양한 방법이 있다. 디바이스 경량 암호/보안 프로토콜 기술로는 암/복호화 및 해시값, MIC 값 등을 생성할 수 있는 암호 알고리즘을 제공하는 게 있는데 다양한 암호 알고리즘을 제공하기 위해 PRESENT나 KATAN 및 KTANTAN, HummingBird, HIGHT, LEA 경량 암호 알고리즘과 AES와 같은 국제표준 암호 알고리즘과 단체표준 암호 알고리즘을 사용할 수 있다.[3]

3.4 통신/네트워킹 보안

두 번째로 통신/네트워킹 보안이다. 이 보안 기술로는 IoT 서비스 구성 디바이스 간 통신을 위한 전용 프로토콜인 CoAP와 클라이언트와 서버 간의 상호 인증을 기반으로 서비스가 제공되는 LwM2M, 전송 계층 프로토콜인 MQTT가 있다.[4]

3.5 데이터 보안

세 번째는 데이터 저장/가공/활용 시 보안이다. 데이터를 처리할 때 나타날 수 있는 프라이버시 침해를 막을 수 있는 기술로 익명성/필명 보안이 있다. 데이터를 제공하는 응답자에게 익명성을 제공하거나 필명을 사용함으로써 일부를 해결하는 기술이다. IoT 서비스에서 센싱된 정보는 서비스 제공을 위해 클라우드에 저장되는데 이 정보를 안전하게 가지고 있기 위해 데이터베이스 암호화 기법 등을 이용해 보호한다. 또한 데이터베이스에 대한 다양한 접근 제어기법을 통해 DB보안을 실현할 수 있다. 최근에는 데이터를 암호화하여 저장할 경우, 성능 지연 문제가 발생하므로 이러한 문제를 해결하기 위해 암호화된 자료를 복호화지 않은 상태에서도 원하는 자료를 검색할 수 있게 해주는 검색 가능 암호 기술(Searchable Encryption Technique)에 대한 연구도 진행되고 있다.

3.6 서비스 보안

네 번째는 서비스 보안이다. 서비스 영역에서의 대표적 보안 이슈로는 서비스에 대한 접근 제어가 있다. 일반적 환경에서 접근 제어를 위해 주로 사용되는 기법으로 접근 제어 주체와 방법에 따라 임의적 접근 제어(Discretionary Access Control), 강제적 접근 제어(Mandatory Access Control), 역할 기반 접근 제어(Role-Based Access Control), 속성 기반 접근 제어(Attribute-Based Access Control) 기법으로 나뉜다. 임의적 접근 제어 기법은 IoT 서비스의 유연성에는 용이하지만, 서비스 제공자의 실수 및 고의로 인한 보안 문제가 생길 수 있다는 단점이 있다. 강제적 접근 제어 기법은 이러한 임의적 접근 제어 기법의 보안 문제를 보완하여 나온 것으로, 서비스 제공자 대신에 공통 플랫폼을 통해 IoT 관리자와 같은 최고권한자가 서비스 사용자에 대한 권한을 설정하는 방식이다. 역할 기반 접근 제어는 비임의적 접근 제어(Non DAC)로 분류되며, 서비스 사용자에게 각각 역할을 부여하고 특정 역할마다 서비스에 대한 접근 권한을 부여할 수 있는 방식으로, 접근 제어 작업을 단순화할 수 있다는 장점이 있다. 속성 기반 접근 제어 기법은 사용자의 인증을 수행하고, 사용자의 권한이 아닌 속성에 따라 접근 제어를 수행하는 기법이다.[5]

3.7 시스템 보안

마지막으로 시스템 보안이다. IoT 디바이스와 AP, 응용 서비스, 데이터베이스 등은 이미 알려진 다양한 악의적 시스템 공격을 감내할 수 있어야 한다. 이를 위해서 안티바이러스 소프트웨어, 방화벽, 샌드박스, 침입 탐지/방지 기법 등 다양한 형태의 시스템 보안 툴을 적용하여 IoT의 시스템 보안성을 강화할 수 있다.[6]

IV. 결론

IoT의 각 요소 기술을 안전하게 보호하는 보안 기술이 존재한다고 하더라도 이를 통합/연동하는 방법이 없다면 보안 취약성이 발생할 수

있다.[3] IoT 서비스는 단일 사업자가 주도하는 것이 아니라 다양한 주체가 공존하는 시장을 형성 하고 있으므로 단일 기술에 대한 보안 취약성을 다루는 것보다 더 심각해질 수 있다.[3]

IoT 서비스는 오픈 데이터 혹은 open API, 매시업 서비스를 지향하므로 각 서비스에 대한 인증/인가/접근 제어/권한 제어 문제뿐만 아니라 통합 시 혹은 통합을 통한 새로운 서비스 제공 시의 동일한 보안 이슈를 다룰 수 있어야 한다.[3] 따라서 보안 기술을 통합하여 사용할 수 있도록 제공하는 서비스를 제안한다. IoT 기기 혹은 IoT 서비스 단위에서 점검을 하되 한 분야의 취약점만이 아니라 보안 점검이 필요한 모든 분야의 취약점을 통합적으로 파악해주는 기능이 필요하다. 이를 구현하기 위해서는 IoT 보안 서비스 제공자들이 통합 서비스를 구축할 수 있도록 조율을 돕는 협력 기구가 있어야 하고, 표준을 제시하는 것이 중요하다.

IoT 환경에서 모든 사물이 안전하게 연결되어 사생활을 침해당할 수 있는 가능성을 차단하는 것이 앞으로 지속적인 IoT의 발전을 위해 중요한 부분이다. IoT가 많은 분야와 융합할 수 있는 만큼 분야별로의 보안 위협 대응 방안이 필요하고, IoT 각 기기마다 효과적인 취약점 진단 방식과 제도적이고 기술적인 해결방안이 조속히 나올 수 있도록 해야 한다.

[참고문헌]

[1] KISTEP InI 14호_ 사물인터넷 발전과 보안의 패러다임 변화_배상태, 김진경

[2] ICT 시사상식 2017, 한국정보통신기술협회, 2016.12

[3] 사물인터넷의 미래, 전자신문사, 2014.11

[4] IoT 보안 기술 동향,電磁波技術 第24卷 第4號,2013.7

[5] 민감한 개인정보 보호를 위한 효율적인 접근제어 기법, 한국통신학회논문지 제32권 제7호(통신이론 및 시스템), 2007.7

[6] 사물인터넷의 미래, 전자신문사, 2014.11

[7] 사물인터넷_정보보호_로드맵,미래창조과학부,2014.10.31

[8] 사물인터넷 보안 위협 동향,kisa,2014.12

2017 한국정보보호학회 동계학술대회

C4 네트워크보안(I)

윤 명 근 (국민대)

- **초음파를 이용한 상용 드론 고도 측정 시스템 스푸핑 공격**
  신정민, 노주환, 신호철, 김용대 (KAIST)
- **WebOS 환경에서의 악성 애플리케이션 자동화 분석**
  박재유, 차상길 (KAIST)
- **다크웹과 표면웹간의 연관성 분석**
  김다솜, 임성희, 임수진, 조은희 (KITRI BoB), 니콜라이 아카티예프 (Horangi Cyber Securi), 유현 (경찰수사연수원)
- **최신 TLS 암호화 프로토콜 웹 서버 적용 의무화 방안 연구**
  김성진 (고려대학교)

초음파를 이용한 상용 드론 고도 측정 시스템 스푸핑 공격

신정민, 노주환, 신호철, 김용대

KAIST 전기 및 전자공학부

Spoofing Attack against Altitude Measurement System of Consumer Drone with Ultrasonic Wave

Jungmin Shin, Juhwan Noh, Hocheol Shin, Yongdae Kim

School of Electrical Engineering, KAIST

요 약

드론에는 자동적인 자세제어를 위한 다양한 센서가 탑재돼어 있는데, 그 중 초음파 센서는 저고도에서의 고도 측정에 주요한 역할을 맡고 있다. 그러나 초음파 센서는 같은 주파수의 초음파를 사용하는 초음파 간섭원에 의해 오작동을 일으킬 수 가능성을 내재하고 있다. 이러한 취약성을 기반으로 하여, 본 논문은 상용 드론 중 하나인 DJI Phantom 4의 고도 측정 시스템을 분석하고, 고도 출력값을 임의로 조작할 수 있는 스푸핑 공격 방식을 제시한다. 제시된 공격 방식의 유효성을 확인하기 위하여 보급형 초음파 센서 모듈을 이용하여 실제 Phantom 4에 대한 스푸핑 공격을 수행하였고, 고도 출력값을 임의로 조작할 수 있음을 실험적으로 확인하였다.[1)]

I. 서론

드론 제작 기술의 발달로 저가형 드론에서부터 전문가용 드론, 산업 및 군사용 드론까지 드론들은 다양한 용도로 사용되고 있다. 자동적인 자세제어시스템 없이 드론을 조종하는 것은 매우 어려운 일이기 때문에 드론에는 다양한 센서들이 탑재되어 있고, 그 중 초음파 센서는 저고도에서의 고도 측정을 위한 핵심 센서로서 초음파 신호를 송신하고 그 메아리가 수신되기까지의 시간을 측정하여 지면과 드론과의 거리를 감지하는 데 이용된다. 고도의 측정은 GPS나 기압 센서로도 수행할 수 있으나 이들 센서들은 저고도(10m 미만)에서 초음파 센서에 비해 정확도가 떨어지기 때문에 초음파 센서 사용의 이점이 있다.

소리 채널의 악용을 통한 센서 공격에 관해서는 Son 등과 Trippel 등이 MEMS 자이로스코프, 가속도계가 소리에 대한 공격에 취약함을 보인 바 있고[1], [2], 초음파 센서에 대해서도 Chen 등이 테슬라 모델S에 탑재된 초음파 센서 스푸핑에 대해서 다룬 바 있으나[3] 드론의 초음파 센서에 대한 공격은 아직 다뤄진 바 없다. 이에 우리는 상용 드론인 DJI Phantom 4의 초음파 센서를 테스트 대상으로 삼아 드론 탑재 초음파 센서에 대한 스푸핑 공격에 대한 연구를 진행하였다.

II. Phantom 4 고도 측정 시스템 분석

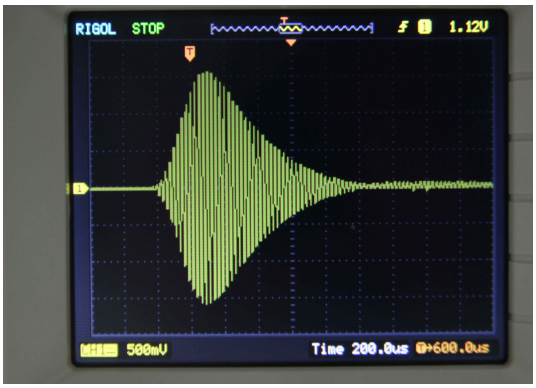
초음파 센서는 기본적으로 특정 파형의 신호를 발생시키고 해당 파형의 신호가 물체에 맞고 다시 돌아오기까지의 시간을 측정하여 이를 통해 물체까지의 거리를 계산하는 방식으로 동작한다. 음속을 340m/s로 가정할 경우, 신호를 송신한 시간을 t_s,[s] 신호를 수신한 시간을 t_l [s]이라고 할 때 물체까지의 거리 l은

1) 이 논문은 2017년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.B0717-16-0119,안전한 드론 서비스를 위한 보안 정보유출 대응 및 ID관리 기술 개발)

$340\times(t_l-t_s)/2$ [m]와 같다.(15℃ 기준).

스푸핑 공격의 기본적인 원리는 위와 같은 현상을 실제 물체 없이 모방하는 것이다. 이를 위해서는 스푸핑 하고자 하는 초음파 센서의 주파수와 주기 등의 특성 대한 분석이 필요하다. 이를 위해서는 초음파 신호를 수신할 수 있는 초음파 수신기가 필요했는데, 우리는 별도의 수신기를 구입하지 않고, HC-SR04 초음파 센서모듈에 부착돼 있는 초음파 마이크를 분리하여 이를 이용하였다.

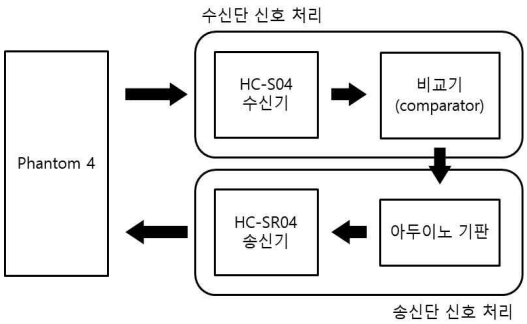
직접적인 측정에 앞서 Phantom 4의 스펙 분석을 수행하였다. 이로부터 Phantom 4의 고도 측정 시스템은 40kHz의 초음파 센서를 사용하며, 최대 10m의 고도까지 측정할 수 있다는 사실을 알 수 있었다.[5] 또한 Phantom 4의 초음파 센서 시스템을 여러 환경에서 테스트해 본 결과, 서로 다른 거리의 여러 물체가 탐지되는 환경에서는 가장 가까운, 즉, 가장 메아리가 뚜렷이 수신되는 물체와의 거리를 표시함도 알 수 있었다. 송출되는 초음파 신호의 파형에 대해서는 초음파 마이크를 이용한 측정 결과 [그림 1]과 같았고, 초음파 펄스의 두께는 1.2ms, 초음파 펄스의 송출 주기는 일정하게 50ms임을 관찰할 수 있었다.



[그림 1] Phantom 4의 초음파 신호 파형

III. 공격 방식

전체적인 공격 방식의 개요는 [그림 2]와 같다. Phantom 4의 초음파 센서 송신부에서 나온 초음파 신호를 HC-SR04 모듈에서 분리한 초음파 수신기를 통해 받고, 비교기를 거쳐 아두이노의 디지털 입력으로 제공한다. 아두이노에는 비교기의 출력 신호를 일정 주기로 읽고 Phantom 4의 초음파 신호가 수신된 경우 미리 입력된 지연 시간 이후에 HC-SR04의 송신기가 초음파 펄스를 송출하게 하는 프로그램을 작성한다.



[그림 2] 공격 방식 개요

1. 수신단 신호 처리

HC-SR04는 초음파 수신기 및 송신기로 구성되어 있다. 그 중 수신기를 HC-SR04로부터 분리하여 Phantom 4의 초음파 신호를 수신하는 데 이용한다. 비교기는 수신단에서 수신한 초음파 신호를 전달받아 특정 세기 이상을 띄는 경우 5V, 아닌 경우 0V로 하는 이진 출력으로 변환한다. [그림 3]는 비교기에서 검출한 신호와 수신단의 신호를 보여준다.



[그림 3] 비교기 검출 신호(노)와 수신단 신호(파)

2. 송신단 신호 처리

비교기에서 받은 이진 출력은 아두이노에 디

지털 입력으로 제공된다. 스푸핑을 위해서는 비교기의 이진 출력을 기반으로 Phantom 4 초음파 센서로부터의 펄스송출시점을 특정한 후 사전설정된 지연시간을 더해 초음파 펄스를 HC-SR04 송신기를 통해 Phantom 4로 되돌려 보내야 한다. 하지만 Phantom 4의 신호 송출 주기는 50ms인 반면, HC-SR04 송신기의 최소 신호 송출 주기는 200ms으로서 단일 송신기로는 50ms마다 초음파 펄스를 송출하는 것이 불가능하였다. 따라서, 4개의 HC-SR04 송신기를 사용하여 순차적으로 신호를 보내도록 처리하였다.

이상의 조치로 수신된 초음파 펄스에 약간의 시간 지연을 가해 Phantom 4에 되돌려 보내는 것은 가능해졌으나, 스푸핑 장치보다 가까운 위치로 물체를 스푸핑하기 위해서는 음의 시간 지연을 가하는 것이 불가능하기 때문에 매우 큰 시간 지연을 가해 다음 번 주기로의 rollover를 일으키는 것이 필요하다. 문제는 이를 위해서는 시간 지연을 가하는 도중에 새로운 펄스 신호의 수신을 처리해야 한다. 멀티스레딩 및 인터럽트 기법을 활용하면 간단하지만 아두이노에서는 이들을 지원하지 않기 때문에 [알고리듬 1]과 같은 단일스레드 루틴을 통해 이를 구현하였다.

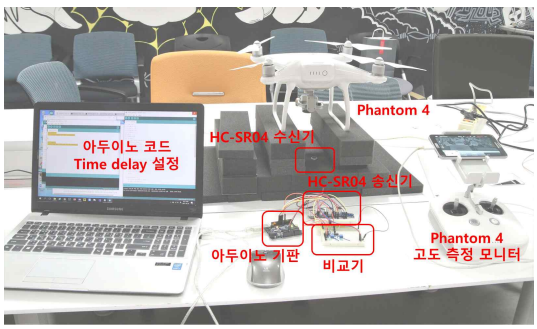
```
int cnt1, cnt2 = 0
int delay = INPUT (0≤INPUT≤100ms)
void loop() {
  if(Phantom 4 초음파 펄스 수신)
    if (cnt1 == 0) cnt1 = delay*10
    else cnt2 = delay*10
  delay(100us)
  if(cnt1, cnt2 중 한 개라도 1일 경우)
    4개의 송신부에 번갈아가며 신호 송신
  if(cnt1 >= 1) cnt1--
  if(cnt2 >= 1) cnt2--
}
```

[알고리듬 1] 송신부의 아두이노 알고리듬

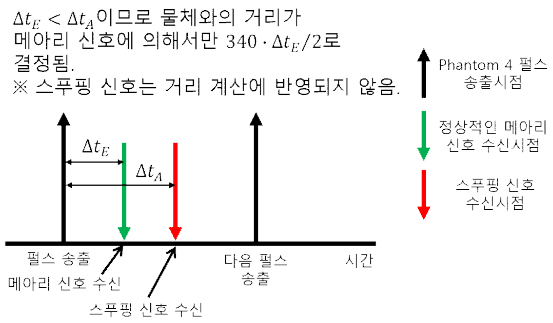
두 개의 카운터 변수를 사용하는 것이 [알고리듬 1]의 핵심인데, 매우 큰 시간 지연을 가하는 경우 가해진 시간 지연이 미처 끝나기 전에 Phantom 4로부터 새로운 펄스 신호가 한 번 더 들어오는 시점을 저장하기 위해서이다. [알고리듬 1]의 방식으로는 세 주기 이상에 걸치는 시간 지연은 올바르게 처리되지 않으나, 한 번의 rollover 만으로 전 지점으로의 스푸핑이 가능해지기 때문에 두 주기 이상에 걸치는 시간 지연을 가하는 것은 필요하지 않다.

IV. 공격 결과

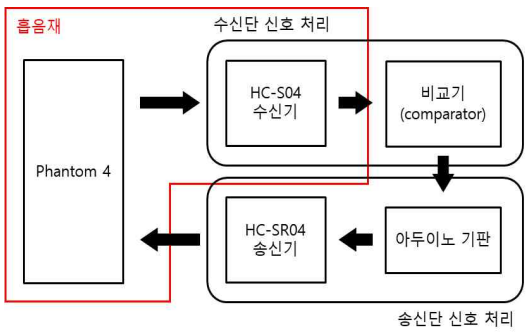
스푸핑 공격의 성공여부를 확인하기 위한 실험환경은 [그림 4]와 같다. 아두이노 코드의 delay값을 조절하면서 Phantom 4의 고도 측정 시스템의 결과값을 관찰하였다.



[그림 4] 스푸핑 공격 실험환경

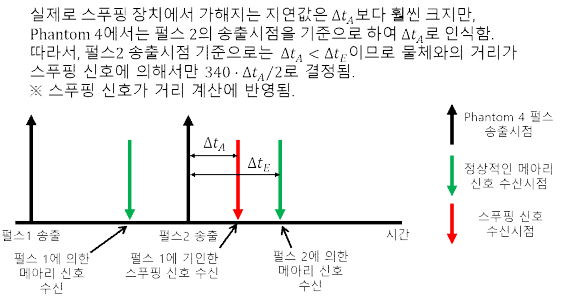


[그림 5] 스푸핑 장치보다 먼 거리로 스푸핑하는 경우의 타이밍 다이어그램



[그림 6] 흡음재를 포함한 플랫폼

Phantom 4 초음파 센서 시스템의 특성 상, 가장 가까운 거리까지의 물체까지의 거리만 표시하게 되기 때문에 스푸핑 장치보다 먼 거리로 스푸핑하면 [그림 5]와 같이 되어 스푸핑 여부를 확인할 수 없다. 이를 해결하기 위해서는 Phantom 4 초음파 센서가 송출한 초음파 펄스에 대한 메아리 신호를 모두 흡수하여 제거해야 한다. 이를 위해 [그림 6]에 표시된 부분에 흡음재를 사용하였다.



[그림 7] 스푸핑 장치보다 가까운 거리로 스푸핑할 경우의 타이밍 다이어그램

반면, Phantom 4의 펄스 발생 주기 이상의 시간 지연을 주는 방식으로 스푸핑 장치보다 가까운 거리로 스푸핑할 경우, 스푸핑 장치에서 발생된 공격 펄스가 정상적인 메아리 신호보다 각 펄스 주기 안에서는 더 빨리 도달하기 때문에 흡음재가 없어도 스푸핑 여부를 확인할 수 있다. [그림 7]이 이와 같은 경우의 상황을 나타내고 있다.

실험 결과, 이상의 방식을 적용하여 제시된 실험 환경에서 Phantom 4의 고도 측정 시스템의 결과값에 임의의 값을 주입하는 데 성공하였다. https://www.youtube.com/watch?v=VPgd4Hc3WdU&feature=youtu.be에서 실험의 데모 영상을 확인할 수 있다.

V. 방어 기법

본 논문에서 제시한 공격들 중 스푸핑 장치보다 가까운 거리로의 스푸핑은 초음파 신호의 주기가 일정해야 하는 것이 필요조건이었다.. 따라서 초음파 신호 생성 주기를 일정하게 하지 않고 랜덤하게 하면 더 가까운 거리로의 스푸핑 공격은 막을 수 있다. 그러나 이는 초음파 센서의 구현을 더 복잡하게 하기에 저렴한 단가가 장점인 초음파 센서에는 적용하는 것이 어려울 수 있다.

VI. 결론

본 논문에서 우리는 상용 드론의 고도 인식 시스템을 초음파 스푸핑 공격을 통해 교란시킬 수 있는 공격방식을 제시하고, 이를 저가의 보급형 초음파 센서 모듈을 이용하여 시판중인 상용 드론인 Phantom 4를 대상으로 입증하였다. 실험 결과 Phantom 4의 고도 인식 시스템의 고도 인식 결과에 임의의 값을 주입하는 것이 가능하였다. 본 논문은 또한 제시한 공격방식 중, 스푸핑 장치보다 가까운 거리로의 스푸핑을 막는 방어 기법과 해당 기법의 한계점 또한 함께 제시하였다.

[참고문헌]

[1] Yunmok Son, Hocheol Shin, Dongkwan Kim, Youngseok Park, Juhwan Noh, Kibum Choi, Jungwoo Choi, and Yongdae Kim,: Rocking Drones with Intentional Sound Noise on Gyroscopic Sensors, 24th USENIX Security Symposium August 12 - 14, 2015

[2] Timothy Trippel, Ofir Weisse, Wenyuan Xu*, Peter Honeyman, Kevin Fu: WALNUT: Waging Doubt on the Integrity of MEMS Accelerometers with Acoustic Injection Attacks, Security and Privacy (EuroS&P), 2017 IEEE European Symposium on

[3] Chen Yan, Wenyuan Xu, Jianhao Liu: Can You Trust Autonomous Vehicles: Contactless Attacks against Sensors of Self-driving Vehicle, DEF CON, 2016

[4] Ultrasonic ranging module HC-SR04, http://www.micropik.com/PDF/HCSR04.pdf

[5] User Manual - DJI phantom 4, https://dl.djicdn.com/downloads/phantom_4/en/Phantom_4_User_Manual_en_v1.0.pdf

WebOS 환경에서의 악성 애플리케이션 자동화 분석1)

박재유*, 차상길**

*한국과학기술원 소프트웨어대학원, **한국과학기술원 정보보호대학원

Automated Analysis of Malicious Applications in WebOS

Jae You Park*, Sang Kil Cha**

*Software Graduate Program, KAIST.
**Graduate School of Information Security, KAIST.

요 약

WebOS는 스마트 가전, IoT 등 다양한 제품에서 운영체제로 널리 사용되고 있다. WebOS에서는, 대부분의 모바일 운영체제가 그러하듯, 단 하나의 악성 앱 만으로도 전체 시스템이 장악될 수 있다. 그럼에도 불구하고 WebOS의 앱 보안에 관한 체계적인 연구는 여전히 미미한 실정이다. 이에 본 연구에서는 WebOS 앱에 대한 정적 및 동적 분석을 수행하여 악성 앱을 자동으로 판단할 수 있는 방법론 및 도구를 제안한다. 이를 위해 우리는 메모리 포렌식 기반의 악성코드 분석 프레임워크를 적용하였으며, 사용자 실험을 통해 제안한 기술이 랜섬웨어 등의 주요 악성코드를 신속하게 탐지하는 데 도움을 준다는 것을 확인하였다. 우리는 본 논문에서 제안하는 도구가 WebOS의 앱스토어 관리 측면에서 높은 업무 효율을 보장할 것으로 기대한다.

I. 서론

사물인터넷 시대의 도래로 생활 곳곳의 모든 장비가 네트워크에 연결되어 온라인으로 동작하고 있다. 대부분의 가전제품에도 모바일 운영체제가 설치되어 있는데, 특히 WebOS는 [1] 스마트 TV나 디지털 사이니지 제품에서 널리 사용되고 있다.

모바일 운영체제의 발전과 더불어 그에 대한 보안 위협은 나날이 증가하고 있다. 특히 PC 환경에서 활개를 치던 악성코드가 이제는 그 영역을 모바일 쪽으로도 확장하고 있는데, 실제로 다양한 스마트폰용 악성 앱이 확산되는가 하면, IoT 장비만을 집중적으로 노리는 Mirai 봇넷이 등장하기도 하였다. 특히 스마트 가전은 사용자의 사생활 속에서 밀접하게 이용되므로 개인정보 노출 등의 피해 발생 가능성이 매우 높다.

1) 본 연구는 방위사업청과 국방과학연구소의 지원으로 수행되었습니다(UD160066BD).

만약 악의적 해커가 위장된 악성 앱을 개발하여 앱스토어에 등록한다면, 사용자는 이를 무의식적으로 설치하여 막대한 피해를 입을 수 있다. 이를 방지하기 위해 안드로이드 앱의 보안성 검토에 관한 연구가 다수 진행되었으며 [2][3], Apple의 경우에도 앱 감사(auditing)를 통해 불필요하게 권한 상승을 시도하거나, 불법적인 메모리 조작이 발생할 경우 스토어 등록을 거부하는 정책을 채택하고 있다 [4].

다른 모바일 운영체제와는 달리, WebOS 앱에 대한 체계적인 연구는 그 활용도 및 파급효과 대비 여전히 미미한 수준이다. 이에 본 연구에서는 WebOS 앱의 감사과정을 자동화할 수 있는 새로운 방법론 및 도구를 제안한다.

II. 관련 연구

본 장에서는 다른 모바일 플랫폼에서의 악성 앱에 대한 분석도구 동향을 살펴보고, WebOS 앱 분석에 대한 선행연구의 한계점을 살펴본다.

2.1 기존의 모바일 악성 앱 분석도구 동향

2017 BlackHat USA에서는 안드로이드 악성 앱을 감사(audit)하는 CuckooDroid 도구가 공개되었다 [5]. 이 프로젝트는 Windows PC 환경의 악성코드를 분석하는 오픈소스 Cuckoo Sandbox를 [6] 안드로이드 에뮬레이터에서 호환되도록 개조한 것으로, 정적 분석과 동적 분석을 모두 지원한다. 하지만 본 연구와는 달리 메모리 분석은 불가능하다는 단점이 있으며, 안드로이드의 Dalvik API 후킹 기반으로 구현되어 WebOS 환경으로의 직접 적용은 불가능하다.

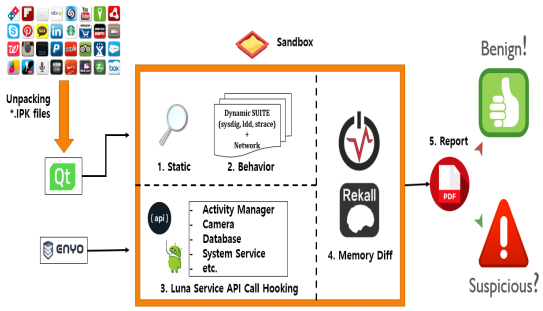
2.2 WebOS 악성 앱에 대한 연구

2017 Hack in Paris에서는 WebOS가 탑재된 스마트 TV를 표적으로 하여, SDK를 통해 악성 앱을 제작하고 이를 앱스토어에 등록함으로써 광범위하게 유통시키는 방식의 공격 시나리오가 제시된 바 있으며, 현재까지도 유효한 공격으로 알려져 있다 [7][8]. 본 연구에서는 실험을 통해 파일 시스템에 접근하여 암호화를 수행하는 랜섬웨어, 네트워크 패킷을 다량으로 송신하는 DoS 공격 등이 실제로 WebOS 위에서 동작 가능함을 직접 확인하였다.

WebOS 관련 다수의 제품을 유통하는 'L 전자'에서는 자사의 WebOS 앱스토어에 등록된 앱에 대한 보안성 검증 방안을 논문으로 발표한 바 있다 [9]. 해당 연구에서는 웹 브라우저 상에서 작동하는 HTML 또는 자바스크립트 코드의 정적 정보를 추출하여 패턴인식 및 머신러닝을 통해 악성 유무를 점검한다. 그러나 이러한 방식은 오직 정적(Static) 정보만을 활용하기 때문에 난독화된 프로그램에 대해서는 동작하기 어려운 한계가 존재한다. 반면, 본 연구에서는 정적 및 동적 분석을 모두 활용하기 때문에 보다 높은 정확도를 기대할 수 있다.

III. 제안 기법

본 논문에서는 최초의 자동화된 WebOS 앱 감사 기법을 제안한다. 제안하는 기법은 정적 분석과 동적 분석을 모두 포함하며, WebOS의 Luna API Hooking, 메모리 포렌식 등을 통합적으로 적용하여 정확도를 높였다.



[그림 1] 제안 기법의 구조

[그림 1]은 제안 기법의 전체적인 흐름도이며, 각 단계별 세부 분석방법을 아래의 절에서 순서대로 각각 소개한다.

3.1 정적 분석

정적 분석 단계에서는 애플리케이션이 배포되는 ipk 형식의 패키지 파일을 대상으로 분석한다. WebOS에서 구동되는 앱은 EnyoJS 프레임워크를 사용하여 제작된 웹 앱과, QT/QML C++을 이용하여 작성하고 컴파일한 ELF 바이너리 형식의 네이티브 앱으로 구분된다. 앱의 형식에 따라 향후 동적 분석 방법이 달라지므로 이 단계에서 언팩을 수행하고 타입을 판단한다. 타입 판별 이후에는 각 타입별로 존재하는 심볼 정보를 추출하고, Yara 룰 기반의 패턴 매칭 및 해쉬비교를 진행한다. 이 단계에서 악성코드로 판별하기 어려운 경우에는 다음의 동적 분석 단계로 넘어가게 되며, 최종 리포트에는 정적 및 동적 분석의 정보가 모두 포함된다.

3.2 동적 행위 분석

동적 분석 과정에서는 실제로 해당 앱을 실행하는 과정이 포함된다. 단, 매번 실제 환경에서 임의의 코드를 실행하면, 악성코드에 의해 테스트베드 자체가 감염될 위험이 있다. 따라서 가상화 솔루션인 VirtualBox를 사용하여 가상머신(WebOS-Ports 기반)을 구동하고 스냅샷 복원 지점을 만든다. 이후 호스트와 VM 사이에 연결을 수립하여 안전한 통신을 맺고(이를

ARES-Tunneling이라 한다), 앱을 VM 내부로 유입시켜 실행한 뒤 시스템콜을 추적함으로써 프로세스 호출이나 네트워크 연결 등 다양한 동적 행위 변화를 기록하는 방식으로 진행한다. 기록된 데이터들은 호스트 시스템으로 전송되어 데이터베이스에 적재된다. 특히 앱 실행 전후의 상태변화를 정상 스냅샷과 비교하여 해당 앱에 의해 어떤 행위가 발생했는지를 선별하여 보여주는 기능을 지원한다.

3.3 Luna API Hooking

일반적으로 WebOS 앱은 하드웨어 등의 하위 계층에 직접적으로 접근할 수 없다. 때문에 프로그래머의 편의를 위해 OS의 여러 가지 서비스와 구성요소를 사용하기 위한 API를 제공하고 있다. WebOS는 'Luna API'를 제공하고 있으며, 앱이 API 메서드 함수를 호출하면 Luna-Bus를 통해 서비스 데몬 프로세스에 메시지를 전달한다. 메시지에는 해당 앱이 요청하고 응답받는 다양한 정보가 포함된다. 이 과정을 관찰하면 어떤 종류의 API가 작동되는지 감지할 수 있다. 앞서 언급한 CuckooDroid [5] 은 Droidmon [10] 을 활용하여 Dalvik API를 관찰하는데, 이에 착안하여 우리는 WebOS의 Luna API 함수를 후킹 하는 방법을 개발하였다. 단, 카메라, 리모컨 등 일부 API의 경우 가상 환경에서 작동하지 않는 한계가 있으므로, 지원 불가한 기능에 대해서는 실제 장비를 사용해야만 정확한 결과를 얻을 수 있다.

3.4 Memory 비교 분석

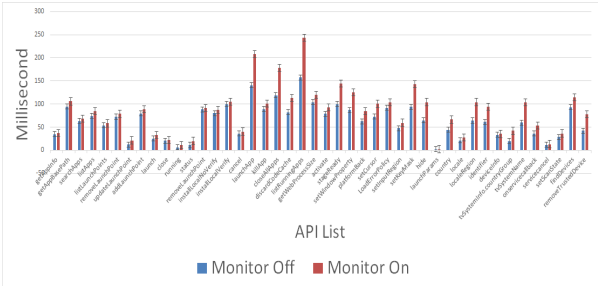
악성코드 분석을 위한 메모리 포렌식은 바이너리 코드에서 알 수 없는 동적 정보나 암호화 비밀키 등을 메모리 분석으로 해결하는 방안이다 [11]. 그러나 메모리 분석 도구인 Volatility [12]는 대부분 Windows 운영체제에 최적화되어 있으므로, WebOS 프로파일은 지원되지 않는다. 본 연구의 대상인 Open WebOS는 리눅스 커널 3.14을 기반으로 하므로, 해당 환경에 부합하는 커널 모듈을 입수하여 직접 빌드한 후 프로파일을 생성했다. 프로파일을 구성요소는 커널 구조가 정의된 vtypes (module.dwarf) 파일과, 디버그 심볼 정보를 담은 System.map 이다. 이를 압축하여 플러그인으로 저장해두면 대상 시스템을 인식하게 된다. 그 결과 Volatility로 WebOS의 메모리 덤프를 분석할 수 있었으며, 악성코드 실행 전후의 메모리 비교 분석을 통해 행위를 파악할 수 있었다.

IV. 적용결과

본 논문에서 제안한 기법을 체계적으로 수행하기 위해 약 1,000 줄의 파이선 프로그램을 제작하였고, 해당 도구의 성능 점검을 위해서 우리는 macOS 환경에서 실험을 진행하였다. 또한 구글 폼을 활용하여 온라인으로 사용성 설문을 진행하였다.

4.1 Overhead 테스트

[그림 2]은 후킹을 실시할 때 전체 44종의 API에서 발생하는 성능 저하의 정도를 요약한다. 파란 막대는 후킹을 사용하지 않은 호출시간이며, 빨간 막대는 후킹 시 지연된 시간이 포함된 값이다. 결과적으로 평균 20ms의 부하가 발생하는 것으로 파악되었으며, 대체적으로 네이티브 앱보다 웹 앱에서 저하가 뚜렷하게 나타났다. 이는 웹앱 관리자(WAM)가 앱 관리 시스템(SAM)을 경유하는 과정에서 지연이 중첩되기 때문인 것으로 보인다.



[그림 2] API Hook Overhead (100회 평균, ms)

하지만 본 기술의 목적은 앱 감사자가 효율적으로 악성 앱을 판별할 수 있도록 도움을 주는 것이며, 앱이 최초 등록될 때에만 사용되므로 20ms의 오버헤드는 큰 문제가 되지 않는다고 판단된다. 또한 다음 절에서 이어지는 사용성 조사를 통해, 우리는 도구의 실익이 어느 정도인지 측정할 수 있었다.

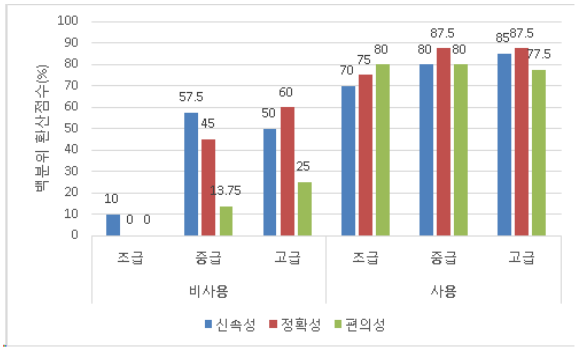
4.2 사용성 설문 연구

본 논문에서 제안한 도구가 실제 유관 업무에 적용되었을 때 기존의 수작업 방식보다 얼마나 효율적인지를 평가하기 위해 필드 연구를 수행하였다. [표 1]은 피험자의 특징이며, 각자의 배경지식, 학력 및 관련 업무 경력을 토대로 세 등급으로 구분하여 참여시켰다.

[표 1] 피험자 개요

전문성	초급	중급	고급
인원수	2	8	4
합계	14		

우리는 이들을 실험군과 대조군으로 나누었는데, 각 그룹에는 등급별 인원이 정확히 절반으로 포함되도록 구성하였다. 실험군에는 3개의 악성(랜섬웨어, 임의 다운로드 시도, 쉘 호출) 및 3개의 정상 앱(메모장, 테트리스, 계산기)과 더불어 우리의 자동화 도구를 제공하였으며, 대조군에는 악성 및 정상 앱 만을 제공하였다. 각자 분석을 수행하도록 한 후 객관식 및 주관식 문항을 제시하여 답변을 수집하였다. 또한, 문제풀이에 소요된 시간을 측정하여 두 집단 간의 신속성을 비교함으로써 도구 사용이 주는 유익을 종합적으로 판단하였다.



[그림 3] 도구의 사용성 설문조사 결과

[그림 3]은 집계된 수치를 백분위로 환산한 결과이며, 이에 따르면 실험군이 대조군보다 평균 78%로 더 신속하게 분석할 수 있었고, 특히 초급 분석가에게 큰 도움이 되었다. 도구 사용 그룹이 시스템에 대해 긍정적으로 인식한 가장 큰 요인은 편의성 측면(10점 만점에 8점)이며, 정확성도 48% 더 우수하게 평가되었다.

V. 결론

본 논문에서는 WebOS 앱 보안감사와 관련하여 메모리 비교분석 기반의 새로운 자동화 분석 기법을 제안하였다. 정적 및 동적 분석을 통합 적용하였으며, API 후킹 기법을 활용한 최초의 WebOS 앱 감사시스템을 구현할 수 있었다. 또한 사용자 연구를 통해 제안된 시스템이 분석가의 업무 효율을 향상하는 것을 확인할 수 있었다. 후속 연구로는 누적된 분석 데이터에 머신러닝 기반 분류 기법을 적용하여 탐지 성능을 향상하는 방안을 제안한다.

[참고문헌]

[1] WebOS-Ports : http://webos-ports.org

[2] Zhang, Yuan, et al. "Vetting undesirable behaviors in android apps with permission use analysis." In Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security. ACM, 2013

[3] Lu, Long, et al. "CHEX: statically vetting android apps for component hijacking vulnerabilities." In Proceedings of the 2012 ACM Conference on Computer & Communications Security. ACM, 2012

[4] Apple inc. “iOS Security White Paper”, 2017

[5] CuckooDroid : https://goo.gl/7FJ1NQ

[6] Guarnieri, Claudio, et al. "The cuckoo sandbox.", 2012

[7] Lee, JongHo, et al. "Are you watching TV now? Is it real?: Hacking of smart TV", Hack in Paris, 2017

[8] 이종호, 김승주, “Open webOS 기반 스마트 기기에 대한 해킹 시나리오 분석”, 한국정보보호학회 동계 학술발표논문집, 2016

[9] Belenko, Vyacheslav, et al. "Security Verification of the Programs in WebOS app Store.", 2014

[10] Droidmon. : https://github.com/idanr1986/droidmon

[11] Amari, Kristine. "Techniques and tools for recovering and analyzing data from volatile memory." SANS Institute, 2009

[12] Volatility : https://goo.gl/TymJ21

다크웹과 표면웹간의 연관성 분석1)

김다솜* 임성희* 임수진* 조은희* 니콜라이 아카티예프** 유현***

*과학기술정보통신부 차세대보안리더양성 프로그램 BoB

**Horangi Cyber Security ***경찰수사연수원

dasomong@gmail.com* shee2520@gmail.com* sjk050505@gmail.com* beensy74@gmail.com*
nikolay@horangi.com** hyeony@gmail.com***

Link Analysis in Dark-web and Surface-web

Da-Som Kim* Sung-hee Lim* Su-Jin Lim* Eun-hee Jo*

Nikolay Akatyev*** Hyeon Yu****

*Ministry of Science and ICT, Security Leader Training Program BoB

**Horangi Cyber Security ***Korea Police Investigation Academy

요 약

마약, 불법거래, 해킹 등 일반적으로는 드러나지 않는 다크웹 채널로 범죄가 증가하면서, 수사기관에서는 이에 대응하기 위해 다양한 방법을 사용해왔다. 하지만 쉽게 서버가 사라지거나 이동을 하며 사이트 가입조건을 점차 까다롭게 하고 있어, 수사기관은 사이트 검색과 가입자 인증 절차 등에 주기적인 모니터링과 많은 시간을 투자하여야 한다. 우리는 다크웹 내의 숨은 정보를 찾아내기 위해 다크웹과 표면웹 간 데이터를 분석해 다크웹 사용자에 대한 추가적인 정보를 얻어내고, 분석자료를 토대로 알고리즘 및 방법을 제안하고자 한다. 본 논문에서는 표면웹상의 커뮤니티(SNS, 메신저, blog)를 활용하여 다크웹(한국어 포럼, 마켓) 사용자의 흔적을 추적할 수 있도록 각 웹 채널간의 연결성을 분석해보려고 한다.

I. 서론

다크웹이란 일반적인 검색엔진이나 브라우저로 접속이 불가한 사이트로, 인터넷 검열이 이루어지는 국가 등에서 사이버상의 자유를 위해 시작되었다가 마약, 총기거래, 비트코인세탁 등 불법거래 채널로 변질되며 부정적인 의미로 인식되고 있다. 다크웹은 16바이트의 주소와 onion 확장자를 주소로 가진다. I2P나 tor 브라우저를 통해서 접속이 가능하며, 여러 프록시를 거치는 특성상 각 단계별 통신 과정을 거치며 복호화해내지 않는 이상 접속자의 IP나 정보를 파악하기가 어렵다. 외부에서 통신내용을 중간에 가로채내더라도, 실제 어느 지점인지 파악이 어려워 수사의 어려움이 따른다. 구체적 제보가 들어오지 않는 이상 수사가 진행되지 않는 행정구조나, 국내에 서버를 두지 않는 사이트의 경우 관할 범위를 벗어나 국내 수사기관들이 난항을 겪고 있다. 더욱이 가상화폐를 주거래 수단으로 사용하면서 사용자의 추적이 더욱 어려워졌다.

주요시장은 마약거래이며 외에도 성매매, 무기, 비트코인 세탁서비스, 해킹툴, 청부살인 등 거래범위가 다양하며, 외국에서는 알파베이를 통한 마약거래 후 해당 마약으로 인한 사망자가 발생하며 이목을 끌었고[1], 최근 국내에서도 한국어로 된 사이트에서 다수의 불법거래가 적발되면서 정부에서도 사업개발과제로 착수하는 등의 노력을 기울이고 있다[2].

본론에서는 관련된 연구와 함께 다크웹전용 크롤링 오픈소스인 Ahmia, Onionscan+OSINT에 대해서 간단히 살펴보고, 직접 크롤링 및 브라우징한 결과를 토대로 분석용 알고리즘을 제

1) 이 논문은 2017년도 차세대 보안리더 양성프로그램(BoB)지원을 받아 수행된 연구임.

안한다.

다크웹과 표면웹 간의 연결성을 보여주는 Link Analysis은 각 채널에서 돌아다니는 계정, 연락처, 비트코인주소 등을 수집하여 일치하는 데이터들을 연결 짓는 것을 말한다. 연결에 어떤 방식으로 사용하는지 본론에서 자세히 이야기한다.

II. 본론

1. 관련 연구

1.1 다크웹 크롤러 오픈소스

Ahmia는 Tor 관련 결과만 반환하는 검색 엔진이며 16년 기준 5000개 이상의 onion index를 생성했다. Ahmia는 대표적인 다크웹 검색 엔진으로, 내부 프로세스 중 하나로 scrapy를 사용하는 크롤러가 존재한다[3].

OSINT의 경우, onionscan을 활용하여 가공과 출력에 초점을 두어 제공을 하는데, 크게 SSH key값 수집, gexf파일로 출력해 Gephi로 분석가능하게 만들어주거나, 복제된 숨겨진 서비스를 찾아내어 차이를 비교해주는 세 가지기능을 추가로 가진다. 여기서 Onionscan은, 주로 tor proxy로 접속해 html을 파싱해 데이터를 분류해내는 기능을 담당한다. 파싱하는 데이터는 googleanalyticsID, 비트코인주소, jpg파일로부터 분석한 exif, PGP키, email 주소이다[4].

이들은 단순히 다크웹 크롤러와 파싱에 집중된 오픈소스들로 데이터 수집에는 유용하지만, 표면웹 간의 비교분석을 제공해주지는 않는다.

1.2 해킹관련 위협 인텔리전스 연구

Eric Nunes외 9명은 해커의 행위를 찾기 위해 해킹 포럼 및 마켓을 중심으로 데이터를 최신 익스플로잇이 올라왔는지, 제로 데이가 개발 중인지 등의 인텔리전스를 제공했다. 마켓과 포럼 각각 데이터를 수집하여, 노이즈를 제거한 후 해커의 행위와 관련된 사례를 분별하는데 학습 데이터로 사용하였다.

마켓 수집 데이터	포럼 수집 데이터
품목 제목, 품목 설명, 공급업체이름, 운송세부사항, 품목 리뷰, 판매된 품목, CVE, 남은 품목, 거래세부사항 등급	주제, 게시물 콘텐츠, 저자, 게시물 작성자, 저자상태, 평판, 주제 관심도

정보의 정확률을 높이기 위해, 노이즈제거를 사용했는데 그 방식으로는 쓰지 않는 말들의 목록을 제거하는 text cleaning, 맞춤법오류/유사단어, large feature space, title feature context의 보존 같은 5가지 방식이 존재하였다[5].

2. 배경

대개 웹 페이지의 배너에서 뽑아올 수 있는 고유 값을 추출하는 데에만 초점이 맞추어져 있거나, 은어수집에 머신러닝을 적용해 자동으로 선별해내는 크롤러에 대한 연구들이 많았다.

그 중 사용자 패턴분석의 경우, 특정 포럼에서 주로 활동하는 사용자의 접속시간, 글 작성시간, 키워드(전문 용어, 습관적으로 사용하는 단어, 자주 공유하는 URL 등)를 수집한 다음, 이를 바탕으로 지표를 생성한다. 지표를 이용해 특정 사용자의 생활패턴, 국가 시간을 통해 나라, 직업을 파악하여 또 다른 지표를 생성하며, 특정 사용자를 찾기 위한 범위를 좁힐 수 있었다[6]. 그렇지만 다크웹에서 특정 사용자를 추정하는데까지 충분한 데이터를 모으기 어려웠고, 우리는 일반적으로는 표면웹에서 홍보를 하고 다크웹을 판매를 하는 형식으로 이루어짐을 이용하여 두 채널간의 연관성을 분석해보고자 한다. 기존 연구들에서 다양한 시도를 해왔었지만 아직까지, 다크웹 사용자들을 통해 키워드나 지표를 만들고 표면웹의 지표와 비교한 경우는 없었다. 이는 한국어 포럼, 마켓처럼 데이터 수집대상의 규모가 작은 경우에 더욱 좋은 결과를 가져다 줄 것이다.

3. 구조 및 실험환경

전체 구조는 데이터를 모으는 크롤러모듈, 데이터를 분석해 구조화 시켜주는 분석모듈 두 가지 구조로 나누었다. 서버에 파싱한 데이터를 Elasticsearch(이하 ES)에 모아, 분석된 내용을 가시화 시켜서 웹서비스로 제공한다. 가시화하는데 일부는 KIBANA를 사용하며, 네트워크형 모식도는 직접 구현하도록 하였다. 크롤러에 사

용한 환경은 아래와 같으며, 아래와 같이 수집한 방법으로 분석시나리오를 작성 하였다.

사용 환경
Elasticsearch 2.3.3
Elasticsearch plugin_kopf
KIBANA 4.2.2 (수집된 데이터 통계용)
Python2.x
Requests_python module 2.18.4
Beautifulsoup(bs4)_python module 0.0.1
Pysqlite 1.0.1

크롤러는 Tor 프록시 설정 후, 다크웹 onion URL을 지정해주면, onion 사이트로 부터 html을 불러와 저장한다. 해당 태그 부분만 파싱해 단어, URL 등 데이터를 가져온 뒤 ES에 구분하여 저장한다. 마지막으로 수집한 데이터들을 json 파일로 출력해 ES로 전송하여준다. 수집한 데이터의 양식은 [index, site_name, id, mobile, pgp_key, bitcoin, collet_link, ...]와 같이 통일한다. 또한 다크웹의 경우 데이터가 사라지기 때문에 html을 통째로 저장해줄 아카이빙 기능이 필요하다. 크롤러가 데이터를 파싱해낼 때, 아카이빙이 함께 이루어지도록 한다.

4. 다크웹과 표면웹의 사례분석

표면웹의 분석대상으로는 주 판매채널인 트위터, 페이스북, 블로그, 카카오톡, 일간베스트저장소, 디시인사이드를 선정하였고, 다크웹의 경우 포럼 3곳, 마켓 3곳[7]을 타깃으로 선정하였다.

크롤러의 경우, 특정 다크웹에서는 계정을 요하는 경우, 해당 계정으로 로그인을 한 후 tor를 태워 접속하도록 설정하였다.

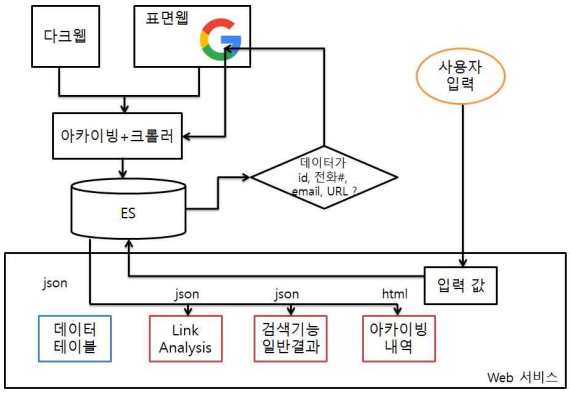
아래는 수집한 사례를 토대로 연관성 분석한 사례이다.[표 1]

사례 1	다크웹 내 특정 id가 여러 전화번호를 사용하거나, 반대로 여러 id가 특정 전화번호를 사용하는 경우. 연락처의 경우, “010-일2삼4-56칠팔” 과 같이 한글과 숫자를 혼용해 검색을 우회하는 경우가 있어, 숫자와 한글을 같은 것으로 인식하도록 한다.
사례 2	트위터는 팔로워와 팔로우를 볼 수 있어, 자기계정 연결한 내용이 파악이 가능하여 연결이 가능하도록 한다.
사례 3	트위터의 계정명이 연락처가 되어 홍보글을 올리는 경우.
사례 4	id중복시 검색을 피하거나 조직인식을 위해 영문+숫자시, 영문을 동일하고 숫자만 다르게 올리는 경우.
사례 5.	다크웹의 PGP 공개키 값, id와 표면웹의 연락처(카카오톡, 위커, 텔레그램)의 수집 : 다크웹의 경우 일반적으로 PGP키로 개인 간 통신을 위해 사용자마다 고유 PGP공개키 및 id가 적힌 개인정보 페이지를 열어볼 수 있게 해놓는다. 이 개인정보 페이지는 URL상에 인덱스 번호로 검색을 하면 특별한 인증절차가 없더라도 조회가 가능한 경우가 많다. 이를 이용해 다크웹의 모든 가입자 명단이 수집가능하다. 또한 다크웹 사이트 마다 id는 다르게 생성하더라도 연락처 정보는 동일한 경향이 있다. 이를 동일한 인물 또는 집단으로 간주하여 link한다.
사례 6	핸드폰 연락처를 수집한 경우, 정보의 유효성을 파악하기 위해, 페이스북이나 카카오톡으로 계정연결을 하여 사용자의 이름을 확인가능하다. 이 경우 수집한 내용 중 정확도 판별이 가능하다. 다크웹의 경우, 가짜정보가 거래되거나 시간이 지난 경우 연락처가 파기될 위험이 있어 수집한 데이터의 유효성을 파악해두는 것이 중요하다. 구/판매자의 정보 확인뿐만 아니라, 개인정보가 유출되어 돌아다니는 경우에도 데이터 유효성검증에 사용할 수 있다.
사례 7	URL을 통한 사이트 간 연결. linked list로 표현. 최초 주소 유포지 및 확산범위 파악에 이용가능하다.
사례 8	비트코인 주소를 통한 연결. 역시 최초 비트코인주소 유포지 및 사건 발생 시점 예측에 도움.
사례	글 게시시간 및 키워드 유사성(예. 단어, 말투). 다수의 오차가 발생하며, 통

9	합해서 보는 경우, 주요 활동 시간 및 요일, 활동안하는 시간, 평소보다 많이 접속한 시간 등 파악하려 그래프로 제공가능.
사례 10	다크웹과 표면웹 간 id가 일치하는 경우를 이용, 다크웹에서 얻어온 계정 명을 blog.naver.com/[id], facebook.com/[id], @[id]와 같은 방식으로 검색하여, 관련 정보를 역시 크롤링해 냄.

[표 1] 사례분석

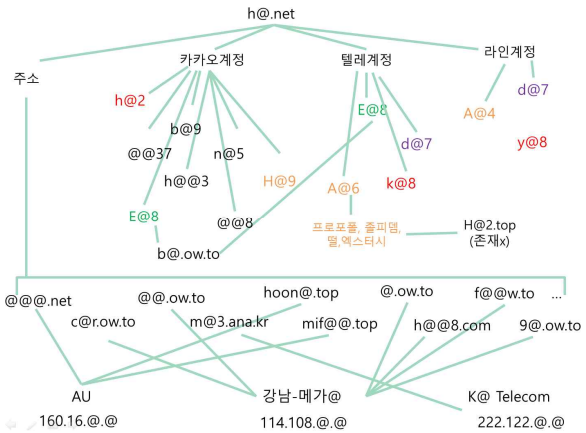
5. 전체 구조 및 알고리즘



[그림 1] 전체 구조 흐름

다크웹/표면웹으로 부터 크롤링을 해두며 전체 사이트를 아카이빙 해둡니다.(아카이빙은 증빙자료로 쓰이며), 수집해둔 데이터 중에 id, 전화, 이메일, URL이 포함된 경우 구글 검색을 통해 다시 아카이빙+크롤링한 내역에 추가되며 계속 탐색합니다. 사용자의 입력으로 검색어가 들어오면 ES조회를 통해 결과를 json으로 받아오고, 4가지 기능으로 반환이 가능합니다.

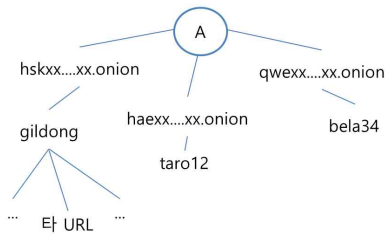
웹 DB의 데이터는 계속해서 변하므로 추가적인 정보만을 걸러 얻어야한다. 매번 전체 크롤링을 하면 시간 및 공간낭비가 되므로, 매번 마지막에 크롤링할 때 가져온 index값을 기록해두었다. 페이지에 정보가 추가되었을 때, 추가된 일부를 선택적으로 가져와 업데이트 하도록 한다. [표 1]의 사례1,5를 통해 여러 [그림 2]과 같이 계정이나 연락처들을 하나의 집단(또는 인물)로 연결 지을 수 있다. 이를 활용하여 id의 규칙성이나 주로 활동하는 시간대 및 말투를 확인할 수 있다. 예로, [표 2]와 같은 방식으로



[그림 2] 실제 사례를 바탕으로 link analysis 걸러지는 데이터들을 도식화 하여 네트워크형으로 연결 지어 나타낸다[그림 3].

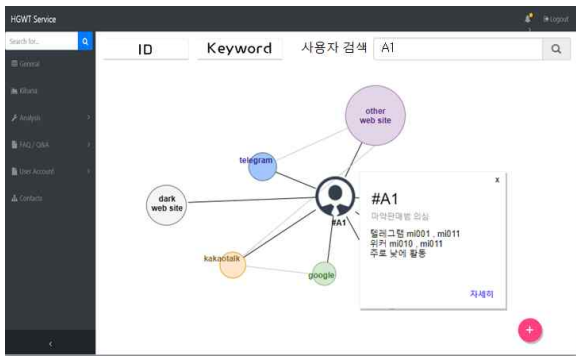
pgp_key	site	user_id	...
A	hskxx....xx.onion	gildong	...
A	haexx....xx.onion	taro12	...
A	qwexx....xx.onion	bela34	...

[표 2] 사용예시 - PGP 키로 질의한 결과



[그림 3] [표 2]를 바탕으로 link analysis

5.2 결과



[그림 4] 메인 화면 예시

메인화면은 단순조회 서비스, 사이트 검색이 가능하고, 실시간으로 올라온 게시 글을 클릭해

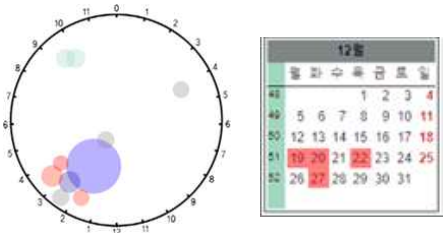
서 해당 사이트로 바로 접속이 가능하다.



[그림 5] 메인 화면에서 사용자 노드를 클릭한 화면 예시

중앙에는 분석 결과를 네트워크로 도식화 시켜서 보여주어 관련된 집단과 정도를 표현한다. 각 노드의 속성과 관계특징은 색상과 모형을 다르게 나타낼 것이다. 동일한 집단으로 묶는 기준은 [표 1]과 같다.

추가적으로 크롤링시 게시물 및 댓글 작성 시간을 수집해내기 때문에, 특정 사용자의 자주 접속하는 요일과 시간대를 유추할 수 있다.



[그림 6] 특정 사용자의 접속시간

III. 결론 및 향후 연구 방향

한국어 중점으로 다룬 것이 없었다. 또한 PGP 키나 비트코인 주소와 같은 내용의 경우, 외국 포럼/시장에서 공동으로 쓰일 수 있기 때문에 보다 많은 주소를 확보할 수 있을 것으로 생각된다. 향후 연구에서는 해당 논문에서 제시한 방법들은 토대로 정확도를 비교해보며, 연관성 분석을 시각화하는데 더욱 중점을 두어 보여줄 것이다. 또한 빠르게 사라지는 서버들 사이에서 크롤링을 통해 모은 데이터들은 수사기관이 과거시점을 조회하는데 중요한 역할을 해줄 수 있다. html을 통째로 가져오기에는 DB가 무거워져서 추가로 얻어낸 정보들은 파싱해내어 저장하는 별도의 기준이 필요해 보인다.

[참고문헌]

[1] 방성수, '인터넷 지하백화점' 다크웹, ChosunBiz, 2017-07-16 07:00, 조선비즈 홈페이지 , http://biz.chosun.com/site/data/html_dir/2017/07/15/2017071501571.html (2017-10-17 방문)

[2] 주성호, '어둠의경로' 다크웹 통한 마약거래…2년간 155명 적발, 뉴스1 홈페이지, 2017-9-27 08:51, http://news1.kr/articles/?3112277 , (2017-10-11 방문)

[3] Pushkar Pathak, Ahmia, https://github.com/ahmia

[4] OSINT, 2016, "Dark web python osint - part one", http://www.automatingosint.com/blog/2016/07/dark-web-osint-with-python-and-onionscan-part-one/, (2017-10-01 방문)

[5] Eric Nunes, Ahmad Diab, Andrew Gunn, Ericsson Marin, Vineet Mishra, Vivin Paliath, John Robertson, Jana Shakarian, Amanda Thart and Paulo Shakarian, "Darknet and deepnet mining for proactive cybersecurity threat intelligence", Intelligence and Security Informatics (ISI), 2016 IEEE Conference on, 2016.

[6] Christopher Ahlberg, Chasing Foxes by the Numbers: Patterns of Life and Activity in Hacker Forums, Blackhat Europe 2016, https://www.blackhat.com/docs/eu-16/materials/eu-16-Ahlberg-Chasing-Foxes-By-The-Numbers-Patterns-Of-Life-And-Activity-In-Hacker-Forums.pdf

[7] 분석에 사용된 한국어 마켓 및 포럼
마켓
hxxp://hixxxxxea5ou4xxy.onion
hxxp://h2xxxxxodqfpzxxc.onion
hxxp://umxxxxxrlqfz7xxw.onion
포럼
hxxp://flxxxxxkmxfnrxxg.onion
hxxp://c2xxxxx6qx6kuxxn.onion
hxxp://khxxxxxlxr3kixxp.onion

최신 TLS 암호화 프로토콜 웹 서버 적용 의무화 방안 연구

김성진*

*고려대학교 컴퓨터정보통신대학원 소프트웨어보안학과

The latest TLS encryption protocol web server research on mandatory application

Sung-Jin Kim*

*Dept of Software Security Engineering, Korea University Graduate

요 약

최근 인터넷과 모바일 기술의 발전으로 웹과 어플리케이션을 이용한 전자상거래, 온라인 업무가 폭넓게 확장되어 가고 있는 반면에 서비스와 기술의 발전에 비해 보안은 상당히 취약하게 노출되어 있는 상황이다. 웹 서버상의 자료를 보호하고 안전한 전자상거래와 업무를 위해서 보안은 선택이 아닌 필수이다. 본 논문은 보안의 대표적인 암호화 프로토콜 방식인 SSL/TLS의 최근 취약점을 소개하고, 국내 웹 서버를 사용하는 웹 브라우저, 어플리케이션을 최신 TLS 암호화 프로토콜로 적용하고 의무화 할 수 있는 연구 방안을 제안한다.

I. 서론

오늘날 인터넷과 모바일 기술의 발전으로 웹 어플리케이션을 이용한 전자상거래와 온라인 업무뿐 아니라 4차 산업시대를 맞아 IT기술이 모든 산업 분야로 확장되어 가고 있다. 하지만 서비스와 기술의 발전에 비해 보안에 대한 대책과 인식은 상당히 취약하다. 지난해 정보보호 응용국제 콘퍼런스(WISA 2016) 논문 발표에 따르면 국내 공기업 291개 중 59%, 100대 사이트 31%가 푸들 취약점을 내포하고 있다[1]. 또한 3년이 지난 2017년 1월 기준, 하트블리드 취약점에 노출된 웹 사이트는 미국이 4만2032개로 가장 많고 한국 1만5380개, 중국 1만4116개, 독일 1만4072개 등이 뒤를 이었다[2]. 인터넷 모바일 기술의 확장에 따른 서버상의 자료를 보호하고 안전한 업무와 개인정보보호를 위해서는 보안은 선택이 아니라 필수사항이다. 국내 공공기관과 기업 그리고 개인의 정보보호를 위한 대표적인 암호화 프로토콜 방식인 TLS(Transport Layer Security)의 최신버전을 웹 서버에 적용하고 의무화 할 수 있는 연구 방안을 제안한다.

II. SSL/TLS

2.1 SSL/TLS 소개

SSL/TLS는 웹사이트와 다양한 어플리케이션에서 보안을 위해 사용하고 있는 대표적인 암호화 방식이다. SSL(Secure Socket Layer)은 인터넷 보안에 대한 관심이 커지면서 1990년대 초 넷스케이프사가 브라우저와 웹 서버 간 암호화 통신을 위해 개발한 암호화 프로토콜이다. 최초 버전인 SSL 1.0은 보안 문제로 발표되지 않았으며 SSL 2.0은 1995년, SSL 3.0은 1996년에 발표되었다. 1999년 SSL 3.1에 해당하고 공식 표준화된 TLS(Transport Layer Security) 1.0이 발표된다. 이후 0.1씩 버전이 올라가며 2006년에 TLS 1.1이 공개되고, 2008년 TLS 1.2로 RFC 5246에 정의되었다. 현재 TLS 1.3의 초안 작업 중에 있다[3].

2.2 SSL/TLS 취약점

1995년 SSL 2.0이 공개적으로 처음으로 발표되었는데, 첫 버전부터 규약 자체의 설계 결함과 실제 구현에서의 오류로 인해 보안 이슈가 끊이지 않고 있다. 최근 3년간의 SSL/TLS의 주요 보안 이슈를 나타낸 표 1을 살펴보면,

표 1. 최근 3년간, SSL/TLS 보안이슈[4]
Table 1. Recent 3 years, SSL/TLS Security Issues[4]

취약점	공격방법
HEIST 공격 (2016. 8)	자바스크립트로 브라우저에 대한 사이드채널 공격을 통해 암호문의 정확한 크기를 알아냄
DROWN 공격 (2016. 3)	SSL 2.0 취약점을 이용하여 TLS 연결 해독 가능
FREAK 공격 (2015. 1)	수출 등급 RSA 사용을 유도하여 brute-force 공격으로 키를 얻어냄
POODLE 공격 (2014. 10)	TLS 연결을 SSL 3.0으로 낮춰 SSL 3.0 취약점을 이용하여 암호문을 해독
Heartbleed 취약점(2014. 4)	Open SSL 라이브러리 오구현으로 웹서버의 시스템 메모리 내용 유출

프로토콜 설계 결함이 있는 SSL 2.0과 SSL 3.0은 사용이 금지되어야 하고, 최신 보안 패치를 적용해야 된다는 것을 알 수 있다.

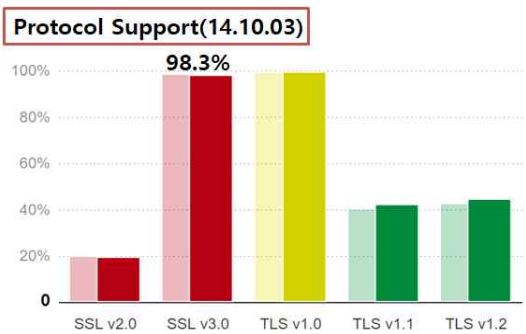


그림 2. SSL/TLS 프로토콜 지원현황(2014.10)[5]
Fig. 2 SSL/TLS Protocol Support Status(2014.10)[5]

2014년 10월 SSL3.0으로 Downgrade 시켜서 자바스크립트를 이용하여 브라우저가 특정 사이트의 인증정보(쿠키, 토큰 등)를 탈취할 수 있으며, 전체시스템에 영향을 미치는 POODLE 공격으로 전 세계적으로 PC 사용자들에게 피해를 입혔다.

그림 2을 살펴보면 POODLE 공격 당시 Trustworthy의 SSL Pulse 프로젝트 조사결과에 따르면, 2014년 10월 기준으로 조사 대상 사이트의 98%(약 15만개)가 보안에 취약한 SSL3.0을 지원하였다.

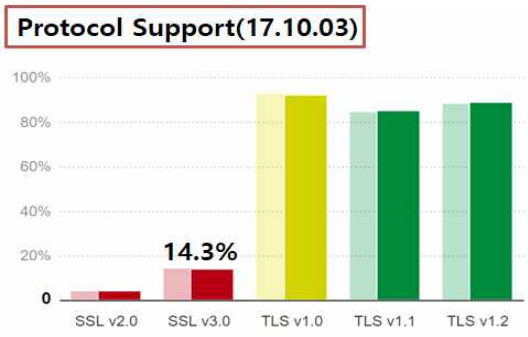


그림 3. SSL/TLS 프로토콜 지원현황(2017.10)[6]
Fig. 3 SSL/TLS Protocol Support Status(2017.10)[6]

그림 3을 살펴보면 현재기준인 2017년 10월, 조사대상 사이트의 14.3%(약 2만개)는 여전히 보안에 취약한 SSL 3.0을 지원하고 있는 실정이다. 최근에는 TLS 1.0과 TLS 1.1조차도 안전하지 않은 것으로 간주되고 있다. 그러므로 TLS 1.2와 그 이후 버전을 사용해야하며 이는 웹 서버가 항상 최신 TLS 버전을 실행해야 함을 의미한다.

3.3 국제동향

하트블리드 취약점이 가장 많았던 미국[2]의 경우, 새로 만들어지는 공공 웹사이트에 HTTPS 암호화 통신이 의무 적용된다. 이 정책에 따라 향후 신규 미국 연방정부 공공 웹사이트는 방문자와 HTTPS 통신을 수행하기 위해 웹 서버에 반드시 SSL 인증서를 적용해야 한다.

기업 사례를 살펴보면, 작년 6월 세계개발자대회(WWDC) 2016에서 애플의 ATS(App Transport Security)를 도입한다고 발표했다. ATS는 iOS 9부터 기본 적용 된 것으로 모든 앱의 인터넷 통신에서 암호화된 HTTPS(HTTP over SSL/TLS) 연결을 사용하도록 강제한다. 현재는 ATS 적용 의무화를 연기하였지만, 새로운 시한을 결정하여 공

지 할 것으로 보인다. 구글은 HTTPS 확산을 위해 미적용 웹페이지에는 '보안상 안전하지 않다(Not Secure)'는 메시지를 표시하고 있으며, 모바일 운영체제인 Android 8.0 Oreo 버전부터는 플랫폼에 SSLv3 프로토콜 지원을 중단하였다. 미국과 글로벌 기업들도 최신 SSL/TLS 프로토콜 설정을 의무화하며 암호화 트래픽 확산을 추진하고 있다.

III. 관련연구

3.1 웹 서버 보안강화 법적 기준 개정

본 논문에서는 웹 서버상의 자료를 보호하고 안전한 전자상거래와 대한민국 국민들의 개인정보보호를 위해 법 개정이 필연적으로 느낀바 표 2와 같이 제안한다.

표 2. 웹 서버 구축의 법률 개정안
Table 2. Law revision of web server construction

현행	저자 개정안
「정보통신망 이용촉진 및 정보보호 등에 관한 법률」 제28조 4. 개인정보를 안전하게 저장·전송 할 수 있는 암호화 기술 등을 보안 조치	「정보통신망 이용촉진 및 정보보호 등에 관한 법률」 제28조 4. 개인정보를 안전하게 저장·전송 할 수 있는 분기별 최신버전 암호화 기술 등을 보안 조치
「개인정보의 기술적/관리적 보호조치 기준」 제6조(개인정보의 암호화) 1. 웹 서버에 SSL(Secure Socket Layer) 인증서를 설치하여 전송하는 정보를 암호화 하여 송/수신하는 기능	「개인정보의 기술적/관리적 보호조치 기준」 제6조(개인정보의 암호화) 1. 웹 서버에 ~~SSL(Secure Socket Layer)~~ 최신버전 TLS(Transport Layer Security)인증서를 설치하여 전송하는 정보를 암호화 하여 송/수신하는 기능

3.2 웹 서버 보안 설정 의무화

웹 서버 최신 TLS 암호화 프로토콜 설정 의무화를 위해 HSTS의 설정도 필수적이다. HSTS(Http Strict Transport Security)은 IETF에서 정의한 표준(RFC6797)의 표준으로 웹브라우저 상에서 세션하이재킹, 프로토콜 다운그레이드 공격 등을 보호하는 웹 보안 정책 보안 메커니즘이다[7]. 최신 TLS버전 기능 구현과 HSTS를 의무화 적용 할 수 있도록 그림4, 5와 같이 설정한다. 그림6은 본 논문에서 제안하는 웹 서버 보안 구성도이다.

```
server {
    listen 80;
    listen 443 ssl;
    server_name site.com
.....
생략
.....
ssl on;
ssl_certificate /etc/pki/tls/certs/site.com.crt;
ssl_certificate_key /etc/pki/tls/private/site.com.key;
ssl_session_timeout 5m;
ssl_protocols all TLSv1.2;
ssl_ciphers 'EECDH+AESGCM:EDH+AESGCM:
AES256+EECDH:AES256+EDH';
ssl_prefer_server_ciphers on;
....
```

그림 4. 최신버전 TLS 적용 예시
Fig. 4 Latest version TLS application example

```
Strict-Transport-Security:
max-age=31536000; --해당 설정 31536000초(1년)
includeSubDomains; --모든 하위 도메인까지 포함
preload --http 접속시, 접속하지 않고 바로 https로 접속
```

그림 5. HSTS 설정 예시
Fig. 5 HSTS setup example

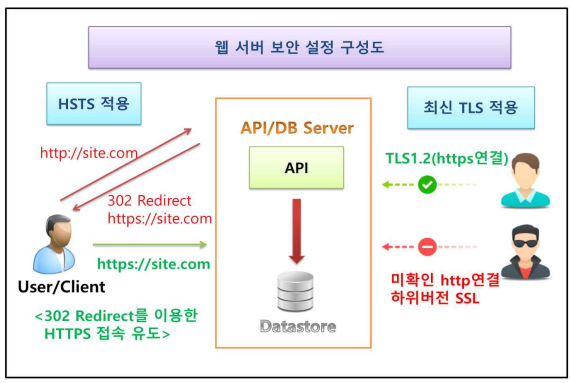


그림 6. 웹 서버 보안 설정 구성도
Fig. 6 Web server security settings diagram

위와 같이 웹 서버 설정 시 여러 클라이언트들은 서버와의 통신을 위해 동일하게 설정할 수밖에 없기 때문에 웹 서버에 HSTS와 최신버전 TLS 설정이 중요하다. 이러한 설정을 위해 「개인정보의 기술적/관리적 보호조치 기준」제6조 개정안을 활용한다. 각기 다른 제조사의 웹 서버마다 적용되는 최신버전 TLS 인증서와 HSTS 설정 가이드를 만들어 배포하고, 먼저 공기업, 준정부기관, 기타공공기관 등으로 우선 확대가 시급해 보인다.

IV. 결론

본 논문에서는 국내 공공기관과 웹 사이트의 심각한 보안 취약성을 조사결과 대안으로 국내 웹 서버를 최신 TLS 암호화 프로토콜 의무화 적용을 위한 연구 방안을 제안해보았다. 글로벌 기업들도 SSL/TLS의 적용 의무화가 확산되고 있는 이 시점에 우리나라도 조속한 대처로 보안 강화에 힘쓰고, 서비스 호환에도 어려움이 없어야 할 것이다.

[참고문헌]

[1] Jongmin Jeong, Hyunsoo Kwon, Hyungjune Shin, Junbeom Hur, "A Practical Analysis of TLS Vulnerabilityies in Korea Web Environment", 17th World Conference on Information Security Applications (WISA), August, 2016

[2] The Shodan analysis released over the weekend is part of the search engine's Heartbleed Report, January, 2017

[3] Wikipidea, "Transport Layer Security" https://en.wikipedia.org/wiki/Transport_Layer_Security, 2017

[4] MONTHLY SOFTWARE ORIENTED SOCIETY 09, pp. 34-43, September, 2016

[5][6] Qualys SSL Labs, SSL Pulse Protocol Support, October, 2017

[7] Hodges, Jeff; Jackson, Collin; Barth, Adam "Section 5.2. HSTS Policy", RFC 6797, IETF, November, 2012

2017 한국정보보호학회 동계학술대회

C5 네트워크보안(II)

서 정 택 (순천향대)

- **SDN 베어메탈 스위치를 위한 네트워크 운영체제의 설치 취약점 연구**
  유봉열, 양경식, 석상윤, 유혁 (고려대학교)
- **VANET 환경에서 해시기반 서명 연구**
  이경모, 이경현 (부경대학교)
- **AEG(Automatic Exploit Generation)에 대한 연구 동향**
  이재융, 석재혁, 이동훈 (고려대학교)
- **커넥티드카(Connected car)통신을 위한 국제 표준화 동향 연구 : 미국, 유럽 표준을 중심으로**
  주경호, 이동훈 (고려대학교)

SDN 베어메탈 스위치를 위한 네트워크 운영체제의 설치 취약점 연구

유봉열 양경식 석상윤 유혁

고려대학교 컴퓨터학과

A Vulnerability Study on Network Operating System Installation for SDN Bare Metal Switches

Bongyeol Yu Gyeongsik Yang Sangyoon Seok Chuck Yoo

Dept. of Computer Science and Engineering, Korea University.

요 약

오픈 네트워킹을 지향하는 SDN 기반의 네트워크에서 전송평면까지 프로그래밍 가능성(programmability)을 부여하기 위한 베어메탈 스위치가 주목받고 있으며 다수의 상용 데이터 센터에 적용되고 있다. 특히 베어메탈 스위치는 사용자가 임의의 운영체제를 선택할 수 있지만 공개 인터페이스 및 프로그래밍이 가능하므로 기존의 닫혀있는 하드웨어 중심의 스위치에 비해 보안 취약성이 매우 높다. 그러나 기존 SDN 네트워크에서의 보안 연구는 베어메탈 스위치 수준의 보안 취약점을 다룬 사례가 매우 드물다. 본 논문은 스위치 운영체제를 설치하기 위한 오픈소스 운영체제인 Open Network Installation Environment를 기반으로 네트워크 운영체제를 설치하는 과정을 분석하고, 간단한 네트워크 공격으로 비정상적인 운영체제가 설치되는 취약성을 제안한다. 특히 KVM 기반 가상화 환경에서 해당 취약성을 DHCP 스푸핑 및 DNS 스푸핑을 기반으로 공격하여 재현한다. 이를 통해 프로그래밍이 가능한 베어메탈 스위치에서 시스템 레벨 소프트웨어에 대한 새로운 수준의 보안 연구가 필요함을 보인다.

I. 서론

전통적인 네트워크 장비는 하드웨어와 벤더 종속적인 NOS(Network Operating System)가 수직적으로 묶여진 형태를 가지고 있다. 이러한 전통적 구조에서는 새로운 네트워크 기능의 추가가 어렵고, 네트워크를 운용하는 데에 설비투자비용(CAPEX) 및 영업비용(OPEX)이 크게 증가한다. 네트워크 자원의 폐쇄성을 해결하고자 오픈 네트워킹 및 SDN (Software-defined Networking) 패러다임이 도입되었고, 하드웨어를 임의 선택하고 사용자가 네트워크 및 서비스에 적합한 소프트웨어를 임의로 선택 및 설치할 수 있는 베어메탈 스위치가 주목받고 있다. [1] 이를 통해 새로운 네트워크 기능 추가가 용이해지고 보다 유연한 네트워크 구조의 실현이 가능하다. 범용 서버를 제어하기 위한 운영체제가 존재하듯, 베어메탈 스위치 또한 스위치를 구동하고 운용하기 위한 운영체제(NOS)를 필요로 한다. 현재 Big switch, Cumulus network 등 다양한 제조사들이 NOS를 출시하고 있고, 오픈소스 프로젝트로는 리눅스 기반의 ONL(Open Network Linux) [2]이 있다.

최초로 베어메탈 스위치를 구동하면 사용자가 구동하는 서비스 및 환경에 적절한 NOS를 선택하여 설치하게 된다. 이를 위해 NOS를 설치하기 위한 오픈소스 소프트웨어인 ONIE (Open Network Installation Environment) [3]가 널리 활용된다. ONIE는 스위치 관리자의 편의를 위해 스위치와 연결되어 있는 물리 저장소를 통한 NOS 설치뿐만 아니라, 네트워크를 통한 NOS의 설치 기능도 제공하며, 순차적으로 로컬 및 네트워크에서 NOS 설치 프로그램(NOS Installer)을 탐색한다. 하지만 이러한 NOS 설치 과정에

서 공격자에 의해 위조된 NOS를 설치하게 된다면, 네트워크 운용자가 해당 스위치로 네트워크를 운용하는 상황에 심각한 위험이 발생한다. 그러나 현재까지 베어메탈 스위치 및 NOS와 관련된 보안 연구는 [4]에서 언급된 취약성 제기 외에 진전된 바가 없다.

본 논문은 ONIE가 NOS를 설치하는 과정을 분석하고 해당 과정에서 발생할 수 있는 취약점 및 그 취약점을 실제로 재현한다. 그리고 해당 취약점에 대한 대응 방안을 논의해본다.

II. ONIE의 NOS 설치 과정

본 절에서는 ONIE가 네트워크에서 NOS Installer를 탐색하고 설치하는 과정을 살펴본다. ONIE는 NOS 설치를 위한 Installer 탐색전에 DHCP를 통해 네트워크 인터페이스 주소들을 설정한다. DHCP 서버가 스위치의 IP를 할당하면 ONIE는 Installer 탐색 과정을 수행하며, 스위치의 물리 저장소 또는 네트워크의 파일 서버를 순차적으로 검색한다.

1) 물리 저장소를 통한 탐색 과정

ONIE는 Installer를 탐색할 때, 몇 가지 이름 규칙을 통해 NOS Installer를 식별한다. ONIE가 식별하는 Installer의 이름 규칙은 표 1과 같은데, NOS가 설치될 베어메탈 스위치의 하드웨어 환경과 적합한 Installer를 찾는다.

ONIE는 베어메탈 스위치에 연결되어 있는 저장소들에게 각각의 이름을 가지는 Installer의 전송을 요청한다. 해당 이름의 Installer가 있을 경우, Installer를 전송 받아 스위치에 설치한다.

순서	NOS Installer 이름
1	onie-installer-<arch>-<vendor>_<machine>-r<machine_revision>
2	onie-installer-<arch>-<vendor>_<machine>
3	onie-installer-<cpu_arch>-<switch_silicon_vendor>
4	onie-installer-<arch>
5	onie-installer

표1. NOS Installer 이름

2) 네트워크의 파일 서버를 통한 탐색 과정

물리 저장소에서 NOS Installer를 찾지 못할 경우, ONIE는 네트워크에 NOS Installer를 요청한다. ONIE는 DHCP 서버로부터 할당 받은 IP 정보의 DNS 서버에 onie-server라는 도메인 네임의 주소를 요청한다. 이 때, onie-server가 네트워크 운영자가 구축한 파일서버의 도메인 네임이 되어야 한다.

onie-server의 주소를 DNS서버로부터 받은 ONIE는 해당 서버에 NOS Installer 탐색을 하는데, 이 과정에서도 표2와 같이 정해진 규칙을 가지고 수행한다. 표2의 순서대로, 해당 경로의 파일 명을 tftp를 이용하여 서버에 요청한다. 서버에 ONIE가 찾는 Installer가 있을 경우, tftp를

순서	경로	NOS Installer 이름
1	onie-server/<switch_mac>/	onie-installer-<arch>-<vendor>_<machine>-r<machine_revision>
2	onie-server/C0A87A59/	onie-installer-<arch>-<vendor>_<machine>-r<machine_revision>
3	onie-server/C0A87A5/	onie-installer-<arch>-<vendor>_<machine>-r<machine_revision>
4	onie-server/C0A87A/	onie-installer-<arch>-<vendor>_<machine>-r<machine_revision>
5	onie-server/C0A87/	onie-installer-<arch>-<vendor>_<machine>-r<machine_revision>
6	onie-server/C0A8/	onie-installer-<arch>-<vendor>_<machine>-r<machine_revision>
7	onie-server/C0A/	onie-installer-<arch>-<vendor>_<machine>-r<machine_revision>
8	onie-server/C0/	onie-installer-<arch>-<vendor>_<machine>-r<machine_revision>
9	onie-server/C/	onie-installer-<arch>-<vendor>_<machine>-r<machine_revision>
10	onie-server/	onie-installer-<arch>-<vendor>_<machine>-r<machine_revision>
11	onie-server/	onie-installer-<arch>-<vendor>_<machine>
12	onie-server/	onie-installer-<cpu_arch>-<switch_silicon_vendor>
13	onie-server/	onie-installer-<arch>
14	onie-server/	onie-installer

표2. 네트워크 파일 서버에 NOS Installer 요청

통해 Installer를 전송받고 NOS를 설치한다.

하지만, 위의 과정을 통해서도 onie-server에서 원하는 NOS Installer를 찾지 못했을 경우, 마지막으로 DHCP를 통해 할당받은 Gateway 주소로 표2의 과정을 tftp를 요청하여 NOS Installer를 탐색한다.

물리 저장소를 통한 탐색과정과 네트워크의 파일 서버를 통해서 탐색한 과정 모두에서 NOS Installer를 발견하지 못했을 때, ONIE는 20초 후 DHCP 서버에 IP 요청 단계부터 반복한다.

III. ONIE의 NOS 설치과정의 취약점

ONIE의 NOS 설치 과정은 ONIE의 Install 모드에서 자동으로 수행된다. ONIE가 네트워크의 파일 서버를 통해 NOS를 설치하게 될 때, 서버의 무결성에 대한 인증 없이 Installer의 파일 이름과 경로명으로만 Installer를 식별한다.

다시 말해, 네트워크 관리자가 의도(구성)한 정상적인 파일 서버가 아니라 제3의 서버를 통해서도 설치 프로그램의 이름과 경로만 규칙에 맞는다면 스위치에 설치될 수 있음을 의미한다.

이에 기반한 취약성을 보이기 위해 본 연구진은 이미 잘 알려져 있는 DHCP 스푸핑 또는 DNS 스푸핑 공격을 통해, ONIE가 네트워크 관리자의 정상 파일서버가 아니라 제3자의 서버로부터 NOS를 설치하는 공격을 보인다. 특히 본 상황을 KVM 기반 가상화 환경에서 재현한다.

1) DHCP 스푸핑 공격

먼저, DHCP 스푸핑 공격을 통해 ONIE가 할당받는 Gateway와 DNS 서버의 주소를 제3자의 주소로 변경한다. ONIE는 앞서 설명했던 것처럼 Installer 탐색 과정 전에 DHCP 서버로부터 IP를 할당 받게 된다. 이 때 공격자에 의해 스푸핑된 DHCP 서버를 통해 IP정보를 할당 받게 된다. 상세한 공격 과정은 다음과 같다.

(1) 공격자는 공격자의 서버에 FTP 서버를 만들고 공격자에 의해 수정된 NOS Installer를 해당 서버에 위치시킨다.

(2) 공격자는 DHCP Spoofing을 위해 기존에 동작하고 있는 DHCP 서버에 DoS 공격을 하여 기존 서버를 마비시킨다.

(3) 스위치가 DHCP를 통해 IP 할당을 요청했을 때, 공격자의 스푸핑된 서버는 공격자를 가리키는 Gateway정보와 DNS 정보를 알려준다.

(4) 스위치는 잘못된 정보의 IP를 할당 받고 NOS Installer 탐색 과정을 수행한다.

(5) 스위치는 공격자의 DNS를 통해 다른 FTP 서버를 가리키거나 Gateway의 FTP 서버에서 공격자의 Installer를 다운 받도록 한다.

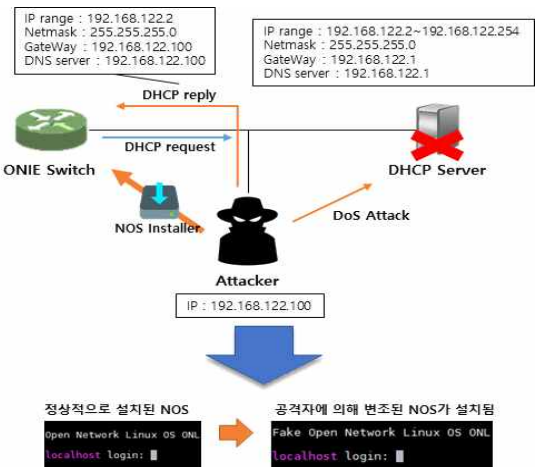


그림 1. DHCP 스푸핑 과정과 설치된 공격자의 NOS

위 과정을 통해 그림1과 같이 공격자에 의한 NOS가 설치됨을 확인할 수 있다.

2) DNS 스푸핑 공격

DNS 스푸핑 공격을 통해서도 해당 스위치에 공격자의 NOS를 할 수 있다. ONIE의 NOS Installer 탐색 과정을 살펴보면, ‘onie-server’라고 지정된 도메인 네임에 FTP 요청을 한다. 이러한 과정을 이용하여 DNS 스푸핑 공격을 수행할 수 있고 수행 과정은 다음과 같다.

(1) 공격자는 공격자의 서버에 FTP 서버를 만들고 공격자에 의해 수정된 NOS Installer를 해당 서버에 위치시킨다.

(2) 공격자는 APR poisoning 공격을 통해 해당 스위치와 Gateway간의 데이터를 스니핑한다.

(3) 공격자는 'onie-server'라는 도메인에 대한 요청에 공격자의 서버의 주소로 응답하도록 DNS 스푸핑을 수행한다.

(4) 스위치가 'onie-server'의 주소를 DNS 서버에 요청할 때, 공격자는 중간에서 공격자의 파일 서버 주소를 알려준다.

(5) 스위치는 공격자의 파일 서버에서 공격자의 NOS Installer를 다운받아 설치한다.

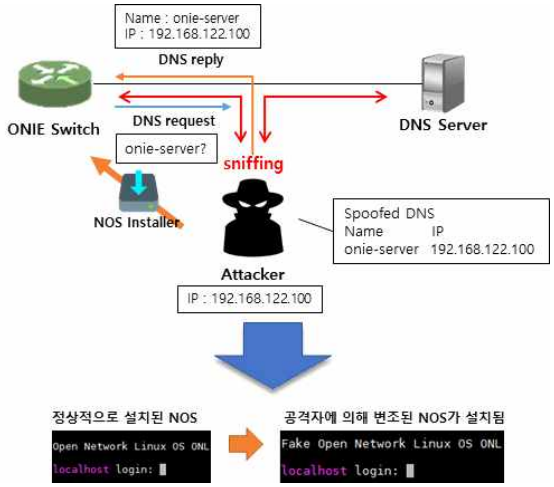


그림 2 DNS 스푸핑 과정과 설치된 공격자의 NOS

위 과정을 통해 그림2와 같이 스위치는 공격자의 NOS를 설치하게 됨을 확인할 수 있다.

IV. 예상 피해 및 예방

베어메탈 스위치는 오픈 네트워크를 위해 도입 되었고 스위치의 모든 동작을 임의 제어가 가능하다. 하지만, 이러한 소프트웨어를 제어하는 운영체제부터 공격자로부터 조작된다면, 언제든지 공격자가 해당 스위치의 동작을 제어할 수 있게 된다. 이럴 경우 네트워크의 흐름 제어뿐만 아니라 네트워크 마비에 이르는 심각한 상황을 초래할 수 있다.

이러한 공격을 막기 위해서는 정해진 순서와 이름 규칙으로만 NOS를 설치하는 ONIE의 방식을 사용자가 제어할 수 있는 방법으로 변경해야 한다. 그리고 NOS Installer의 파일 서버에 대한 인증 과정이 없기 때문에, 공격자의 파일 서버에서 NOS Installer를 다운 받아 올 수 있다. 또한, 공격자에 의해 변경된 NOS의 Installer가 파일 이름만 같다면 별도의 인증 없이 설치되기 때문에 파일 서버의 인증 외에 NOS Installer를 인증할 수 있는 프로토콜의 도입이나 하드웨어 기반의 Secure boot [3] 등의 추가적인 모듈 인증 메커니즘이 필요하다.

V. 결론

본 논문에서는 베어메탈 스위치에서 NOS를 설치하기 위한 플랫폼인 ONIE의 NOS 설치 과정을 살펴보고 그 과정 중에 발생할 수 있는 취약점에 대해 분석하였다. 특히 DHCP 스푸핑과 DNS 스푸핑에 의해 공격자의 NOS가 쉽게 베어메탈 스위치에 설치됨을 확인할 수 있었다.

또한 베어메탈 스위치와 NOS는 아직 도입 단계에 있으며 그 보안성에 대해서는 논의된 바가 거의 없다. 따라서 본 논문의 연구 결과를 통해 NOS 설치 과정에서의 기본적인 공격에도 취약점이 있음을 확인할 수 있었고, 베어메탈 스위치의 보안성 향상에 대한 많은 논의가 필요할 것으로 보인다.

VI. Acknowledgement

이 논문은 2017년도 정부의 재원으로 국가보안기술연구소의 지원을 받아 수행된 연구임 (2017-089, SDN 스위치 OS 분석 기술 연구)

[참고문헌]

[1] Sherwood, Rob. "Tutorial: White box/bare metal switches." Open Networking User Group meeting, New York. 2014.

[2] Open Network Install Environment. [Online]. Available: https://github.com/opencomputeproject/onie [retrieved: October, 2017]

[3] Open Network Linux. [Online]. Available: https://github.com/opennetworklinux/ONL [retrieved: October, 2017]

[4] Pickett, G. "Staying persistent in software defined networks." Black Hat Briefings (2015).

VANET 환경에서 해시기반 서명 연구

이경모*, 이경현**

*부경대학교 정보보호학협동과정

**부경대학교 IT융합응용공학과

A Study on Hash-based Signature in VANET Environment

*Interdisciplinary Program of Information Security, Graduate School, Pukyong National University.

**Dept of IT Convergence and Application Engineering, Pukyong National University.

요 약

디지털 서명은 진정성, 무결성, 부인방지를 제공하는 중요한 기능을 한다. 상용 디지털 서명체계의 대부분은 소인수분해 및 이산로그 문제를 해결하는 어려움에 기반한다. 하지만 양자컴퓨팅을 사용한 Shor 알고리즘을 이용하면 위와 같은 문제가 다항시간 내 해결 가능하므로 양자컴퓨팅 활성화 이후에도 안전한 서명체계가 필요하게 되었다. 또한 이러한 Ad-hoc 환경으로 대표되는 VANET 환경 등 디바이스가 적은 리소스를 가지는 환경에서도 사용가능해야 한다. 본 논문에서는 다양한 해시기반 서명체계를 소개하고 각 서명체계별 성능구성 파라미터를 제시하며 이를 기반으로 VANET 등 자원 제한적 환경에서의 해시기반 서명체계 적용 가능성을 제시한다.

I. 서론

디지털 서명은 데이터의 진정성(authenticity)과 무결성(integrity)및 부인방지(non-repudiation)를 제공하고 식별 및 인증 프로토콜에 사용되고 있다[1]. 일반적으로 공개키 암호시스템 기반의 서명으로 RSA[2], DSA[3], ECDSA[4]이 사용된다. [2][3][4]의 안전성은 큰 합성수를 소인수분해하는 것과 이산로그 문제를 해결하는 어려움에 기반을 두고 있다.

현대의 공개키 암호시스템에서 사용하는 수학적 어려운 문제에 대해 Shor[5]와 Grover[6]는 각각 양자 컴퓨터를 사용하여 효율적으로 해결 할 수 있는 알고리즘을 제시하였다. Shor[5]는 양자컴퓨터가 소인수분해와 이산로그 문제 해결의 어려움에 기반한 공개키 암호시스템의 문제들을 다항시간 내 문제해결이 가능함을 입증하였다. 따라서 언급한 기존의 수학적 문제에 기반한 서명체계를 대체할 새로운 서명체계가 필요하게 되었다. 한편으로는 다양한 분야의 Ad-hoc 환경 중 하나인 VANET(Vehicle Adhoc Network)에서도 차량에서 사용할 적은 리소스와 빠른 서명 및 검증 속도를 가진 안전한 서명기법이 요구되었다. 본 논문의 구성은 다음과 같다. 2장에서는 양자컴퓨터 활성화 이후에도 사용 가능한 서명체계의 종류 및 특징을 소개하고 3장 본론에서는 VANET 환경에서 적용 가능한 해시기반의 서명체계를 소개한다. 4장에서는 해시기반의 서명체계별 파라미터에 따른 성능구성을 제시한다. 5장에서는 VANET 환경에서의 해시기반 서명의 적용 가능성을 제시한다.

Ⅱ관련 연구

2장에서는 양자 컴퓨팅 환경이 활성화 된 이후 현재 공개키 기반의 서명체계를 대체하여 사용 가능한 서명체계 후보에 대해 소개한다.

◆ 격자기반 암호

격자기반 암호[7]은 격자 문제를 기반으로 하는 암호시스템으로써 키 설정 알고리즘은 상대적으로 간단하고 효율적이며 서명은 다변수 기반 암호 및 코드 기반 암호와 비교하여 빠르며 짧은 서명을 가진다.

◆ 다변수 기반 암호

다변수 기반 암호[7]은 유한체 상에서 다변수 함수의 해를 구하는 것이 NP-Hard 문제임을 기반으로 하며 매우 짧은 서명길이와 일반적으로 짧은 공개키를 사용한다.

◆ 코드 기반 암호

코드 기반의 암호[8]는 송신자가 메시지에 의도적인 오류를 만들어 전송할 때, 수신자는 생성 매트릭스를 알고 있는 수신자만이 오류탐지 후 복원이 가능하고 이를 모르는 수신자는 메시지를 복원하는 것이 NP-hard 문제임을 기반으로 한다.

Ⅲ 본론

해시기반 서명은 다른 대체 가능한 후보 체계와 다르게 큰 정수 연산을 필요로 하지 않고 단지 안전한 암호학적 해시만을 요구하며 자원 제약적인 디바이스에서도 활용이 가능하다[9]. 이 장에서는 제한적 자원을 가진 디바이스에서도 사용가능한 해시 기반의 서명체계를 제시한다.

◆ BiBa One-time Signature(BiBa OTS)

Perrig은 "Bins and Balls"의 의미를 가진 일회성 서명 체계를 제안하였다[11]. BiBa 서명 체계의 주요 이점은 빠른 검증속도 및 작은 서명크기이고 상대적으로 서명생성 오버헤드 및 공개키 크기가 크다는 특징이 있고 주요 아이디어는 생일 역설(birthday paradox)을 이용한다.

◆ Lamport One-time Signature(OTS)

1979년 Lamport[12]는 일방향 해시함수에 기반한 일회성 서명체계를 발표하였다. 이 때 공개키는 비밀 입력값에 대해 두개의 해시 출력값으로 구성되며, bit 0에 서명할 경우 첫 번째 출력의 Pre-image를 공개하고, bit 1에 서명할 경우 두 번째 Pre-image를 공개하였다. 비밀키는 오직 한번만 사용할 수 있다는 특징이 있다.

◆ Merkle Signature Scheme(MSS)

Merkle[13]은 현재 MSS라 불리는 새로운 서명 체계를 제안하였다. 즉 높이 h의 이진 해시 트리를 사용하여 2^h개의 OTS 키 쌍을 인증하였고 OTS 비밀키는 MSS의 비밀키로, OTS 공개키의 해시값은 각 리프노드로, 계산된 트리의 루트노드는 MSS의 공개키로 각각 사용하였다. 총2^h개의 메시지에 서명이 가능하였고 실제로 [9]에서는 AVR 플랫폼을 기반으로 MSS를 구현하였다.

◆ Winternitz OTS(WOTS)

Mekle의 최초 발표 후 Lamport OTS[12]를 개선한 WOTS[13]가 발표되었다. 메인 아이디어는 해시함수 f를 사용하며 서명할 비트의 수 t에 대해 $w=2^t$로 설정하고 $Y=f^{w-1}(x)$를 공개키로 공개하고 서명은 서명하고자 하는 메시지에 따라 함수의 반복을 결정하여 서명을 메시지와 함께 공개한다. 검증은 서명으로부터 검증 값을 계산한 뒤 공개키 값과 비교하여 서명을 검증한다.

◆ Winternitz OTS$^{\$}$($WOTS^{\$}$)

이후 Johannes Buchmann가 WOTS의 변형된 형태인 $WOTS^{\$}$[14]를 발표하였다. $WOTS^{\$}$ 메인 아이디어는 입력값 x에 대해 키 k를 사용하여 i번 반복한다는 것이다.

◆ Winternitz OTS^{+}($WOTS^{+}$)

이후 발표된 $WOTS^{+}$[15]는 비트마스크를 사용하여 해시 함수의 Collision Resistant 성질을 요구하던 기존 WOTS[13]에서 PRF(Pseudo random Function)와 Second-Preimage Resistant의 속성을 요구한다는 차이점이 있다.

◆ Hash to Obtain Random Subset(HORS)

HORS[11]은 BiBa OTS[11]에서 업데이트 되었다. HORS와 유사한 Bos and Chaum[16]에서는 해시함수의 일방향성(One-wayness)의 성질과 집합 T에 대해 랜덤한 부분집합을 선택하는 전단사 함수 S (Byjective function)를 사용하였다. 이후 HORS는 [16]의 효율성을 개선하기 위해 함수 S를 해시함수 H로 변경하여 Subset-Resilience 특성과 일방향 함수 f을 이용하여 서명을 구성하였다.

◆ HORST

HORST[11]는 HORS의 공개키 크키와 서명의 크기를 줄이기 위해 실행시간을 Trade-off 및 트리의 형태로 구성하였다. HORST에서는 HORS의 공개키 크기 nt를 n비트로 감소시키면서 실행시간이 증가하도록 구성하였다.

◆ SPHINCS

SPHINCS[16]은 총 높이 h로 구성된 하이퍼트리 구조로써 각 계층의 높이 h/d로 구성된 d 계층을 가진다. 내부적으로는 $WOTS^+$[15]와 FTS(Few time Scheme)인 HORST[11]를 사용한다.

◆ XMSS

XMSS[17]과 MSS[13]의 주요 차이는 XMSS의 경우 OTS 대신 $WOTS^+$[15]을 사용하고 $WOTS$ 공개키를 압축하기 위해 $l-tree$로 리프노드를 구성하며 $l-tree$ 및 트리내부에 리프 구성을 위한 해시함수 계산 전 비트마스크를 적용한다는 점이다. SPHINCS[16]와 차이점은 XMSS의 경우 Stateful 방식이라는 점과 하이퍼트리가 아닌 단일 트리라는 점이다.

Ⅳ성능 결정 파라미터

본 장에서는 VANET 환경에서의 해시기반 서명체계 구성을 위해 각 서명체계별 성능구성 파라미터를 소개한다.

WOTS[13]에서는 w를 사용하여 한 번에 서명할 수 있는 비트 수를 결정하였다. w는 WOTS 계열의 서명체계인 [13][14][15]에서 해시함수의 반복 횟수를 결정하며 값을 크게 하면 서명이 짧아지는 반면 키 생성과 서명 검증 속도가 느려지는 특성을 갖는다.

<표1> SPHINCS-256 Parameters (for 128-bit Post-quantum security)

이름	값	의미
n	256	bitlength of hashes in HORST and WOTS
m	512	bitlength of the message hash
h	60	height of the hyper-tree
d	12	layers of the hyper-tree
w	16	Winternitz parameter used for WOTS signatures
t	2^{16}	number of secret-key elements of HORST
k	32	number of revealed secret-key elements per HORST signature

표<1>의 경우 SPHINCS-256의 파라미터 SET으로 서명의 크기는 41KilloByte(KB), 공개키와 비밀키의 크기는 각각 1056 Byte, 1088 Byte를 가진다. 이후 제안된[18]에서는 키 쌍의 크기는 기존 SPHINCS-256의 1KB에서 32-64 Byte로, 서명의 크기는 30KB까지 감소되고 서명 및 검증시간도 단축 가능함을 제시하였다.

<표2> 서명기법별 성능 처리시간 및 길이

플랫폼	Intel i5
서명체계	XMSS (w_{16},H_{20})
서명 길이	2500 Byte
공개키 길이	64 Byte
서명시간	7 ms
검증시간	0.52 ms

표<2>에서 XMSS는 [17]의 성능분석을 참고하여 작성하였다. 위 표에 따라 서명검증 시간은 XMSS의 경우 WAVE 표준에서의 ECDSA(4.2ms)보다 효율적이였다.

Ⅴ 결론

본 논문에서는 각 해시기반 서명기법의 특징

과 성능 결정 파라미터 및 Ad-hoc 환경 중 VANET에서의 해시기반 서명 체계의 적용 가능성에 대해 알아보았다. 향후 VANET을 비롯한 Ad-hoc 환경에서 해시기반 서명 체계의 도입시 기초 자료로써 본 논문의 활용을 기대한다.

Acknowledgement

이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2017-0-00156, 블록체인의 정보보호 프레임워크 및 평가 방법 개발)

[참고문헌]

[1] D.S.Milojicic at al. “Peer to Peer Computing”, HP Laboratories Palo Alto HPL-2002-57, March, 2002.

[2] R. L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM 21(2):120 - 126, 1978.

[3] T. Elgamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In Advances in Cryptology - CRYPTO ’84, number 196 in LNCS, pages 10 - 18. Springer, 1985.

[4] D. Johnson and A. Menezes. The Elliptic Curve Digital Signature Algorithm (ECDSA). Technical Report CORR 99-34, University of Waterloo, 1999.

[5] P. W. Shor. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. In Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1994), pages 124 - 134. IEEE Computer Society Press, 1994.

[6] Lov K. Grover. A fast quantum mechanical algorithm for database search. In Proceedings of the Twenty-Eighth Annual Symposium on the Theory of Computing, pages 212{219, New York, 1996. ACM Press.

[7] Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, Daniel Smith-Tone, NISTIR 8105: Report on Post-Quantum Cryptography

[8] Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. volume 8042 of Lecture Notes in Computer Science, pages 40-56. Springer, 2013.

[9] Rohde, Sebastian, et al. "Fast hash-based signatures on constrained devices." CARDIS. Vol. 8. 2008. (Vol. 8, pp. 104-117).

[10] Hülsing, A., Gazdag, S. L., Butin, D., & Buchmann, J. Hash-based Signatures: An Outline for a New Standard.

[11] Reyzin, Leonid, and Natan Reyzin. "Better than BiBa: Short one-time signatures with fast signing and verifying." Australasian Conference on Information Security and Privacy. Springer, Berlin, Heidelberg, 2002. p. 144-153.

[12] Lamport, Leslie. Constructing digital signatures from a one-way function. Vol. 238. Palo Alto: Technical Report CSL-98, SRI International, 1979.

[13] Merkle, Ralph C. "A certified digital signature." Conference on the Theory and Application of Cryptology (pp. 218-238). Springer, New York, NY, 1989..

[14] Buchmann, Johannes A., et al. "On the Security of the Winternitz One-Time Signature Scheme." Africacrypt 11 (2011): 363-378.

[15] Hülsing, Andreas. "W-OTS+-Shorter Signatures for Hash-Based Signature Schemes." Africacrypt 7918 (2013): 173-188.

[16] Bernstein, Daniel J., et al. "SPHINCS: practical stateless hash-based signatures." Annual International Conference on the Theory and Applications of Cryptographic Techniques(pp. 368-397). Springer, Berlin, Heidelberg, 2015.

[17] Buchmann, Johannes, Erik Dahmen, and Andreas Hülsing. "XMSS-a practical forward secure signature scheme based on minimal security assumptions." Post-Quantum Cryptography (2011): 117-129.

[18] Aumasson, Jean-Philippe, and Guillaume Endignoux. "Improving Stateless Hash-Based Signatures."

AEG(Automatic Exploit Generation)에 대한 연구 동향

이재융*, 석재혁*, 이동훈*
*고려대학교 정보보호학과

Research Trends on AEG

Jae-Yung Lee*, Jae Hyuk Suk*, Dong Hoon Lee*
*Korea University, Graduate School of Information Security

요 약

AEG은 취약점 분석 및 익스플로잇 과정을 자동화시킨 기술이다. 이는 Fuzzer와 비슷한 개념이지만 정확도가 더 뛰어나며 취약점을 이용하여 익스플로잇 과정까지 자동화한다는 점에서 Fuzzer와 차이가 있다. 본 논문에서는 AEG의 연구 동향 파악 및 해당 기술의 이론과 구현에 대해 설명하고, 한계점을 언급함으로써 향후 연구 방향에 대해서도 다루도록 한다.

I. 서론

현대 사회에서 컴퓨팅파워가 발전하면서 사람들은 기계에 점점 의존하게 된다. 이에 따라 수동으로 처리되던 일들이 기계에 의해 자동으로 처리할 수 있는 경우가 많다. 기계는 하드웨어와 소프트웨어로 구성된 하나의 시스템에 의해서 동작한다. 그리고 소프트웨어는 하드웨어를 제어하는 등 시스템에서 중요한 역할을 한다.

그러나 기존의 소프트웨어는 기능성에 더 많은 초점을 두고 있으며 상대적으로 보안성 측면에서는 많이 고려되지 않기 때문에 취약점이 발생하는 경우가 종종 있다. 공격자는 이러한 취약점을 악용해 악성코드를 만들어 사용자가 실행하게 만들거나, 직접 시스템에 침투하여 시스템의 가용성을 크게 저하시키거나 중요한 정보를 탈취 할 수 있기 때문에 문제가 된다. 따라서 소프트웨어의 취약점을 찾는 것은 매우 중요하다고 할 수 있다.

소프트웨어 악성코드 탐지 방안에 관한 이전의 연구는 기계학습을 이용하였다. 이 방법은 악성코드 데이터 셋을 학습데이터로 사용하여 각각의 특징(feature)을 추출한 후 이를 분류기에 학습시키고, 새롭게 유입된 프로그램의 특징을 분류기에 대입하여 악성코드 여부를 판단한다. 그러나 이 방법은 많은 양의 학습 데이터가 요구되고 변종이 아닌 신종 악성코드에 대해서는 탐지율이 떨어진다는 한계점이 있다.

위와 같은 이유로 신종 악성코드는 상세 분석이 요구될 수 있으나, 이는 기본적으로 시간이 많이 소요되며 수동의 분석과정 자체가 쉽지 않다. 소프트웨어의 취약점을 정확히 분석하고 패치하기 위하여 필요한 기술 중 하나가 AEG이며 이는 취약점 분석을 자동화시킨 기술이다. 이를 실제로 적용한 사례는 2016년

이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2017-0-01864,(안전성 연구 3세부) 암호 소프트웨어 안전성 연구)

DARPA (Defense Advanced Research Projects Agency)에서 최초로 주최한 CGC(Cyber Grand Challenge)이다. CGC는 인공지능 해킹대회이며 CMU (Carnegie Mellon University)의 MAYHEM[5]이 1등을 차지한 바 있다.

AEG는 자동 취약점 분석, 익스플로잇 생성, 취약점 원인 파악, 그리고 이의 보완 패치까지 수행하는 것을 목표로 하고 있으나 현재까지는 익스플로잇을 생성하여 이를 실행하는 수준에 머물러 있다.

본 논문의 2장에서는 AEG에 대한 설명과 배경 지식 및 기존의 관련 연구를 살펴본 뒤 구현과정에 대해 서술하고 3장에서는 관련 연구들의 한계점을 정리하며, 마지막으로 4장에서는 향후 계획 및 결론을 서술하며 마무리한다.

II. Automatic Exploit Generation

위에서 언급하였듯이 AEG란 프로그램의 취약점 분석 및 익스플로잇을 자동화하는 기술이다. AEG 연구의 기본적인 개념은 '1) 취약점의 위치를 파악하고, 2) 이에 도달하기 위한 조건(Ex:입력 값)과 3) 익스플로잇 가능여부를 파악' 하는 것이다. 이를 구현하기 위하여 필요한 개념인 실행경로식(Path Formulas)과 기호실행(Symbolic Execution)에 대해서는 뒤에서 서술할 예정이다.

2.1 Fuzzer와 AEG의 차이

기호실행과 실행경로식을 설명하기에 앞서 본 절에서는 Fuzzer와 AEG의 차이를 서술하고자한다. 현재 보안 분야에서 많이 이용되는 자동 취약점 탐지 기술 중 하나로는 퍼징(Fuzzing) 기법이 있다. Fuzzer도 AEG와 유사하게 취약점을 자동으로 분석하지만, 생성하는 입력 값이 실제로는 사용할 수 없는 경우가 대다수이며 Fuzzer의 경우에는 취약점의 자세한 정보를 얻을 수 없다는 차이점이 있다. 이러한 이유로 AEG는 Fuzzer 보다 높은 효율성을 가진다고 할 수 있다. [표 1]에서는 Fuzzer와 AEG의 차이점을 정리한다(효율성이란 전체 입력 값 생성 수 대비 유효한 입력 값의 비율을 의미).

[표 1] Fuzzer와 AEG의 차이

	Fuzzer	AEG
생산성	높음	낮음
효율성	낮음	높음
선호 분석 대상	소스코드	바이너리 코드
취약점 발견 방법	무작위 입력 후 취약점 분석	메모리, 명령어 모니터링등을 통한 분석

2.2 실행경로식[1]

경로식이란 프로그램의 실행 경로를 수식으로 표현한 것이다. 뒤에서 설명할 기호실행 입력 값은 변수가 아닌 수식을 대입해, 수식을 만족하는 입력 값을 얻을 수가 있다. AEG에서는 취약점 위치까지의 도달 경로를 실행경로식으로 구성한다.

2.3 기호실행[2]

기호실행은 입력을 구체적인 값으로 주는 것이 아닌 일종의 기호로 주고 프로그램을 실행하는 기법이다. 앞서 생성한 실행경로식을 기호실행의 입력 값으로 하여 익스플로잇을 생성할 수 있다.

기호실행을 통하여 입력 값을 생성할 수 있는 프로그램으로는 KLEE, Angr, Z3[3] 등이 있다. [그림 1]은 Z3 라이브러리를 이용한 기호실행 기법의 예제이다.

```
1 from z3 import *
2 array = [BitVec('a%i'%i,8) for i in range(0,17)]
3 s = Solver()
4 s.add(((array[3] * array[6]) ^ (array[2] * array[2])) == 7320)
5 s.add(((array[11] * array[14]) ^ (array[9] * array[11])) == 15882)
6 s.add(((array[14] * array[15]) ^ (array[11] * array[4])) == 11789)
7 print s.check()
8 print s.model()
```

8번 줄 → 만족하는 입력 값 출력

6번 줄 → Path Formulas

[그림 1] z3라이브러리를 이용한 예제

2.4 AEG 관련 연구

AEG를 적용하고자 하는 대상 프로그램의 소스코드까지 가지고 있는 경우는 많지 않고, 실행파일만 있는 경우가 더 많기 때문에 AEG 연구는 소스코드 뿐 아니라 바이너리 레벨에서도 적용할 수 있게끔 연구되어 왔다. 따라서 AEG 연구는 소스코드가 함께 있는 경우와 바이너리코드만을 가지고 있는 경우의 두 가지 경우로 진행되어 왔다. 본 절에서는 각 경우에 해당하는 [4]와 [5]의 구현 방식, 기여도, 한계점

과 [11]의 필요성과 한계점에 대하여 서술하며 마친다.

[4]에서 소개하는 구현 방식은 [그림 2]와 같다:

1. 바이너리코드 추출을 위하여 GCC를 활용하며, 중간언어(Intermediate Representation, IR)을 추출하기 위하여 LLVM을 활용한다.

2. 기호실행 기법을 중간언어에 적용하여 취약점의 위치를 대략적으로 파악한 후 해당 위치에 도달할 수 있는 실행경로를 찾는다.

3. 분석 대상의 프로그램을 실행하여 동적 분석을 진행하며 이 때 얻는 정보는 스텍 구조, 리턴주소 값 등이다.

4. 2, 3번의 정보를 이용하여 실행 가능한 입력 값(익스플로잇)을 생성한다.

[4]는 소스코드 기반의 AEG를 연구하였으며, 이를 이용하여 버그를 찾고 익스플로잇 생성까지 자동화를 시켰다는 점에서 기여도를 가진다. 그러나 이의 한계점은 생성한 익스플로잇이 BOF(Buffer Over-Flow)와 FSB(Format String Bug) 공격기법만을 대상으로 하였기 때문에 이외의 공격기법에 대하여서는 익스플로잇의 생성이 불가능하다는 점이다.

[5]에서 소개하는 구현 방식은 [그림 3]과 같다:

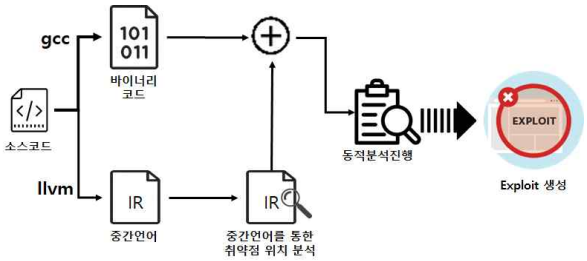
1. MAYHEM[5]에서는 익스플로잇을 생성하기 위하여 두 개의 모듈을 사용한다: Concrete Executor Client(CEC)는 명령어를 제어흐름을 변경하는 명령어(Ex: jmp, call)가 실행될때 마다 이를 분석 모듈인 Symbolic Executor Server(SES)에 전달하며, SES에서는 전달받은 명령어를 중간언어로 변환한다.

2. SES는 변환된 중간언어에 만족하는 실행경로식을 생성하고 이를 기호실행에 입력하여 익스플로잇을 생성한다.

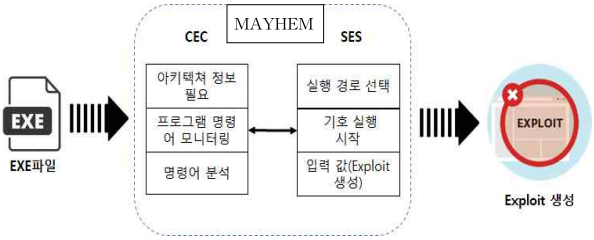
[5]은 바이너리코드의 분석만으로도 익스플로잇을 생성할 수 있음을 보인 연구라는 점에서 기여도를 가진다. 그러나 이의 한계점은 (1) 소규모 프로그램에만 적용이 가능하며, (2) 멀티쓰레드 프로그램에는 적용할 수 없고, (3) 메모리 소비가 심하며, (4) DEP(Data Execution Protect), ASLR(Address Space Layout Randomization) 등의 보호기법이 적용된 경우 해당 기법을 적용하기 어렵다는 점이다.

[5]에서 제시하는 취약점 분석은 바이너리 그 자체만으로는 분석이 어려울 수 있으나, 이를 중간언어로 변환할 수 있는 경우 분석의 난이도가 낮아지게 된다. 이를 위하여 바이너리를 중간언어로 변환하는 도구(Ex:BAP, PyVEX, BINSEC)가 연구된 바 있지만, 중간언어 변환 결과가 정교하다고 할 수는 없다.

이에 따라 [11]에서는 위와 같은 변환 도구의 정확도 측정방안을 제시 및 구현한 결과를 제시하였다. 그러나 이는 현재까지 x86 아키텍처에서만 적용이 가능하고 소수점 처리 명령어에 대해서는 고려하지 않았다는 점에서 개선의 여지가 있다.



[그림 2] 소스코드 기반 AEG의 동작 과정



[그림 3] MAYHEM의 동작 과정

III. 기존 AEG 연구의 한계점

본 장에서는 2장 4절에서 소개한 연구들 이외에 다른 기존 연구들의 내용을 간략하게 서술하며 이의 한계점을 서술한다([표 3] 참고).

[표 3] 기존 연구 한계점

기존 연구	내용(위) / 한계점(아래)
[6]	최초로 AEG 문제를 Verification 문제로 간주
	● 주어진 API 모델 상에서만 동작

	● 실제 Shell을 실행하는 익스플로잇에는 부적합함 ● 생성된 입력 값이 유효하지 않을 수 있기 때문에 익스플로잇이 동작하지 않을 수 있음
[7]	패치에서부터 자동으로 익스플로잇을 생성하는 최초의 시도
	● 이미 패치가 완료된 취약점에 대해서도 탐지함 ● 생성된 익스플로잇은 단순히 프로그램을 동작을 중지시킬 뿐으로 Control-Flow Hijacking같은 공격은 할 수 없음
[8]	ROP(Return Oriented Programming) 공격이 가능한 코드 자동 생성
	● ret 명령어가 없는 코드 재사용 공격은 고려하지 않음
[9]	제어흐름 공격이 가능한 취약점을 찾고 다형성(polymophic)이 적용된 익스플로잇 생성 도구 구현
	● ASLR, DEP가 적용되어 있는 경우에는 해당 기법을 적용 할 수 없음 ● 모든 공격 가능 경로를 탐색하지는 않음
[10]	Data-Oriented Attack의 익스플로잇 생성을 자동화하였으며, 해당 공격이 제어흐름 보호기법을 우회할 수 있음을 증명함
	● 분석 대상 프로그램의 취약점 위치를 미리 알고 있어야 함

IV. 결론

2005년부터 시작된 AEG 연구는 초기에는 API를 기반으로 익스플로잇을 생성하는데 그쳤으나, 그 이후 종류가 다양해짐에 따라 패치 기반, 크래시(Crash) 기반, 소스코드 기반, 바이너리코드 기반으로 분석 및 익스플로잇을 생성할 수 있게 되었다. 3장에서 확인할 수 있듯이 AEG 분야에서는 아직 해결해야할 한계점이 많이 있다.

향후에는 더 많은 종류의 익스플로잇을 고려하며 제약 조건 및 한계점을 해결하는 방향으로 연구할 예정이며 최종적으로는 AEG의 자동 취약점 패치에 관한 연구를 수행할 예정이다.

[참고문헌]

[1] Path Fomulas. http://secuinside.com/archive/2016/2016-2-6.pdf

[2] Symbolic Execution. https://en.wikipedia.org/wiki/Symbolic_execution

[3] python z3 library. https://github.com/Z3Prover/z3

[4] Avgerinos, S.K Cha, B.L.T.Tao and D.Brumley. AEG : Automatic Exploit Generation. NDSS, 2011

[5] S.K Cha, T.Avgerinos, A.Rebert and D.Brumley. Unleashing MAYHEM on Binary Code. IEEE Symposium on Security and Privacy, 2012

[6] V.Ganapathy, S.A.Seshia, S.Jha, T.W.Reps and R.E.Bryant, Automatic Discovery of API-Level Exploits, ICSE 05, May.

[7] D.Brumley, P.Poosankam, D.song and J.Zheng. Automatic Patch-Based Exploit Generation is Possible: Techniques and Implications. IEEE Symposium on S&P, 2008

[8] E.J. Schwartz, T.Avgerinos and D.Brumley. Q : Exploit Hardening Made Easy. USENIX Security Symposium, 2011

[9] H.Wang, P.R.Su, Q.Li, L.Y.Ying, Y.Yang and D.G.Feng. Automatic Polymorphic Exploit Generation for Software Vulnerabilities. International Conference on Security and Privacy in Communication Networks, 2013

[10] H.Hu, Z.L.Chua, S.Adrian, P.Saxena and Z.K.Liang. Automatic Generation of Data-Oriented Exploits. USENIX Security Symposium, 2015

[11] S.M.Kim, M.Faerevaag, M.Jung, S.Jung, D.Y.Oh, J.H.Lee, S.K.Cha, Testing Intermediate Representations for Binary Analysis, ASE, 2017

커넥티드카(Connected car)통신을 위한 국제 표준화 동향 연구 : 미국, 유럽 표준을 중심으로1)

주경호* 이동훈*

*고려대학교 정보보호대학원

Study on International Standardization for Connected Car Communication: Focusing on US and European Standards

Kyungho Joo* DongHoon Lee*

*Graduate School of Information Security, Korea University.

요 약

ITS(Intelligent Transportation System)는 미래 교통시스템의 안정성(Safety), 이동성(Mobility) 및 지속가능성(Sustainability)을 혁신적으로 개선시킬 수 있는 융합 시스템이다. 하지만 도로를 주행하는 차량은 다양한 국가에서 생산되며, 구매자 및 운전자 역시 다양한 국가에 존재한다. 즉, 효율적인 ITS 구축을 위해서는 국가별로 요구되는 ITS 서비스 및 요구사항들이 반드시 표준화 과정에 조화롭게 반영되어야 한다.

본 논문에서는 ITS구축을 위한 미국과 유럽의 표준화 단체들이 정의하고 있는 커넥티드카(Connected car) 통신의 표준화 동향을 비교 분석한다. 특히 IT 산업을 주도하는 미국과 자동차 산업을 주도하는 유럽의 커넥티드카 통신을 위한 표준화 동향을 비교함으로서 표준화 단체별 요구사항의 공통점 및 차이점을 분석한다.

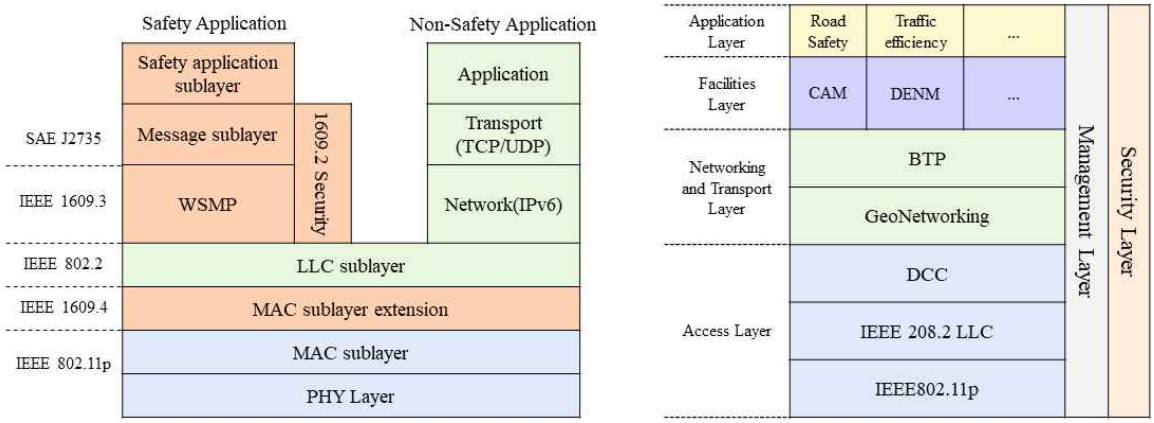
I. 서론

최근 출시되는 차량들은 기존의 단순한 운송수단으로서의 역할을 넘어 주변 환경과의 연결성이 확대되고 있으며, 운전자의 주행 환경을 보조하는 다양한 기술들이 차량에 탑재되고 있다. 이와 같은 기술 발전의 흐름은 ITS(Intelligent Transportation System)구축을 목표로 진행되고 있다. ITS는 미래 교통시스템의 안정성(Safety), 이동성(Mobility) 및 지속가능성(Sustainability)을 혁신적으로 개선시킬 수 있는 융합 시스템이다.

한편, 안전(Safety)하고 효율적인(Efficient) ITS 환경 구축을 위해 차량과 주변 통신 인프라 간의 V2I(Vehicle to Infrastructure)통신, 차량과 차량 간의 V2V(Vehicle to Vehicle)통신에 대한 다양한 연구들이 진행되고 있다. 한편, 이와 같은 연구들을 기반으로 미국 및 유럽을 포함한 세계 각국에서는 커넥티드카(Connected Car)에 사용되는 통신 시스템의 표준을 제정하기 위한 다양한 움직임이 진행되고 있다.

하지만 도로를 주행하는 차량은 다양한 국가에서 생산되며, 구매자 및 운전자 역시 다양한 국가에 존재한다. 만약 국가 혹은 지역마다 ITS 환경에서 제공되는 서비스 및 요구사항들이 효과적으로 통합되지 않는다면, 해당 국가 혹은 지역의 변동사항을 반영하는 과정에서 추가적인 비용이 발생하게 된다. 반대로, 표준화된 하드웨어와 소프트웨어 모듈은 ITS환경의 구성요소들을 제조 및 개발하는 과정에서 비용감소에 가장 집적적인 영향을 줄 수 있다. 또한, 다양한 IT서비스를 디자인 및 구현 과정에서 발생 가능한 위험(Risk) 및 위협(Threat)을 효과적으로 관리할 수 있을 뿐만 아니라 Assurance, Trustworthiness를 만족하는 시스템을 구축할 수 있다. 따라서 효율적인 ITS환경구축을 위해서는 국가별로 요구되는 ITS 관련 서비스 및 요구사항들이 반드시 표준화 과

1)이 논문은 2017년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(R7117-16-0161, 자율주행 스마트자동차용 이상징후 탐지 핵심기술개발)



(a) IEEE의 WAVE 표준 (a) ETSI의 ETSI ITS 표준

[그림 1] IEEE의 WAVE 표준과 ETSI의 ETSI C-ITS 표준의 통신 프로토콜 스택

정에 조화롭게 반영되어야 한다. 또한, 개발자 역시 표준에 대한 심도 있는 이해를 통해 하드웨어 및 소프트웨어 모듈을 구현해야 한다.

본 논문에서는 IT 산업을 주도하고 있는 미국과 자동차 산업을 주도하고 있는 유럽의 ITS관련 표준화 동향을 비교 분석한다. 이를 통해 미래 ITS 구축을 위해 표준화 단체별 요구사항의 공통점 및 차이점을 분석한다. 논문의 기여도는 다음과 같다.

· 기존에는 개별적으로 분석이 진행되었던 미국과 유럽의 커넥티드카 통신표준을 표준 스택(Stack)의 레이어별로 통합하여 비교, 분석하였다.

논문의 구성은 다음과 같다. 2장에서는 커넥티드카 통신 표준을 정의하는 표준화 단체에 대해 살펴본다. 3장에서는 표준 스택의 레이어별 특징 및 두 표준의 공통점 및 차이점을 분석한다. 4장에서는 표준분석 및 이에 대한 선제적 대응의 필요성에 대해 살펴본다.

II. 배경지식

2.1 관련 국제기구

ITS 환경구출을 위한 커넥티드카 통신 표준화 과정은 서로 다른 국가 및 지역에 따라 다양한 레벨에서 정의되고 있다. 대표적인 표준화 단체는 IEEE(Institute of Electrical and Electronics), ETSI(European Telecommunications Institute), SAE(Society of Automotive Engineers) International 등이 있다.

SAE International는 미국 자동차 기술자 협회로 처음 설립되었으며, 다양한 업계의 엔지니어링 전문가를 위한 표준 개발 기관이다. 특히, 자동차, 항공 우주 및 상업용 차량과 같은 운송 산업에 중점을 두고 있다[1].

IEEE는 1963 년 미국 전기 기술 연구소(Institute of Electrical Engineers)와 라디오 엔지니어 협회 (Institute of Radio Engineers)의 합병으로 설립되었으며, 전 세계 160 개국 이상에 42만명 이상의 회원을 보유한 세계에서 가장 큰 기술 전문가 협회이다[2].

ETSI(European Telecommunications Standards Institute)는 유럽의 통신 업계 (장비 제조업체 및 네트워크 사업자)의 독립적 인 비영리 표준화기구이다. ETSI는 고정, 모바일, 라디오, 수렴, 방송 및 인터넷 기술을 포함하여 정보통신기술(ICT)에 대해 전 세계적으로 적용되는 표준을 제정하고 있다[3].

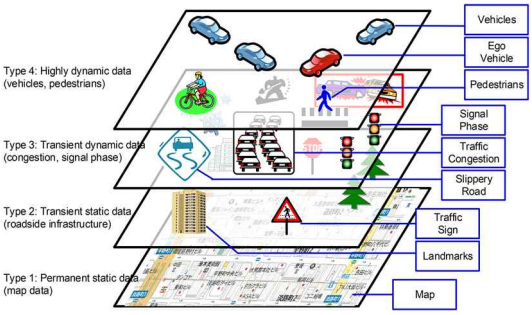
III. IEEE vs. ETSI

본 절에서는 IEEE에서 정의한 WAVE 표준과 ETSI에서 정의한 ETSI ITS 관련 표준에 대해 살펴본다. 3.1에서는 WAVE 표준과 ETSI ITS표준의 대표속성에 대해 살펴본다 3.2 에서는 WAVE와 ETSI 표준을 요약하고 두 표준의 차이점에 대해 살펴본다.

3.1 WAVE vs. ETSI C-ITS

IEEE의 WAVE 표준 및 ETSI의 C-ITS 표준 스택은 [그림 1]에 나타나 있다. 유사한 기능을 수행하는 레이어별 표준의 특징을 살펴보면 다음과 같다. 본 논문에서는 ETSI ITS 표준을 ETSI 표준으로 표기한다.

· **PHY Layer**



[그림 2] LDM(Local Dynamic Map) 개념도

IEEE와 ESTI는 모두 물리계층(PHY Layer)의 표준으로 IEEE 802.11p의 PHY layer표준을 정의하고 있다. 802.11은 무선 근거리 통신망(WLAN)을 구현하기 위한 표준이며, 802.11p는 차량 통신을 지원하기 위한 수정본이다. ETSI는 IEEE 802.11p 표준을 따르되 WAVE와는 다른 주파수 대역을 할당하고 있다. 또한, WAVE와는 달리 ETSI 표준에서는 물리계층의 개념이 PHY란 용어대신 Access Layer에 포함된 개념으로 사용되고 있다.

· MAC(Medium Access Control) Layer

WAVE 표준에서 정의된 MAC Layer역시 ETSI에서는 Access layer에 포함되는 개념으로 사용되고 있다. WAVE 표준과 ETSI 표준에서는 MAC layer 또한 IEEE 802.11p의 MAC Layer표준을 따르고 있다. WAVE 표준의 MAC layer는 CSMA/CA(Carrier sense multiple access with collision avoidance)[4]를 사용한다.

반면, ETSI는 CSMA/CA와 더불어 DCC (Decentralized Congestion Control)를 사용하여 네트워크상의 트래픽을 관리한다. DCC는 네트워크상의 혼잡(Congestion)을 관리 위해 TPC(Transmit Power Control)알고리즘을 이용하여 노드들이 전송하는 신호의 전력(Power)을 적응적(Adaptive)으로 조절한다. 또한, TRC(Transmit Rate Control)알고리즘을 이용하여 노드들이 생성하는 패킷의 생성빈도를 조절한다.

한편, WAVE와 ETSI 모두 IEEE 802.2 LLC를 부계층(Sub layer)으로 사용한다. LLC(Logical Link Control)는 소프트웨어적 구성을 통해 시스템이 다양한 MAC 부계층 프로토콜을 사용할 수 있도록 한다. 이를 통해 [그림 1]에서 응용계층(Application Layer)에서 사용되는 Safety application 과 Non-safety application에 사용되는 서로 다른 토폴로지가 호환될 수 있으며, LLC를 통해 ETSI역시 IP(Internet Protocol) 서비스를 제공할 수 있다.

- Network 및 Transport Layer

Network 및 Transport Layer 표준을 정의하는 IEEE 1609.3에서는 WAVE 통신의 데이터 교환을 위한 주소 체계 및 라우팅 방법을 정의하고 있다. 한편, ETSI는 GeoNetworking을 정의하고 있다. GeoNetworking 은 GPS 좌표를 통해 단일 홉(Single-hop) 및 다중 홉(Multi-hop) 통신의 라우팅(Routing) 방법을 제공한다.

- SAE J2753 과 Facilities Layer

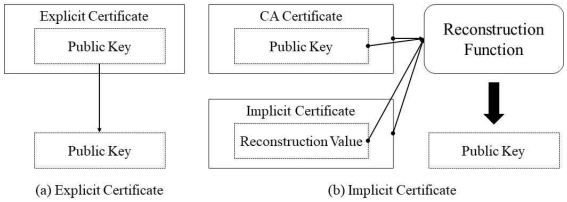
IEEE가 WAVE 표준에서 사용되는 V2V, V2I의 통신스펙을 정의하고 있지만, 메시지의 내용(Content)은 SAE가 정의하고 있다. 특히, WAVE 표준에서는 차량이 주기적으로 전송하는 안전 메시지를 BSM(Basic Safety Message)으로 정의하며, 이는 SAE J2753 표준으로 제정되었다. BSM은 1초당 10회 주기로 전송되는 메시지이며, 전송하는 차량의 상태정보(e.g 차량의 위도·경도·고도 정보, 차량의 크기, 속도, 가속도 방향, 제동시스템 상태 등)를 포함하고 있다.

한편, ETSI는 설비계층(Facilities Layer)에서 BSM과 유사한 기능을 수행하는 CAM(Cooperative Awareness Messages)과 DENM(Decentralized Environmental Notification Messages)을 정의한다. CAM은 초당 1~10회 주기로 차량의 상태정보(위도, 경도, 고도, 차량의 타입)를 전송한다. 반면, DENM은 특정 사건(event)이 발생 하였을 때에만 메시지를 전송하는 Event-driven 방식으로 전송되며, 도로위의 돌발 상황(급제동, 도로공사, 위험물의 위치) 에 대한 정보를 전송한다.

또한, ETSI 표준에서는 CAM과 DENM을 이용하여 LDM(Local Dynamic Map)을 생성한다. LDM은 동적으로 변하는 도로위의 정보를 CAM과 DENM을 이용해 업데이트 한다. 이를 통해, 차량 및 노변기지국(RSU)이 자체적으로 지도정보를 로컬에 저장하여 추후에도 사용할 수 있도록 한다. LDM에 대한 개념은 [그림 2]에 나타타 있다.

- Application Layer

WAVE 및 ETSI는 응용 계층(Application layer)을 통해 ITS 환경에서 사용되는 기본적인 어플리케이션의 요구사항을 정의하고 있다. WAVE는 이를 Safety application과 Non-safety application으로 구



[그림 3] 명시적 인증서(Explicit Certificate)와 암시적 인증서(Implicit Certificate) : (a) 명시적 인증서, (b) 암시적 인증서

분 하였으며, ETSI는 이는 Road Safety(Driving Assistance), Traffic efficiency, Other applications로 구분하여 정의하고 있다.

- 보안 계층(Security Layer)

WAV E표준과 ETSI 표준의 보안계층에서 정의된 항목들은 상당부분 유사하며, 공통적으로 보안 컴포넌트(Security Component), 보안 헤더(Security Header), 인증서 포맷(Certificate format), 보안 프로파일(Security Profile)을 정의하고 있다. 또한, ECC(Elliptic Curve Cryptograph)를 사용하여 보안 서비스를 제공하도록 정의하고 있다.

특히, WAVE 및 ETSI 표준에서는 기존의 명시적 인증서(Explicit certificate) 대신 암시적 인증서(Implicit certificate)를 사용한다는 특징이 있다. 암시적 인증서는 기존의 명시적 인증서와는 달리 인증서를 통해 공개키가 재구성(Reconstruct) 되는 특징이 있다. 이에 대한 설명은 [그림 3]에 나타나 있다. 명시적 인증서 및 암시적 인증서는 모두 식별 데이터, 공개키, 디지털 서명으로 구성되어 있다. 특히, 암시적 인증서는 재구성 함수 (Reconstruction Function)를 통해 공개키를 재구성 하여 사용하기 때문에 인증서의 길이가 ECC 암호시스템의 비트수와 같다. 즉, 160비트의 ECC 암호시스템에서 사용되는 암시적 인증서의 길이는 160비트이다. 반면, 대표적인 명시적 인증서의 구성 형태인 X.509는 order of 1KB 이며, 최대 8000bit까지 커질 수 있다[5].

3.2 요약 및 차이점

WAVE 표준과 ETSI 표준은 기존에 존재하는 IEEE의 무선통신 표준을 기반으로 ITS 환경에 특화된 표준이 기관별로 각각 정의되는 형태이다. 또한, 유사한 기능을 담당하더라도 서로 다른 용어를 통해 계층을 구분함으로서 차별성을 두고 있다.

물리계층에서는 두 개의 표준모두 IEEE 802.11p PHY 표준을 사용하지만 할당하는 주파수 대역에서 차이점이 존재한다. MAC 계층역시 두 개의 표준모두 IEEE 802.11p MAC 표준인 CSMA/CA를 사용하나, ETSI 표준의 경우 DCC를 추가하여 보다 ITS 환경에 특화된 혼잡 컨트롤 기법을 정의하고 있다. 또한, ETSI 표준에서는 라우팅을 위해 노드의 위치정보를 통한 라우팅 기법인 GeoNetworking을 사용하는 특징이 있다.

한편, WAVE 표준과 ETSI 표준 모두 차량의 상태정보를 공유하기 위한 표준을 정의하고 있다. WAVE 표준은 BSM을 이용해 차량의 상태정보를 주기적으로 전송한다. 반면, ETSI 표준은 주기적으로 메시지를 전송하는 CAM 뿐만 아니라, Event-driven 방식으로 전송되는 DENM이 존재하는 차이점이 있다. 또한 ETSI 표준에서는 CAM과 DENM을 이용하여 LDM을 추가적으로 생성한다.

WAVE 표준과 ETSI 표준의 보안서비스는 ECC를 기반으로 이루어진다는 공통점이 있으며, 상당부분 유사한 구조와 기능을 가진다. 특히, 기존의 명시적 인증서가 아닌 ITS 환경에 특화된 암시적 인증서가 사용된다는 특징을 가지고 있다.

IV. 결론

본 논문에서는 지능형 미래 교통 시스템인 ITS 구축을 위한 국제적인 표준화 동향에 대해 살펴보았다. 5G로 대표되는 커넥티드카 통신 및 지능형 교통시스템으로의 변화는 피할 수 없는 흐름이다. 이러한 흐름에 맞춰 IT 산업을 주도하는 미국과 자동차 산업을 주도하는 유럽의 커넥티드카 통신표준화 동향을 주의 깊게 살펴볼 필요가 있으며, 이를 통해 다양한 ITS 서비스에서 발생 가능한 위험관리 및 위협평가가 이루어져야 한다.

[참고문헌]

[1] http://www.sae.org/[Accessed : 2017-10-28]

[2] https://www.ieee.org/about/index.html [Accessed : 2017-10-28]

[3] http://www.etsi.org/ [Accessed : 2017-10-28]

[4]https://en.wikipedia.org/wiki/Carrier-sense_multiple_access_with_collision_avoidance [Accessed : 2017-10-28]

[5] https://en.wikipedia.org/wiki/Implicit_certificate [Accessed : 2017-10-28]

2017 한국정보보호학회 동계학술대회

D1 해킹 및 취약점(I)

김 경 곤 (고려대)

- **DTLS 기반의 CoAP 보안 메커니즘 분석 및 성능평가**
  한상우, 조정모, 박창섭 (단국대학교)
- **난독화 방법에 따른 Clang 정적 분석 도구의 성능 영향 분석**
  진홍주, 박문찬, 이동훈 (고려대학교)
- **프린터 관리 시스템의 취약점 분석**
  지우중, 이경문, 이병천 (중부대학교)
- **다크 웹 크롤링을 위한 지능형 크롤러 설계**
  노인우, 권구환, 임을규 (한양대학교)

DTLS 기반의 CoAP 보안 메커니즘 분석 및 성능평가*

한상우, 조정모, 박창섭

단국대학교

DTLS-based CoAP security mechanism analysis and performance evaluation

Sang-Woo Han, Jung-Mo Jo, Chang-Seop Park

Dankook University

요 약

자원 제약적 IoT 환경에 최적화 된 표준 프로토콜 CoAP(Constrained Application Protocol)은 IoT 환경 내의 센서노드(CoAP Server) 그리고 인터넷 상의 클라이언트(CoAP Client) 간의 웹 기반 통신을 지원한다. 특히, CoAP에서는 응용 데이터의 보호를 위해 DTLS(Datagram TLS)의 사용을 권고하고 있다. 본 논문에서는 CoAP과 CoAP에서 지원하는 CoAP-DTLS 보안 모드별 분석과 Cooja 시뮬레이션을 통한 성능평가를 수행한다. 또한 보안 모드가 적용될 수 있는 환경인 IETF에서 제시하는 7가지의 활용사례(Use Cases)에 대하여 분석하고 활용사례 별로 적절한 보안모드를 제시한다.

I. 서 론

CoAP(Constrained Application Protocol) [1]은 자원이 제약적인 IoT 환경에서 사용하기 위한 경량화된 웹 전송 프로토콜이다. LoWPAN이 연동되는 CoAP Client-Server에 대한 보안 잇슈로는 센서노드(CoAP Server) 자체와 인터넷상의 클라이언트(CoAP Client) 간의 종단간 보안(End-to-End Security)이 요구되는 데에 있다. 이를 위해 IETF에서는 LoWPAN 연동 IoT 환경에서 보안 메커니즘이 적용되는 7개의 활용사례(Use Case) [2]를 기술하고 있으며, CoAP에 의해 전달되는 응용 데이터 보호를 위한 DTLS(Datagram TLS) [3]의 사용과 이에 대한 실효성 검증연구들이 진행되고 있다 [4, 5, 6]. 본 논문에서는 안전한 CoAP 응용을 위한 표준 프로토콜인 DTLS의 보안모드에 대한 분석 그리고 각 보안모드가 적용될 수 있는 응용환경 (7 Use Cases)에 대한 분석을 각 보안모드의 성능평가를 기반으로 진행한다. 2장에서는 LoWPAN 연동 IoT 아키텍처 및 CoAP 보안을 소개하고 3장에서는 DTLS 기반 CoAP 보안 메커니즘을 보안모드 별로 분석한다. 또한, Use Case 별로 적절한 보안모드를 제시한다. 4장에서는 보안모드 별 성능평가를 Contiki Cooja 시뮬레이터 [7]를 이용하여 진행하며 마지막으로 5장에서 결론을 맺는다.

* 본 연구는 미래창조과학부 및 한국 인터넷 진흥원의 “고용계약형 정보보호 석사과정 지원사업”의 연구결과로 수행되었음. (과제번호 H2101-17-1001)

II. LoWPAN 연동 IoT 아키텍처에서의 CoAP 보안

2.1 LoWPAN 연동 IoT 아키텍처 환경

IETF는 IoT를 지원하는 다양한 프로토콜 표준을 제정하고 있지만, IoT 환경의 복잡성 그리고 IoT 기반 서비스의 다양한 요구사항에 기인하는 기술적 도전 과제들이 남아있다. IETF에서 제시하는 IoT 무선 센서 네트워크는 [그림 1]에서와 같이 자원제약적인 LoWPAN으로 규정하고 있으며 IEEE 802.15.4는 MAC/PHY 계층의 사실 상의 표준이다 [8, 9]. 듀얼 스택을 탑재한 6LBR (6LoWPAN Border Router)은 6LoWPAN 기반 LoWPAN의 인터넷 연결을 가능하게 하며, 응용계층에서의 CoAP은 DTLS에 의해 보호된다. IETF ACE 워킹그룹에서는 [그림 1]에서와 같이 LoWPAN 연동 IoT 환경에서 보안 메커니즘이 적용되는 7개의 Use Case들을 규정하고 있다.

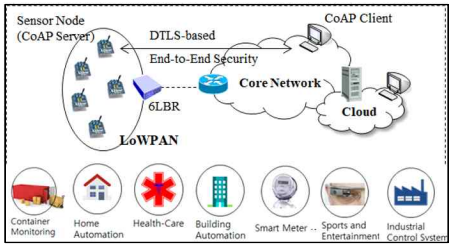


그림 1. LoWPAN 연동 IoT 및 IETF's 7 Use Cases

2.2 CoAP과 DTLS

IoT 환경에서의 표준 웹 전송 프로토콜인 CoAP은 자원 제약적인 디바이스(Constrained Device)와의 정보교환을 위해 경량화된 프로토콜이다. CoAP 종단간에 CoAP Request 및 CoAP Response 메시지의 교환을 통해서 CoAP 클라이언트는 특정 명령어를 센서 디바이스에게 전송하거나 또는 센싱정보를 CoAP 서버로부터 전달받게 된다. 특히, LoWPAN 환경의 특성을 고려하여 CoAP에서는 CoAP 메시지의 신뢰성 있는 전송을 위해 Exponential-Backoff 방식에 기반을 둔 Stop-and-Wait 전송 프로토콜이 사용된다.

DTLS는 IoT 클라이언트-서버 모델에서의 안전한 CoAP 기반 응용 보안 메커니즘의 사실 상의 표준이다. CoAP-DTLS에서 권고되는 보안모드(Security Mode)는 PSK(Pre-Shared Key), RPK(Raw Public Key) 및 ECDSA 기반 Certificate가 있다. 하지만, PSK 및 RPK의 경우 센서노드(서버)와 클라이언트 간의 일대일 사전 보안설정의 어려움으로 확장성이 떨어지고, 반면에 ECDSA 기반 Certifcate의 경우는 확장성 측면에서는 우수하지만 DTLS Full Handshake의 경우 ECC 계산 및 인증서 검증에 너무 많은 시간이 소요되는 단점을 가지고 있다.

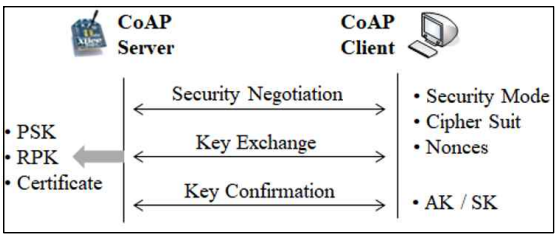


그림 2. DTLS Handshake 프로토콜

DTLS 보안모드는 CoAP 종단간의 상호인증 및 세션키 도출을 위해서 [그림 2]에서와 같이 3단계로 구성되는 DTLS Handshake 프로토콜의 세부동작을 규정한다. 보안협상(Security Negotiation) 단계를 통해서 보안모드 그리고 사용할 Cipher Suit 및 Nonce들이 결정된다. 키 교환(Key Exchange) 단계에서는 보안모드에 따른 인증키(Authentication Key, AK) 및 이를 기반으로 세션키(Session Key, SK)가 도출되며, 마지막으로 키 확인(Key Confirmation) 단계에서 상호인증이 수행된다. 이 모든 과정이 성공적으로 종료되면 도출된 세션키를 이용하여 CoAP 메시지에 대한 보호작업이 DTLS Record 프로토콜을 통해 실행된다.

III. DTLS 기반의 CoAP 보안 메커니즘 분석

본 장에서는 DTLS 보안모드의 특성과 IETF ACE 워킹그룹이 규정하는 LoWPAN 연동 IoT 환경에서 보안 메커니즘이 적용되는 7개의 Use Case 별로 적합한 보안모드를 제시해본다.

3.1 DTLS 보안모드 분석

PSK 모드는 CoAP 클라이언트와 CoAP 서버간에 사전에 대칭키가 공유되어 DTLS Handshake 프로토콜이 진행된다. 4장의 성능평가를 통해서 보면 3개의 보안모드 중에서 처리시간 측면에서 가장 효율적인 보안모드이다. 하지만, CoAP 클라이언트와 CoAP 서버간에 대칭키의 사전 공유를 위해서는 별도의 키 공유 아키텍처 및 프로토콜이 요구된다.

Certificate 모드와 RPK 모드에서는 상호인증을 위해 ECDSA 기반의 공개키와 개인키가 요구된다. 4장의 성능평가에서 이들에 대한 처리시간이 가장 오래 소요되는 이유는 EC (Elliptic Curve) 덧셈연산이 수반되기 때문이다. Certificate 모드에서는 CoAP 클라이언트 및 서버의 공개키를 X.509 v3 형식의 인증서 형태로 발급하여야 함을 전제로 하고 있다. 따라서, 인증서를 발급해 주는 인증기관의 역할을 담당하는 제3의 서버가 존재해야 하며 인증서를 안전하게 발급받기 위한 별도의 보안 메커니즘이 존재해야 한다. 특히, DTLS Handshake가 진행되는 과정에서 상대방의 인증서를 전달받게 되는 쪽은 인증서의 유효성을 확인해야 하기 때문에 OCSP(On-line Certificate Status Protocol) [10]를 별도로 기동시켜야 하는 추가의 부담이 생기게 된다. 또한, 인증서의 크기가 수백 바이트에 달하기 때문에 IEEE 802.15.4 기반의 LoWPAN 환경에서는 프레임의 길이가 127바이트이어야 하는 제약에 걸려 단편화(Fragmentation) 또는 압축(Compression)의 과정을 거치기 때문에 성능 면에 있어서는 다른 2가지 모드들에 비해 현저히 떨어지게 된다. 하지만, Certificate 모드에서는 일단 인증서가 개별적으로 발급이 된 이후에는 임의의 CoAP 클라이언트와 CoAP 서버 간에는 별도의 추가 절차 없이 DTLS Handshake가 진행될 수 있다.

RPK 모드는 Certificate 모드가 가지는 2가지의 단점, 첫째 인증서의 길이가 길다는 단점, 둘째 별도의 OCSP가 기동되어야 한다는 단점을 보완하기 위한 보안모드이다. 즉, CoAP 클라이언트와 CoAP 서버는 각각 상대방의 공개키를 사전에 공유하고 있다는 가정을 하고 있다. 만약, 특정 CoAP 서버가 다수의 클라이언트와 DTLS Handshake를 기동하는 응용환경에서는 결국 다수의 상대방 공개키가 사전에 공유되어 있어야 한다. 이는 결국 PSK 모드에 내재된 동일한 문제점을 가지게 된다. 하지만, Certificate 모드와 함께 RPK 모드는 PSK 모드가 제공하지 못하는 Source 인증이 요구되는 응용환경에서는 필수적인 보안모드이다. 따라서, Source 인증이 요구되지만 인증서 사용이 성능 상의 제약으로 가해지는 경우에만

활용이 가능하게 된다.

3.2 Use Case 별 보안모드 선정

본 절에서는 3.1절에서 분석된 3가지 보안모드의 특성과 4장의 성능평가에 따라서 IETF 7가지 Use Case 별로 적절한 보안모드에 대한 선정이 논의된다. Use Case들은 자원 제약적 특성, 연결 설정의 빈도. 연결 상태 지속 시간, 사용 환경을 고려하여 보안 모드를 적용할 PSK 사용 그룹, RPK 사용 그룹, Certificate 사용 그룹으로 구분할 수 있다.

3.2.1 PSK 사용 그룹

PSK 사용 그룹은 자원 제약적 특성과 연결 설정의 빈도를 기준으로 결정되는 그룹이다. 4.2 절의 Handshake 완료 시간에 의하면 RPK와 Certificate에 비하여 PSK의 연결 설정에 필요한 시간이 상대적으로 빠른 것을 볼 수 있다. 연결 설정의 빈도가 잦을수록 RPK와 Certificate의 경우 필요한 시간이 높은 비율로 증가하기 때문에 PSK 사용을 권고한다. 또한 4.3 절에 의하면 PSK RAM / ROM 소모량이 가장 적다. 이에 따라 PSK 사용 그룹에는 연결 설정이 빈번하게 이루어지는 환경이나 센서의 자원이 제약적인 환경에 적합하다. 사용자의 조작에 따라 연결 설정의 빈도가 결정되는 Home Automation, Sport/Entertainment 과 일반적으로 센서의 RAM이 10킬로바이트 미만인 Health-Care가 PSK 사용 그룹에 속하게 된다.

3.2.2 RPK 사용 그룹

RPK 사용 그룹은 연결 상태의 지속 시간과 사용 환경에 의해 결정되는 그룹이다. 3.1 절에 의하면 RPK는 별도의 OCSP가 없이 Source 인증이 필요한 환경에서 적합하다. 별도의 OCSP가 필요하지 않다는 것은 통신이 필요한 개체 간 공개키를 사전에 공유하고 있기 때문에 Source 인증이 가능하다는 것을 말한다. 또한 4.3 절에 의해 연결 설정에 필요한 시간이 크기 때문에 연결 상태가 계속 유지되는 환경에 적합하다. 이에 따라 단일 그룹 내의 보안 통신이 필요한 Smart Metering, Building Automation, Industrial Control System이 RPK 사용 그룹에 속하게 된다.

3.2.3 Certificate 사용 그룹

Certificate 사용 그룹은 RPK과 같은 특징을 갖지만 다 그룹 간의 보안통신이 필요한 경우에 사용되는 환경이다. 다 그룹간의 Source 인증의 경우, 사전에 공개키가 공유되어야 하는 RPK가 부적합하기 때문에 개인키·공개키를 담당하는 인증기관이 필요하게 된다. 다 그룹 간 보안통신이 필요한 Container Monitoring과, 빌딩 내에 여러 그룹이 존재하고 Building Administrator가 Building의 여러 그룹 내의 센서 관리의 역할을 수행하는 Building Automation의 경우 Certificate 사용 그룹에 속하게 된다.

IV. 보안모드 별 성능 분석

본 장에서는 CoAP에서 지원하는 DTLS의 세가지 모드인 PSK, RPK, Certificate 모드들을 Contiki OS 상의 Cooja Simulator를 통하여 성능 분석을 실시한다. 성능 측정 요소는 각 모드별 DTLS Handshake 소요 시간, Server 노드의 RAM / ROM 소모량으로 구성된다.

4.1 실험 환경

본 논문은 IoT 환경 상의 센서 노드(CoAP Server)와 인터넷 상의 클라이언트(CoAP Client)간의 안전한 통신을 위한 DTLS의 각 모드 별 성능 측정을 목표로 하여 실험 환경을 구성하였다. 성능 측정을 실행한 시뮬레이션 환경은 [표 1]과 같다. Cooja 시뮬레이터 내에 Border Router와 CoAP Server를 배치하였으며 Border Router에 tunslip6 명령어를 통하여 외부의 CoAP Client와 통신이 가능한 IPv6 주소로 매핑하였다. Cooja 상의 각 노드들(CoAP Server, Border Router)은 WSN Applications을 위한 저전력 모듈인 MSP430 시리즈의 Wismote에 컴파일하여 테스트를 진행하였다.

표 1. 시뮬레이션 환경

환경	설명
OS	Contiki 3.0
Simulator	Cooja
Sensor Node	Wismote(TI MSP 430)
DTLS	TinyDTLS 0.8.2 psk 및 uECC / Certificate

4.2 DTLS Handshake 소요시간

DTLS Handshake 완료 시간은 전체적인 성능의 척도로써 CoAP 클라이언트가 DTLS 연결 시작의 첫 번째 메시지인 CLIENT HELLO 메시지 전송 시간부터 CoAP 서버의 FINISH 메시지의 수신까지의 시간을 측정하였다.

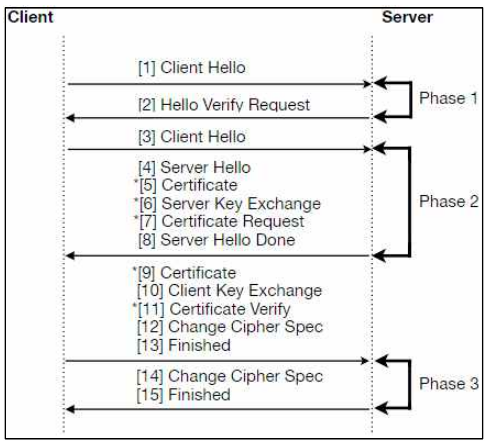


그림 3. DTLS Handshake Message

[그림 3]은 DTLS Handshake 메시지의 전 과정으로 Phase 1은 [그림 1]의 보안 협상 단계의 초반부로 클라이언트가 서버에게 DTLS Handshake를 시작함을 알리고 서버는 그에 대한 응답을 하게 된다. Phase 2는 보안 협상 단계의 후반부와 키 교환 단계

의 초반부로 이루어져 있으며 클라이언트와 서버 간 보안 협상을 완료하고, 서버는 클라이언트에게 세션 키를 도출하는데 사용되는 파라미터를 전송한다. 마지막으로 Phase 3는 키 교환단계의 후반부와 키 확인 단계로 이루어져 있으며 클라이언트는 서버에게 받은 파라미터를 기반으로 세션키를 생성하고 서버가 사용해야 할 파라미터를 전송하여 서버도 동일한 세션키를 생성하게 된다. 생성된 세션키를 사용하여 키 확인 단계를 거치고 클라이언트와 서버 간 DTLS 통신이 이루어지게 된다. *은 RPK와 Certificate 모드에서만 발생하는 메시지이다.

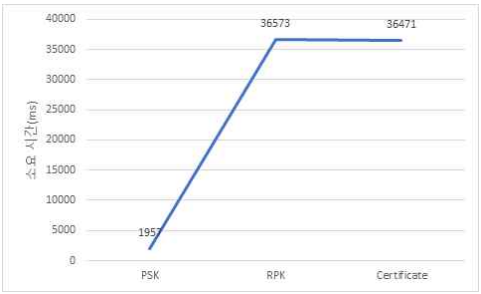


그림 4. DTLS Handshake 소요 시간

[그림 4]를 보면 PSK의 경우, 사전에 대칭키가 공유되어 있어 별도의 계산 시간이 필요하지 않기 때문에 다른 두 방식에 비하여 상당히 빠른 소요 시간을 보인다. RPK와 Certificate 방식의 경우 클라이언트와 서버의 상호 인증을 위한 EC(Elliptic Curve) 덧셈 연산이 수행되기 때문에 PSK에 비하여 상당한 시간이 소요된다.

4.3 RAM / ROM 소모량

본 절은 실제 사용되는 센서 노드의 하드웨어 자원과 DTLS 모드 별 RAM / ROM 소모량을 비교하여 사용 적합성을 논의한다. 본 성능 평가에 사용된 센서 노드는 16비트 TI MSP 430 시리즈의 Wismote를 사용하였다. Wismote는 램 16 킬로바이트, 롬 256 킬로바이트의 크기 제한이 있다. DTLS 모드 별 RAM / ROM 소모량을 알기 위해 Cooja에서 컴파일의 산출물인 .wismote 오브젝트 파일과 size 명령어를 사용하였다. 오브젝트 파일의 세션 정보를 출력하는 size 명령어는 해당 파일의 text, data, bss 세션 크기를 반환한다. text는 CPU가 실행할 Code부를 나타내고, data는 프로그램에 선언된 초기화 된 변수를 나타낸다. 마지막으로 bss는 값을 설정하지 않은 배열이나 Null 포인터와 같이 초기화 되지 않은 데이터로 구성된다.

표 2 ROM / RAM 계산 표(단위 : bytes)

인증 모드	text	data	bss	**ROM**	**RAM**
PSK	71513	342	8324	71855	8666
RPK	81963	350	11084	82313	11434
Certificate	83441	374	11084	83815	11458

[표2]의 출력된 text, data, bss 크기 값을 기준으로 ROM = text + data , RAM = bss + data로 계산하였다. 세 가지 모드 모두 Wismote의 사양에 적합하다고 평가되었다. 하지만 Wismote보다 더 자원이 제약된 sky mote 와 z1 mote와 같은 모트의 경우, 10 킬로바이트 미만의 RAM 제한을 가지고 있어 RPK와 Certificate의 경우 mote의 성능에 따라 사용이 제한될 수 있다고 보인다.

V. 결 론

본 논문에서는 LoWPAN이 연동되는 IoT 네트워크 아키텍처 관점에서 CoAP 프로토콜에 적용되는 DTLS 보안모드에 대한 분석 및 성능평가를 통해 IETF Use Case 별로 적정한 보안모드를 제시하였다. PSK, RPK 및 Certificate 보안모드는 자체적인 장단점을 가지고 있기에 각각의 적용 대상에 대한 선정은 향후 CoAP 기반의 다양한 응용환경에서의 보안 메커니즘 설계에 중요한 지침으로 활용될 수 있을 것으로 판단하고 있다.

[참고문헌]

[1] Z. Shelby, K. Hartke, and C. Bormann, “The Constrained Application Protocol,” IETF RFC 7252, June 2014.
[2] L. Seitz, S. Gerdes, G. Selander, M. Mani, and S. Kumar, “Use Cases for Authentication and Authorization in Constraint Environments,” IETF RFC 7744, Jan. 2016.
[3] E. Rescorla and N. Modadugu, “Datagram Transport Layer Security,” IETF RFC 6347, Jan. 2012.
[4] S. Raza, H. Shafagh, K. Hewage, R. Hummen, and T. Voigt, "Lithe: Lightweight Secure CoAP for the Internet of Things," IEEE Sensors Journal, 13(10), 3711-3720, October 2013.
[5] S. Raza, L. Seitz, D. Sitenkov, and G. Selander, "S3K: Scalable Security with Symmetric Keys - DTLS Key Establishment for the Internet of Things," IEEE Transactions on Automation Science and Engineering, 2016.
[6] R. Hummen, J. Ziegeldorf, H. Shafagh, S. Raza, and K. Wehrle, “Towards Viable Certificate-based Authentication for the Internet of Things,” in Proc. of the 2nd ACM Workshop on Hot Topics on Wireless Security and Privacy, pp. 37-42, 2013.
[7] Contiki Community, Cooja Simulator, http://www.contiki-os.org/start.html.
[8] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, “Transmission of IPv6 Packets over IEEE 802.15.4 Networks,” IETF RFC 4944, Sep. 2007.
[9] IEEE std. 802.15.4-2011, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), Standard for Information Technology Std., June 2011.
[10] S. Santesson, M. Myers, R. Ankney, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", IETF RFC 6960, 2015

난독화 방법에 따른 Clang 정적 분석 도구의 성능 영향 분석

진홍주*, 박문찬*, 이동훈*

*고려대학교 정보보호학과

Analysis of Detection Ability Impact of Clang Static Analyzer by Obfuscation Method

Hong Joo Jin*, Moon Chan Park*, Dong Hoon Lee*

*Graduate School of Information Security, Korea University

요 약

사물 인터넷 시장 규모가 급격하게 성장함에 따라 임베디드 시스템에서 가장 많이 쓰이는 언어인 C/C++ 언어의 사용도 증가하고 있다. C/C++ 언어의 코드 품질을 향상시키고 개발 비용을 절감시키기 위해 소프트웨어 생명주기의 전반부에서 수행 가능한 소프트웨어 검증기법인 정적 분석을 사용하는 것이 좋다. 많은 프로그램들이 정적 분석을 사용하여 소프트웨어의 안전성을 검증하고 있으며 많은 정적 분석 도구들이 사용, 연구되고 있다.

본 논문에서는 오픈 소스 프로그램인 Clang 정적 분석 도구를 사용하여 검증된 시험 코드에 대한 보안 약점 검출 성능을 확인하고, 난독화가 적용된 시험 코드에 대한 정적 분석 결과와 비교하여 난독화 방법에 따른 Clang 정적 분석 도구의 성능 영향을 분석한다.

I. 서론

각종 산업과 일상생활에 널리 활용되고 있는 사물인터넷(Internet of Things) 시장 규모는 매년 크게 성장하고 있으며, 사물인터넷 기기의 수 또한 급격하게 증가하고 있다[1]. IEEE Spectrum(Interactive: The Top Programming Languages 2017)에 의하면 임베디드 시스템에서 C/C++ 언어가 가장 많이 사용되고 있으며[2], 앞으로도 임베디드 소프트웨어 개발자를 위한 프로그래밍 언어로써 가장 선호하는 경향이 지속될 것이다.

하지만 C/C++ 언어는 버퍼 오버플로, 취약한 라이브러리의 사용, 안전하지 않은 포인터의 사용 등 보안 취약점을 가진다. 이러한 언어 규칙 위반을 탐지하는데 정적 분석 도구를 사용하는 것이 효과적이다. 정적 분석 도구는 많은 수의 프로그램을 비교적 짧은 시간에 분석할 수 있는 장점이 있지만, 코드가 커지거나 경로(path)가 복잡해지면 정적 분석 도구의 성능에 따라 보안 약점을 검출 못할 수 있다. 그렇기 때문에 소스 코드 난독화가 적용된 코드의 경우 일반 소스 코드보다 정적 분석이 어려워진다.

이 논문에서는 소스 코드 난독화가 적용 된 시험 코드를 이용하여 Clang 정적 분석 도구의 검출 능력이 달라짐을 확인하고, 이를 분석하여 성능 개선 방향을 제시한다.

II. 배경지식

2.1 Clang 정적 분석 도구

LLVM[3] 컴파일러의 Clang은 C/C++ 언어를 대상으로 특정 언어나 아키텍처에 종속적이지 않은 중간 언어(intermediate representation)를 생성하는 프론트엔드이다. 또한 Clang[4]은

이 성과는 2017년도 과학기술정보통신부의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2017R1A2B3009643)

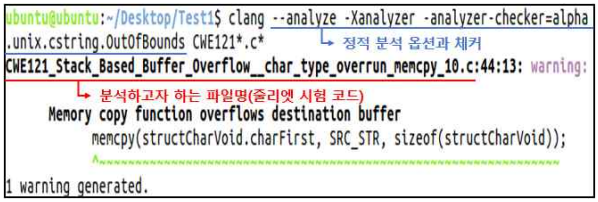


Fig. 1. Clang의 보안 약점 검출 예

C/C++ 프로그램에 대하여 정적 분석을 수행할 수 있는 모듈을 가지고 있으며, 이를 이용해 소스 코드가 가진 보안 약점을 검출할 수 있다. 정적 분석 옵션에는 체커(checker)를 선택하는 부분이 있으며, 분석하고자 하는 보안 약점에 알맞은 체커를 선택하여 정적 분석을 수행한다. 체커는 특정 보안 약점에 대해 정적 분석을 수행하는 기능 별로 분류되어 있으며 8개 체커 집합에 속한 95개 체커가 존재한다. 체커의 개발 방법과 체커 관련 소스 코드들은 공개되어 있으며 새로운 체커의 개발과 기존 체커의 보완 및 수정은 계속해서 진행되고 있다. Fig. 1.은 alpha.unix.cstring.OutOfBounds 체커를 이용해 버퍼 오버플로 보안 약점을 찾아 경고를 표시하는 예이다.

2.2 줄리엣 시험 코드

본 논문에서 사용한 정적 분석 대상 코드는 줄리엣 시험 코드(Test Suite v1.2 for C/C++)[5]로 NIST가 SAMATE[6] 프로젝트에서 정적 분석 도구들의 성능을 평가하기 위해 사용한 시험 코드이다. 줄리엣 시험 코드는 CWE(Common Weakness Enumeration)[7]를 기준으로 119개 보안 약점 집합으로 분류 되어있다. 각 집합의 1번 코드는 특정 보안 약점을 가지는 기본 코드이고, 코드 번호가 커질수록 이 기본 코드에서 제어 흐름이나 데이터 흐름의 복잡도가 증가하여 정적 분석이 어려운 코드가 만들어진다. 제어 흐름의 경우 조건문이나 반복문을 추가하여 경로가 복잡하게 만드는 방식을 사용하고, 데이터 흐름의 경우 지역 변수, 전역 변수, 함수의 매개변수 및 인수 등 사용하는 데이터에 접근하기 위한 경로가 복잡해지는 방법을 사용한다.

2.3 난독화

난독화 방법에 따른 Clang의 정적 분석 성능 영향을 분석하기 위해 줄리엣 시험 코드에 난독화를 적용한다. 난독화 방법은 적용 대상에 따라서 바이너리 난독화와 소스 코드 난독화로 나눌 수 있다. Clang은 소스 코드에 대하여 정적 분석을 수행하므로 줄리엣 시험 코드에 소스 코드 난독화를 적용하였다. 소스 코드 난독화 기술은 구획 난독화(Layout obfuscation), 데이터 난독화(Data obfuscation) 제어 흐름 난독화(Control flow obfuscation), 예방 난독화(Preventive obfuscation)로 구분할 수 있다[8].

줄리엣 시험 코드를 난독화하기 위해 Stunnix[9] C++ 난독화 도구를 사용하였다. 다른 난독화 도구들의 경우 난독화 정도를 설정하여 난독화를 수행하기 때문에 구획 난독화와 데이터 난독화 등 여러 난독화 방식이 같이 적용된다. Stunnix의 경우 식별자(Identifier), 문자열과 상수 값, 공백 문자와 주석 등에 대한 세부적인 난독화 옵션을 선택할 수 있었기 때문에 구획 난독화와 데이터 난독화를 분리하여 진행할 수 있었다. 이 논문에서는 원본 줄리엣 시험 코드와 구획 난독화, 데이터 난독화가 각각 적용된 줄리엣 시험 코드 3개의 집합을 정적 분석 대상 코드로 사용한다.

구획 난독화는 함수나 변수 등 식별자 이름을 변환하는 방식의 난독화 방법이다[10]. 프로그램의 성능에 적은 영향을 주는 장점이 있고, 간단하게 구현할 수 있다. Stunnix를 이용하여 줄리엣 시험 코드의 식별자 치환, 주석 제거, 공백 문자 섞기 등의 난독화 기법을 적용하였다. 데이터 난독화는 데이터의 형태나 구조를 변환하는 변수 분할, 문자열 암호화, 인코딩 형식 변환 등의 방법을 사용한다[10]. 데이터가 사용되는 목적을 감출 수 있는 장점이 있다. Stunnix를 이용하여 줄리엣 시험 코드의 문자열 및 정수, 상수 데이터를 변환하는 난독화 기법을 적용하였다.

III. 줄리엣 시험 코드에 대한 정적 분석

정적 분석은 리눅스 운영체제에서 수행하였

Table 1. 원본 및 난독화된 줄리엣 시험 코드에 대한 Clang의 정적 분석 결과(감소율은 감소한 경우만 표시)

CWE 이름	원본 시험 코드	검출수 (%)	구획 난독화(%)	감소율	데이터 난독화(%)	감소율
CWE-121 Stack Based Buffer Overflow	4,968	864(17.4)	674(13.6)	3.8	864(17.4)	-
CWE-122 Heap Based Buffer Overflow	5,922	1,011(17.1)	583(9.8)	7.3	1,011(17.1)	-
CWE-124 Buffer Underwrite	2,048	466(22.8)	324(15.8)	7.0	466(22.8)	-
CWE-126 Buffer Overread	1,452	27(1.9)	21(1.4)	0.5	27(1.9)	-
CWE-127 Buffer Underread	2,048	484(23.6)	332(16.2)	7.4	484(23.6)	-
CWE-188 Reliance on Data Memory Layout	36	18(50.0)	14(38.9)	11.1	18(50.0)	-
CWE-194 Unexpected Sign Extension	1,152	243(21.1)	189(16.4)	4.7	243(21.1)	-
CWE-195 Signed to Unsigned Conersion Error	1,152	366(31.8)	270(23.4)	8.4	366(31.8)	-
CWE-242 Use of Inherently Dangerous Function	18	0	0	-	0	-
CWE-369 Divide by Zero	864	54(6.3)	42(4.9)	1.4	54(6.3)	-
CWE-377 Insecure Temporary File	144	36(25.0)	0	25.0	36(25.00)	-
CWE-400 Resource Exhaustion	720	240(33.3)	216(30.0)	3.3	240(33.3)	-
CWE-401 Memory Leak	1,658	94(5.7)	56(3.4)	2.3	94(5.7)	-
CWE-404 Improper Resource Shutdown	384	48(12.5)	44(11.5)	1.0	48(12.5)	-
CWE-415 Double Free	962	462(48.0)	185(19.2)	28.8	462(48.0)	-
CWE-416 Use After Free	459	210(45.8)	0	45.8	210(45.8)	-
CWE-457 Use of Uninitialized Variable	948	472(49.8)	146(15.4)	34.4	472(49.8)	-
CWE-467 Use of sizeof on Pointer Type	54	54(100)	0	100	54(100)	-
CWE-468 Incorrect Pointer Scaling	37	0	0	-	0	-
CWE-469 Use of Pointer Subtraction to Determine Size	36	0	0	-	0	-
CWE-476 NULL Pointer Dereference	348	204(58.6)	109(31.3)	27.3	204(58.6)	-
CWE-561 Dead Code	2	1(50.0)	1(50.0)	0	1(50.0)	-
CWE-562 Return of Stack Variable Address	3	3(100)	2(66.7)	33.3	3(100)	-
CWE-563 Unused Variable	512	230(44.9)	134(26.2)	18.7	230(44.9)	-
CWE-571 Expression Always True	16	1(6.3)	1(6.3)	0	0	6.3
합 계	25,943	5,588(21.5)	3,343(12.9)	8.6	5,587(21.5)	-

으며 LLVM/Clang 3.8.0 버전의 정적 분석 도구를 사용하였다. Clang의 95개 체커 중 22개의 체커가 줄리엣 시험 코드의 보안약점을 검출할 수 있었고, 119개의 CWE로 분류 된 시험 코드 집합 중 25개의 집합, 총 25,943개 시험 코드에 대하여 정적 분석을 수행하였다. Fig. 1. 같이 특정 시험 코드 파일에서 보안 약점을 Clang의 체커가 검출하면 파일 이름과 위치, 해당 위치의 코드를 출력하여 표시해준다. 또한 체커마다 출력하는 경고 메시지를 가지고 있기 때문에 해당 코드 부분에서 어떤 보안 약점이 검출되었는지 알 수 있다.

3.1 원본 줄리엣 시험 코드 정적 분석

원본 줄리엣 시험 코드에 대하여 정적 분석을 수행한 결과는 Table 1.의 2, 3열에서 확인할 수 있으며 25,943개 코드 중 21.5%인 5,588개 코드에 대하여 보안 약점을 검출하였다. Clang은 상대적으로 복잡도가 낮은 하위 시험 코드(1번~18번)에 대해서는 대부분 검출할 수 있었지만 그 이상의 상위 번호에 대해서는 검출 능력이 떨어지는 것을 확인할 수 있었다. 이는 시험 코드의 제어 흐름 또는 데이터 흐름이 복잡해질수록 Clang이 보안 약점을 검출하지 못하는 것을 보여준다.

3.2 난독화가 적용 된 시험 코드 정적 분석

구획 및 데이터 난독화 방법에 의해 난독화 된 시험 코드에 대한 정적 분석 결과는 Table 1.의 4~7열에서 확인할 수 있다. 구획 난독화가 적용 된 시험 코드 중 3,343개의 보안 약점을 검출하였고 원본 코드에 비해 평균 8.6% 정도로

```
void CWE571_Expression_Always_True__two_equals_two_01_bad()
{
    /* FLAW: This expression is always true */
    if (2 == 2)
    {
        printLine("Always prints");
    }
}
```

```
void CWE571_Expression_Always_True__two_equals_two_01_bad()
{if((0xe43+4357-0x1f46)==(0x22+881-0x391)){printLine(
"%x41%x6c%x77%x61%x79%x73%x20%x70%x72%x69%x6e%x74%x73");}}
```

Fig. 2. 원본 시험 코드와 데이터 난독화가 적용 된 시험 코드(Expression Always True)

보안 약점 검출 수가 감소하였다. 줄리엣 시험 코드의 1번부터 18번까지의 하위 번호에 대해서는 구획 난독화가 적용 되더라도 큰 영향 없이 검출되었다. 하지만 전역 변수를 사용하거나 전처리문으로 선언 된 상수를 사용하는 등 데이터 흐름이 변화된 코드의 경우 Clang의 검출 성능에 영향을 주었다. 전역 변수의 경우 이 변수를 사용하는 영역이 많기 때문에 이를 추적하는 경로의 깊이(Depth)나 경우가 증가할 수 있다. 정적 분석 도구는 한정된 깊이만큼만 분석을 수행하기 때문에 검출 결과에 영향을 준 것으로 보인다. 또한 전처리문으로 선언 된 상수의 경우 분석 수행 시 전처리문을 먼저 처리하지 않고 소스 코드부터 분석하는 방식을 사용하기 때문인 것으로 판단된다.

또한 25개 보안 약점 중 Expression Always True에 대한 정적 분석 결과만 데이터 난독화의 영향을 받았다. Fig. 2.는 원본 줄리엣 시험 코드와 데이터 난독화가 적용 된 시험 코드 부분이다. 조건문의 조건 값 2가 난독화 되었고 경로 변화는 없다. 피연산자와 연산자의 복잡도가 늘어남에 따라 연산을 수행하지 못하고 정적 분석을 했기 때문에 해당 보안 약점을 검출하지 못한 것으로 보인다.

IV.결론

이 논문에서는 검증된 시험 코드를 이용하여 Clang의 정적 분석 성능을 확인하고, 구획 및 데이터 난독화가 적용된 시험 코드에 대한 검출 결과를 확인하였다. 또한 정적 분석 결과에 대한 분석을 통해 Clang이 난독화로 인해 보다 더 분석이 어려운 환경에서 보안 약점 검출 성능이 어떻게 저하되는지 확인하였다.

난독화 방법에 따른 Clang의 성능 영향을 분석한 결과를 Clang의 체커 개발 및 성능 개선 과정에서 참고할 수 있을 것이라 생각한다. 더 나아가 난독화 된 시험 코드가 가지는 특성을 이용하여 정적 분석 도구의 평가 기준으로 사용하거나 C/C++ 언어에 대한 시험 코드 개발 과정에서도 활용될 것으로 기대한다.

[참고문헌]

[1] DYOGRAM, http://www.dyogram.com/2017/04/internet-of-things-iot-market-potential-trends-in-2017-and-beyond/.

[2] IEEE Spectrum, https://spectrum.ieee.org/static/interactive-the-top-programming-languages-2017.

[3] Lattner. C. and Adve. V., “LLVM: A compilation framework for lifelong program analysis & transformation.” In Code Generation and Optimization, 2004., CGO, 2004. International Symposium on. IEEE, pp75-86, 2004.

[4] Clang, https://clang-analyzer.llvm.org/.

[5] SAMATE, Juliet Test Suite v1.2 for C/C++ User Guide, National Security Agency.

[6] SAMATE, https://samate.nist.gov/Main_Page.html.

[7] CWE, https://cwe.mitre.org/.

[8] Christian Collberg, Clark Thomborson, Douglas Low, A Taxonomy of Obfuscating Transformations, Technical Report #148, Department of Computer Science, The University of Auckland, 1997.

[9] Stunnix, http://stunnix.com/prod/cxxo/.

[10] W. Gregory, General Method of Program Code Obfuscation. PhD thesis, Institute of Engineering Cybernetics, Wroclaw University of Technology, 2002.

프린터 관리 시스템의 취약점 분석

지우중, 이경문, 이병천

중부대학교 정보보호학과

Security Vulnerability of Printer Management System

Woojoong Ji, Kyungmoon Lee, Byoungcheon Lee

Department of Information Security, Joongbu University.

요 약

디지털화가 가속되면서 학교나 도서관 등 공공기관에서 디지털 정보의 이용이 높아지고 있으며 프린트 서비스의 요구도 점점 증대하고 있다. 함께 사용하는 공공 PC에서의 프린터 사용료에 대한 과금은 관리자에게 꽤나 까다로운 일인데 이러한 까다로운 과금 업무를 자동화할 수 있도록 하는 프린트 관리 솔루션이 개발되어 현재 널리 사용되고 있다. 하지만 서비스 개발자 및 프린터 관리 솔루션 개발자들의 보안개발에 대한 인식은 아직 취약한 부분이 많이 있는 것으로 보인다.

이 논문에서는 현재 공공기관에서 사용되고 있는 프린터 관리 서비스의 취약성을 분석하였다. 로그인과 프린터 과금 금액 등 중요한 정보를 전송하는 경우 HTTPS 등의 보안통신을 적용해야 하지만 HTTP 평문통신을 사용하여 도청공격에 취약하며, 수신하는 정보가 실제 상대방에서 전송되어온 것인지에 대한 인증 및 검증이 필요한데 이런 과정을 생략함으로써 프록시를 이용한 변조공격이 가능함을 확인하였다. 아울러 우리나라의 취약점 신고 포상제도가 널리 취약점을 찾아내고 빠르게 개선하려고 하는 본래의 목적에 적합하지 못하다고 생각하여 이를 개선할 수 있는 방안을 제시하였다.

I. 서론

최근 공공 도서관에서는 도서관 보상금 제도[1,2,3]를 도입하여 시행하고 있는데 도서관의 소장자료 중 디지털 형태로 구매할 수 없는 자료를 출판년도와 무관하게 디지털화하고 발행 후 5년이 지난 자료는 도서관간에 전송하여 볼 수 있게 하되, 이용에 대하여 도서관이 소정의 보상금을 지불하도록 하는 제도이다. 서비스에 활용되는 수많은 저작물들에 대해 보상금을 지급하여 저작물을 적법하게 활용할 수 있도록 함으로써 이용자들이 저작권법을 지키도록 함과 동시에 자료 사용의 편의성을 도모하기 위한 시스템이다. 이 제도를 적용하기 위해 도서관은 저작물을 프린트하는 경우 사용내용에 대한 적절한 과금 시스템을 준비하여야 한다.

또한 공공장소에서 사용하는 프린터 자체의 관리도 매우 까다로운 일인데 소모품 관리, 프린터 정비, 사용내역 관리, 복사 및 인쇄 사용량에 대한 과금 등이 필요하다. 이러한 까다로운 프린터 관리 업무를 자동화할 수 있는 솔루션들이 개발되어 사용되고 있는데, 그 중 S사의 제품은 대학교 대략 40곳, 공공도서관 40여곳 등에서 사용되고 있다. 이 시스템에서는 사용자들이 계정을 생성하고 돈을 예치한 다음 프린트한 페이지의 수만큼 금액이 차감되는데 특별한 하드웨어 장치의 도입이 필요없이 네트워크 연결만 되면, 장소에 상관없이 프린트가 가능하다는 장점이 있다.

우리는 S사의 프린터 관리 시스템을 패킷분석, 중간자공격 등 네트워크 공격 기법을 이용하여 분석한 결과 보안통신을 사용하지 않아 로그인, 과금 등의 중요 정보가 쉽게 도청되는 취약점이 있으며, 전송되는 정보에 인증이 제공되지 않아 공격자가 프록시 중간자공격으로 데이터를 쉽게 변조할 수 있음을 확인하였다. 공격자가 남의 로그인 정보를 취득하면 남의 계정으로 프린터 서비스를 이용할 수 있고, 과금 데이터를 변조할 수 있게 되면 돈을 지불하지 않고 프린터 서비스를 이용하거나 충전 금액을 공격자가 임의로 변조하는 것도 가능하게 될 것이다.

II. 프록시를 이용한 중간자 공격

HTTP 프로토콜을 사용하면 서버와 클라이언트 사이에 데이터가 평문으로 전송되기 때문에 공격자가 중간에 패킷을 가로챈 다음 악의적인 행동을 할 수 있다. 먼저 본 논문에서 사용되는 공격기법과 용어들에 대해 설명하고자 한다.

2.1 도청 공격

가장 먼저 이루어져야 할 공격은 통신 내용을 도청할 수 있는 스니핑인데 이것은 네트워크를 통해 오고가는 패킷들을 제 3자가 동의 없이 패킷을 가로채서 보는 것을 말한다. 스니핑을 위해서는 통신 패킷이 공격자를 거쳐가도록 해야 하는데 본 논문에서는 ARP 스푸핑을 이용한다.

스푸핑이란, 악의적인 공격자가 자신의 신분을 속이는 것을 말한다. 스푸핑의 기법에는 ARP 스푸핑, IP 스푸핑, 쿠키 스푸핑, DNS 스푸핑 등의 기법들이 존재한다. 공격자는 ARP 스푸핑을 통해 공격 대상의 ARP 테이블을 변조시켜 그 컴퓨터의 모든 네트워크 패킷은 공격자의 컴퓨터를 거쳐가게 되고 공격자는 해당 패킷에 대해서 정상적으로 처리해 줌으로써 공격 대상 사용자는 아무런 의심없이 네트워크 통신을 계속 사용하게 된다.

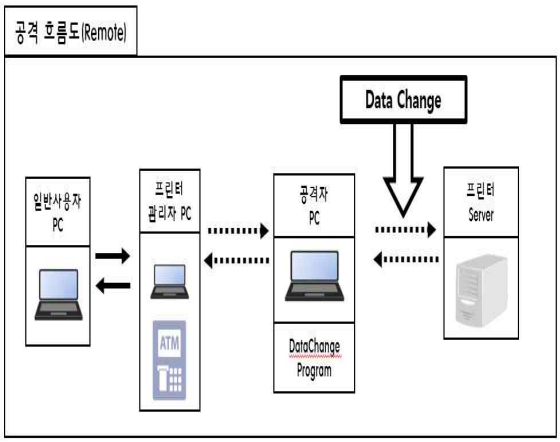
2.2 HTTP 중간자 공격

중간자 공격[4]이란 서버와 클라이언트 사이의 중간에 프록시 서버를 운영하면서 서버와 클라이언트 사이에서 HTTP 통신을 맺어 통신을 중개하면서 사용자의 프린터 과금 정보를 서버에 그대로 전달하도록 하는 공격 기법으로 사용자가 프린터를 사용하기 위해 지불한 금액 정보를 탈취할 수 있는 방법이다. 해당 취약점을 적용하기 위해 직접 프로그램을 구현하였으며 여기에서는 이것을 이용하여 HTTP 프록시 공격을 수행하는 사례를 보인다.

2.3 공격 시나리오

공격자가 운영하는 HTTP 프록시는 서버와 클라이언트 사이의 HTTP 통신을 중개하는 역할을 한다. [그림 1]에 전체적인 공격 시나리오를 도시하였다.

공격자가 ARP 스푸핑을 이용하여 프린터 관리 PC를 감염시킨다. 공격이 수행되면 공공기관에서 운영되는 프린터 관리 PC는 공격자의 PC를 거쳐서 프린터 서버로 통신을 하게 된다. 그 다음 프록시를 동작시키고 사용자가 프린터를 하기 위해 금액을 지불할 때까지 대기한다. 대기하고 있던 프록시에서 해당 사용자가 프린터를 이용하기 위한 금액을 지불하는 데이터가 거치게 될 때 해당 금액을 변조하여 실제 프린터 서버로 전송하게 된다. 만일 서버에서 클라이언트의 신분을 확인하고 통신데이터에 대한 인증을 검증한다면 이런 공격을 찾아내어 에러메시지를 내고 중단시킬 수 있겠지만 서버에서는 이러한 검증 시스템이 구현되지 않아 이런 중간자 공격은 훌륭하게 동작하게 된다.



[그림 1] 프록시를 이용한 공격

III. 프린터 관리 서비스 공격 사례

3.1 ID/Password의 노출 사례

일반적으로 ID/Password의 로그인 정보는 정보 노출을 방지하기 위해 HTTPS 프로토콜로 보내는 것이 일반적이다. 하지만 테스트중인 프린터 관리 서비스에서는 HTTPS 보안통신을 적용하지 않아 로그인 정보, 프린터 업무에 대한 정보, 금액 정보 등 모든 데이터가 HTTP 프로토콜로 서버와 데이터를 주고받는다. 그렇게 되면 프록시를 이용한 중간자 공격으로 통신 내용을 쉽게 도청할 수 있다.

```
POST /JPAMgr/servlet/InitServlet HTTP/1.1
Host: 220.81.62.73:7070
Connection: keep-alive
Content-Length: 51
Cache-Control: max-age=0
Origin: http://220.81.62.73:7070
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/61.0.3163.100 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/
webp,image/apng,*/*;q=0.8
Referer: http://220.81.62.73:7070/JPAMgr/servlet/InitServlet
Accept-Encoding: gzip, deflate
Accept-Language: ko-KR,ko;q=0.8,en-US;q=0.6,en;q=0.4
Cookie: JSESSIONID=B5A3BCD69A129712C0FA7FDFBA8946BC

svc=LOGIN&loginId=fafa9121&passwd=        &x=4&y=12HTTP/1.1 200 OK
```

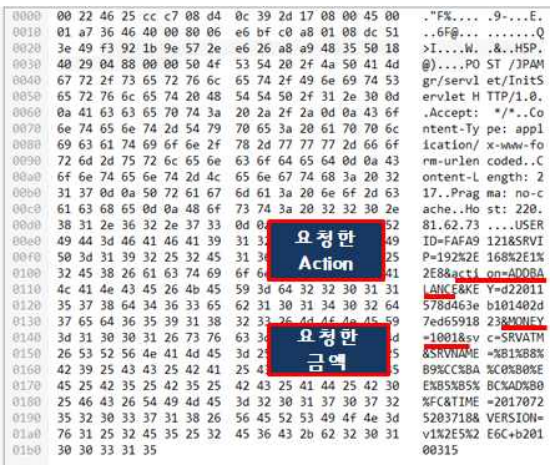
[그림 2] 프린터 서비스 로그인 정보 도청

3.2 과금 금액 노출 및 변조 사례

사용자가 프린터 관리 서비스에 로그인하게 되면 현재 사용자 계정의 충전 금액이 서버로부터 전송되어온다. 사용자가 ATM 기기를 이용하여 금액을 충전하게 되면 충전금액을 포함하는 평문의 데이터가 프린터 관리 PC에서 서버로 전송된다. 사용자가 프린트 작업을 수행하는 경우 현재의 충전금액에서 사용량을 차감하고 남은 충전금액을 서버로 전송하게 된다. 이러한 모든 통신과정이 평문상태로 전송되며 서버에서는 검증을 하지 않고 그대로 받아들이기 때문에 공격자가 마음대로 데이터를 변조할 수 있다.



(a) 사용자의 금액 충전



(b) 공격자에 의한 충전금액 변조

[그림 3] 프린터 관리 시스템의 금액 변조

[그림 3]은 사용자가 금액 충전시 공격자 PC에서 데이터를 원하는 금액으로 변조하여 프린터 서버로 전송하는 사례를 보여준다. 서버는 데이터의 인증성 여부를 검증하지 않고 그대로 받아들이기 때문에 사용자 계정에는 변조된 금액이 충전되는 것을 확인할 수 있다.

3.3 취약점 개선 방안

여기에서 확인된 취약점은 네트워크 기반의 클라이언트/서버 환경의 서비스에서 가장 기본적인 보안 대책인 HTTPS 통신을 사용하지 않는 것과 서버가 인가된 클라이언트에서 보내온 데이터인지 확인하지 않는 문제점 때문에 발생한다고 볼 수 있다.

이를 해결하기 위해서는 서버에 인증서를 설치하고 HTTPS 보안통신을 사용하도록 서버를 구성해야 할 것이다. 서버가 프린터 관리 PC를 인증하기 위해서는 프린터 관리 PC를 서버에 초기 등록시 인증서를 발급하여 장착하도록 하고 인증서로 서명된 통신만을 받아들이도록 운영해야 한다. 또한 프린터 관리 PC의 시스템보안, 해킹대응을 위해 인증서의 저장 및 서명작업은 하드웨어보안모듈을 이용하도록 구성할 수 있을 것이다.

IV. 취약점 신고 결과

이러한 취약점을 확인한 후 해당 취약점이 업체에 피드백되고 빠르게 개선하도록 하기 위해 KISA[5]에 취약점 제보를 하였다. KISA에서는 신규 취약점을 빠르게 발굴하고 개선해나갈 수 있도록 하기 위해 취약점 신고포상제[6]를 실시하고 있는 것으로 알고 있다. 그러나 취약점 담당자로부터 받은 응답은 “실제 서비스 중인 웹사이트나 시스템(서버, 네트워크, 보안장비 등)에 특정 데이터를 전송하여 영향을 줄 우려가 있는 서비스 취약점”은 평가 및 포상 대상에서 제외된다는 것이었다.

[한국인터넷진흥원-안내] 공동인쇄구역 취약점 분석(17-550)

보낸사람 ☆업무용(취약점신고)<vuln@krcert.or.kr>

받는사람 < @naver.com>

안녕하세요. 한국인터넷진흥원 취약점 담당자입니다.

신고해주신 " 취약점 분석"은 아래와 같은 사유로
평가 대상에서 제외되었음을 알려드립니다.
* 실제 서비스 중인 웹사이트나 시스템(서버, 네트워크, 보안장비 등)에 특정 데이터를 전송하여 영향을 줄 우려가 있는 서비스 취약점은 평가 및 포상 대상에서 제외

앞으로도 적극적인 신규 취약점 발굴 및 신고활동 부탁드립니다. 감사합니다.

기타 문의사항이 있으시면 연락주시기 바랍니다.(분석가 :

감사합니다.

[그림 4] KISA 취약점 포상 제외 사항

실제 서비스 중인 웹사이트나 시스템에 특정 데이터를 전송하여 영향을 줄 수 있다는 것 자체가 취약점이 있다는 것인데 이를 포상금을 떠나 취약점으로 신고받지 않고 평가대상에서 제외된다는 것은 애초의 취약점 신고포상제의 취지와 맞는지 고려해보아

야 한다.

애초에 버그바운티(Bug Bounty) 제도란 소프트웨어 또는 웹 서비스의 취약점을 찾아낸 사람에게 포상금을 지급하는 제도로서 신고포상제 운영이 자체 발굴보다 보안 연구비를 절감하고 대량의 취약점을 발견하는데 효율적이기 때문에 많은 기업들이 운영하고 있다. 외국에서는 기업의 서비스나 제품 등을 해킹해 취약점을 발견한 화이트해커에게 포상금을 지급하는 버그바운티 제도를 통해 빠르게 보안 패치를 적용하고 있다.

하지만 국내에서는 2012년 이 제도를 도입했지만 취약점을 밝히는 것을 꺼려하는 기업 문화와 보안 불감증, 보안 취약점이 발견된다면 부정적인 이미지로 비춰질 것을 우려된다는 등의 이유로 포상금도 적고 신고 건수도 저조한 상태이다. 공공기관인 KISA가 취약점 신고포상제를 시행한다면 이 제도에 부정적인 기업입장을 우선시하기보다는 취약점을 빠르게 발굴해내고 기업에서도 빠르게 보안패치를 해나갈 수 있도록 압박하는 애초의 목적에 맞게 운영했으면 하는 바람이다.

V. 결론

우리가 일상적으로 사용하는 많은 서비스, 어플리케이션, 솔루션 등이 아직까지 보안 프로토콜인 SSL/TLS, HTTPS를 적용하지 않고 운영되는 경우가 많으며, 클라이언트/서버 환경에서 인가된 클라이언트가 보내는 데이터인지를 서버에서 확인하지 않는 등 취약한 방법으로 운영되는 사례가 많다는 것을 확인하였다. 서비스, 어플리케이션, 솔루션 개발자들뿐만 아니라 이러한 제품을 취급하는 회사들은 이러한 취약성들에 대해 인식하고 보안기술을 적절히 활용하여 신뢰성 있는 제품을 구현하려는 노력이 필요하다고 생각된다. 공공기관인 KISA가 운영하는 취약점 신고포상제는 이 제도에 부정적인 기업입장을 우선시하기보다는 취약점을 빠르게 발굴해내고 기업에서도 빠르게 보안패치를 해나갈 수 있도록 압박하는 애초의 목적에 맞게 운영했으면 하는 바람이다.

[참고문헌]

[1] 저작권법, 법률 제14634호

[2] 이영아, “개정 저작권법상 도서관 보상금 제도”, 문화관광부

[3] 정경희. "도서관보상금제도의 운영성과에 대한 분석." 한국문헌정보학회지, 49.4 (2015.11): 265-288.

[4] Man-in-the-middle attack, https://www.owasp.org/index.php/Man-in-the-middle_attack

[5] KISA 한국인터넷진흥원, https://www.kisa.or.kr

[6] 한국인터넷진흥원, S/W 신규 취약점 신고포상제 운영 안내서

다크 웹 크롤링을 위한 지능형 크롤러 설계

노인우*, 권구환**, 임을규
*한양대학교 컴퓨터소프트웨어학과
**한양대학교 정보보안학과
한양대학교 컴퓨터소프트웨어학과

Intelligent Crawller for Dark Web Crawling

In-Woo Ro*, Ku-Hwan Kwon**
*Division of computer and software, Hanyang University.
**Division of information Security, Hanyang University.
Eul Gyu Im
Division of computer and software, Hanyang University

요 약

웹 브라우저는 개발자 모드에서 네트워크 로그를 HAR(HTTP Archive) 포맷으로 저장하는 기능을 제공한다. 본 논문에서는 HTTP Archive의 구조와 이를 파싱하여 Python 소스코드 형태의 Crawling Library를 자동 생성하는 기법에 대해 설명한다.

I. 서론

데이터 활용 및 분석은 더 이상 데이터 마이닝이나 분석전문가의 영역이 아니며 학생부터 연구자, 웹 서비스 개발을 하는 기업에 이르기까지 다양한 계층에게 있어 기본적으로 요구되는 기술이 되었다. 데이터를 활용하기 위해서는 우선 데이터를 보유해야 하지만 필요로 하는 데이터가 항상 public한 데이터 셋 형태로 존재하지는 않는다. 따라서 많은 경우 웹 크롤링을 통해서 필요로 하는 데이터를 웹에서 수집한다.

이 때 두 가지 문제가 존재한다. 하나는 웹 서비스 데이터 배포 방식이 업데이트를 겪으면서 변경 될 수 있다는 점이며(Crawler 입장에서는 입력 데이터의 무결성을 보장 받지 못 한다), 나머지 하나는 일부 웹 서비스의 경우 크러울러 개발을 위해서 서비스 구조 분석에 많은 시간을 소요하는 형태로 개발된다는 점이다.

본 논문에서는 이를 해결하기 위한 방법으로 HTTP Archive 분석을 통한 Intelligent Crawler 자동생성 기법과 응용을 통한 활용 방법으로서 Dark Web 대상 크롤링을 제시한다.

II. 관련연구

신재호 외 2명은 자바스크립트를 사용하여 링크 관계를 설정하고 있는 인터넷 쇼핑몰에 대하여 기존의 오픈소스 기반의 웹 크롤러들에 의해 수집되지 못하여 크롤링이 불가능한 문제점을 착안하여 자바스크립트 에뮬레이터를 사용한 웹 페이지 크롤링 기법을 제안하였다.[1] 해당 논문에서는 사용자가 직접 웹 페이지로부터 링크 정보를 추출하고 특정 사이트의 네비게이션 정보를 XPath로 특정한 후, 크롤러에게 XPath를 따라가면서 크롤링을 진행하도록 하는 방법을 사용하였다.

Tianjun Fu 외 2인은 인터넷의 성장으로 인한 다양한 범죄의 문제가 되고 있는 다크 웹(Dark Web)에 대하여 상용화 되고 있는 웹 크롤링 기술로는 웹에 대한 수집이 어렵다는 점을 착안하였다. 따라서 다크 웹 포럼 컨텐츠를 수집하기 위한 새로운 크롤링 시스템을 제안하

였다.[2] 해당 논문에서는 URL 토큰 앵커, 텍스트 링크 수준에 따라 크롤링 공간 URL의 순서의 BFS와 DFS를 이용하여 크롤링을 진행한다. 또한 업데이트된 컨텐츠를 식별하기 위해 랩퍼(wrapper)를 사용하여 컬렉션을 업데이트 하기 위해 증분 크롤링을 이용한다.

서동민 외 1인은 빅데이터 분야가 성장함에 따라 수집된 데이터가 매우 다양하고 데이터의 양이 기하급수적으로 증가하고 있다. 따라서 크롤러 중요성이 증가함에 따라 수집된 웹문서의 효율적인 관리와 중복된 웹 문서를 수집하는 문제점을 착안하여 분산형 웹 크롤러를 제안하였다.[3] RSS와 Google Search API를 통해 기존 웹 크롤러의 문제를 해결하고 RMI와 NIO을 활용하여 서버와 클라이언트 간 네트워크 연결을 최소화 하여 수집 속도가 빠른 크롤링 기능을 제공한다.

III.HTTP Archive

[그림 1]은 HAR의 구조를 나타내며, Internet Explorer와 Chrome 등의 브라우저에 기본 내장된 개발자 도구는 네트워크 로그를 HAR(HTTP Archive) 포맷으로 출력하는 기능을 제공한다. 개발자 도구에서 네트워크 로그를 저장하면 .har 확장자의 json 형태의 로그가 출력된다. har 로그는 로그 자체에 대한 메타 정보와 통신 내용에 대한 entries 정보로 구성된다. 메타 정보에는 생성 브라우저 종류와 버전, 생성 시간과 관련된 정보가 포함되며, entries 부분에는 각각의 entry가 시작 시간, 소요시간, request와 response 4개의 세부 데이터를 포함한다. entry 내의 request와 response 아이템에는 개발자 도구를 통해서 확인 가능한 모든 데이터가 포함되어 있다. 이는 개발자가 har 로그를 확인함으로써 실제 사이트 동작을 개발자 도구를 통해서 실시간 관찰한 것과 같은 양의 정보를 얻을 수 있다는 것이며, 나아가서 parsing을 통해서 크롤러 개발에 필요한 데이터를 자동 추출하고, 크롤링 코드 작성 까지도 반자동화하는 것이 가능하다는 것을 의미한다.

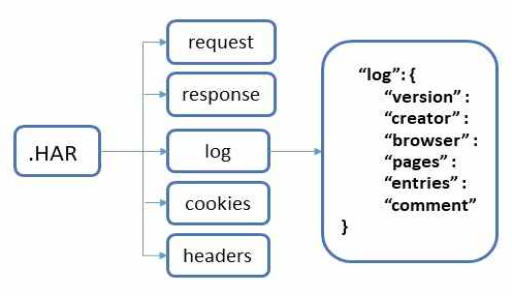


그림 1. HTTP Archive Format(HAR)

IV. Generating Crawling Library: HAR2LIB

본 장에서는 지능형 크롤러의 예시로서 har 로그 파일을 로드하여 필요 정보를 파싱하고, 이를 바탕으로 크롤링 파이썬 코드를 자동 생성하는 har2lib 패키지를 소개한다.

다음 [그림 2] 는 본 논문에서 구성한 HAR2Lib에 대한 플로우차트(flow-chart)를 나타낸다.

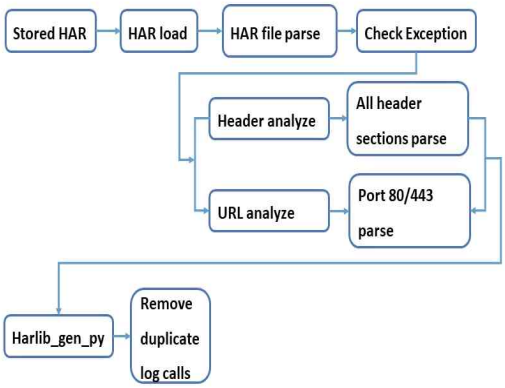


그림 2. HAR2lib.py FlowChart

4.1 Parsing HAR Files

HAR2LIB 패키지에서는 harlib 클래스를 제공하며, 객체 생성 후 har 로그 파일을 로드하고 파싱한다. 파싱 단계에서는 1) 예외 처리 2) 헤더 분석 3) URL 분석으로 나눠서 진행한다. 헤더 분석 단계에서는 har에 포함된 모든 헤더 섹션을 대상으로 파싱을 수행하며, URL 분석 단계에서는 80/443 포트 중 다음 항목들만을 대상으로 파싱을 수행한다.

- url
- url_protocol
- url_domain_port
- url_domain
- url_port
- url_path
- url_arguments
- url_title

4.2 Generating Crawling Library in Python

파싱이 완료되면 harlib._gen_py 메소드를 호출하여 해당 사이트에 대한 크롤링 메소드를 담고 있는 python 클래스를 작성한다. 메소드는 har 로그 상에서 중복 호출이 있을 수 경우 최초 호출된 것을 기준으로 메소드 이름이 중복되지 않게 생성한다. 크롤러 개발에 있어서 가장 시간 소요가 큰 부분은 HTTP 헤더를 브라우저에서 전송하는 것과 동일하게 manipulate하는 부분인데, 이 과정을 harlib가 자동으로 수행한다. 생성된 클래스는 어디 까지나 크롤링에 필요한 메소드를 포함할 뿐이며, 실제로 이를 호출하는 비즈니스 로직은 직접 개발해야 한다. 이는 크롤링에 필요한 데이터를 코드로 작성하는 부분을 자동화 하는 것과, 크롤링 시나리오를 짜는 것은 별개의 이슈이기 때문이다.

V. Crawling Scenario with Intelligent Crawler

harlib가 크롤링 시나리오를 직접 제시하지는 않지만 automated delay와 같이 크롤링 시나리오를 작성하는데 도움을 주는 기능은 있다. 본 절에서는 웹서비스 데이터 크롤링 시나리오에서 고려해야 하는 사항과 이에 대한 해결책으로서의 크롤링 시나리오에 대해서 설명한다.

5.1 Anti-Crawling Method for Web Service

웹 서비스 개발에서 과도한 데이터 크롤링을 하는 비정상 사용자를 감지하고 자동으로 black list를 만들어 관리하는 것은 서비스 운용 차원에서나 기업 재산 보호 차원에서 매우 중요하다. 본 절에서는 Anti-Crawling 기법들을 소개한다.

5.1.1 HTTP 헤더 확인

일반적인 크롤러의 경우 브라우저에서 사용하는 http 헤더를 사용하지 않는다. 웹서비스 서버에서는 클라이언트에서 보내오는 요청의 http 헤더를 확인하여 User-Agent 등의 값이 정상적으로 들어있는지 확인하여 정상 사용자와 크롤러의 요청을 구분한다.

5.1.2 Access Frequerncy Recognition(접근 빈도 인식)

크롤러를 통해서 웹서비스의 전체 데이터를 복제하려는 공격자의 경우, multi process 혹은 multi device를 사용하여 고속으로 데이터를 수집한다. 이는 대상이 되는 웹서비스의 데이터를 복재할 뿐 아니라, 서비스 자체의 성능을 저하시키거나 심할 경우 마비시키기도 한다. 접근 빈도 인식 기법은 특정 클라이언트가 데이터에 접근하는 요청의 빈도를 백엔드에서 인식하고 일정 threshold를 넘어서면 IP 차단 등의 기법으로 대응하는 방식이다. 하위 섹션에서 설명하는 접근패턴인식 방식과의 병행 없이 이 방식만을 사용할 경우 정상적인 사용자에 대해 크롤러인 것으로 False-Positive 오류를 발생시킬 수 있다. 예를 들어 하나의 IP 공유기를 사용하는 사무실에서 특정 업무를 보기 위해 직원들이 한 서비스에 동시 접속할 경우 웹서버는 하나의 IP에서 동시에 일반적인 수준의 수십배에 달하는 요청이 들어오는 것을 보고 공격자로 오인 가능하다.

5.1.3 Access Pattern Recognition(접근 패턴 인식)

공격적인 크롤링을 수행하는 공격자는 해당 웹서비스에서 필요로 하는 핵심 데이터를 사전에 정의하고 불필요한 html, js, css 등의 리소스를 요청하지 않고 특정 데이터를 요청하는 ajax 호출만을 크롤러로 구현한다. 이에 대한 방어가 없을 시, 공격자에 의해 웹서버는 특정 ajax 호출만을 연속적으로 대량 처리해야 한다. 예를 들어 대한민국 소재의 560만개 지번에 대한 부동산 전수 데이터를 보유한 웹서비스의

데이터를 모두 복재하려는 공격자라고 가정하자. 해당 웹서비스는 지번을 검색하면 html, js, css 파일을 클라이언트에 넘기고, html table form을 채워주는 데이터를 ajax 호출로 동적 완성한다. 공격자는 ajax 호출만을 python 등의 스크립트로 구현하고, 공개된 지번 주소 목록을 통해 확보하여 병렬적으로 서버에 데이터를 요청할 수 있다. 주의 깊은 공격자라면 서버의 처리 성능 한계에 근접하지 않도록 하겠지만 그렇지 않다면 자신이 원하는 데이터를 최대한 빠른 시간 내에 얻을 수 있도록 웹서비스가 마비되기 직전 까지 수집 프로세스를 병렬화 할 것이다.

접근 패턴 인식은 상기 접근 빈도 인식과 더불어 동적 분석을 통한 Anti-Crawling 기법이다. 클라이언트가 정상적으로 요청해야 하는 리소스에 대한 호출 없이 ajax 호출만을 연속적으로 한다면 이를 블록한다. 접근 패턴 인식 방식을 고도화 하여 적용한 웹서비스의 경우는 사용자 UX 관점에서 서비스를 일련의 연속적인 요청들의 집합으로 보고, 같은 집합 내에 속하는 요청과 응답을 Cookie에 특정 Hash 값을 포함시키는 방식으로 chaining하여 이 중 일부 요청을 별도로 할 수 없도록 구성한다.

5.2 Intelligent Crawler against Anti-Crawler Method

앞 4장에서 설명한 har2lib를 통해서 생성된 지능형 크롤러 라이브러리를 활용할 경우, 앞서 설명한 3가지 anti-crawling 방식에 다음과 같이 모두 대응 가능하다.

5.2.1 HTTP 헤더 복제

har2lib에 의해서 생성된 모든 요청 method는 정상적인 브라우저에 의해 요청한 것과 동일한 http request header를 자동으로 탑재한다. 웹서비스는 http 헤더 확인으로 크로울러와 브라우저를 구분할 수 없다.

5.2.2 접근 빈도 자동 설정

har2lib는 har 분석 과정에서 각 method 별로 실제 소요된 시간을 파싱하여 method 별로 automated dealy time value를 설정한다. 모든 method는 auto dealy 옵션을 True로 줄 경우 실제 소요된 것과 같은 delay를 sleep 함수를 통해 manipulate한다. 이 설정을 적용한 후 다중 IP 프록시 서버와 결합하면 IP 당 접근빈도 한계를 넘기지 않는 크롤링 유닛으로 구성된 병렬 크롤링 네트워크를 구성 가능하다.

5.2.3 접근 패턴 가이드

har2lib는 비록 크롤링 시나리오를 직접 작성해주지는 않지만, ajax 호출 복제만으로 구성된 시나리오가 접근패턴인식에 의해서 블록 당할 경우에 대해서도 가이드 형태로 지원한다. har 로그 자체가 정상적인 접근 패턴을 기록한 것이기 때문에 이를 바탕으로 접근 패턴을 별도의 도표로 요약 가능하다.

VI. 결론 및 향후 계획

har2lib를 이용해서 기존의 크롤링 방식으로 데이터를 수집하지 못 하는 웹 서비스에 대해 데이터를 수집하는 실험을 수행한다. 이는 실험 방식의 객관성 뿐 아니라 합법적인 테두리 내에서 수행해야 하는 이슈로 적절한 대상과 실험 방식을 선정하는 대로 수행 할 예정이다.

[참고문헌]

[1] 신재호, 김태우, 김철연. “XPath를 기반으로 한 특정 온라인 쇼핑몰 크롤링 기법 구현”, 한국정보과학회눈문지, pp. 240-242, 2014

[2] Tianjun Fu, Ahmed Abbasi, Hsinchun Chen. “A focused crawler for Dark web forums” Journal of the American Society for Information Science and Technology, Volume 61, pp.1213-1231, 2010

[3] 서동민, 정한민. “빅데이터 분석 서비스 지원을 위한 지능형 웹 크롤러”, 한국콘텐츠학회논문지, vo1. 13, pp.575-584, 2013

2017 한국정보보호학회 동계학술대회

D2 해킹 및 취약점(II)

강 병 탁 (에이아이스페라)

- **다중 바이트 간 독립성 검정을 이용한 소프트웨어 잡음원의 엔트로피 추정방법**
  박호중, 김원태, 염용진, 강주성 (국민대학교)
- **기존 NTRU 전력 분석 방법 실제 적용**
  안수정, 김수리. 김희석, 홍석희 (고려대학교)
- **안드로이드 앱의 호출 흐름 무결성 검증 기법**
  최준태, 조홍래, 하동수, 오희국 (한양대학교)
- **기계학습을 이용한 행위 기반의 악성코드 판별 모델**
  박의성 (목포대학교), 진수현 (동의대학교), 지한별, 이주연,
  국광호 (서울과학기술대학교)

다중 바이트 간 독립성 검정을 이용한 소프트웨어 잡음원의 엔트로피 추정방법

박호중, 김원태, 염용진, 강주성†

국민대학교 정보보안암호수학과 / 금융정보보안학과

An entropy estimation method of software noise source by using the independence test for multi-byte

Hojoong Park, Wontae Kim, Yongjin Yeom, Ju-Sung Kang†

Dept. of Information Security, Cryptology, and Mathematics / Financial information security, Kookmin University

요 약

암호시스템의 안전성을 보장받기 위해서는 잡음원으로부터 충분한 엔트로피를 확보하는 것만큼 잡음원의 엔트로피를 정확히 추정하는 것이 중요하다. 엔트로피 수집에 사용되는 대부분의 잡음원이 하드웨어에 의존적이지만 하드웨어 잡음원 수집에 제한적인 환경에서는 소프트웨어 잡음원을 이용하여 시스템의 엔트로피를 확보한다. 한편, 현존하는 NIST SP 800-90B의 엔트로피 추정 알고리즘은 비트 단위의 하드웨어 잡음원에 초점이 맞추어져 있으며, 메모리의 제약성 때문에 워드 단위로의 확장성이 현실적으로 불가능하다. 본 논문에서는 소프트웨어 잡음원의 특성에 맞추어 워드 단위에서도 엔트로피 추정이 가능한 방법을 새롭게 제안한다. 먼저 다중 바이트 간 독립성을 평가할 수 있는 통계적 검정법을 제안하고, 이 검정법과 정보이론을 바탕으로 하여 워드 단위의 엔트로피를 추정할 수 있는 알고리즘을 제시한다.

I. 서론

암호시스템의 안전성을 보장받기 위해서는 잡음원으로부터 충분한 엔트로피를 수집하는 것뿐만 아니라, 엔트로피의 잘못된 추정으로 인한 시스템의 취약성[1]을 방지하기 위해 잡음원의 엔트로피도 정확히 추정되어야 한다. 엔트로피 수집에 사용되는 대부분의 잡음원은 열 잡음원(Thermal noise), 양자, 방사성 붕괴 등과 같이 하드웨어에 의존적이지만[2], IoT와 같이 하드웨어 잡음원 수집에 제한적인 환경이 증가하면서 소프트웨어 잡음원에 대한 연구가 증가하고 있다. 대표적인 소프트웨어 잡음원에 관한 연구로는 소프트웨어 잡음원의 활용에 대한 [3]과 CMVP IG의 Heuristic 분석을 활용하여 소프트웨어 잡음원의 엔트로피를 추정하는 방법에 대한 연구인 [4]와 소프트웨어 잡음원 평가에 적합한 알고리즘을 제시한 [5] 등이 있다.

한편, 기존 잡음원의 엔트로피 평가방법인 NIST SP 800-90B[2]는 비트 단위의 하드웨어 잡음원에 초점이 맞추어져 있으며, 메모리의 제약성 때문에 워드 단위로의 확장성이 현실적으로 불가능하다. 본 논문에서는 소프트웨어 잡음원의 특성에 맞추어 워드 단위에서도 엔트로피 추정이 가능한 방법을 새롭게 제안한다. 먼저 다중 바이트 간 독립성을 평가할 수 있는 통계적 검정법을 제안하고, 이 검정법과 정보이론을 바탕으로 하여 워드 단위의 엔트로피를 추정할 수 있는 알고리즘을 제시한다.

† 교신저자

II. 다중 바이트 간 독립성 검정 및 엔트로피 추정 알고리즘의 설계 원리

Ⅱ장에서는 논문에서 제안하는 엔트로피 추정 알고리즘에 적용된 이론적 원리에 대해 작성한다. 제안하는 알고리즘은 임의의 바이트 크기를 가진 워드에 적용할 수 있지만, 본 논문에서는 설명의 편의를 위해 1 워드의 크기를 4바이트로 가정하고 설명함을 밝힌다. 제안하는 알고리즘은 가설검정, 상호 독립, 비모수적 통계적 방법, 독립성과 엔트로피의 관계식, 이 네 가지 원리를 이용하여 설계되었다. 다음 소단원부터는 제안하는 알고리즘에 적용된 원리에 대해 작성한다.

2.1 가설검정(Hypothesis test)

가설검정은 설정된 귀무가설(이하, H_0)에 대한 반증을 찾아내는 방법으로, 통계적 검정은 H_0를 '표본수열이 랜덤하다'로 설정하고, 이에 상정하는 H_1을 '표본수열이 랜덤하지 않다'로 설정하여 H_0에 반대되는 증거를 찾아내는 가설검정법이다. 본 논문에서 제안하는 알고리즘에 적용되는 가설검정은 다음과 같다.

H_0: 워드 내 모든 바이트가 상호독립이다.
H_1: 워드 내 모든 바이트가 상호독립이 아니다.

본 논문에서 제안하는 알고리즘은 카이제곱 통계량을 이용한 독립성 검정방법으로 유의수준(α)이 10^{-3}으로 설정된 가설검정이 진행된다.

2.2 상호 독립(Mutually independent)

상호독립은 모든 사건이 다른 부분 사건들과 독립을 의미한다. 이를 수식으로 표현하면 다음과 같다.

$$P\left(\bigcap_{i=1}^{4} A_i\right) = \prod_{i-1}^{4} P(A_i)$$

제안하는 알고리즘은 상호독립을 이용한 독립성 검정으로 두 바이트 간 독립성 검정 6회($_4C_2 = 6$), 세 바이트 간 독립성 검정 4회($_4C_3 = 6$), 네 바이트 간 독립성 검정 1회($_4C_4 = 1$)로 진행된다. 11회의 독립성 검정에서 모두 독립으로 판정된 경우에 대해서만 알고리즘은 잡음원을 독립으로 판정한다.

2.3 비모수적 방법(Nonparametric statistical method)

비모수적 방법이란 모집단에 대한 정보가 없는 분포를 추정할 때 사용하는 방법이다[6]. 잡음원의 분포는 미지의 분포이기 때문에 비모수적 방법을 적용하여 잡음원의 엔트로피를 추정하는 것이 타당하다. 제안하는 알고리즘에서 비모수적 방법이 적용된 부분은 독립성 검정으로, 모수적 방법과 가장 큰 차이점은 자유도 계산에 있다. 모수적 방법을 이용한 독립성 검정의 자유도 계산은 일반적으로 범주의 수에서 1을 뺀 만큼이 검정의 자유도로 계산되었으나, 비모수적 방법에서는 미지의 분포에 대한 추정이기 때문에 범주의 수에서 추정된 모수의 수를 빼야 한다. 비모수적 방법을 적용하였을 때 자유도 계산은 다음과 같다.

자유도 = # (범주의 수) - # (추정된 모수의 수)

2.4 독립과 엔트로피와의 관계

출력된 잡음원 워드의 각 바이트 위치에서 발생하는 값은 알 수 없으므로, 워드로 출력된 잡음원의 바이트 위치는 확률변수 X_1, X_2, X_3, X_4로 표현할 수 있다. 이때 4개의 확률변수에 대한 결합 확률분포(Joint probability distribution)의 엔트로피는 주변 확률분포(Marginal probability distribution)의 엔트로피의 합보다 작거나 같다. 또한, 등호는 확률변수 X_1, X_2, X_3, X_4가 독립일 때 성립한다[7].

$$H_\infty(X_1, X_2, X_3, X_4) \leq H(X_1) + H(X_2) + H(X_3) + H(X_4) \cdots (1).$$

제안하는 알고리즘은 정보 이론적으로 증명된 수식 (1)에 근거하여 소프트웨어 잡음원의 엔트로피를 추정한다.

III. 제안하는 엔트로피 추정 알고리즘

제안하는 다중 바이트 간 독립성 검정 및 엔트로피 추정 알고리즘은 Ⅱ장에 작성한 원리에 의해 설계되었다. 제안하는 엔트로피 추정은 크게 단일 바이트의 엔트로피 추정, 독립성 검정, 엔트로피 추정의 세 단계로 구성된다. 독립성 검정은 두 바이트, 세 바이트, 네 바이트 순으로 진행되며, 모두 독립으로 판정된 경우에는 수식 (1)의 등호 조건을 만족한다고 판단하여 추정된 네 바이트의 엔트로피를 더하여 출력한다. 이때, 독립으로 판정되지 않은 경우 다음 단계로 진행하지 않고, 독립으로 판정된 바이트 엔트로피 합의 최댓값을 워드의 엔트로피로 추정한다. 제안하는 독립성 검정 알고리즘과 이를 활용하여 다중 바이트의 엔트로피를 추정하는 알고리즘에 대한 설명을 위해 세 바이트에 대한 독립성 검정 및 엔트로피 추정방법을 각각 [표 1]과 [표 2]에 예시로 제시하였다.

[표 1] 세 바이트의 독립성 검정 알고리즘

알고리즘 1. 세 바이트의 독립성 검정

입력 : 소프트웨어 잡음원 표본수열 S

출력 : 독립/종속

※ 가정 : 두 바이트 간에서 독립으로 판정

1. 세 바이트 간 독립성 검정 단계

$1 \le i,j,k \le 4, (i \ne j \ne k)$인 임의의 i,j,k 열에 대한 X_i, X_j, X_k의 독립성 검정.

1.1 $l,m,n \in \{0,1,...,255\}$일 때, 난수열에서 $x_{s_l}^i, x_{s_m}^j, x_{s_n}^k$가 발생할 비율 $p_{s_l}^i, p_{s_m}^j, p_{s_n}^k$과 그 쌍이 발생할 기댓값을 계산.

$E_{s_l,s_m,s_n} = p_{s_l}^i p_{s_m}^j p_{s_n}^k \times$ (표본수열의 길이)

1.2 기댓값이 5 미만인 바구니는 하나의 바구니로 통합하여 최종 바구니 수 q를 계산하고, $1 \le i \le q$에 대해 순서대로 E_i로 명명.

1.3 카이제곱 통계량을 계산.

$$T = \sum_{i=1}^{q} \frac{(O_i - E_i)^2}{E_i}.$$

1.4 $T > \chi_\alpha^2(df)$면 종속으로 판정.

이때 자유도는 $df = q - 766$.

[표 2] 세 바이트의 엔트로피 추정 알고리즘

알고리즘 2. 세 바이트의 엔트로피 추정

입력 : 소프트웨어 잡음원 표본수열 S

출력 : 워드의 최소 엔트로피

※ 가정 : 두 바이트 간에서 독립으로 판정

1. 세 바이트 간 독립성 검정 단계

$1 \le i,j,k \le 4, (i \ne j \ne k)$인 임의의 i,j,k에 대한 X_i, X_j, X_k의 독립성 검정.

1.1 모두 종속으로 판정된 경우, 두 바이트 엔트로피 합의 최댓값을 출력.

$H_\infty(X_1,X_2,X_3,X_4) =$

$\max(H_\infty(X_i) + H_\infty(X_j)), 1 \le i,j \le 4.$

1.2 일부만 종속으로 판정된 경우, 독립으로 판정된 세 바이트 엔트로피 합의 최댓값을 출력.

$H_\infty(X_1,X_2,X_3,X_4) =$

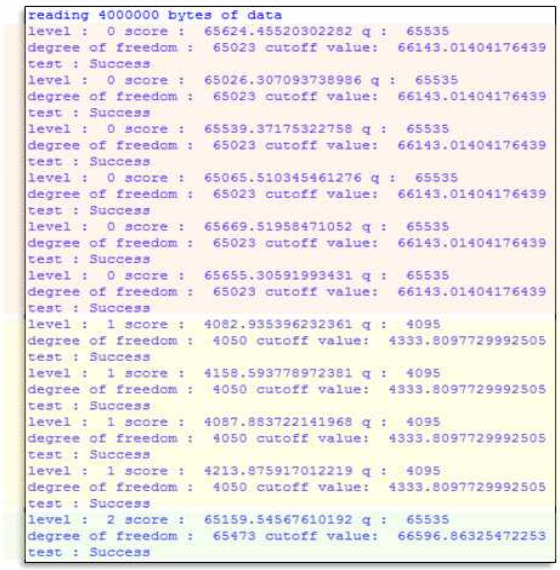
$\max(H_\infty(X_i) + H_\infty(X_j) + H_\infty(X_k))$

1.3 모두 독립으로 판정

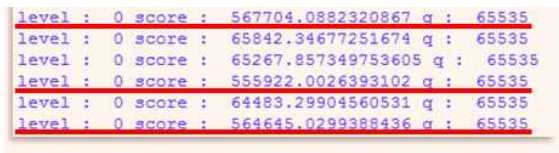
네 바이트 간 독립성 검정 진행.

IV. 알고리즘 유효성 검증 실험

Ⅳ장의 유효성 검증 실험은 설계한 알고리즘이 데이터의 독립성을 정확히 판정하는지 확인하는 실험이다. 알고리즘이 정확히 동작을 하는지 확인하기 위해 이상에 가까운 난수를 대조군으로 설정하였다. 이때 대조군으로 사용된 데이터는 QRNG와 /dev/urandom의 출력 난수로 각각 400만 바이트씩 연속적으로 수집하였다. 유효성 검증의 실험군은 소프트웨어 잡음원 중 Windows의 CryptGenRandom을 선정하였다. CryptGenRandom은 사용자가 요구하는 크기의 난수를 출력하는 특징이 있다. 이러한 특징을 활용하여 CryptGenRandom을 실제 4바이트씩 출력하는 소프트웨어 잡음원이라 가정하고 실험에 활용하였다. 또한 소프트웨어 잡음원을 수집할 때 고려사항 중 하나인 수집 간격[5]을 2초로 고정하여 CryptGenRandom 400만 바이트를 수집하였다.



[그림 1] QRNG의 독립성 검정 결과



[그림 2] CryptGenRandom의 독립성 검정 결과

실험결과 대조군인 QRNG, /dev/urandom 출력과 같이 이상에 가까운 난수에 대해서는 제안하는 알고리즘이 독립으로 판정함을 확인하였다. 이 결과는 제안하는 알고리즘이 이상적인 난수에 대해 정확히 동작함을 보여준다. 또한 이와 같이 알고리즘에 의해 최종 독립으로 판정된 워드의 엔트로피는 네 바이트의 엔트로피 합으로 추정된다. [그림 1]은 QRNG 출력 난수열에 대한 실험결과이다.

[그림 2]는 실험군인 CryptGenRandom에 대한 실험결과를 보여준다. 우리의 방법으로 수집한 CryptGenRandom은 [그림 2]에 나타난 것처럼 두 바이트 간 독립성 검정에서 일부가 종속으로 판정되었다. 이와 같이 일부가 종속으로 판정된 경우에는 독립으로 판정된 두 바이트의 엔트로피 합 중 최댓값을 잡음원의 엔트로피로 추정한다. 또한, 우리의 방법으로 수집한 CryptGenRandom 수열에서는 인접 바이트 간에서 종속성이 존재하는 특징을 추가적으로 확인할 수 있었다.

V. 결론

본 논문에서 제안하는 새로운 다중 바이트의 엔트로피 추정방법은 다중 바이트 간 독립성을 평가하는 새로운 통계적 검정법과 정보이론을 바탕으로 하여 워드 단위의 엔트로피를 추정하는 방법이다. 이는 현재 평가할 수 없는 다중 바이트에 대한 엔트로피 추정과 함께 워드 단위로 출력되는 소프트웨어 잡음원의 엔트로피 추정에 활용될 것으로 기대한다.

Acknowledgement

이 논문은 2017년도 ETRI 부설연구소의 지원(No.2017-075, OS 잡음원 패턴 추출을 통한 종속성 분석기법 연구)과 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No.2017-0-01857, 난수발생기 및 임베디드 기기 안전성 연구)

[참고문헌]

[1] Y. Yeom and J. S. Kang, “Probability distribution for the Linux entropy estimator”, Discrete Applied Mathematics, 2016.

[2] M. S. Turan, E. Barker, J. Kelsey, K. A. McKay, M. L. Baish, and M. Boyle, “Recommendation for the Entropy Sources Used for Random Bit Generation”, NIST Special Publication(Second DRAFT) 800-90B, 2016.

[3] TTA, “운영체제별 잡음원 수집 및 응용 지침”, 2013.

[4] 김예원, 염용진, 강주성, “소프트웨어 잡음원에 대한 평가방법 연구”, 2016한국정보보호학회동계학술대회, 2016.

[5] TTA, “소프트웨어 환경에서의 엔트로피 검증 알고리즘”, 2017.

[6] P. H. Kvam and B. Vidakovic, “Nonparametric Statistics with Applications to Science and Engineering”, John Wiley and Sons, 2007.

[7] T. M. Cover and J. A. Thomas, “Elements of Information Theory”, Second edition, John Wiley & Sons, 2006.

기존 NTRU 전력 분석 방법 실제 적용*

안수정*, 김수리*, 김희석**, 홍석희*

*고려대학교 정보보호대학원, **고려대학교 사이버보안학과

Apply existing power analysis attack against NTRU on Open source

Soojung An*, Suhri Kim*, Heeseok Kim**, Seokhie Hong*

*Grduate school of information security, Korea University

**Department of cyber security, Korea University

요 약

NTRU는 1996년 제안된 Short Vector Problem을 기반으로 하는 래티스 기반 암호로 양자 컴퓨터에 안전할 뿐만 아니라 RSA와 ECC에 비해 속도가 빨라 Post Quantum Cryptography로 활발히 연구되고 있다. NTRU가 다양한 환경에 적용되면서 부채널 분석에 대한 안전성 분석의 필요성이 대두된다. 하지만 이전 연구들은 암호에 특허가 걸려있어 실제에 적용한 분석이라고 말하기 힘들다. 본 논문에서는 2017년에 공개된 소스코드를 이용해 기존 공격 방법을 적용하고 분석한다.

I. 서론

오늘날 공개키 암호 시스템으로는 Rivest, Shamir, Adleman이 제안한 RSA와 Neal Koblitz와 Victor S. Miller가 독립적으로 제안한 타원 곡선 암호가 있다. 최근 양자 컴퓨터에 대한 연구가 활발히 진행되면서, 기존의 암호를 대체할 암호 시스템이 요구되고 있다. 양자 컴퓨터에서 가장 효율적으로 수행할 수 있는 알고리즘이 소인수분해 알고리즘으로, 소인수 문제를 기반으로 한 암호 시스템은 더 이상 사용이 불가해진다. 이에 대비하기 위해 포스트 양자 암호(Post Quantum cryptography)에 대한 연구가 진행 중이다. 포스트 양자 암호는 RSA, ECC와 같이 고정 비트를 사용하지만 기반 이론을 푸는 효율적인 양자 알고리즘이 개발이 안되 양자 컴퓨터에도 안전한 암호를 의미한다. 포스트 퀀텀 암호는 크게 래티스 기반 암호, 다변수 다항식 기반 암호, 해쉬 기반 암호, 코드 기반 암호, 아이소제니 암호가 있다.

NTRU는 Jefferey Hoffstein 등이 1998년 처음으로 소개한 공개키 암호 시스템이다[1]. NTRU는 래티스 기반 암호로 포스트 양자 암호 시스템으로 연구되고 있다. NTRU의 장점은 RSA보다 높은 보안강도에서 키 사이즈가 더 작고, 빠르다는 장점이 있다.

1996년 NTRU가 처음 제안된 이후로 안전성을 분석은 꾸준히 진행되고 있다. 하지만 소스코드에는 특허가 있어 소요시간 분석, 단순 전력 분석, 상관계수 전력 분석, 충돌쌍 분석 같은 부채널 공격을 적용한 연구들은 직접 구현한 결과물을 바탕으로 분석했다[2][4][5]. 하지만 2017년에 NTRU의 소스코드를 공개하였다. 본 논문에서는 Song et al. 제안한 단순 전력 분석

* 이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2017-0-00520, (ICT 기초연구실) SCR-Friendly 대칭키 암호 및 응용모드 개발)

방법(SPA)과 상관계수 전력 분석(CPA)을 공개된 실제 코드에 적용하고 그 결과를 논의하고자 한다[2].

본 논문의 구성은 2장에서는 NTRU 공개키 시스템의 전체적인 과정과 사용되는 컨볼루션 연산을 정리하고 부채널 공격에 대해 간략히 소개한다. 3장에서는 실제 코드에 이전 연구에서 제시한 방법 중 SPA는 적용 불가함을 보이고, CPA의 경우 표준에 맞추어 적용했을 때의 공격 수행 횟수를 계산한다. 4장에서는 결론과 향후 연구 방향을 제시한다.

II. 배경지식

1.1 NTRU 공개키

NTRU 공개키 암호는 NP-hard문제인 Shortest Vector Problem에 기반한 암호이다.

암호화와 복호화 과정은 환 $R=Z[X]/(x^N-1)$에서 연산을 한다. R의 원소 f는 $N-1$차 수이고, 정수 계수를 갖는다.

두 다항식 f와 g의 곱은 $*$로 표현한다.

$$f(X)=f_0+f_1X+f_2X^2+\cdots+f_{N-1}X^{N-1}$$

$$g(X)=g_0+g_1X+g_2X^2+\cdots+g_{N-1}X^{N-1}$$

$$h(X)=f(X)*g(X),$$

$$h_k=\sum_{i=0}^{k} f_i g_{k-i}+\sum_{i=k+1}^{n-1} f_i g_{n+k-1}$$
$$=\sum_{i+j\equiv k \bmod n} f_i g_j$$

NTRU는 4개의 (N, d, p, q), $\gcd(p,q)=1$를 만족하는 정수 파라미터를 사용한다. p와 q는 소수일 필요는 없지만 서로소 관계에 있고, $p \ll q$를 만족해야한다.

1.1.1 키 생성

NTRU 암호화 시스템에 사용되는 개인키와 공개키가 생성되는 방식은 다음과 같다. 우선 R에서 삼진다항식 f와 g를 뽑는다. 이때 f는 계수가 1인 항의 개수는 d_f개 여야하고, -1 항의 개수는 d_f-1개가 되도록 뽑는다. 이는 f는 mod p와 q상에서 역원이 존재해야하기 때문이다. g는 계수 1 항의 개수가 d_g개, -1 항도 d_g개를 만족하도록 뽑는다. f를 뽑았으면, $f_p^{-1}*f \equiv 1 \bmod p$와 $f_q^{-1}*f \equiv 1 \bmod q$를 만족하는 f_p^{-1}, f_q^{-1}을 구한다. 공개키는 $h=pf_q^{-1}*g \bmod q$를 사용하고, f와 f_q^{-1},f_p^{-1}까지를 개인키로 사용한다. 코드에서는 f를 f_1,f_2,f_3로 나누어서, $f=f_1f_2+f_3$로 표현한다.

1.1.2 암호화

암호화는 R에서 1과 -1계수항의 개수가 d_m개인 평문 m과, d_r개인 임의의 삼진다항식 r을 골라, $e=r*h+m \bmod q$를 계산하면 된다.

1.1.3 복호화

복호화를 할 때는 $a=f*e \bmod q$를 계산하고, $a*f_p^{-1} \bmod p$를 한다. 그러면,

$a=f*e=r*h*f+m*f=pr*g+m*f \bmod q$가 되고, $a*f_p^{-1}=m \bmod p$가 된다.

1.2 부채널 분석

부채널 분석이란 암호 시스템을 구현한 장치로부터 전력, 시간, 전자파, 열 등과 같은 부수적인 정보를 추출하고 해석하여 비밀 정보를 알아내는 분석 방법을 말한다. 부채널 분석의 한 종류로 전력 분석(Power analysis)이 있다[3].

전력 분석은 1998년 Kocher이 제안한 분석 방법으로 암호 장치를 수행할 때 장치에서 사용하는 소비하는 전력을 측정하여 분석에 사용한다[3]. 이때 소비 전력은 장치가 수행하는 연산에 의존한다는 것이 전제 조건이 되겠다. 전력 분석의 종류로는 크게 단순 전력 분석(Simple Power Analysis), 차분 전력 분석(Differential Power Analysis), 상관계수 전력 분석(Correlation Power Analysis)이 있다.

단순 전력 분석이란 특정 연산을 할 때 나타나는 파형의 패턴을 찾아내어 비밀 정보를 복원하는데 사용하는 방법을 의미한다. 차분 전력 분석이란 암호 동작 중에 키와 관련된 연산의 중간 값을 추측하고 이를 이용해 비밀 정보를

추출하는 분석 방법이다. 상관계수 전력 분석은 암호 알고리즘에서 중간 값을 예상하여 중간 값과 실제 전력 파형 사이의 상관계수 값을 계산하는 방법이다. 소비 전력은 연산과 관련된 정보라는 가정을 바탕으로, 중간 값을 옳게 추측했다면 그 상관계수가 높게 나타나 비밀정보를 복원할 수 있다.

III. NTRU 전력 분석 공격 방법

공격자는 개인키를 알아내기 위하여 개인키가 사용되는 부분인 복호화 부분을 타겟으로 한다. 개인키를 곱하는 $a = f * e \bmod q$ 연산을 공격 타겟으로 한다.

NTRU 분석하여 알아내려는 개인키는 정확히 말하면, 개인키의 0이 아닌 계수들의 위치를 배열에 저장해두는데, 그 값들의 차이를 도출하는 것이 목표이다. 예를 들면, 다항식 x^3+x+1이라면, 1의 계수의 위치는 배열 b에 $b[3] = \{0, 1, 3\}$라고 저장되어 있겠다. 그러면 분석을 통해 알아내는 값은 $(1, 2)$이다. 차이를 알아내면, 다항식에서 0이 아닌 계수가 처음 나타나는 위치 값은 전수조사를 통해 알아낸다. 이 때 차이를 $b[1]-b[0]=b1$이라고 표현한다.

3.1 단순 전력 분석 공격 방법

Song et al.이 제안한 단순 전력 분석 방법은 $0+x\,(x \neq 0)$인 연산의 파형과 $x+y\,(x, y \neq 0)$ 연산의 파형 패턴이 다르다는 가정에서 시작한다. 패턴을 분석하여 개인키를 복원하는 방법이고, 선택 평문공격을 하면 가능하다. 예를 들어, $0+x$ 패턴을 ZN, $x+y$의 패턴을 NN이라 하고, 개인키를 x^3+x+1라고 한다. 예시에서는 개인키와 곱할 $e=x^3+x^2+x+1$의 계수를 모두 1로 채워 0인 수가 없도록 한다. 그리고 저장할 배열 t를 0부터 $2N-1$까지를 0으로 초기화한다. 연산 첫 단계에서는 $t[b[0]]$에 해당하는 위치부터 곱하는 e의 계수를 더한다. 그러면 파형의 패턴은 ZN, ZN, ZN, ZN이다. 다음으로는 e를 $t[b[1]]=t[1]$부터 더할 것이기 때문에 파형이 NN, NN, NN, ZN이다. 마찬가지로 마지막 연산에서는 $t[4]$부터 더하므로, NN, NN, ZN, ZN이다. 두 번째 단계에서 ZN 형태의 패턴이 한번 발생하는데 이는 $b1(=b[1]-b[0])$이 1임을 의미한다. 세 번째에서 ZN이 두 번 발생하고 $b2(=b[2]-b[1])$는 2라는 의미이다. 개인키의 1의 위치가 저장된 배열 값의 차이를 알았으므로, 전수조사로 첫 번째 위치를 찾으면 개인키를 복원할 수 있다.

Song et al.에서는 위 공격의 대응방법으로 처음에 ZN패턴이 나타나지 않도록 t를 0이 아닌 다른 값으로 초기화 할 것을 제안한다.

3.2 상관계수 전력 분석 공격 방법

Song et al.에서 제안한 상관계수 전력 분석 방법은 입력되는 평문을 알고, 두 번째 cycle부터 평문의 첫 번째 값이 이전에 더해놓은 평문의 몇 번째 값과 더해지는가를 중간 값으로 이용하여 해밍 디스턴스($HD = HW((c_{k+b1}) \oplus (c_{k+b1} + c_k))$) 식을 세워 상관계수 전력 분석을 시행한다. k값을 바꿔가며 상관계수 값이 가장 높게 뜨는 k를 찾는다. 그렇게 찾은 k값은 1이 저장된 배열 사이의 차이가 되겠다.

상관계수 전력 분석을 막기 위하여 제안한 대응 방법으로는 두 가지가 있다. 첫 번째는 배열 b에 저장된 순서를 무작위로 바꾸는 방법이고, 두 번째는 결과를 저장할 배열의 초깃값을 0이 아닌 임의의 수로 초기화하는 방법이다. 위 방법은 배열 사이의 차이를 알아내도 개인키 복원이 불가능하게 만든다.

IV. NTRU 전력 분석 적용

최근 공개된 코드에 Song et al.에서 제안한 방법을 적용하기 위해서는 삼진 다항식을 사용하므로 -1과 1 모두를 복원해야하고, 매번 리덕션을 수행하는 것을 고려해야한다. 또한 개인키를 f_1, f_2, f_3로 나누어 있다는 것도 기존 분석과 차이점이다. 본 장에서는 실제 코드에의 적용 방법을 논의한다.

4.1 단순 전력 분석 방법 적용과 실제

공개 코드를 보면, 배열 N만큼을 0으로 초기화한 후, 개인키 배열에 평문을 더해줄 때 N을 넘어가면 바로 mod N이 취해진 배열에 더해지도록 구현되어 있다. 그러므로 $0+x$ 연산은 첫

번째 덧셈 과정에만 일어나고, 이후로는 모두 $x+y$ 에 해당하는 연산만 있을 것이다. 그러므로 [2]에서 제안한 SPA 공격을 실제 구현하기 위해서는 선택 평문을 신중하게 조절하여 수행해야한다.

4.2 상관계수 전력 분석 적용과 실제

Song et al.에서 제안한 공격은 실제 코드에도 적용할 때 개인키 f를 f_1, f_2, f_3로 나누어 연산함을 고려해야한다. 연산은 Algorithm 1과 같다.

Algorithm 1. 암호문과 개인키 곱셈 과정

Input: 암호문 $e(X)$와 개인키 $f(X)=f_1(X)*f_2(X)+f_3(X)$, q

Output: $e(X)*f(X)$ 연산 결과를 저장한 배열 c

1. $t \leftarrow e*f_1$
2. $t \leftarrow t*f_2$
3. $t2 \leftarrow e*f_3$
4. for $0 \leqq i < N$ do
5. $\quad c[i] = (t[i]+t2[i]) \wedge (q-1)$
6. end for

또한 f_1, f_2, f_3는 삼진 다항식이다. 그러므로 실제에서는 1의 위치 뿐 만 아니라 -1의 위치도 알아내야한다. 소스코드에서는 계수 위치를 저장하는 배열에 1의 위치를 먼저 저장하고 그 뒤에 -1의 위치를 저장하는 것을 고려하여 -1의 위치를 찾고, 1의 위치를 찾는 순으로 진행하면 된다.

Song et al.이 제안한 방법을 실제 코드에 적용하기 위해서는 f_1*e과 f_3*e에 적용하고, 알아낸 f_1을 이용해 $(f_1*e)*f_2$ 부분을 분석해야한다. 그러므로 실제에 적용하기 위해서는 적어도 세 번의 분석을 해야 개인키 값을 알아낼 수 있을 것이다.

V. 결론

NTRU가 처음 제안된 이후 다양한 부채널 분석 방법이 연구되었다. 이전 연구들은 소스코드가 공개되지 않아 직접 구현한 것들을 바탕으로 부채널 분석을 했기 때문에 기존 방법을 실제에 적용하기 위해서는 고려해야할 요소들이 있다. 더해서 코드를 바탕으로 향상 시킬 수 있는 방법도 있을 수 있다. 또 공개된 정보를 바탕으로 충돌쌍 공격[4], 소요시간 분석[5] 이외에도 다양한 부채널 분석 방법을 고민해볼 수 있을 것이다.

[참고문헌]

[1] Hoffstein, J., Pipher, J., & Silverman, J. H. (1998, June). NTRU: A ring-based public key cryptosystem. In International Algorithmic Number Theory Symposium (pp. 267-288). Springer, Berlin, Heidelberg.

[2] Song, J. E., Han, D. G., Lee, M. K., & Choi, D. H. (2009). Power analysis attacks against NTRU and their countermeasures. Journal of the Korea Institute of Information Security and Cryptology, 19(2), 11-21.

[3] Kocher, P., Jaffe, J., & Jun, B. (1999). Differential power analysis. In Advances in cryptology—CRYPTO'99 (pp. 789-789). Springer Berlin/Heidelberg.

[4] Zheng, X., Wang, A., & Wei, W. (2013). First-order collision attack on protected NTRU cryptosystem. Microprocessors and Microsystems, 37(6), 601-609.

[5] Vizev, N. V. (2007). Side Channel Attacks on NTRUEncrypt (Doctoral dissertation, Bachelor's thesis, Technical University of Darmstadt, Germany, 2007. Available on http://www. cdc. informatik. tu-darmstadt. de/reports/reports/Nikolay_Vizev. bachelor. pdf).

안드로이드 앱의 호출 흐름 무결성 검증 기법*

최준태, 조홍래, 하동수, 오희국†
한양대학교 컴퓨터공학과

A Method of Verifying Integrity of Function Call on Android App*

Joontae Choi, Honglae Jo, Dongsoo Ha, Heekuck Oh†
Department of Computer Science and Engineering, Hanyang University.

요 약

대다수 안드로이드 앱 공격은 사용자 정의 함수나 API 함수의 후킹을 통해 이뤄지며, 이를 통해 검증 루틴이 우회되거나 악성코드가 삽입된다. 특히 올해에 주요 API 함수 후킹을 통해 대다수 안드로이드 보호 서비스를 우회하는 기법이 공개되었기에, 함수 호출 흐름의 무결성 검증 기법이 더욱 요구되고 있다. 본 논문에서는 후킹 기법을 함수 주소 수정과 명령어 수정을 통한 방식으로 분류하여, 이를 기반으로 각각의 무결성 검사 기법을 제시한다. 그리고 라이브러리와 앱 코드의 무결성 두 가지 측면으로 검사하는 기법을 제안한다. 제안하는 기법을 이용하여 앱 자체에서 사용되는 함수들의 무결성을 검증할 수 있으며 이를 통해 공격을 방어할 수 있다.

I. 서론

바이트코드 기반인 안드로이드 애플리케이션은 역공학에 취약하며 프로그램 수정이 용이하다. 그래서 공격자는 비교적 쉽게 애플리케이션의 소스코드를 분석할 수 있으며, 프로그램의 흐름을 변경하거나 애플리케이션의 코드를 수정하여 악성코드를 삽입하고 재배포할 수 있다.

공격자는 흐름 변경 및 악성코드 삽입을 위하여 여러 가지 방법을 사용하고 있으며, 그 중 대표적인 방법은 애플리케이션 동작 과정에서 실행되는 함수를 후킹하는 것이다.

여기서 말하는 후킹은 운영 체제나 응용 소프트웨어 등의 각종 컴퓨터 프로그램에서 소프트웨어 구성 요소 간에 발생하는 함수 호출, 메시지, 이벤트 등을 중간에서 바꾸거나 가로채는 명령, 방법, 기술이나 행위를 말한다.

공격자는 후킹을 사용하여 애플리케이션의 여러 검증 절차를 무시하고 넘어갈 수 있다. 또한 애플리케이션의 정상 흐름에 있는 함수를 후킹 후 코드를 삽입하여 위장하는 방식으로 분석가에게 혼란을 주어 분석을 지연시키게 할 수 있다.

현재 마켓에서는 애플리케이션의 모듈을 후킹하여 시스템이나 응용프로그램의 동작을 변경할 수 있게 해주는 프레임 워크 애플리케이션인 'Xposed'를 배포 중이다. 또한 금융 앱을 사용할 시 루팅 탐지 과정을 우회할 수 있도록 변경해주는 'rootcloak' 애플리케이션도 쉽게 설치하여 사용할 수 있다. 이 애플리케이션들을 악의적으로 사용한다면 금전적 손실을 일으킬 수 있다.

* 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 대학ITCT연구센터육성 지원사업의 연구결과로 수행되었음(IITP-2017-2014-0-00636)
이 논문은 2015년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (NRF-2015R1D1A1A09058200)
† 교신저자 : hkoh@hanyang.ac.kr

그러므로 애플리케이션의 흐름 제어 및 악성 코드의 삽입 여부를 검사하기 위하여 애플리케이션 동작 과정에 사용되는 함수들의 후킹 여부를 분석하는 무결성 검증이 필요하다.

논문의 구성은 다음과 같다. 본 논문의 2장에서는 후킹에 관한 배경지식을 서술한다. 3장에서는 안드로이드 환경에서 사용하는 후킹 방법에 대해서 크게 두 부분으로 나누어 조사한다. 4장에서는 조사한 후킹 방법을 통한 무결성 방법을 제시하고, 마지막으로 5장에서 논문의 결론을 맺는다.

II. 배경 지식

후킹 기술은 다양한 목적으로 사용된다. 대표적으로 프로그램의 데이터를 감시하기 위한 목적으로 키보드나 마우스 등의 이벤트를 가로채서 다른 구성요소의 동작을 모니터링 하는 기능을 추가하기 위해서 사용된다. 또한, 악의적인 용도로 컴퓨터의 메모리 정보, 키보드나 터치의 입력 정보 및 함수들의 결과를 변경하거나 원하는 곳으로 빼돌리기 위해서 사용된다. 그리고 프로그램의 흐름을 변경하기 위하여 기존 API함수를 후킹하여 전혀 다른 함수로 흐름을 변경하거나 새로운 기능을 추가하기 위한 용도로 사용된다.

안드로이드 애플리케이션의 공격자가 함수 후킹을 사용하는 이유는 다음과 같다.

첫째, 애플리케이션의 여러 검증 절차를 무시하고 넘어갈 수 있게 한다. 만약 애플리케이션이 루팅 탐지, 브레이크 포인트 탐지 등과 같은 기능이 추가되어 있으면 후킹을 통하여 그 기능들이 포함된 함수들을 지나치게 할 수 있다.

둘째, 안드로이드 애플리케이션의 원하는 흐름 시점에 악의적인 행위를 추가할 수 있다. 공격자가 원하는 흐름에서의 API 함수 혹은 실행되는 함수를 후킹하여 대상 함수가 호출되는 특정 시점에 악성코드가 실행이 되게 할 수 있다.

셋째, 삽입된 코드 및 함수를 일반적인 함수 혹은 안드로이드 API함수들로 위장할 수 있게 한다. 분석가들이 코드를 분석할 때, API함수를

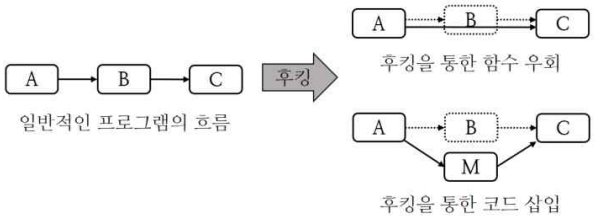


그림 1. 전반적인 후킹의 개요

보게 되면 예상된 결과가 나올 것이라 추측하게 되어 지나친다. 이러한 분석가들의 특성을 겨냥하여 일반적인 함수 혹은 API함수들로 위장된 함수는 분석가로 하여금 분석에 혼란을 줄 수 있다.

III. 후킹의 분류 및 방법

본 논문에서는 안드로이드 환경에서 사용되는 함수 후킹 방법은 크게 두 가지로 나눈다. 이 장에서는 그 두 가지 후킹 방법에 분류하고, 각 후킹 방법의 차이점과 공통점을 분석한다.

3.1 함수의 참조 주소 수정을 통한 후킹 방법

안드로이드 시스템에서 애플리케이션을 실행하면 프로그램 로더가 적재되고 실행된 후에 애플리케이션이 실행된다. 적재된 프로그램 로더는 애플리케이션에서 쓰이는 다른 모든 공유 라이브러리를 찾고 로드한다. 적재시키는 과정 중에 후킹을 할 수 있다.

첫째, 프로그램 로더는 로드할 공유 라이브러리의 이름으로 찾는다. 만약 기존에 존재해야 할 라이브러리 위치에 같은 이름으로 하여 임으로 만들어진 라이브러리가 존재한다면 프로그램 로더는 그 라이브러리를 로드할 것이며, 내부에 있는 공유 라이브러리 함수 또한 임의로 작성된 함수들이 실행된다. 이 방법은 '라이브러리 인젝션'이라고 불린다.

둘째, 프로그램은 함수 검색 시 라이브러리의 로드 순서에 따라 함수를 순차적으로 검색하지만, 라이브러리의 로드되는 순서를 변경할 수 있다. 환경변수 'LD_PRELOAD'를 정의한다면 프로그램이 라이브러리를 로드할 때 PRELOAD로 설정된 라이브러리들을 우선적으로 로드하게 된다. 하지만 PRELOAD로 설정된 라이브러리에 기존에 존재하는 함수의 심볼과

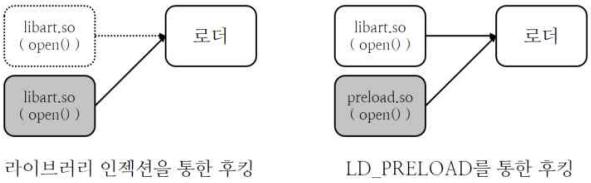


그림 2. 라이브러리 수정을 통한 후킹

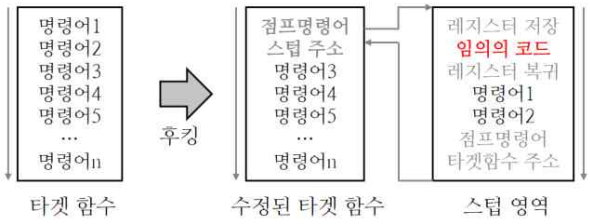


그림 3. 명령어 수정을 통한 후킹

같은 함수가 존재한다면 함수 검색과정에 의해 함수를 오버라이드 하게 된다. 이 방법을 통해서 공격자가 미리 작성해둔 라이브러리를 PRELOAD하여 기존 함수를 후킹할 수 있다.

셋째, 애플리케이션은 동작 과정에서 외부라이브러리를 호출하면 GOT(Global Offset Table)을 참조하여 해당 주소로 점프한다. 여기서 GOT는 프로시저들의 주소를 담고 있으며, 만약 공격자가 GOT의 일정 함수의 주소를 악의적인 주소로 변경을 한다면 애플리케이션은 PLT를 통해서 악의적인 곳으로 점프하게 된다. 이러한 방식으로 함수의 주소를 변경하여 후킹을 할 수 있다.

3.2 함수의 명령어 수정을 통한 후킹 방법

안드로이드는 일반적으로 비교적 전력 소모가 적고 파이프라인의 고속화를 위하여 명령어 길이가 고정적인 ARM아키텍처를 사용한다. 그러므로 바이트 코드로 작성된 안드로이드 애플리케이션은 실행 중에는 ARM의 기계어로 번역되어 실행된다. 기계어로 번역된 코드를 실행되기 전에 함수의 기계어를 수정 혹은 추가하여 실행 흐름을 변경할 수 있다.

함수의 기계어 수정을 통한 후킹은 다음 과정을 통해서 이루어진다. 먼저 함수의 심볼을 검색하여 함수의 시작주소를 찾아낸다. 그 후 찾은 함수의 프롤로그 명령어를 점프하는 명령어 및 점프하는 주소로 변경한다. 원하는 명령어를 실행한 후에는 수정되기 전의 명령어를 실행한 후 다시 정상 흐름으로 돌아가게 명령어를 수정한다.

이 방법을 '인라인 후킹'이라고 하며, DBI (Dynamic Binary Instrumentation)이라고도 불린다. '인라인 후킹'을 사용할 때에는 제약 사항이 존재한다. 만약 수정할 프롤로그 명령어 중 점프 명령어가 존재한다면 스텝 영역에 추가된 프롤로그 명령어 수행과정에서 점프하기 때문에 후킹이 불가능하다. 또한, 수정할 프롤로그 명령어 중 상대 주소를 사용하여 데이터를 로드하는 명령어가 존재한다면 상대 주소의 데이터를 미리 계산하여 절대값을 사용하거나 스텝 영역 일부분에 데이터를 복사하여 명령어의 오프셋을 변경하는 방법을 사용하여야 한다.

기계어를 변경하는 것뿐 아니라 애플리케이션의 apk 압축파일을 압축 해제하여 바이트코드를 수정하여 원하는 명령어로 점프하도록 흐름을 변경할 수 있다. 바이트코드를 변경하여 후킹하는 방법은 앞에서 기술한 기계어를 변경하여 후킹하는 과정과 같다.

IV. 후킹 기반의 참조 무결성 검사

공격자는 앞 장에서 분석한 후킹 방법을 사용하여 함수를 후킹하며, 일반적인 공격자는 하나의 방법만 사용하지 않고 여러 방법을 혼합하여 사용한다. 그러므로 여러 후킹 방법에 대하여 여부를 파악해야 하며 본 논문에서는 다음과 같이 애플리케이션의 무결성 검사를 제시한다.

먼저 애플리케이션에서 사용하는 라이브러리가 올바른 라이브러리인지 확인해야 한다. 애플리케이션이 실행되는 시점에 애플리케이션에서 사용되는 라이브러리의 의존성 검사를 수행하고 수행되는 라이브러리의 함수 혹은 라이브러리 파일의 해쉬값을 구한다. 구한 후 데이터베이스에 저장하며 프로그램 수행과정 중 라이브러리를 로드하고 해쉬값을 비교한다.

그 후 동적으로 함수의 코드가 변화되는지 확인한다. 동적으로 함수의 코드는 변화될 수는 있지만, 만약 8바이트 이상 변경이 되거나, PC

분류	후킹 방법	검사 방법
함수의 주소 변경	라이브러리 삽입을 통한 후킹	• 라이브러리의 해쉬값을 통한 방어
	PRELOAD를 통한 후킹	
	GOT영역 수정을 통한 후킹	• mprotect() 함수 감지를 통한 방어
함수의 명령어 변경	프롤로그 수정을 통한 후킹	• 코드 변조 및 스택추적을 통한 방어
	바이트 코드 변경을 통한 후킹	• 함수의 해쉬값을 통한 방어

표 1. 후킹 방법 별 무결성 검사 방법

에 로드를 하는 명령어 혹은 롱점프 명령어가 포함이 되어 있다면 후킹이 된다는 것을 알 수 있다. 명령어가 변경된다는 것을 판별하기 위해서는 스택을 추적하여 확인할 수 있다.

부가적으로 프로그램 내부에서 mprotect()함수를 사용하는 것을 방지해야 한다. 일반적으로 함수의 주소를 담고 있는 GOT영역이거나 프로그램의 코드 영역은 write권한이 없다. 그래서 GOT영역을 수정하거나 프로그램의 코드 영역을 수정하기 위해서는 페이지에 write권한을 부여하기 위해서 mprotect()함수를 사용해야 한다. 일반적인 프로그램에서 mprotect()함수를 사용하지 않으므로 mprotect()함수의 사용 여부로 후킹 여부를 판단할 수 있다.

제안한 기법을 통하여 후킹 기반의 참조 무결성 검사를 할 수 있지만, 공격자는 이 무결성 검사 루틴 또한 후킹하여 우회할 수 있다. 그러므로 이 검사하는 루틴을 프로그램에 삽입하기 위해서는 소스코드의 난독화 및 보호가 필요할 것으로 보인다.

V. 결론 및 향후 연구계획

본 논문에서는 함수들의 후킹을 기반으로 하는 무결성 검증을 위하여 안드로이드 환경에서의 후킹 방법들을 조사하였다. 다음 조사를 통해서 각 후킹 방법들의 공통점을 분석하였으며 분석의 결과를 토대로 무결성 검사 기법을 제시한다.

제시하는 기법은 애플리케이션에서 사용되는 라이브러리의 무결성 검사, 코드의 변화를 동적으로 감시, 그리고 mprotect() 함수의 사용 여부 순으로 검사한다.

향후 안드로이드 애플리케이션의 후킹 여부 탐지 뿐 아니라 더욱 더 견고한 애플리케이션을 위하여 후킹을 방지하는 안티 후킹 플랫폼을 연구를 진행할 계획이다.

[참고문헌]

[1] http://terms.tta.or.kr/dictionary/searchList.do

[2] S.-T. Sun, A.Cuadros, and K. Benznosov, “Android rooting: Methods, detections, and evasion,” in Proc. of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices, Octover 2015,

[3] Kyeonghwan Lim, Younsik Jeong, Seong-je Cho, Minkyu Park, and Sangchul Han, “An Android Application Protection Scheme against Dynamic Reverse Engineering Attacks,” Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, vol. 7, no. 3, pp. 40-52, September 2016.

[4] W. Zhou, Y. Zhou, X. Jiang, and P. Ning, “Detecting repackaged smartphone applications in third-party android marketplaces,” in Proc. of the 2th ACM conference on Data and Application Security and Privacy, February 2012.

[5] http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0234b/i1010871.html

기계학습을 이용한 행위 기반의 악성코드 판별 모델

박의성* 진수현** 지한별*** 이주연*** 국광호***

*목포대학교 정보보호학과

*동의대학교 소프트웨어공학과

*서울과학기술대학교 글로벌융합산업공학과

A Study on Model of Behavior-based Malicious code Identification using Machine Learning Techniques

Uiseong Park* Su Hyeon Jin**

Han Byeol Ji*** Ju Yeon Lee*** Kwang-Ho Kook***

*Department of Information Security, Mokpo National University

**Department of Software Engineering. DongEui University

***Department of Industrial and Information Systems Engineering, Seoul National University of Science and Technology

요 약

최근 악성코드를 이용한 공격이 증가하며, 여러 악성코드 생성 도구를 통해 생성된 다양한 변종/신종 악성코드가 늘어나고 있다. 기존 행위 분석 기법의 낮은 정확도를 보완하고 많은 양의 악성코드에 대응할 수 있도록 분석 기법의 효율성을 높이기 위해 본 논문에서는 기계학습의 방식 중 로지스틱 회귀분석과 의사결정나무 알고리즘을 사용한 악성코드 분석 기법을 제안한다.

I. 서론

악성코드는 컴퓨터 또는 시스템에서 악의적인 행위를 수행하는 명령어 집합을 의미하며, 목적에 따라 바이러스, 웜, 스파이웨어, 트로잔 등으로 분류될 수 있다[1]. 최근 악성코드를 이용한 공격이 증가하고 있으며 AhnLab ASEC REPORT VOL.82의 집계한 바에 따르면 2016년 10월 한 달간 탐지된 악성코드 수는 970만 118건으로 전월 899만 9000건에 비해 약 7.8% 증가하였다[2]. 또한 여러 악성코드 생성 도구를 통해 다양한 변종/신종 악성코드를 생성시킬 수 있으므로 늘어나는 악성코드에 대한 빠른 대처가 필요하다.[3]

일반적으로 백신은 악성코드의 바이너리 형태를 가공하여 시그니처를 추출하고 데이터베이스에 등록하여 비교/탐지 하는 시그니처 기반 탐지방법을 사용한다. 새로운 악성코드들이 등장할 경우를 대비해 지속적인 시그니처 데이터베이스의 업데이트를 시행하지만 기하급수적으로 증가하는 모든 악성코드에 대해 시그니처를 생성하고 데이터베이스를 업데이트하기에는 어려움이 있다. 또한, 악성코드가 난독화 되어 있는 경우 정확한 시그니처를 추출하는데 어려움이 있으므로 효과적인 탐지가 힘들다.[4] 따라서 이러한 문제점 해결을 위해 행위 기반 탐지방법에 대한 연구가 필요하다.

본 논문에서는 False-positive 탐지를 줄여 정확도를 높이기 위한 방법으로 로지스틱 회귀분석과 결정트리 알고리즘을 이용하여 행위 기반의 악성코드 판별 모델을 제안하고자 한다.

II. 관련연구

2.1 악성코드 분석 방법

악성코드 분석 방법에는 크게 정적 분석과 동적 분석이 있다. 정적 분석 방법은 악성코드를 실행시키지 않고 바이너리 파일 자체를 분석하여 정보를 얻는 것을 의미한다. 정적 분석 방법을 통해 얻을 수 있는 정보는 악성코드가 가지는 문자열 정보, 해시 값, DLL 파일의 종류 등 다양한 정보를 얻을 수 있다. 동적 분석 방법은 악성코드를 실행시켜 행위를 분석하는 것으로 파일 생성, 조작한 레지스트리, API 호출 등의 정보를 얻을 수 있다. 그리고 얻어진 정보들에 다양한 알고리즘을 적용하여 악성코드 분류를 진행한다.[5]

2.2 API 추출을 통한 행위기반 분석 방법

[6]에서는 악성코드의 API Call 정보를 이용한 행위 분석 기반 방법론을 제시하였다. 악성코드로 의심되는 프로그램의 Import table에서 추출한 API 목록을 토대로 네 가지의 행위로 분류한다. 분류 기준은 네트워크 행위, 레지스트리 관련 행위, 파일 입출력 행위, 기타 행위로 나누어 프로그램이 어떠한 행위를 수행하는지 분석한다. 행위 분석이 완료되면 이미 알려져 분석되어 데이터베이스 형태의 악성코드들과 비교 분석을 수행한다.

이때, 행위별 API 리스트가 비슷할 경우 같은 행위를 하는 것으로 간주하고 같은 그룹으로 분류한다. 그리고 그룹별로 Jaccard Similarity Coefficient를 적용하여 유사도를 판별한다. 이 후 유사도에 부가적 행위를 구분할 수 있도록 가중치를 적용시켜 최종 유사도를 구한다. 이를 통해 악성코드를 판별하고 분류할 수 있다.

하지만 이 방법론의 경우 악성코드와 유사한 행위를 수행하는 정상 프로그램을 악성코드로 탐지하는 오류를 범한다. 즉, 목록별로 유사도를 탐지하기 때문에 비슷한 API 목록을 사용하는 정상 프로그램의 유사도가 악성코드와 비슷하게 측정되어 False-positive가 발생한다.

2.3 머신러닝 기법을 적용한 탐지 방법

[7]에서는 악성코드 여부 판별을 위한 기본적인 정보를 얻기 위해 프로그램의 디스어셈블과 [6]의 방법처럼 프로그램이 사용하는 API를 추출하여 프로그램을 분석한다. 이 후 악성코드 여부 판별을 위해서는 분석자가 스스로 판별해야하는 불편함이 존재하기 때문에 이를 기계학습 기법을 통해 보완하는 방법을 제시하였다.

기계학습은 기존 여러 악성코드에서 얻은 정보들을 트레이닝 데이터로 지정하여 학습하고, 차후 샘플들로 테스트 데이터를 구성하여, 유사도에 따라 클러스터링을 수행한다. 클러스터링 알고리즘으로는 K-Means, EM(Expectation - Maximization) 알고리즘을 사용하였다. 결과적으로 이 방법은 K-Means 알고리즘의 사용으로 유사도에 따라 악성코드 여부를 판별하기 때문에 판별에 대한 정확도가 떨어지는 단점이 존재한다.

2.4 분석 알고리즘

본 논문에서는 지도 학습 알고리즘 중 로지스틱 회귀 분석(Logistic Regression)과 의사결정나무(Decision Tree)를 분석기법으로 사용하였다.

로지스틱 회귀분석은 존재(1)/부재(0)의 형태로 되어 있는 종속변수에 대하여 특정 조건을 의미하는 한 개 이상의 독립변수들을 가지고 이를 추측하기 위해 회귀분석을 하는 것이다. 즉, 종속변수가 구간 비율척도 변수가 아닌 0과 1의 값을 가지는 것이 다른 회귀분석과의 차이점이다.

의사결정나무는 의사결정규칙(Decision Rule)을 트리구조로 도표화 하여 분류와 예측을 수행하는 분석 방법이다. 분류와 예측의 과정이 나무구조에 의한 추론규칙에 의해서 표현되므로 과정을 쉽게 이해할 수 있다는 장점이 있다.

본 논문에서는 정확한 악성코드 탐지를 위해 악성 및 정상 파일의 두 가지 특성을 구분하여 악성코드와 일반코드를 대표하는 각각 변수들을 설정하였고, 로지스틱 회귀분석과 의사결정

나무를 이용해 분석을 시행하였다. 각각의 변수는 API목록을 추출하여 악성 및 정상 행위를 하는 API의 존재를 판별하는 것이 아닌 API를 호출하는 순서 즉 API 호출 시퀀스를 하나의 룰으로 설정하였다.

III. 악성코드 판별 모델

3.1 악성코드 판별 방법론

로지스틱 회귀분석은 종속변수가 0과 1의 값을 가지는 점으로 인해 악성 코드와 정상 파일을 악성(1)/정상(0) 두 가지 값으로 구분할 수 있어 로지스틱 회귀분석을 사용하여 기계학습을 진행하였다.

3.2 기계 학습 절차

기계 학습은 malwr.com에서 정상 및 악성 샘플 데이터를 수집, 데이터 전처리, 기계학습, 결과 해석 순으로 진행하였다.

단계	방법
수집	malwr.com에서 샘플 데이터 수집
데이터 전처리	악성 및 정상 API 호출 시퀀스 룰 셋을 통한 Ground Truth 추출
기계학습	Logistic Regression, Decision Tree 적용
결과해석	Decision Matrix 분석 결과 해석

[표 1] 기계 학습 절차

3.3 데이터 수집

malwr.com에서 virustotal 결과를 기준으로 정상코드 및 악성코드 샘플을 구분하여 각각 500개의 표본 데이터를 수집하였다.

3.4 데이터 전처리

악성코드 실행 파일은 기능에 필요한 API를 순서대로 호출하며 작동하기 때문에 악성코드를 실행할 때도 일련의 순서대로 API를 호출한다. 악성코드들의 기능은 유사한 면이 많기 때문에 API들이 비슷한 호출 시퀀스를 가질 것이다. 따라서 악성 코드에서 실행되는 API 호출 시퀀스(malware rule) 68개와 정상 파일의 API 호출 시퀀스(clean rule) 68개를 변수로 선정했다. 선정한 룰의 예시는 [표 2]와 같다.

malware rule	['RegOpenKeyEx@SubKey=(Software\\\\Microsoft\\\\Windows\\\\CurrentVersion\\\\Run)', 'RegSetValueEx']
clean rule	['RegOpenKeyExW@SubKey=(Software\\\\Microsoft\\\\COM3)','RegOpenKeyExW@SubKey=(Software\\\\Microsoft\\\\COM3)']

[표 2] API 호출 시퀀스 변수 예시

malware rule의 예시는, 현재 로그인한 사용자에 대해서 해당 레지스트리에 등록한 프로그램이 자동 실행된다. 컴퓨터 시작 시 실행되는 것이 아닌, 사용자가 로그인 할 경우에 실행되는 레지스트리 등록에 관련된 룰이며, 악성코드들이 주로 사용하는 레지스트리 경로이다. clean rule의 예시는, 유용한 Windows Registry key로써, COM+ 카탈로그와 관련된 여러 항목이 기록되는 하이브이다. 악성 행위와 관련성은 적기 때문에, clean rule로 분류 하였다.

트레이닝 데이터 총 1000개(악성 샘플 500개 + 정상 샘플 500개)를 136개(악성 변수 68개 + 정상 변수 68개)변수 각각에 대해 큰 범주의 시퀀스가 동일한 경우(파일 생성 등) 1점, 상세 시퀀스가 동일한 경우(A 파일 생성 등) 3점을 부여하여 Ground Truth를 만들었다.

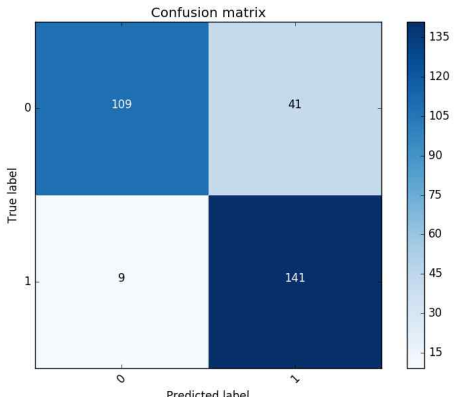
IV. 기계학습 결과

4.1 분석 및 결과

정상 샘플 데이터 150개와 악성 샘플 데이터 150개를 새로 수집하여 해당하는 악성 및 정상 API 호출 시퀀스 일치 점수(x_value)와 샘플 데이터의 악성 및 정상 여부에 대한 정답 값(y_value)을 생성하였다. x_value는 분석데이터로 활용하여 77%는 트레이닝 데이터로, 23%는 테스트 데이터로 사용하였다. Ground Truth를 기준으로 악성일 확률이 높았던 결과에 따라 테스트 데이터의 결과를 예측하여 악성 및 정상 여부를 판단한다.

4.2 성능 평가

모델의 정확도(Accuracy)를 구해 성능을 평가 하였다.



[그림 1]

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)}$$

[그림 1]과 같이 Logistic Regression의 경우 약 TP=141, TN=109, FP=9, FN=41 으로 $Accuracy = \frac{(141+109)}{(141+109+9+41)} = 0.833$이다. 계산 결과 약 83.3%의 정확도를 얻었다. 위의 성능평가를 통해 기계학습을 이용한 행위 기반의 악성코드 판별모델은 악성코드 탐지 성능 향상에 충분한 도움이 될 수 있음을 증명한다.

TP:악성코드를 악성코드로 예측/TN:비악성코드를 비악성코드로 예측/FN:비악성코드를 악성코드로 예측/FP:악성코드를 비악성코드로 예측.

V. 결론

본 논문에서는 로지스틱 회귀분석 알고리즘을 이용하여 행위 기반의 악성코드 판별 모델을 제안하고 실험결과를 분석하였다. 해당 탐지 모델은 악성코드에서 API를 호출하는 순서 및 연관성을 기반으로 Logistic Regression 과 Decision Tree 알고리즘을 결합한 새로운 방법론을 제시하였다. 모델의 정확도(Accuracy)를 측정한 결과, 83.3%의 정확도를 통해 해당 판별 모델이 악성코드 탐지 성능 향상에 도움이 될 수 있음을 확인하였다.

하지만, 몇 가지 부족한 요소로 인하여 추가적인 연구가 진행되어야하는 부분이 있다. 첫째로 약 1000개의 샘플데이터와 68개의 탐지 룰을 활용하여 실험을 진행하였으므로 좀 더 많은 수의 데이터와 다양한 탐지 룰이 필요하다. 둘째로 clean rule의 경우 특정 API의 존재 여부만으로 악성여부에 대한 기준을 내리는 방법론에는 불분명한 판단이 존재한다. 셋째로 실험에 사용했던 트레이닝 데이터는 2016년에 발생했던 악성코드이고 테스트 데이터는 2017년에 발생한 악성코드이다. 하지만, 2017년에 발생한 신종 악성코드의 경우 해당 행위를 탐지 못할 가능성이 있다. 악성 및 비악성 코드에 대한 샘플데이터 확보와 실시간 생성되는 악성코드 정보가 업데이트 될 수 있다면, 지속적인 피드백을 통하여 모델의 성능을 향상시킬 수 있다. 또한 해당 연구에서는 가장 적합하다고 판단되는 Logistic Regression 과 Decision Tree 분석기법을 결합한 모델을 제시하였다. 이 외에도 다양한 분석방법에 대한 실험이 필요하다.

[참고문헌]

[1] 황호・문대성・김익균 (2016). “악성코드 동적 분석을 위한 효율적인 다중실행경로 탐색법.” 정보보호학회논문지

[2] ASEC (2016.10). “Asec report vol. 70.”

[3] 박재우・문성태 외 5인 (2011). “문자열과 API를 이용한 악성코드 자동 분류 시스템.” 보안공학연구논문지

[4] 윤재성・장재욱・김휘강 (2014). “행위기반의 프로파일링 기법을 활용한 모바일 악성코드 분류 기법.” 정보보호학회논문지

[5] 박정빈・한경수 외 2인 (2015). “악성코드 분류를 위한 중요 연산부호 선택 및 그 유용성에 관한 연구.” 정보과학학회논문지

[6] 김태근・임을규 (2011). “악성코드 탐지 정확성 향상을 위한 행위 별 API List 비교 분석.” 한국통신학회 학술대회논문집

[7] 민승욱・조형진 외 2인 (2013). “안드로이드 악성코드 분석 및 기계학습 기법을 이용한 탐지 방법.” 정보과학회논문지

버에 파일이 올라갔을때 암호화가 되며, 이 경우 REST로 전송되는 과정에서는 평문이 그대로 노출될 수 있고, REST 종단간 암호화를 적용한다 하더라도, 결국 서버측에서 재암호화를 위해서 평문을 그대로 볼 수 있다는 보안 문제가 존재한다. 따라서, 본 논문에서는 클라이언트 측에서 파일을 암호화하여 올리는 방식을 고려하였다. 클라이언트 측의 암호화를 통하여 데이터를 보호할 수 있으며, 서버 측에서는 임의로 암호화된 파일을 확인할 수 없다.

일반적으로 클라우드에는 다양한 종류의 파일이 업로드 될 수 있으며, 그 중에서 특히 강력한 보안이 필요한 파일이 있을 수 있다. 이 과정에서, 각각의 클라이언트 별로 서로 다른 암호키로 적용하여 파일을 업로드하는 경우, 파일 공유 시 문제가 발생할 것이다. 예를 들어, 특정 클라이언트에서 특정키로 암호화하여 올린 데이터의 경우, 다른 클라이언트에 파일을 공유하려면 결국 암호키를 알려줘야 한다는 문제가 있다. 이러한 문제를 해결하기 위하여, 키 관리 서버를 적용하였다. 본 논문에서는 (그림 1)에서 제안하는 방식과 같이 별도의 키 관리 서버를 통하여 키를 관리할 수 있는 방법을 제안하고자 한다.



[그림 1] OpenStack Swift와 키 관리 서버

(그림 2)의 프로토콜 설명은 다음과 같다.

① 사용자는 키관리 서버(KMS)에 사용자 정보와 함께 키를 요청한다.

② 키 관리 서버는 사용자 정보를 확인하고, 사용자에 권한에 대응하는 암호화된 전달한다.

③ 사용자는 클라이언트의 키와 키 관리서버에서 전달받은 키를 XOR하여 암호키를 생성한다.

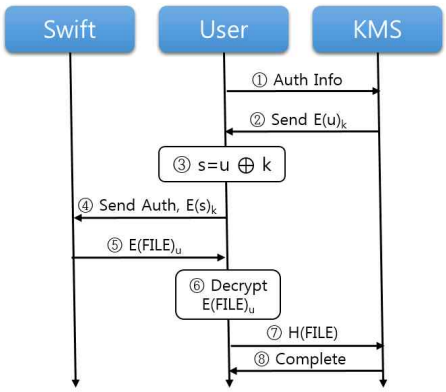
④ 사용자는 Swift 서버에 파일을 요청한다.

⑤ Swift 서버는 사용자의 신원 확인 후, 암호화된 파일을 사용자에게 전달한다.

⑥ 사용자는 앞서 생성한 암호화 키를 통하여, 파일을 복호화하고, 원본 파일을 획득한다.

⑦ 파일의 해쉬값을 KMS서버에 전송하여, 소유권의 정당성 확인 및 감사증적에 활용한다.

⑧ KMS 서버는 확인이 완료됨을 알린다.



[그림 2] Swift 키 생성 프로토콜

IV. 결론

본 논문에서는 OpenStack Swift의 객체 스토리지 환경에서서의 클라이언트 측면의 암복호화 구조와 프로토콜을 제안하였다. 제안된 구조 및 프로토콜은 클라이언트 측의 오버 헤드가 다소 발생할 수 있으나, Swift 서버와의 REST 통신과정에서 어떠한 정보도 노출시키지 않으며, 서버측의 암호화 비용을 절감시킬 수 있다는 장점이 있다. 또한, 키관리 서버를 통하여 개체간 파일 공유가 가능하다는 장점이 있다.

ACKNOWLEDGMENT

이 논문은 2017년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No.2017-0-00207,클라우드 기반 지능형 영상보안 인큐베이팅 플랫폼 개발). 교신저자 (박남제, namjepark@jejunu.ac.kr).

[참고문헌]

[1] Surekha Thota Email author, Ravi Prakash Reddy Induri, Raghavendra Kune, "Split key management framework for Open Stack Swift object storage cloud", CSI Transactions on ICT, pp.1-10, 2017.

[2] Ayache M, Erradi M, Freisleben B (2015) Access control policies enforcement in a cloud environment: openstack. In: IEEE 11th international conference on information assurance and security (IAS)

2017 한국정보보호학회 동계학술대회

D3 해킹 및 취약점(III)

이 경 문 (중부대)

- **심볼 기반 악성코드 분류 모델**
  이은지(아주대학교), 이태진 (호서대학교), 곽진 (아주대학교)
- **침해 전파과정에서 악성행위 탐지 및 분석**
  정의엽 (고려대학교), 문의성 (광운대학교), 이용민 (경찰대학교), 장진호 (호서대학교)
- **NTRU 서명 시스템에 대한 오류 주입 공격 동향 및 구현에 관한 연구**
  장호철(호서대학교), 김기현 (앤앤에스피), 하재철 (호서대학교)
- **원격제어 소프트웨어의 위협 도출 및 공격 시나리오 연구**
  양산휘, 이아름 (세종대학교), 박인혁, 조재일 (순천향대학교), 이승명 (대구카톨릭대학교), 이영주 (선린인터넷고등학교), 신정훈 (한국정보기술연구원), 김경곤 (고려대학교)

심볼 기반 악성코드 분류 모델

이은지*, 이태진**, 곽진***

*아주대학교 컴퓨터공학과 정보보호응용및보증연구실

**호서대학교 정보보호학과

***아주대학교 사이버보안학과

Symbol-based Malware Classification Model

Eun-Ji Lee*, Tae-Jin Lee**, Jin Kwak***

*ISAA Lab., Department of Computer Engineering, Ajou University.

**Department of Information Security, Hoseo University.

***Department of Cyber Security, Ajou University.

요 약

탐지를 피하기 위해 난독화, 안티 VM 등의 기술을 통해 발전된 형태로 진화하고 있는 악성코드 탐지를 위해 변종 악성코드를 분류하기 위한 방안 연구가 필요하다. 하지만, 기존의 악성코드 분석을 위한 PE 헤더 분석 기반, 시그니처 기반 등의 기술은 난독화된 변종 악성코드를 분류하는데에 한계가 있다. 또한, 기존 악성코드 분류를 위해 동적 또는 정적 feature를 분석하는 방안의 연구가 진행 중이다. 따라서, 기존 악성코드 분석 및 분류 연구를 기반으로 난독화 기술이 적용된 변종 악성코드를 고려하여 효율적인 변종 악성코드 분류 방안을 연구할 필요가 있다. 이에 따라, 본 논문은 심볼 기반 악성코드 분류 모델을 제안한다. 이는 난독화된 악성코드 바이너리파일에서 특정 심볼이 많이 나타나며, 변종 그룹 내 Strings 분포가 유사한 특성을 이용해 악성코드 변종 그룹을 분류하는 것을 목표로 한다.

I. 서론

최근 APT, 랜섬웨어 등 지능화된 악성코드가 지속적으로 출현하고 있다. 또한, 악성코드는 탐지를 피하기 위해 코드 난독화, 안티 VM 등의 기술을 통해 발전된 형태로 진화하고 있다[1].

하지만, 기존의 악성코드 정적 분석을 위한 PE 헤더 분석 기반, 시그니처 기반 등의 방식은 난독화에 따른 변종 악성코드 분류에 한계가 있다[2]. 또한, 변종 악성코드 분류를 위한 연구로는 주로 악성코드의 동적 또는 정적 feature를 조합하여 비교 분석하는 방안으로 진행되고 있다[3,4].

이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2017R1E1A1A01075110).

따라서, 기존의 악성코드 탐지 또는 악성코드 분류 연구를 기반으로 난독화 기술을 적용하여 진화하고 있는 악성코드를 고려하여 효율적인 변종 악성코드 분류 방안 연구가 필요하다.

이에 따라, 본 논문은 심볼 기반 악성코드 분류 모델을 제안한다. 이는 악성코드 난독화시 바이너리파일에서 특정 심볼이 많이 나타나며, 변종 악성코드 그룹에서 특정 Strings 분포가 유사한 특성을 기반으로 한다[4]. 또한, 악성코드 그룹 별 악성코드 바이너리 파일에서 추출한 특정 심볼 문자열의 비율에 대한 유사도 분석을 통해 변종 악성코드를 분류할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 기존 악성코드 분류 연구를 분석한다. 3장에서는 제안하는 모델을 설명하고 4장에서 실험 결과를 분석하며, 5장에서 결론을 맺는다.

II. 관련연구

2.1 feature 기반 악성코드 분류 연구

악성코드 변종 또는 신종 탐지를 위한 악성코드 분류연구가 지속적으로 진행되고 있으며, 이는 주로 동적 또는 정적 feature 기반으로 연구되고 있다[3,4].

이 중, M. Ahmadi 등[4]은 악성코드 분류를 위해 Hex view 기반의 feature와 Assembly view 기반의 feature를 고려하였다. 해당 연구에서 고려한 몇 가지 feature들에 대한 설명은 아래와 같다.

❑ 엔트로피(Entropy)

불일치 정도를 측정하는 엔트로피를 이용해 난독화 가능성을 감지할 수 있다. 슬라이딩 윈도우 방식을 이용해 악성코드를 일련의 엔트로피 수치로 나타내고, 이에 대한 통계적 측정 값을 통해 엔트로피를 계산한다.

❑ 메타데이터(Metadata)

Hex view 또는 Assembly view 각각 에서는 파일 크기, 주소 값 등의 메타데이터를 추출할수 있으며, 악성코드 메타데이터의 특성을 분석할 수 있다.

❑ 심볼(Symbol)

난독화의 예로, 특정 주소를 호출하는 부분이 난독화되는 경우 해당 데이터에 대한 위치를 알 수 없다. 이러한 난독화된 파일에서 +, @, * 등과 같은 심볼이 많이 나타난다.

❑ 섹션(Section)

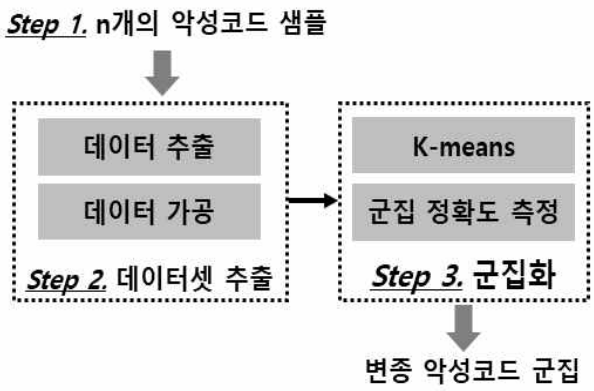
PE 파일은 .text, .edata, .rdata 등의 섹션으로 이루어져있다. 패킹 등의 기술로 섹션을 수정할 수 있으며, 이러한 이유로 악성코드의 섹션 특성을 추출할 수 있다.

III. 제안기법

악성코드 분석 시 난독화 기술이 적용된 변종 악성코드를 고려할 때 효율적으로 분석이 가능한 feature로써, 심볼을 악성코드 분류를 위한 feature로 선정할 수 있다. 본 논문은 변종 그룹 내 Strings의 분포가 유사하게 나타나는 특징을 이용하여, 변종 그룹 별 임의의 심볼 문자열 분포를 유사도 분석하는 방안을 통한 악성코드 분류 연구를 진행한다.

3.1 제안하는 모델

제안하는 심볼 기반 악성코드 분류 모델은 [그림 1]과 같으며, 각 동작 단계에 대한 설명은 아래와 같다..



[그림 1] 제안 모델

Step 1. 악성코드 바이너리 파일 입력

악성코드 n개의 샘플 바이너리 파일을 입력한다.

Step 2. 데이터셋 추출

❑ 데이터 추출

입력된 악성코드 샘플 파일에서 임의의 심볼 기반의 데이터셋을 추출한다. 우선, feature로 사용할 임의의 심볼들을 목록화한다.

이때, 목록화한 각 심볼을 포함하는 임의의 문자열들을 악성코드 샘플 파일에서 추출할 수 있다. 이를 통해 각 악성코드 샘플 파일 별 심볼 문자열 개수를 데이터로 추출한다.

❑ 데이터 가공

데이터 추출단계를 통해 추출된 비선형 데이터를 가공하여 범주화된 형태로 변환할 필요가 있다. 이를 위해, 추출된 데이터인 악성코드 샘플 파일 별 심볼 문자열 개수를 각 심볼 별 비교하여 임의의 범위로 분류한다. 이때, 기존의 심볼 문자열 개수를 분류한 범위의 중간 값으로 설정함으로써 데이터를 가공할 수 있다.

Step 3. 군집화

악성코드 변종여부를 분석하기 위해 유사도 분석을 통해 유사한 특성을 갖는 악성코드 별로 군집화하는 단계이다. 데이터셋 추출단계에서 추출된 데이터셋을 기반으로 k-means 클러스터링 알고리즘을 이용하여 각 심볼별 데이터 간 유사도 분석 및 군집화한다. 결과적으로, 기존에 분석된 백신명을 기반으로 군집 정확도를 측정한다.

IV. 실험 결과 및 분석

실험을 위해 변종 악성코드를 포함하는 악성코드 바이너리 샘플 1,000개를 선정하여, 제안하는 악성코드 분류 모델의 입력 값으로 사용하였다. 또한, 제안하는 모델의 각 동작 단계에 사용된 도구는 [표 1]과 같다.

[표 1] 각 동작단계 별 사용 도구

동작 단계		도구
데이터셋 추출	심볼 문자열 추출	Strings
	심볼 문자열 개수 추출	c언어 기반 구현 프로그램
군집화		SPSS

4.1 데이터셋 추출 결과

❑ 데이터 추출

데이터 추출을 위해 우선, 임의의 심볼을 선정할 필요가 있으며, 본 연구를 위해 @, !, *, % 등 29개의 심볼을 선정하여 진행하였다.

다음으로, Strings 도구를 이용해 각 샘플 파일별 심볼 문자열을 추출한다. 이때, Strings 도구는 바이너리 파일 내 문자열들을 찾아 출력해주는 프로그램이다.

이후, Strings 도구를 이용해 추출한 심볼 문자열을 기반으로 각 샘플 파일별 각 심볼 문자열 개수를 추출한다. 이에 따른 데이터 추출 단계를 통해 추출된 데이터셋의 예는 [표 2]와 같다.

[표 2] 데이터 추출 단계 결과의 예

파일	심볼x	심볼y	…	심볼m
파일1	x1	y1	…	m1
파일2	x2	y2	…	m2
파일3	x3	y3	…	m3
. . .	. . .	. . .	. . .	. . .
파일n	xn	yn	…	mn

❑ 데이터 가공

데이터 추출 단계에서 추출된 비선형 데이터를 범주형 값으로 변환하기 위해, 샘플 파일 별 심볼 문자열 개수를 임의의 범위로 분류하였다. 이후, 기존의 심볼 문자열 개수를 분류한 범위의 중간 값으로 선정 및 변환하여 데이터를 가공하였다.

이에 따른, 심볼 문자열 개수를 분류한 범위별 선정한 데이터 가공 결과의 예는 [표 3]과 같다.

[표 3] 데이터 가공 결과의 예

임의의 심볼 문자열 개수 범위	데이터 가공 값
1~10개	5
11~30개	20
31~50개	40
51~100개	75
101~200개	150
· · ·	· · ·
801개 이상	1,000

4.2 군집화 결과

우선, 기존 백신회사에서 분석한 각 파일별 백신명을 기반으로 1,000개의 샘플 파일을 재분류한다. 재분류한 결과 1,000개의 샘플 파일을 63개의 악성코드 군으로 분류할 수 있었다.

다음으로, 추출한 데이터셋을 기반으로 유사도 분석 및 군집화하기 위해 SPSS 도구를 이용해 k-means 클러스터링을 적용하여 진행하였다. k-means 클러스터링은 k개의 중심 값을 선정하여 중심 값을 기준으로 가까운 거리의 데이터들을 군집화하는 군집화 알고리즘이다[5]. 이때, 앞서 기존 백신명에 따라 분류한 결과에 따라 군집수 k는 63개를 지정하여 진행하였다.

마지막으로, 군집화 단계를 통해 얻은 군집화 결과의 정확도를 측정하기 위해 군집 정확도 산정식을 이용해 진행하였다[6]. 군집 정확도 산정식은 각 군집별 군집에 속한 패밀리 수 및 각 군집별 패밀리와 기존 군집별 패밀리 간 일치/비일치 여부의 비율을 통해 측정된다. 이때, 산정식에 사용되는 패밀리 수는 각 샘플 파일 수로 나타낼 수 있다. 이에 따른, 군집 정확도 결과는 [표 4]와 같다.

[표 4] 군집 정확도 결과

총 패밀리수	일치 패밀리수	비일치 패밀리수	군집 정확도
1000	859	141	85.9%

V.결론

본 논문은 심볼 기반 악성코드 분류 모델을 제안하였다. 이는 악성코드 바이너리 파일로부터 추출한 특정 심볼 문자열의 분포를 악성코드 파일별 유사도 분석 및 군집화함으로써 변종 악성코드 군을 분류할 수 있다.

제안하는 모델을 통해 난독화가 적용된 변종 악성코드 분석 및 탐지에 제한적이었던 기존의 악성코드 분석 방안을 개선할 수 있다.

[참고문헌]

[1] Smita Naval, Vijay Laxmi, Muttukrishnan Rajarajan, Manoj Singh Gaur and Mauro Conti, “Employing Program Semantics for Malware Detection”, IEEE Transactions on information forensics and security, Vol. 10, No. 12, December. 2015.

[2] watchguard, “Internet Security Report”, January. 2017.

[3] Ying Fang, Yu-feng Huang and Bo Yu, “A New Malware Classification Approach based on Statistical Feature”, International Conference on Computer Science and Technology (CST 2017), July. 2017.

[4] M. Ahmadi, D. Ulyanov, S. Semenov, M. Trofimov and G. Giacinto, “Novel Feature Extraction, Selection and Fusion for Effective Malware Family Classification”, CODASPY ’16, March. 2016.

[5] Jain, Anil K, “Data clustering: 50 years beyond K-meansq”, Pattern Recognition Letters 31, June. 2010.

[6] http://challenge.cisc.or.kr

침해 전파과정에서 악성행위 탐지 및 분석

정의엽*, 문의성**, 이용민***, 장진호****

*고려대학교 컴퓨터공학과

**광운대학교 전자공학과

***경찰대학교 법학과

****호서대학교 정보보호학과

Lateral Movement Analysis

Eui-Yub Jung*, Eui-Seong Moon**, Yong-Min Lee***, Jin-Ho Jang****

*Dept. of Computer Science, Korea University

**Dept. of electric Enginerring, Kwangwoon University

***Dept. of Law Korean National Police University

****Dept. of Information Security, Hoseo University

e-mail : moon7392@naver.com

요 약

최근 발생하는 침해사고들의 경우 사용되는 기법들이 보다 다양해지고 있으며, 기존의 보안 솔루션으로는 탐지하기 힘든 사례들이 빈번히 발생하고 있다. 결과적으로 기업이 침해사고를 인지하는 시점은 침해사고로 인해 피해를 입은 후가 대부분이고, 이는 탐지 메커니즘이 기본적으로 침해사고의 유입 단계에 초점을 맞추고 있기 때문이다. 본 논문에서는 침해 유입 차단을 통한 원천적인 방어가 아니라 침해 전파과정에서 침해사고를 인지하여 피해를 최소화할 수 있는 방안을 강구하고자 한다. 이를 위해 전파과정에서 주로 사용되는 공격기법을 행위 별 최소 단위로 분류하여 각각의 행위분석 결과의 조합을 통해 공격행위와 정상행위를 구분 지을 수 있는 Lateral Movement Intelligence를 생성하고, 실제 공격 시나리오에 적용하여 침해전파 과정에서 침해사고를 탐지 및 대처할 수 있는 가능성을 제시한다.

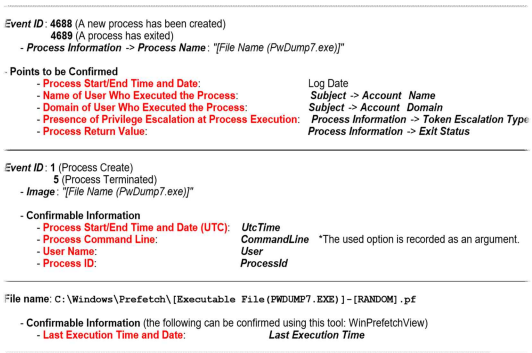
I. 서론

침해사고의 단계는 크게 나누면 침해유입, 침해전파, 침해실행, 침해지속의 4단계로 나눌 수 있다. 이 중 침해 사고의 원천 차단을 위한 유입단계에서의 방어를 위한 연구는 활발히 진행되고 있으나, 공격기법의 고도화 및 지능화에 의해 이러한 접근이 현실적으로 효용성을 드러내지 못하는 사례가 더욱 많아지고 있다. 최근에 이슈 된 '인터넷나야나' 사건에서 공격자는 유입 후 153대의 추가 피해서버가 발생하는 침해전파과정을 성공시켰지만 해당 업체는 유입단계에서의 탐지에 실패하고, 핵심 정보 유출이 발생한 뒤에 침해를 인지하였다. 이러한 케이스에 대하여 보다 효율적인 보안기법을 마련하기 위해, 본 논문에서는 침해전파단계 (이하 Lateral movement)에 초점을 둔다. Lateral movement의 탐지에 성공할 경우, 유입단계 공격은 허용하더라도 최종적으로 공격자가 원하는 정보유출 및 데이터 파괴와 같은 궁극적인 피해는 막을 수 있어 최근의 공격 트렌드를 고려하였을 때 보안 측면에서 충분히 가치있다고 볼 수 있다. 본 논문은 침해내부전파를 탐지하기 위한 방안을 도출하기 위해 관련연구를 조사, 보안 실무자들에게 제공할 수 있는 Lateral Movement Intelligence의 구축, 최종적으로 대

표적인 침해전파 사례의 탐지 가능성을 살펴보았다.

II. 관련연구

기존의 Lateral movement를 대상으로 한 연구의 경우, Lateral movement에 사용되는 기법들을 세분화하여 각각을 특정해낼 수 있는 시스템 상의 변화에 초점을 둔다. JPCERT의 경우 원격 명령 실행, 계정 인증 값 탈취, 파일 공유 등 세부 기법을 분류한 뒤 그에 사용되는 Mimikatz, WCE, Net, At, PsExec와 같은 툴들의 관련 아티팩트를 정리하였고, EU-CERT의 경우 Active Directory 환경에서의 인증 정보 탈취 관련 아티팩트를 유사한 기준 아래 정리하였다.



(그림 1) PwDump7 실행 아티팩트 (JPCERT)

이러한 기존 관련연구는 주로 개별적인 아티팩트에 초점을 맞추고 있다. 그러나 Lateral movement의 경우 세부 기법에 있어서 악성이 아닌 윈도우즈 기본 관리 툴을 이용하는 경우가 많다. 따라서 개별적인 아티팩트만을 가지고 침해 전파를 판단하는 것은 높은 오탐률을 가질 위험이 있고, 이를 낮추기 위해서는 개별 아티팩트의 연결을 통해 행위의 문맥을 파악하는 것이 중요하다. 이 점을 염두에 두고 본 논문에서는 실제 공격을 수행하고 시간 순으로 정리하여 단일 행위가 아닌 시스템 상의 일련의 행위를 탐지의 기준으로 두었다. 추가적으로 기본 아티팩트, 추가 확보 가능한 아티팩트 등 다양한 흔적을 수집하고, 이를 탐지율을 보완하기 위한 데이터로 활용하고자 하였다.

III. Lateral Movement Intelligence

3.1. Intelligence 생성을 위한 환경 구축

Intelligence 생성에 앞서 그에 필요한 데이터에는 일련의 행위 안의 각 행위 단계 별 시스템 변화를 나타내는 로그 등 실시간 정보가 포함된다. 이를 수집하기 위해, 공격의 대상이 되는 환경을 구축하고, 각 공격기법을 시도한 뒤 다양한 로그 수집, 모니터링 툴 등을 이용하여 데이터를 확보하였다. 내부망에서의 호스트에 사용되는 운영체제는 기업에서 높은 점유율을 보유하고 있는 Windows 7을 사용하였고, Active Directory와 같은 환경에 이용되는 서버의 운영체제는 Windows 2012 R2를 사용하였다.

3.2. Intelligence 구축을 위한 공격기법 분류

Lateral Movement Intelligence를 생성하기 위해 기존의 APT 보고서들이 발표한 Lateral Movement 공격기법을 행위 별 최소 단위로 분류하여 각각의 행위를 분석했다. 개별적인 행위에 대한 흔적 수집 시 최대한 많은 정보를 확보하기 위해 이벤트 뷰어, Sysmon, Procexp, Procmon 등 다양한 수집도구를 사용했다.

공격기법	공격도구	행위
원격실행	PsExec Wmic Powershell Wmiexe WinRS At	타 PC에 프로세스 생성 및 원격 실행
정보탈취	PWDump7 PWDumpX QarkDump Mimikatz WCE WBPV RDPV	SAM 인증 정보 추출, 메모리로부터 패스워드 혹은 해시 추출
공유	net share icacls net use	파일 복사를 위한 공유
권한상승	bypassuac_comhijack bypassuac_eventvwr bypassuac_injection bypassuac_vbs MS10_092 MS14_058 MS15_051 MS16_032 MS16-135	감염PC에서의 추가적인 행위를 위한 권한획득
계정생성	net user	공격에 사용될 계정생성

(표 1.) 실험에 사용된 공격 기법

(표 1)의 공격들을 수행하여 흔적들을 수집하고 발생 시간을 고려한 흔적 간의 조합을 통해 공격행위와 정상행위를 구분 짓는 데에 사용될 수 있는 Lateral Movement Intelligence를 약 50개 생성했다.

IV. Lateral Movement 탐지 검증

본 장에서는 Lateral movement의 대표적인 시나리오를 가정하고, 실험을 통해 확보한 Intelligence가 이를 탐지하는 데에 활용할 수 있는지 확인하고자 한다.

4.1 원격실행 프로세스의 인증정보 탈취 시도

Lateral movement의 필수적인 과정 중 하나는 네트워크 내의 한 호스트에서 다른 호스트의 인증 정보를 획득하여 감염을 이동시키는 과정이다. 이를 위해 시나리오에서는 내부망 내의 한 단말이 감염되었을 때에, 그 단말로부터 다른 호스트의 계정 정보에 접근을 시도하여 추가적인 계정 정보를 획득하고자 하는 상황을 가정한다.

이러한 시나리오에서 인증 값 탈취를 탐지하기 위한 Lateral Movement Intelligence의 데이터는 아래 (표 2)와 같이 mstsc.exe외의 프로세스가

C:\Users\Host1\AppData\Roaming\Microsoft\Credentials

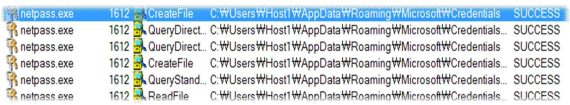
경로에 접근하고, 이벤트로그 Security ID 4648(자격증명을 통한 로그인시도) 및 4624(로그인 성공)가 남지 않는 경우이다. 아래의 수집 도구 및 이벤트 로그, Sysmon 데이터를 통해 이를 확인할 수 있다.

수집도구	흔적
	내용
Procmon	C:\Users\Host1\AppData\Roaming\Microsoft\Credential
	자격증명이 저장되어진 특정 경로에 mstsc.exe 프로세스를 제외한 다른 프로세스의 접근유무 파악
이벤트뷰어	4648 (자격증명 통한 로그인시도)
	자격증명을 통해 로그인 시도하지 않아 남지 않음.
	4624 (로그인 성공)
	로그인을 안 하기 때문에 남지 않음.
Sysmon	3
	Network통신이 이루어지지 않아 남지 않음.

(표 2) 인증 값 탈취 관련 이벤트 로그

상세히 살펴보면, 공격자는 Netpass.exe 프로그램을 통해서, 자격증명에 접근해 타 PC의 계정 ID, PW를 탈취한다. 이 때 Netpass.exe는 자격증명이 저장된

C:\Users\Host1\AppData\Roaming\Microsoft\Credentials 의 경로로 접근한다.



(그림 2) Promon으로 수집한 Netpass.exe 흔적

(그림 2)은 Netpass.exe가 해당 경로에 접근하는 흔적을 Procmon을 통해서 잡아낸 것이다. 이러한 행위가 발생하였을 때에 (표 2)의 이벤트 로그 데이터를 살펴보면, 인증 정보에 접근하는 프로세스가 mstsc.exe가 아니고 Sysmon 이벤트 3, 이벤트로그 Security 4648, 4624가 발생하지 않으므로 행위를 공격행위로 탐지할 수 있다.

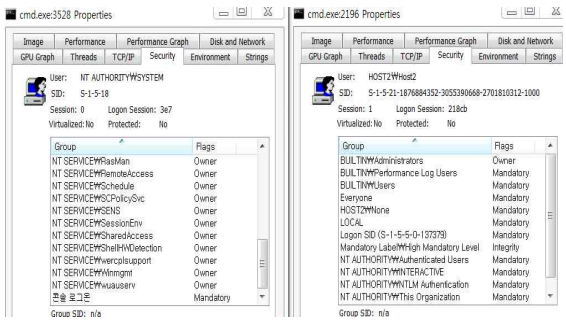
다만, 다양한 시스템 환경을 고려하였을 때 이러한 행위를 하는 정상 프로세스가 존재할 가능성이 있으므로, 이 전의 행위를 분석에 포함시킨다면 보다 정확한 탐지의 기준을 세울 수 있을 것이다. 예를 들어, 계정 인증 정보에 접근하는 프로세스가 타 PC로부터 원격실행이 되었음을 확인할 수 있다면, 이 행위가 정상행위가 아닌 침해전파과정에서의 악성행위임을 판단하는 것에 더 높은 신뢰성을 보장할 수 있다.

실행된 프로세스와 관련된 데이터 중 원격실행 여부는 해당 프로세스의 접근 권한 그룹을

통해 확인할 수 있다. 윈도우즈의 프로세스는 부모 프로세스가 자식 프로세스에게 자신의 그룹 정보를 복사한다. 따라서 특정 쉘로부터 실행된 인증 정보 탈취 프로세스를 자식 프로세스라고 할 때 자식 프로세스의 접근 권한을 확인하면 RemoteAccess 등 로컬에서 실행되는 프로세스와 확연히 다른 정보를 가지게 된다. 해당 사실 확인을 위한 정보는 아래의 수집도구를 통해 가능하다.

수집도구	흔적
	내용
Procexp	접근 권한 정보 전자서명 여부
	해당 프로세스의 접근 권한 그룹을 통해 원격 실행 여부 파악
Procmon	네트워크 통신 정보
	원격 실행 관련 정보 파악

(표 3) 원격실행여부 파악을 위한 데이터



(그림 3) 프로세스 접근 가능 사용자 정보의 비교
(좌: 원격 실행, 우: 로컬 실행)

이와 같이 두 가지 단일 행위의 연결을 통해서 프로세스가 실행되고 있는 시스템 상에서의 문맥을 더 상세하게 확인하고, 이러한 일련의 행위를 탐지의 기준으로 잡는다면 오탐율을 낮출 수 있다. 이에 덧붙여, 프로세스의 전자서명 정보 및 프로세스의 가계도로부터 이러한 문맥에 필요한 정보를 더 상세히 확인할 수 있다.

V. 결론

본 논문에서는 전파과정에서 사용되는 공격 기법을 분류하고, 각 기법의 행위가 생성하는 흔적들을 모아 일련의 행위로 연결하여 Lateral Movement Intelligence를 생성 및 특정 시나리오를 기반으로 정상행위와 악성행위를 구분 지을 수 있는 가능성을 제시했다.

기존의 침해유입 관련 연구와 달리 침해전파 과정에서 탐지를 하는 이러한 시도는 침해사고의 경위가 더 복잡해짐에 따라 유용하게 쓰일 수 있다. 기업의 입장에서도 침해사고가 발생하였을 때에 그 피해를 최소화하고, 사고 경위 분석을 위한 포렌식 준비도의 관점에서도 유용하게 활용될 수 있을 것이다. 또한, 이러한 데이터를 보다 정제 및 가공하여 시각화를 통해 보여준다면 분석가 및 기업 보안 실무자의 입장에서도 침해사고 경위 파악에 있어 의미 있는 시야를 제공할 수 있을 것이다.

[참고문헌]

[1] JPCERT, “Detecting Lateral Movement through Tracking Event Logs”, Japan Computer Emergency Response Team Coordination Center, 2017. Retrived from https://www.jpcert.or.jp/english/pub/sr/ir_research.html

[2] CERT-EU, “Detecting Lateral Movements in Windows Infrastructure”, CERT-EU Security Whitepaper, 2017, Retrived from http://cert.europa.eu/static/WhitePapers/CERT-EU_SWP_17-002_Lateral_Movements.pdf

[3] Brian Baskin, 2015, 7, “Noriben: Your Personal, Portable, Malware Analysis Sandbox”, Black Hat US 2015

[4] kbandla.(2013),“APTnotes”,
“https://github.com/kbandla/APTnotes”, (accessed Sep 30, 2017)

[5] Junghoon, Oh(2015), “A Forensic Analysis of APT Lateral Movement in Windows Environment”, Ahnlab, https://www.first.org/resources/papers/conference2014/a-forensic-analysis-of-apt-lateral-movement-in-windows-environment.pptx (accessed Sep 27, 2017)

- 본 논문은 차세대 보안리더 양성 프로그램 BoB(Best of the Best)의 지원을 받아 작성되었습니다.

NTRU 서명 시스템에 대한 오류 주입 공격 동향 및 구현에 관한 연구

장호철*, 김기현**, 하재철*

*호서대학교, **(주)앤앤에스피

Fault Analysis for NTRU Signature System

Ho-Cheol Jang*, Ki-Hyun Kim**, Jae-Cheol Ha*

*Hoseo University, **NNSP

요 약

포스트 양자 암호는 기존 RSA와 같은 시스템을 위협하는 양자 시스템으로부터의 공격에도 안전한 차세대 암호이다. 하지만 포스트 양자 암호도 실제 디바이스에 구현할 때 부채널 분석에 대한 취약점이 존재한다. 이에 따라 부채널 분석 기술에 대한 대응 기법이 필요한 실정이다. 따라서 본 논문에서는 NTRU 서명 시스템에 대한 오류 주입 공격 기법에 대해 알아보고자 한다. 또한 소개한 오류 주입 기법을 실제 구현, 시뮬레이션 하여 실질적인 위협이 됨을 설명하고 이에 대응하고자 한다.

I. 서론

기존 공개키 기반의 서명 시스템의 안전성은 충분히 큰 수에 대한 소인수 분해 문제 혹은 이산로그 문제 등 수학적 난제에 그 기반을 두고 있다. 그러나 양자 컴퓨팅 기술이 개발, 보급되면 현재 기술로는 해결하기 어려운 수학적 난제들을 빠른 시간 안에 풀 수 있게 된다. 1995년 Shor는 소인수분해 문제와 이산 로그 문제를 양자 컴퓨터로 해결하기 위한 알고리즘을 제시한 바 있다[1]. Shor의 알고리즘은 현재 사용되고 있는 RSA, DSA, ECDSA 와 같은 시스템들을 실질적으로 위협할 수 있는 알고리즘이다. 따라서 이러한 위협에 대응하기 위해 포스트 양자 암호 시스템에 대한 연구가 필요하다.

포스트 양자 시스템이 이론적으로 안전하다 할지라도, 이를 실제 디바이스에 구현할 때 구현상의 취약점이 존재할 경우 부채널 분석이나 오류 주입 공격에 의해 비밀 키와 같은 민감한 정보가 노출될 가능성이 있다. 오류 주입 공격은 알고리즘이 동작하는 일련의 과정 중 특정 부분에 오류를 주입하여 데이터나 클록에 변화를 주고, 이로부터 얻는 부가 정보를 이용하여 민감한 정보를 찾아내는 공격이다.

포스트 양자 시스템 중 격자 기반의 NTRU 서명 시스템에 대한 오류 주입 공격은 세부 알고리즘에 따라 그 형태를 달리 한다. 본 논문에서는 NTRU 서명 시스템에 대한 오류 주입 기법을 소개하고 이를 구현하였다.

II. NTRU 서명 시스템

NTRU 서명 시스템은 격자 기반 포스트 양자 암호시스템의 오픈 소스 공개 키 암호 시스템이다.

2.1 NTRUSign

NTRUSign은 가장 기본적인 NTRU 서명 시스템으로 키 생성 알고리즘, 서명 알고리즘,

검증 알고리즘 총 3개의 알고리즘으로 구성되어 있다. NTRUSign 키 생성 알고리즘은 [그림 1]과 같고, 서명 알고리즘은 [그림 2]와 같다.

Key Generation

1. 정수 $N, q, N, B \geq 0$, 이진 다항식 df, dg, trinary 다항식 d, NTRU 타입(standard 또는 transpose)
2. $i = B$, $i \geq 0$ 일 때 a., b., c. 반복
   a. df, dg에 대한 랜덤 이진 다항식 $f, g \in R$ 선택, $d+1$에 대한 trinary 다항식 f, g 선택. (f는 NTRU 타입 standard 일 때 R_q상에서 역을 가져야 하며, g는 NTRU 타입 transpose 일 때 R_q상에서 역을 가져야 함.)
   b. $f \star G - F \star g = q$를 만족하는 작은 다항식 $F, G \in R$ 계산
   c. $f_i = f$, NTRU 타입이 standard일 때, $f_i' = F_i, h_i = f_i^{-1} \star g \bmod q$
      NTRU 타입이 transpose일 때, $f_i' = g_i, h_i = f_i^{-1} \star F_i \bmod q$, $i = i-1$
3. 공개 파라미터 : 공개키 $h = h_0$, 입력 파라미터
4. 개인(비밀) 파라미터 : f_i, f_i', h_i $(i = 0..B)$

[그림 1] NTRUSign 키 생성 알고리즘

Signing

1. Digital document D, 개인 키 집합 $\{f_i, f_i', h_i\}$ 입력
2. $r = 0$
3. $s = 0, i = B$, $m_0 = H(D \| r)$, $m = m_0$
4. $i \geq 1$일 때 다음 a., b., c.을 계산
   a. $A = \left[\frac{-f_i' \star m}{q}\right], B = \left[\frac{f_i \star m}{q}\right], S_i = A \star f_i + B \star f_i' \bmod q$ 계산
   b. $m = s_i \star h_i - h_{i-1} \bmod q$
   c. $s = s + s_i, i = i-1$
5. $A = \left[\frac{-f_0' \star m}{q}\right], B = \left[\frac{f_0 \star m}{q}\right], s_0 = A \star f_0 + B \star f_0' \bmod q$, 서명 $s = s + s_0$ 계산
6. 서명 검사를 위해 다음 a., b.를 계산
   a. $b = \| s, s \star h - m_0 \bmod q \|$
   b. 만약 $b \geq N$라면, $r = r+1$ 후 단계 3으로 이동
7. (D, r, s) 출력

[그림 2] NTRUSign 서명 알고리즘

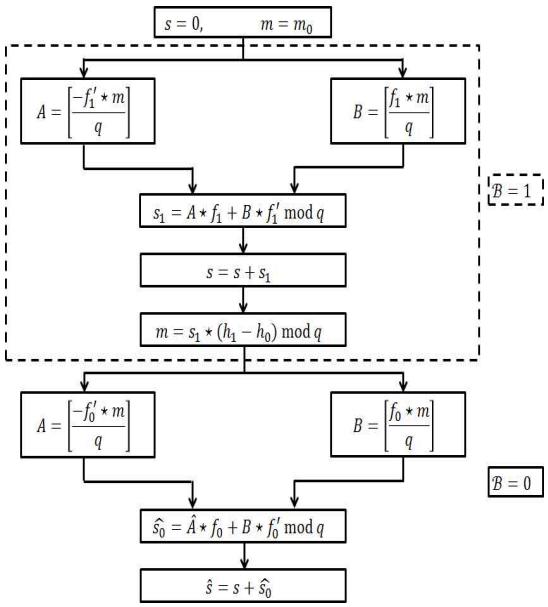
NTRUSign 검증 알고리즘은 서명 알고리즘의 출력인 (D, r, s)와 공개 키 h를 입력으로 $m = H(D \| r)$ 을 연산한 후 $b = \| s, s \star h - m_0 \bmod q \|$ 에 대해 $b \leq N$이면 유효한 서명임을 나타내는 "valid"를 출력하고, 아닐 경우 "invalid"를 출력한다.

III. 오류 주입 공격

오류 주입 공격은 암호 알고리즘을 실제 구현한 디바이스에서 구동시킬 때 오류를 발생시켜 나타나는 부가 정보를 분석하여 비밀 키와 같은 데이터를 알아내는 기법이다.

3.1 NTRUSign에 대한 오류 주입 공격

NTRUSign에 대한 오류 주입 공격은 서명 과정 중에 오류를 주입하여 서명을 위조하고, 출력 값으로부터 비밀 키를 찾아내는 기법을 사용한다.



[그림 3] NTRUSign 서명 흐름도

공격자가 서명 s_0를 계산하기 전, A가 계산되는 과정에 오류를 주입할 수 있다고 가정한다. $\hat{s} = s_1 + \hat{s_0}$ 이고, 여기서 $\hat{s_0} = \hat{A} \star f_0 + B \star f_0' \bmod q$, $\hat{A} = A + \epsilon \bmod q$ 이다. 공격자는 오류 서명과 정상 서명의 차이가 $\triangle s = \hat{s} - s = (s_1 + \hat{s_0}) - (s_1 + s_0) = \hat{s_0} - s_0$ 이고, 이는 곧 $\triangle s = \epsilon \star f_0 \bmod q$ 가 된다. 따라서 $f_0 = \epsilon^{-1} \star \triangle s \bmod q$ 를 계산해 비밀 키를 얻어낼 수 있다. 모든 가능한 경우에 대한 오류 ϵ을 반복 시행하여 f_0를 얻은 공격자는, $g_0 = f_0 \star h \bmod q$ 를 계산할 수 있다.

IV. 시뮬레이션

NTRU 서명 시스템 및 오류 주입 공격 구현

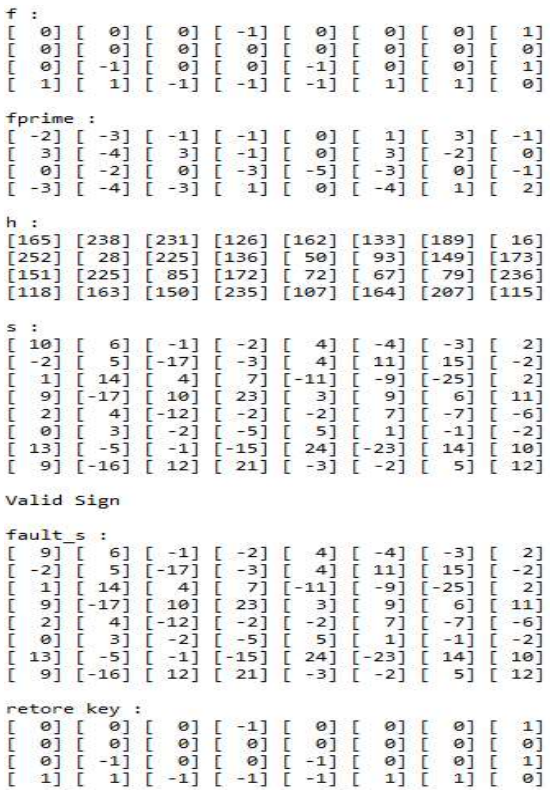
환경은 다음과 같다.

-개발 환경: Windows 7 Enterprise K

-개발 도구 : Eclipse

-개발 언어 : JAVA

다음 [그림 4]는 NTRUSign 및 NTRUSign에 대한 오류 주입 공격 결과를 구현한 화면이며, 데이터의 첫 32바이트를 출력한 결과다.



[그림 4] 구현 결과

[그림 4]의 데이터는 모두 처음 32바이트만 출력한 데이터이다. 정상 서명 s에 대한 연산을 완료하고, A에 대한 오류 주입을 반복 시행하여 오류 서명 $\hat{s}$과의 차분을 통해 실제 비밀 키가 복구될 수 있음을 확인하였다.

V.결론

본 논문에서는 격자 기반 포스트 양자 암호 시스템인 NTRUSign 시스템에 대한 오류 주입 기법을 소개하였다. 또한 서명 시스템에 대한 오류 주입 공격을 구현, 시뮬레이션을 통해 시스템에 대한 공격이 실제로 가능함을 보였다.

NTRUSign에 대한 오류 주입 기법은 서명 생성 단계에서 오류를 주입하고 오류 서명과 정상 서명의 차이로부터 비밀 키를 얻어내는 공격이다.

따라서 NTRU 서명 시스템을 구현할 때 오류 주입 분석에 대응할 수 있는 대응기법에 대한 연구가 활발히 진행되어야 한다.

[참고문헌]

[1] P.Shor, “Algorithms for quantum computation: Discrete logarithms and factoring,” Proceedings of the 35th Annual Symposium on Foundations of Computer Science, pp. 124-134, Nov. 1994

[2] Leo Ducas, Thomas Prest, and Vadim Lyubashevsky, “Efficient identity-based encryption over NTRU lattices“, Asiacrypt 2014.

[3] A. A. Kamal and A. M. Youssef. Fault analysis of the NTRUSign digital signature scheme. Cryptography and Communications, 4(2):131 - 144, 2012.

[4] Hoffstein, J., Graham, N., Pipher, J., Silverman, J., Whyte, W.: NTRUSign: Digital signatures using the NTRU lattice. Draft 2, NTRU Cryptosystem Inc. (2002).

원격제어 소프트웨어의 위협 도출 및 공격 시나리오 연구

양산휘* ,이아름* ,박인혁** ,조재일** ,이승명*** ,이영주****
,신정훈***** ,김경곤*******
*세종대학교 정보보호학과, **순천향대학교 정보보호학과,
***대구가톨릭대학교 정보보호학과, ****선린인터넷고등학교,
*****한국정보기술연구원(Best of the Best), ******고려대학교 정보보호대학원

Threat Deduction and Attack Scenarios for Remote Access Software

Sanhwi Yang*, Arem Lee*, InHyeok Park**, Jail Jo**, Seungmyung Lee***, Youngjoo Lee****, Jeonghoon Shin*****, Kyounggon Kim******
*Dept. of information security,Sejong University.
**Dept. of information security,Soonchunhyang University.
***Dept. of information security,Deagu Catholic University.
****Sunrin Internet Highschool.
*****KITRI (Best of the Best).
******Dept. of information security,Korea University Graduate School.

요 약

IT 기술의 발전함과 동시에 편리성을 추구하는 소프트웨어 기술이 발전하고 있고 원격제어 소프트웨어가 화두 되고 있다. 원격제어 소프트웨어는 시간과 공간적인 제약을 극복하여 보다 효율적인 환경을 만들어준다. 하지만 원격제어 소프트웨어의 보안적인 측면 또한 크게 대두되고 있다. 본 논문에서는 원격제어 소프트웨어 대표적인 프로토콜과 보안 위협 및 공격 시나리오를 도출하고 원격제어 소프트웨어를 보호하기 위한 대응 방안을 제시함으로써, 안전한 원격제어 소프트웨어 프로그램의 안전성을 향상시킬 수 있을 것으로 기대한다.

I. 서론

원격제어 소프트웨어는 원격지에서 자신의 PC 환경을 이용하거나, PC들을 관리하기 위해 개발, 발전되어 왔다. 현재는 파일전송, 관리, 고장 난 PC의 원격 진단 및 시스템 관리자의 클라이언트 원격 관리, 원격 강의 시스템 등 다양한 곳에서 이용되고 있다.[1]

원격제어 소프트웨어란 뷰어가 제어하는 측, 서버가 제어 당하는 측이라고 정의하였을 때, 뷰어와 서버 사이의 통신 시, 서버의 화면을 공

유하거나, 서버를 대상으로 원격제어 기능을 제공하는 소프트웨어를 말한다.

대표적인 소프트웨어로 T제품, M제품, G제품 등 원격 데스크톱이 사용되고 있으며, 이를 위해 프로그램 별로 상이한 프로토콜과 암호화를 제공한다. 본 논문에서는 원격제어 소프트웨어에서 사용하는 프로토콜을 살펴보고 원격제어 소프트웨어의 공격 기술과 시나리오를 도출하고 이로 인한 파급력과 원격제어 소프트웨어를 보호하기 위한 대응 방안을 제시한다.

II. 관련 연구

원격제어 소프트웨어와 관련한 대표적인 프로토콜을 알아보고 기술에 대해서 살펴본다.

2.1 RDP 프로토콜

RDP(Remote Desktop Protocol) 프로토콜은 Windows를 포함하는 Microsoft OS에서 제공하는 원격 데스크톱 프로토콜을 이용하여 접속하는 방식이다. Windows 운영체제에서 원격제어 기능을 자체적으로 제공하기 때문에 PC 상에 프로그램을 원격제어 소프트웨어를 설치할 필요가 없다. 또한 자체적으로 종단 간에 TLS(Transport Layer Security)를 적용하여 전송구간에 대한 암호화를 지원한다.[2]

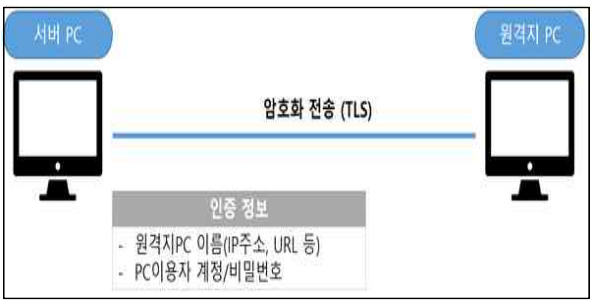


그림 1 RDP 접속방식

2.2 RFB 프로토콜

RFB(Remote Frame Buffer) 프로토콜은 VNC(Virtual Network Computing) 프로그램에서 사용하는 프로토콜이다. RFB는 프레임 버퍼 수준에서 동작하고 GUI 기능을 이용하여 원격 시스템에 접근할 수 있도록 해주는 구조이다.[3]

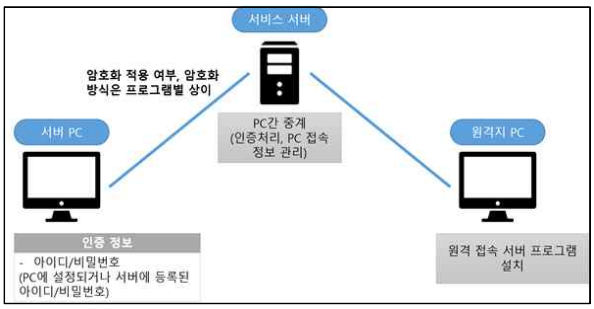


그림 2 RFB 접속방식

2.3 프로토콜 분류 및 특징

RDP 방식은 외부 네트워크에서 접속 시 공인 IP나 URL이 필요하지만 RFB 방식은 PC 정보(IP, 포트번호) 없이 중계서버에서 아이디와 비밀번호만으로 접속하도록 연결해주기 때문에 다수의 이용자들이 선호하고 있다. 다음 아래 표 1은 RDP, RFB의 접속방식 및 암호화, 특징이다.

분류		접속방식	암호화	특징
RDP		Windows RDP	TLS	RDP에서 제공하지 않는 기능 외부 네트워크에서 접속 시 공인 IP 필요
RFB	직접 접속	VNC 자체제작	프로그램별상이	프레젠테이션 출력 등 다양한 기능 제공
	서버 경유			서비스 서버를 경유해서 PC 접속(원격 PC의 IP주소 없이도 접속 가능)

표 1 원격제어 소프트웨어 프로토콜 분류 및 특징

III. 위협분석

PC의 접근 권한을 넘겨주는 원격제어 소프트웨어의 특성상 발생할 수 있는 권한 상승 및 탈취 취약점을 분석하고 이를 통해 원격제어 소프트웨어의 공격 시나리오를 도출한다.

3.1 RCE(Remote code execution) 취약점

원격제어 소프트웨어의 취약점이 존재할 경우 공격자 PC가 피해자 PC에 취약 코드를 지닌 프로그램에 대해서 원격 명령 실행 공격 코드를 삽입하여 전송한다. 공격자 PC가 피해자 PC의 권한을 획득한다.

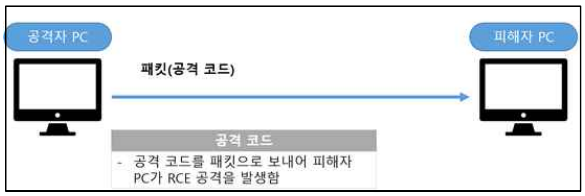


그림 3 RCE 공격

3.2 LPE(Local privilege escalation) 취약점

원격제어 소프트웨어의 취약점이 존재하고 접근이 가능할 경우 시스템에서 일반 사용자의 권한을 가진 상태에서 관리자의 권한으로 상승시키는 취약점이다.

3.3 취약한 소스 코드

원격제어 소프트웨어에서 취약한 소스 코드를 사용할 경우 취약점이 발생하여 공격 시나리오에 이용될 수 있다. 다음 그림 4는 취약한 소스 코드이다.

```
void main(){
        char buf[256];
        int sockfd;

        ...

        read_string(sockfd, buf);

        ...

}

void read_string(int socketfd, char *buf){
        int size;

        ...

        recv(sockfd, buf, 4, 0);
        size = atoi(buf);

        if(size > 256){         // INTEGER UNDERFLOW!!
                printf("OVERFLOW!!");
                return;
        }

        recv(sockfd, buf, size, 0); // BUFFER OVERFLOW

        ...

        return;

}
```

그림 4 취약한 소스 코드

위 소스에서 size를 입력받고 조건문을 이용하여 string의 크기를 제한하는데, size에 음수가 들어갈 경우에 조건문을 우회할 수 있다.

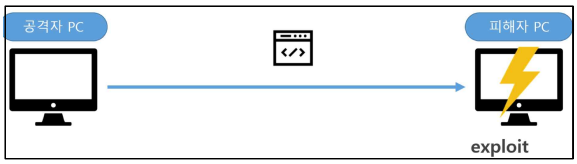
아래 recv 함수에서 0xffffffff 만큼 입력받게 되어서 원래 버퍼보다 더 큰 사이즈의 값을 넣어 buffer overflow를 발생시킬 수 있다.

IV. 공격 시나리오

위의 언급한 취약점을 이용하여 나오는 공격 시나리오는 다음과 같다.

4.1 플레이어 공격 시나리오

원격제어 소프트웨어 중 일부는 녹화 기능을 제공하지만 녹화된 파일은 자체 동영상 플레이어를 가지고만 재생할 수 있다.



4.2 상담원 인증 우회 공격 시나리오

원격제어 소프트웨어의 인증 우회 취약점을 가지고 있을 경우 인증 우회를 통해서 공격자가 인증을 우회하여 상담원의 권한을 획득할 수 있고 상담원의 권한으로 피해자의 PC에 랜섬웨어 등 다양한 악성코드를 실행시켜 피해를 입힐 수 있다.

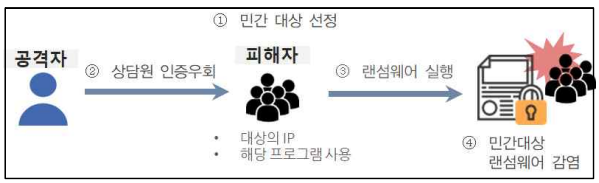


그림 6 상담원 인증 우회 공격 시나리오

4.3 국방 공격 시나리오

PMS(Patch Management System)란 네트워크상에 존재하는 모든 서버와 사용자 PC의 운영체제 및 애플리케이션에 대한 업데이트 관리자가 일괄적으로 관리 통제하는 보조 소프트웨

어이다. PMS Server의 PMS Solution으로 사용하는 백신 프로그램에는 해당 원격 관리 기능을 제공한다.[4] 이를 이용해 원격제어 모듈을 탑재 한 백신 프로그램의 취약점을 이용한 공격이 가능하다.

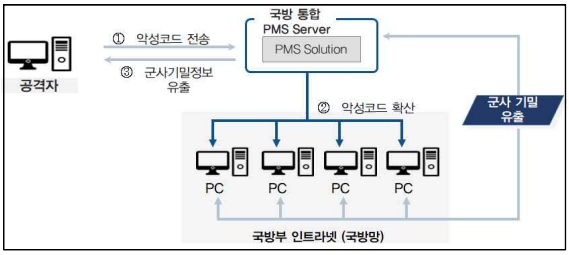


그림 7 국방 공격 시나리오

V. 대응방안

사용자의 입력이 프로그래머의 의도대로 입력받을 수 있도록 검증하는 루틴이 필요하며, 고정된 버퍼에 입력을 받기보단 동적 할당을 통한 버퍼에 입력을 받아서 stack에서 발생할 수 있는 치명적인 취약점을 예방해야 한다. 또한, 프로그램 내에서 외부 라이브러리나 프로그램을 사용하는 경우 취약한 버전을 사용할 시 알려진 취약점으로 공격이 가능할 수 있기 때문에 프로그램의 이슈사항을 주기적으로 확인하여 업데이트하는 것이 매우 중요하다.

본 논문에서는 여러 가지 원격제어 소프트웨어 취약점과 공격 시나리오에 대해서 알아봤다. 원격제어 소프트웨어가 취약할 경우 백도어를 자기 PC에 설치하는 것이랑 동일시되기 때문에 원격제어 소프트웨어의 지속적인 연구와 효과적인 대응 방안 마련이 시급하다.

이 논문은 차세대 보안리더 양성 프로그램(KITRI)의 지원을 받아 작성하였습니다.

[참고문헌]

[1] 이 철, 김상철 김중환, 원격 제어 프로그램의 설계 및 구현, 2001.

[2] Microsoft. Remote Desktop Protocol(RDP) Features and Performance. 2009

[3] Tristan Richardson, The RFB Protocol, 2009.

[4] 산업자원부. Remote Control 기술을 이용한 유,무선 PMS(PDA Managerment System) 개발

2017 한국정보보호학회 동계학술대회

D4 디지털 포렌식

김 종 성 (국민대)

- **사이버 범죄 예방을 위한 사이버 CPTED 모델**
  주민희, 박무규, 이경호 (고려대학교)
- **Volume Shadow Copy의 포렌식적 활용**
  박은후, 이세영, 이종혁, 김청우, 이상범, 최형석, 김영효, 최건희, 김한기, 김종성 (국민대학교)
- **웹 캐시의 위치 기반 서비스 정보를 활용한 위치추적 방안 연구**
  김채린 (고려대학교), 송인석 (세종대학교)
- **IoT 디바이스의 네트워크 정보 이력관리를 통한 사고분석 활용 방안**
  장대일, 김태은, 김환국 (한국인터넷진흥원)

사이버 범죄 예방을 위한 사이버 CPTED 모델

주민희*, 박무규*, 이경호*

*고려대학교 정보보호학과

Cyber-CPTED Model for Cyber Crime Prevention

Minhee Joo*, Mookyu Park*, and Kyungho Lee*

*Institute of Cyber Security & Privacy (ICSP), Korea University.

요 약

최근 사이버범죄의 증가로 인해 사이버공간뿐만 아니라 현실세계에 대한 피해도 증가하고 있다. 이러한 사이버범죄의 경우, 일반범죄가 가지는 지리학적 특징 또는 범죄자의 심리적 특징 확인이 어렵다는 점에서 예측이 어렵다. 하지만, 사이버범죄자는 범죄 대상에 대한 공격효과가 크고 저비용일 때 공격을 선택하게 된다. 즉, 사이버범죄의 대상이 되는 목표물이 사이버공격에 대한 저항 값을 크게 가지는 경우, 사이버범죄에 대한 예방이 가능하다. 본 논문은 건축학에서 많이 사용되는 범죄예방 환경설계(CPTED: Crime Prevention through Environmental Design)의 개념과 사이버공간 계층에 대한 개념을 결합하여 사이버공간에서 사이버범죄를 예방할 수 있는 모델인 사이버 CPTED을 제안한다.

I. 서론

과거의 사이버공격은 소프트웨어, 하드웨어 파괴, 네트워크 마비등과 같은 사이버공간에 제한되어 발생하였다. 하지만, 최근에는 직접적으로 현실 세계에 피해를 주는 방향으로 발전하고 있다. 대표적인 사례는 금품갈취가 목적인 WannaCry와 같은 랜섬웨어다. 또한, 사이버공간이 소셜네트워크와 IoT(Internet of Things)와 같은 real world와 가까워지면서 사이버 공격은 사회적 혼란을 야기하는 원인이 되고 있다.

이러한 사이버 공격은 국가적 이벤트 기간일 때, 빈번하게 발생한다. 2016년 미국 대통령선거에서 Hillary Clinton 후보에 대한 러시아의 사이버 공격, 프랑스의 대통령 선거기간 중 period of election silence 에 발생한 fake news 등이 대표적이다. 이러한 공격들은 정보유출, 거짓정보 생산들 인간의 합리적 의사결정을 방해하는 요소를 기반으로 한다. 특히, 이러한 공격으로 통해 발생하는 정보의 출처가 보안등급이 높은 정부기관의 데이터베이스의 경우, 이에 대한 정치적, 사회적 그리고 경제적 피해는 크다. 2017년 열린 RSA conference에서 미국 국토안보부 분과 위원장인 Michael McCaul 하원의원이 미국 정부는 21세기 위협에 20세기 기술과 19세기 제도를 가지고 대응하고 있다고 비판하였다.

경찰청 통계에 따르면 2016년의 사이버범죄는 2014년도에 비해 약 40000건 이상 증가하는 추세를 보였다. 또한, 현재의 정책적 한계로 인해 범죄 예방보다는 범죄 발생 후 검거에 집중하고 있다. 하지만, 현실 세계에서 일반범죄를 예방하기 위해서 활용하는 방안이 범죄 예방 환경 설계(CPTED, Crime Prevention Through Environmental Design)이다. CPTED는 건축 환

경 설계를 통하여 범죄를 예방하는 범죄 대응 방안이다. 본 논문은 이러한 CPTED의 개념을 활용하여 사이버 범죄 및 사이버 공격을 예방 가능한 사이버범죄 예방 환경 모델을 제안한다.

Ⅱ. 배경 및 관련연구

본 연구는 CPTED를 사이버공간에 적용하기 위해 사이버공간 계층을 활용한다. 이 단락에서는 사이버공간 계층에 대한 개념과 최근에 연구된 CPTED에 대하여 설명한다.

A. 사이버공간 계층

사이버공간은 인터넷, 통신, 네트워크, 임베디드 프로세스 등을 포함하는 상호 의존적인 환경으로 구성되어 있다. Joint Publication 3-12R Cyber Operation에 따르면, 사이버공간은 물리적 계층(physical layer), 논리적 계층(logical layer) 그리고 페르소나 계층(persona layer) 총 3가지 계층들로 구성된다. 물리적 계층은 지리학적 요소와 물리적 네트워크요소로 구성되는 계층이다. 논리적 계층은 물리적 계층에서 설정된 노드 간 논리적 연결을 통하여 네트워크로 구성된 계층이다. 페르소나 계층은 인간과 인지 측면을 포함하는 계층으로 전자메일 주소, IP 주소 등의 개인 식별정보가 포함된다[1].

David Clark은 사이버공간을 물리적 계층, 논리적 계층, 정보 계층(Information Layer) 그리고 최상위-사람 계층(Top-people layer) 4가지 계층으로 설명하였다. Logical Layer와 Physical Layer는 JP 3-12R의 Layer들과 그 의미와 역할은 같다. 하지만, JP 3-12R에서 하나로 구성하였던 페르소나 계층은 정보 계층과 최상위-사람 계층으로 구분되었다. 정보 계층은 사이버공간에서 저장, 전송 및 변환되는 정보가 존재하는 계층이다. 최상위-사람 계층은 사이버 경험에 참여하는 사람들로 구성되며, 서비스 및 기능을 사용하여 의사소통을 하고 정보를 사용하여 의사결정을 내리고 계획을 수행하는 계층이다[2][3].

2011년 미국 국토안보부에서 발표한 보고서 (Enabling Distributed Security in Cyberspace)에 따르면, 사이버공간 계층을 사이버 에코시스템(cyber ecosystem)으로 확장하여 설명하고 있다. 사이버 에코시스템은 이 시스템은 사람, 프로세스, 기술의 복잡한 상호 작용을 해결하기 위해, 하드웨어, 케이블, 사람, 건물, 모든 물리적, 실제 항목에 연결되어 있는 15개의 계층으로 구성되어 있다[4].

본 연구는 사이버공간의 물리적 계층, 논리적 계층 그리고 페르소나 계층과 사이버 에코시스템을 결합하여 사이버 CPTED에 대한 모델에 대하여 설명한다.



〔그림 1〕 사이버공간 생태계

B. CPTED(Crime Prevention Through Environmental Design)

CPTED는 주요 선진국을 중심으로 보편화되고 있는 기법으로 여러 학문간 연계를 통해 도시 및 건축 공간 설계 시 범죄기회를 제거하거나 최소화하는 방향으로 계획, 변경함으로써 범죄 및 불안감을 저하시키는 원리이자 실천전략이다. 미국, 영국, 호주, 네덜란드, 일본 등에서는 관련분야의 역사가 오래되었고 도시, 건축뿐 아니라 정보화시대에 발맞춰 컴퓨터 분야 등에서 많이 사용하고 있다. CPTED의 원리는 접근통제와 감시를 중점으로 하고 있다.[1]

경찰청 생활안전국은 발생한 범죄에 대한 사후 대응과 경찰 등 형사사법기관을 중심으로 통치적인 전통적인 접근 보다는 범죄위험요소에 대한 사전적 대응과 환경에 중심이 되는 협

치적인 중심을 강조하는 CPTED의 정책을 연구하였다.[2] CPTED를 고려할 때 주요원리는 자연감시, 접근통제, 영역성 강화, 활동의 활성화, 유지관리가 있다. 자연감시는 주변을 잘 볼 수 있고 은폐장소를 최소화 시킨 설계이다. 접근통제는 외부인과 부적절한 사람의 출입을 통제하는 설계이다. 영역성 강화는 공간의 책임의식과 준법의식을 강화 시키는 설계이다. 활동의 활성화는 자연 감시와 연계된 다양한 활동을 유도하는 설계이다. 마지막으로는 유지관리이다. 지속적으로 안전한 환경 유지를 위한 계획이다.

Carri Casteel은 위와 같은 주요원리를 고려하여 CPTED 접근이 광범위 하여 모든 설정 및 결과에 적응할 수 있다고 설명하였다.[3] CPTED는 범죄의 예방의 강도를 줄이기 위한 효과적인 접근 방법이라고 하였다. 결론으로 CPTED가 다양한 분야의 환경에서 개별 구성요소와 효과에 대한 연구가 필요하다고 하였다. 사이버범죄의 피해는 사람의 자산이나 정보를 훔쳐갈 수 있다.

사이버공간에서의 사이버범죄가 증가하고 있는 추세에서 사이버범죄를 예방하는 모델 구축이 필요하다. 이미 많은 연구에서 CPTED에 대한 범죄예방 효과를 입증해왔다.[4] 본 연구에서는 사이버공간과 유사한 물리적인 환경의 설계를 빗대어 사이버공간 범죄 예방 모델을 구축한다.



〔그림 2〕 CPTED 기본원리

Ⅲ. 사이버범죄 예방 모델

CPTED의 효과를 극대화시키기 위해 도시공간의 환경적 패턴이나 분석을 통해 환경요인들을 조작함으로써 범죄를 예방하는 대안이 필요하다. 대부분의 범죄는 일정한 시공간적 발생패턴과 피해대상의 반복성을 내포하고 있기 때문에, 이를 근거로 범죄기회를 제공하는 상황적 요인들을 사전에 통제하거나 제거함으로써 범죄를 예방하는 것이 CPTED의 핵심이다. 사이버범죄는 익명성이다. 자신을 밝히지 않는 중요한 특징이 있다. 사이버공간에서의 범죄의 예방방법은 IP주소나 기존의 공격을 했던 방법을 사용해야한다. 사이버공간은 실제 세계보다 가상공간이다. 현실의 모든 규제로 부터 자유롭다는 인식이 사람들에게 가지고 있어서 범죄를 일으키기 쉽다. 이와 같이 CPTED도 보이지 않는 환경공간에서 익명으로 사람들에게 범죄를 하는 것이기 때문에, 사이버 공간과 똑같다. 사이버공간에서도 공격할 틈이 없는 공간을 통제하고 한번 들어온 공격패턴이나 비슷한 공격들은 예방해야 한다. 사이버범죄와 현실세계의 범죄를 비교하여 현실세계 환경을 고려하여 CPTED의 주요원리들과 사이버 생태계에서의 주요역할들을 보아 사이버 범죄예방 모델을 제안하였다.

CPTED의 주요원리들을 고려하여 사이버 에코시스템에 맞춰 예방 모델을 제안하였다. CPTED의 다섯 가지의 원리인 감시, 통제, 강화, 분석, 예방을 사이버 생태계에 접목시켰다. 사이버공간에서의 감시 계층이 존재한다. 이 계층에서는 사람이 계속해서 자신의 정보가 범죄에 노출되었는지 확인하는 단계이다. 사이버공간을 전체로 보았을 때, 공격을 할 수 있는 공간을 최소화하는 단계이다. 이 단계에서는 매 시간 감시를 하고 언제 어느 곳에 악성코드가 들어와서 공격에 대한 피해를 최소화하기 위해 사이버범죄 감시단계가 필요하다. 네트워크의 접근과 데이터의 송수신을 관리하는 논리계층이 사이버공격을 통제한다. 미리 있던 사이버공격에 대한 데이터들을 저장해 유사한 사이버공격이 발생할 경우 통제한다. 많은 컴퓨터 바이

러스들이 데이터들을 통해 들어온다. 그 데이터들을 분석하고 전송하는 레이어에서 공격이 의심되는 파일이 있다면 이 레이어에서 통제한다. 사이버공간 강화단계에서는 컴퓨터의 프로그램을 설치할 때, 각각의 윈도우나 어플리케이션들은 각각의 방화벽이나, 보안 프로그램들을 설치한다. 또한, 각 프로그램에 취약한 취약점들이 다 존재하여 그 것에 대한 보안점이나 공격들을 막고 있으므로 사이버 공간을 강화할 수 있다. 사이버공간에서의 분석은 컴퓨터의 언어로 해야 한다. 그러하여 컴퓨터의 언어인 기계언어로 많은 데이터들이 기계언어로 소프트웨어에 전달된다. 이러한 언어로 사이버범죄에 관련된 공격들을 바탕으로 더 다양한 사이버 범죄에 대한 기계언어 분석이 가능하다. 최종적으로 사이버범죄를 예방해야 하는 것이 목표이기 때문에 사이버범죄를 예방하기 위해 가장 필요한 단계로 앞에서 말했던 단계들을 바탕으로 사이버공간에서 범죄에 대한 예방하기 위해서 계속해서 사이버공간의 환경을 조성한다.

[표 1] 사이버범죄 예방 모델

사이버범죄 감시	공격을 할 수 있는 공간을 최소화하는 단계
사이버공격 통제	데이터, 바이러스로부터 공격 통제하는 단계
사이버공격 강화	프로그램의 방화벽이나 설치과정에서 각 프로그램에 대한 방어 단계
사이버범죄 분석	기계언어로 사이버범죄 분석 단계
사이버범죄 예방	사이버공간의 안전한 환경 조성 단계

Ⅴ. 결론

사이버공간에서의 사이버범죄가 증가하는 요즘 사이버범죄가 일어난 후에 잡는 경우가 대부분이다. 이것을 방지하고자 사이버공간의 범죄를 사전에 예방하고자 모델을 구축하였다. 건축학이나 도시 계획 분야에서 이미 시행하여 실제 사례로 범죄가 예방된 CPTED를 바탕으로 사이버공간의 물리적 계층, 논리적 계층, 정보 계층으로 나누어 모델을 구축하였다. 미국에서는 CPTED를 IoT(Internet of Things)에 접목하여 사이버공간에서의 범죄를 예방하고 있다. 우리나라는 다른 나라에 비해 사이버범죄가 나날이 늘어나고 있다. 우리나라도 미국과 같이 사이버공간에서 표1과 같은 모델로 범죄를 예방해야한다.

[참고문헌]

[1] OPERATIONS, Cyberspace. Joint Publication 3-12 (R). Joint Chief of Staffs, 2013.

[2] CLARK, David. Characterizing cyberspace: past, present and future. MIT CSAIL, Version, 2010, 1: 2016-2028.

[3] BARFORD, Paul, et al. Cyber SA: Situational Awareness for Cyber Defense. Cyber Situational Awareness, 2010, 46.1: 3-13.

[4] PHILIP, R., et al. Enabling Distributed Security in Cyberspace. Departament of Homeland Security, 2011.

[5] Donghyun Kim, Songmi Park, Improving community street lighting using CPTED: A case study of three communities in Korea, In Sustainable Cities and Society, Volume 28, 2017, Pages 233-241

[6] Park. Y oung Ju. (2010). A study on the effects of the Crime prevention through environmental design (CPTED). Korean Police Studies Review, 9(1), 55-82.

[7] Carri Casteel, Corinne Peek-Asa, Effectiveness of crime prevention through environmental design (CPTED) in reducing robberies, In American Journal of Preventive Medicine, Volume 18, Issue 4, Supplement 1, 2000, Pages 99-115

[8] Yu, Kwang-Heum, Cho, Young-Jin, Ji, Hye-Ran & Son, Dongpil. (2017). An Analysis of the Effectiveness of Crime Prevention Through Environmental Design Project, JOURNAL OF THE ARCHITECTURAL INSTITUTE OF KOREA Planning & Design, 33(5), 13-20

Volume Shadow Copy의 포렌식적 활용

박은후*, 이세영*, 이종혁*, 김청우*, 이상범*,
최형석*, 김영효*, 최건희*, 김한기**, 김종성*,**
*국민대학교 정보보안암호수학과, **국민대학교 금융정보보안학과

Forensic Application of Volume Shadow Copy

Eun-hu Park*, Se-young Lee*, Jong-hyeok Lee*, Chung-woo Kim*,
Sang-beom Lee*, Hyeong-seok choi*, yeong-hyo kim*,
Keon-hee choi*, Hangi Kim**, Jongsung Kim*,**
*Dept. of Information Security, Cryptology and Mathmathics, Kookmin University.
**Dept. of Financial Information Security, Kookmin University.

요 약

볼륨 섀도 복사는 Windows XP 이후 Windows에서 제공하는 백업 서비스이다. Windows는 일정 주기마다 차등 백업 방식으로 복원지점을 자동 생성하며, 이는 사용자가 수동으로 생성할 수도 있다. 악의적으로 사용한 파일을 삭제하기 전 복원지점이 생성된다면, 복원지점으로 돌아가 파일을 복원하는 방식으로 볼륨 섀도 복사를 포렌식적으로 활용할 수 있다. 반면, 이전 복원지점으로 컴퓨터 상태를 되돌려 증거인멸의 수단으로도 사용될 수 있다. 본 논문은 볼륨 섀도 복사의 포렌식적 의미, 활용, 악용가능성을 살펴보고, 그 개선 방향에 대해서 기술한다.

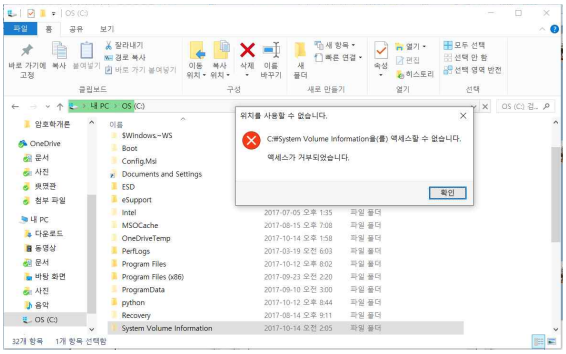
I. 서론

Windows Vista 이전에 사용되던 섀도 복사는 파일 단위로의 시스템 파일 백업은 가능하지만 사용자가 생성한 파일은 백업할 수 없었다. Windows Vista 이후, 볼륨 단위의 백업 방식으로 바뀐 볼륨 섀도 복사는 사용자 파일을 비롯한 볼륨의 모든 파일을 백업할 수 있게 되었다. 또한 볼륨 섀도 복사는 Windows 자체에서 제공하는 서비스이기 때문에 다른 백업 방식과는 다르게 따로 설치하지 않아도 활성화 되는 경우가 많다[1]. 백업 정보들은 복원이 가능하며, 그 중 일부는 '%SYSTEMROOT%\system32\Restore\filelist.xml' 파일에 남아있다. 이 파일은 모니터링 할 목록을 설정하는 파일로, 생성, 변조, 삭제하는 이벤트가 발생할 때마다 이벤트 내역, 경로, 파일의 복사본 등 포렌식적으로 중요한 정보가 백업된다[2].

Windows는 자동 혹은 수동으로 볼륨 섀도 복사 백업을 하는데, 자동으로 백업되는 경우 사용자도 알아채지 못할 수 있다. 특정파일이나 폴더 또는 시스템의 변경내용을 백업하기 때문에, 시스템 변경 이전의 데이터를 제공할 수 있다. 또한 삭제된 데이터를 찾아내 복구할 수 있으며, 이를 이용해 특정 데이터를 숨기거나 삭제 했는지에 대한 여부를 파악할 수 있다[3].

'%HOMEDRIVE%\System Volume Information'은 백업 정보를 담고 있으며 [그림 1]과 같이 접근권한 없이 접근할 수 없다. 복원지점을 생성하면 기본적으로 파일들이 저장되는 경로인 '%HOMEDRIVE%\System Volume Information_restore{GUID}\RP#'[1] 내에 있는 파일들은 권한을 가지고 있더라도 변조하거나 삭제하기 어렵다. 파일 내 정보는 가시적이지 않은 형식으로 존재하기 때문에, 전문가가 아니면 그 파일만을 보고 그 파일이 어떤 정보를 담고 있는지 알기는 어렵다. 접근이 어렵고 볼륨 섀도 복사본에 관련된 파일을 변조할 수 없다는 것은 다른 의미로는 정보의 무결성이 어느 정도 보장된다는 것을 의미하며 포렌식적 증거로써 의미를 가진다.

1) #은 복원지점 생성 번호



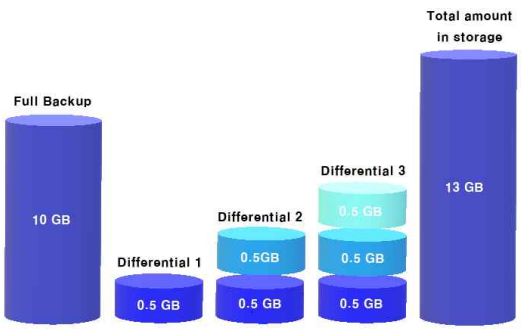
[그림 1] 권한이 없이는 접근 불가능한 System Volume Information 폴더

본 논문의 2장에서는 볼륨 섀도 복사로 백업이 되는 데이터에 대해 살펴보며, 3장에서는 백업된 데이터가 포렌식적으로 활용되거나 악용될 수 있는 상황을 가정하여 실습한다. 마지막으로 4장에서 결론과 함께 볼륨 섀도 복사의 개선 방향을 제시한다.

II. 백업되는 정보

볼륨 섀도 복사의 백업은 전체 백업을 한 뒤, 차등 백업을 하는 방식으로 이루어진다. 이 때 차등 백업이란, [그림 2]과 같이 전체 백업을 한 시점을 기준으로 백업하려는 시점까지 수정 혹은 추가된 데이터만을 백업하는 방법이다[4].

볼륨 섀도 복사는 드라이버를 설치할 때와 같이 컴퓨터에서 특정 변경을 감지할 때나, 지정된 주기마다 이전 백업 시점에서 변경된 내용을 복원지점 폴더인 '%HOMEDRIVE%\System Volume Information' 에 차등 백업한다. 기본적으로 Windows는 복원에 가장 필요로 하는 정보만을 백업하고 있으며, 백업을 하는 정보로는 레지스트리, 사용자 프로파일, COM+ DB, WFP.dll 캐시, WMI DB, IIS Metabase, file.xml에 <Include>등이 있다[2].



[그림 2] 차등 백업 이미지

III. 포렌식에서의 활용 및 한계

본 장에서는 볼륨 섀도 복사가 포렌식적으로 활용될 수 있는 시나리오와 악의적으로 사용될 수 있는 시나리오를 보인다.

3.1 복원지점 생성 시간의 활용

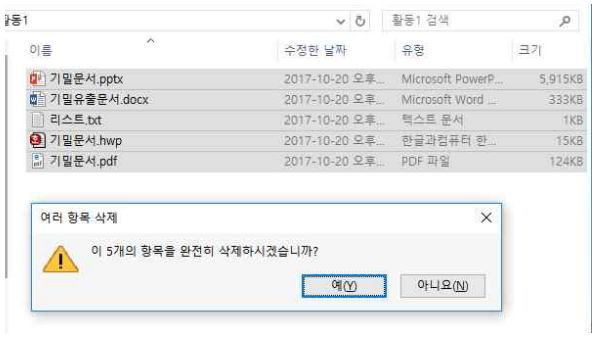
볼륨 섀도 복사는 Windows에서 기본적으로 활성화 되어 있는 기능으로, 고의적으로 설정을 바꾸지 않는 이상 시스템 복원지점이 자동적으로 생성된다. 다음은 악의적인 비밀문서를 생성한 뒤 영구 삭제했지만, 삭제 전에 볼륨 섀도 복사본이 생성된 사례를 가정한 것이다.

2017년 10월 20일 이모씨는 A회사의 비밀문서를 유출하기 위해 한글, 워드, 데이터베이스, 텍스트문서, 엑셀, 파워포인트 등 다양한 파일의 비밀문서들을 작성한다. 이 후 영구삭제를 통해 자신의 행위를 은폐한다.

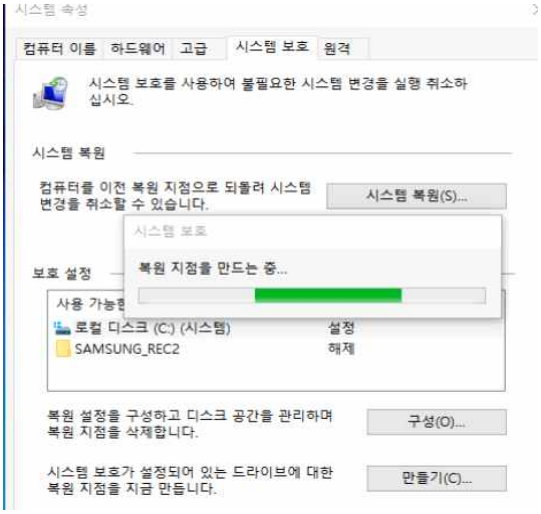
위의 시나리오를 재현하기 위해 [그림 3]와 같이 2017년 10월 20일 오후 3시 50분에 여러 가지 형태의 비밀문서를 생성했다. 이모씨는 자신의 악의적 행위가 발각될 것을 염려하여 4시에 문서에 대한 완전삭제를 [그림 4]와 같이 시행하였지만, 3시 55분에 자동으로 복원지점이 생성되었다고 가정하자 ([그림 5]). 이 때, 실험에서는 수동으로 복원지점을 생성하였지만, 시나리오에서는 Windows가 자동으로 복원지점을 생성하여 사용자는 알아채지 못했다고 가정한다.



[그림 3] 현존하는 데이터 확인



[그림 4] 생성된 파일 완전삭제



[그림 5] 복원지점 생성

파일의 복원지점을 선택하여 백업된 지점을 보기 위해 Shadow Explorer을 사용했다. Shadow Explorer은 생성된 볼륨 새도 복사본을 이용하여 정상적으로 사용했던 날짜를 선택하여 그 지점의 파일들을 볼 수 있는 도구다. [그림 4]에서 완전 삭제된 비밀문서들을 [그림 6]과 같이 Shadow Explorer에서 확인가능하며, 파일을 추출, 복구할 수 있다.



[그림 6] 비밀문서 복원 가능 확인

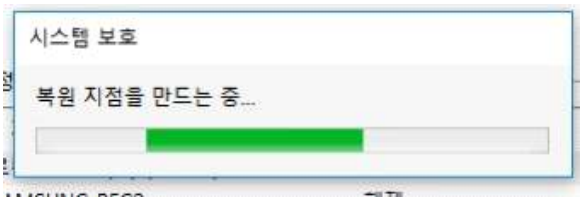
이와 같이 파일 생성시점과 삭제시점 사이에 볼륨 새도 복사본이 생성되는 경우, 포렌식 수사관은 툴을 사용하여 파일들을 복원할 수 있다. 그리고 복원한 데이터를 범죄의 증거로 확보할 수 있다.

3.2 복원지점의 악용

볼륨 새도 복사를 악용한다면 증거인멸과 같은 문제점이 있을 수 있다. 다음은 범죄자가 볼륨 새도 특징을 알고 있고, 이를 이용하여 증거인멸의 방법으로 사용하려고 한 사례를 가정한 것이다.

2017년 10월 20일 XX회사의 개인정보를 해킹한 김모씨는 약 10만명의 개인정보를 각종 비밀문서를 저장했다. 하지만 이러한 행각이 발각될 것을 대비하여, 범죄자는 경찰이 출두하기 전 비밀문서를 생성하기 이전의 복구지점으로 돌아가 경찰 수사망에서 벗어나려고 했다.

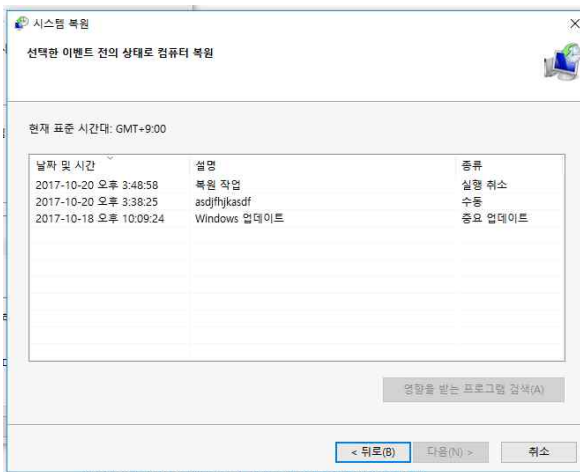
위의 시나리오를 재현하기 위해 선택하여 돌아갈 수 복원지점이 존재하도록 [그림 7]과 같이 수동으로 복원지점을 생성했다. 사건에서 범죄자가 저장한 비밀문서를 [그림 8]과 같이 엑셀파일과 텍스트 파일로 작성하여 C드라이브에 저장했다. 그리고 [그림 9]과 [그림 10]와 같이 증거인멸을 위해 생성해둔 복구지점을 선택하여 컴퓨터를 복구해보았다.



[그림 7] 복원지점 생성



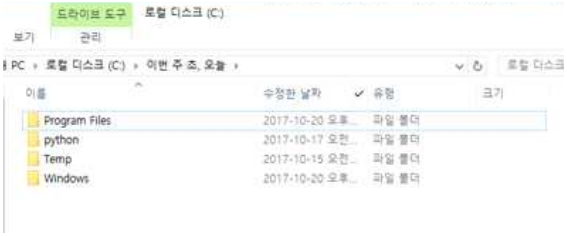
[그림 8] C 드라이브에 존재하는 비밀문서



[그림 9] 복원지점 선택



[그림 10] 복구지점으로 복구



[그림 11] C드라이브에 존재하지 않는 비밀문서

그 결과, 저장되어있던 비밀문서는 존재하지 않음을 [그림 11]과 같이 확인했다. 위의 시나리오를 통해, 볼륨 섀도 복사는 복구지점을 생성하여 그 시점으로 복구만 될 뿐 복구되기 이전 지점에 대한 정보는 존재하지 않음을 확인했다. 따라서 볼륨 섀도 복사의 존재를 알고 범행 이전의 시점으로 컴퓨터를 복구하는 경우, 증거인멸의 위험성이 있을 수 있다. 컴퓨터를 포맷하는 방법은 증거를 인멸한 의심을 쉽게 받을 수 있는데 비해, 백업파일로부터 복원을 한다면 정상적으로 사용하던 컴퓨터로 오인할 가능성이 생기기 때문에 충분히 악용이 가능할 것으로 보인다.

위의 상황을 방지하기 위해서는 복구기능을 사용하는 시점에 대한 차등 백업 또한 저장하는 기능이 필요하다. 하지만 이 방법은 불필요한 저장 공간을 차지하므로 일정기간이 지나면 자동 삭제되어야 한다. 저장 공간의 부족으로 모든 정보를 저장하기 어렵다면 복구지점으로 복원한 흔적만이라도 쉽게 알아낼 수 있어야 할 것이다.

IV. 결론

볼륨 섀도 복사는 Windows에서 기본적으로 제공하는 백업 방법이다. Windows 설치 관리자에 의해 프로그램이 제거되거나 자동 업데이트 될 때와 같이 시스템의 변화가 생길 때, 혹은 Windows에서 기본적으로 설정된 주기적인 시기에 자동으로 차등 백업을 시행한다.

볼륨 섀도 복사본의 백업 정보를 담고 있는 '%HOMEDRIVE%\System Volume Information' 파일은 백업 정보를 담고 있으며, 접근권한이 없으면 접근할 수 없다. 접근권한을 가지고 있더라도 파일을 가공하거나 삭제하기는 어렵다. 백업된 정보들은 가시적인 형태로 존재하지 않기 때문에 파일 내에서 원하는 정보를 알아내고 변조하는 것 또한 어렵다. 따라서 볼륨 섀도 복사는 정보의 무결성을 보장되기 때문에 포렌식적으로 의미를 가진다.

볼륨 섀도 복사본 생성이 자동으로 이루어지는 것은, 다른 의미로는 사용자의 행위가 의도치 않게 저장될 수 있다는 것을 의미한다. 복원지점이 생성되면 '%HOMEDRIVE%\System Volume Information_restore{GUID}\RP#' 폴더 안에 백업과 관련 있는 정보들이 각종 파일형태로 생성된다. 단순 삭제한 파일은 복사본으로 백업되고, 휴지통을 거치지 않은 영구 삭제한 파일은 파일 복사본으로 백업되지는 않지만 삭제한 기록이 남는다. 때문에 그 증거는 볼륨 섀도 복사본을 복원하여 확보할 수 있다.

필요한 정보가 있는 시점에 복원지점이 생성되는 경우, 그 지점으로 복원시켜 증거를 확보할 수 있다. 하지만 동시에 복원지점을 악용하여 컴퓨터 사용자가 증거가 없는 복원지점으로 돌아가는 경우 증거를 인멸할 가능성도 있다. 이를 방지하기 위해서는 볼륨 섀도 복사는 복원에 대한 로그를 자세히 기록해야하며, 저장 공간이 충분한 경우 복원이전지점에 대한 차등백업을 시행할 필요가 있다.

[참고문헌]

[1] 섀도 복사본, "https://ko.wikipedia.org/wiki/%EC%84%80%EB%8F%84_%EB%B3%B5%EC%82%AC%EB%B3%B8"

[2] 시스템 복원지점 포렌식 분석, "http://forensic-proof.com/archives/2854"

[3] 볼륨 섀도 카피 분석, "http://secgora.tistory.com/2"

[4] VSS 백업, "https://www.ibm.com/support/knowledgecenter/ko/SSGSG7_6.3.0/com.ibm.itsm.db.sql.doc/dps_con_bup_vss.html"

웹 캐시의 위치 기반 서비스 정보를 활용한 위치추적 방안 연구

김채린*, 송인석**
고려대학교 수학과, 세종대학교 컴퓨터공학과

A Study on Location Tracking Using Location-Based Service Information of Webcache

Chaerin Kim*, Inseok Song**
*Mathematics, Korea University., Computer Engineering, Sejong University.

요 약

최근 개인의 위치 정보를 활용한 다양한 서비스들이 웹 브라우저를 통해 제공되고 있으며, 태블릿 PC와 경량 노트북의 점유율이 높아지며 PC의 휴대성이 높아져 의미 있는 위치 정보가 남게 되었다. 또한 Windows 10 운영체제에서 제공하는 GPS 위성 신호 기반 위치 정보가 활용되면서 기존에 활용되던 네트워크 기반의 위치 정보보다 더 정확한 정보를 얻을 수 있다. 이에 본 논문에서는 Windows 10 기반 PC에 남는 웹 기반 LBS의 캐시 파일들과 위치 정보의 종류를 분석하고 있으며 이 정보는 위치 정보가 핵심이 되는 다양한 수사에 활용가능하다.

I. 서론

최근 서비스의 형태가 고객 맞춤형으로 진화하면서 이용자의 나이, 성별, 관심 분야와 같은 일반 개인 정보들 외 위치 정보의 활용도가 높아지고 있다. 위치 정보를 활용한 위치 기반 서비스(Location Based Service, 이하 LBS) 산업의 경우 국내 사업자가 약 980명으로 증가하고 있으며[1], 세계 시장은 2019년에 약 414.6억 달러에 이를 것으로 예상되고 있다.[2]

일반적으로 많이 사용되는 LBS는 웹 브라우저로 제공되며, 원활한 서비스 제공을 위해 캐시를 사용하고 있다. 따라서 캐시를 사용하는 데이터는 사용자의 PC에 남게 되고, 디지털 포렌식 관점에서 연쇄범죄나 조직범죄와 같이 위치 정보가 중요한 사건의 경우 용의자가 위치하였던 실제 위치, 검색한 위치 정보는 수사에 핵심적인 요소가 될 수 있다.

기존 PC에서 위치를 특정하는 기술인 Network Location 기술은 VPN을 통해 외국 IP로 우회한 경우 사용자의 위치 특정에 한계가 존재한다. 그러나 최근 태블릿 PC가 활성화 되면서 Windows 10 운영체제 이후 GPS 기능이 PC에 탑재되었고, 이 기능을 활용하면 더 정확한 디바이스의 위치 정보를 얻을 수 있게 되었다.

따라서 본 논문은 Windows 10 운영체제 이후 PC에서 수집할 수 있는 웹캐시에 남아 있는 위치 정보를 확인하고, 분석할 수 있는 방안을 제시한다. 본 논문의 2장에서는 위치 정보에 대한 개요와 분류, 웹캐시에 남는 위치 정보와 관련된 연구들을 살펴보고, 3장에서 다양한 시나리오에서의 위치 정보 분석 결과를 보여준다. 4장에서는 분석한 내용의 효율성 검증을 한 후, 5장에서 결론과 추후 연구 방향에 대해서 제시한다.

II. 관련 연구

2.1 위치 정보에 대한 개요

“위치정보의 보호 및 이용 등에 관한 법률”

에서 정의하고 있는 “위치 정보”란 이동성이 있는 물건 또는 개인이 특정한 시간에 존재하거나 존재하였던 장소에 관한 정보를 말한다. 뿐만 아니라 단독 데이터로 위치 정보를 특정할 수 없는 경우 다른 정보들과 결합하여 그 위치를 파악할 수 있는 정보도 포함한다. 위치 정보는 실시간 행적 추적, 수사 시 용의자의 범행 경로 및 주 활동지 파악 등에 활용될 수 있다는 점에 의거하여 디지털 포렌식적 활용 가능성이 있다.

위치 정보와 관련된 선행 연구로는 스마트폰 지도 애플리케이션 이용 시 아티팩트 및 위치 정보에 대한 분석[3], GPS 모듈을 사용하는 내비게이션 애플리케이션에 대한 연구[4]가 있다. 또한 추출한 위치 정보를 활용하여 경로 패턴을 학습하고 이를 통해 개인 경로 모델을 생성하는 연구[5]도 진행되었다. 그러나 이와 같은 선행연구는 대부분 모바일 디바이스에서의 위치 정보를 기반으로 하고 있다.

2.2 스마트폰 환경에서 위치 정보

스마트폰 환경에서 위치 정보는 측정 방식에 따라 네트워크 기반, 위성 신호 기반, Wi-Fi 신호 기반으로 나눌 수 있다.[6]

측정 방식	설 명
네트워크 기반	이동통신사의 기지국과 통신하는 경우 그 위치 값 (Cell 기반)
위성 신호 기반	GPS 위성에서 송신하는 신호를 바탕으로 계산
Wi-Fi 신호 기반	Wi-Fi AP(Access Point)의 위치를 조회하여 위치 값 측정

2.3 PC 환경에서 위치 정보

PC 환경에서 위치 정보는 스마트폰 환경에서의 분류 기준과 비슷하지만, 기지국과 통신하지 않으므로 네트워크 기반의 방식을 제외하고 위성 신호 기반, Wi-Fi 신호 기반 두 가지로 나눌 수 있다. Windows 10 운영체제 이전의 PC는 Wi-Fi 신호를 기반으로 위치 정보를 측정했으나 Windows 10 운영체제 이후는 위성 신호를 기반으로 하는 위치 정보 측정 기술을 탑재하고 있다.

2.4 웹캐시 상에서의 위치 정보

캐시는 저장하는 위치에 따라 클라이언트 캐시, 프락시 캐시, 웹캐시로 나눠진다. 그 중 웹캐시는 웹 서버의 성능을 향상시키고 응답 시간을 줄이기 위해 메모리에 저장되는 것을 말한다.[7] 따라서 웹캐시는 이미지, 텍스트, HTML, XML 파일과 같은 형태로 캐시 데이터의 저장 위치, 다운로드 URL, 다운로드 시각 정보를 저장한다. 이러한 정보의 분석을 통해 특정 시간에 접속한 사이트와 접속 당시 화면들을 확인할 수 있으므로 웹 브라우저를 통한 사용자의 행위 분석 시 캐시의 활용도가 높아지고 있다.

웹 기반 LBS의 경우 위치 기반 광고, 날씨, 주변 정보 검색을 위해 위치 정보가 사용된다. 이러한 특성 때문에 계속해서 위치 정보를 변경하는 것이 아니라 접속된 시간 동안 동일한 위치 정보를 받아서 활용한다. 따라서 초기 접속할 때 받아온 위치 정보를 웹 캐시의 형태로 저장한 후 사용할 가능성이 높다.

최근 데스크탑 PC의 이용률은 낮아지는 반면 태블릿 PC와 더불어 노트북 PC를 휴대하는 사람들이 늘어나고 있어,[8] 앞으로 PC에도 의미 있는 위치 정보가 많이 남아있게 될 것이다. 하지만 위치 정보와 관련된 선행 연구들은 대부분이 모바일 디바이스를 기반으로 하고 있으며 웹 캐시는 고려하지 않는다. 이에 본 논문에서는 PC에 남는 아티팩트 중에서도 웹 기반 LBS를 사용할 때 남게 되는 웹캐시에 중점을 두고 위치 정보에 대한 분석을 진행하였다.

III. 웹 기반 LBS 분석

3.1 분석 환경

사용자는 운영체제나 웹 브라우저에서의 설정을 통해 위치 정보 사용 여부를 선택할 수 있다. 그러나 통상적으로 LBS 이용 시에는 서비스가 사용자의 위치 정보에 접근할 수 있도록 위치 정보 사용 활성화가 필요하다. LBS 이용 시 일시적으로 위치 정보 사용을 활성화 시키는 사용자와 상시적으로 활성화 시키고 있는 사용자 모두 존재하기 때문에, 각 디바이스에는

사용자의 성향에 따라 상이한 위치 정보 설정이 존재할 수 있다. 따라서 본 논문에서는 사용자 디바이스 내 존재하는 LBS 사용 흔적을 분석하여 디지털 포렌식적 활용 가능성을 제시하고자 하며, 모바일 위치 정보에 집중된 다수 선행 연구와 달리 PC 내 저장된 웹캐시 상 위치 정보를 중점적으로 다룰 것이다. 본 논문에서 웹 기반 LBS 사용 흔적을 분석하기 위한 환경은 다음과 같다.

운영체제	Windows 10
인터넷 브라우저	Google Chrome 61.0.3163.100
	Microsoft Edge 40.15063.674.0

단순 지도 이용 서비스, 딜리버리 서비스, 숙박 서비스로 카테고리를 분류하여 각 카테고리별 분석 대상 서비스를 선정하였다.

서비스 유형	웹 기반 LBS
지도	구글 맵
	네이버 맵
	다음 맵
딜리버리	배달의 민족
숙박	여기어때

3.2 분석 결과

위에서 선정된 웹 기반 LBS를 실행하고 그에 따라 생성되는 사용 흔적 중 위치 정보가 남아 있는 파일은 대부분 캐시 파일이며 그 파일명은 아래와 같다.

웹 기반 LBS	캐시 파일명
구글 맵	ko.html
네이버 맵	geo2address
다음 맵	getUnionResponse
배달의 민족	쿠키 파일로 존재
여기어때	쿠키 파일로 존재

캐시 파일명은 브라우저와 관계없이 동일하지만, 운영체제와 브라우저의 위치 서비스 기능 설정에 따라 파일에 저장된 위치 정보의 종류가 달라진다. 각각의 분석 결과는 다음과 같다.

3.2.1 Microsoft Edge

Microsoft Edge(이하 Edge)는 Windows 10의 위치 서비스 기능과 직접 연동된다. 따라서 웹 기반 LBS에 접속하면 Windows 10의 위치 서비스 기능을 활성화 시킬 수 있도록 권한을 요청한다. 이를 허용하면 GPS 신호 기반 위치 정보를 활용할 수 있고, 거부하면 LBS를 이용할 수 없다. 즉, Edge에서는 Wi-Fi 신호 기반의 위치 정보와 관계없이 위성 신호 기반의 위치 정보만 활용한다.

3.2.2 Google Chrome

Google Chrome(이하 Chrome)에서 웹 기반 LBS를 사용할 경우 Windows 10의 위치 서비스 기능의 활성화 여부에 따라 사용되는 위치 정보의 종류가 달라진다.

구글 메인 페이지는 최초 접속하거나 위치 정보를 한 번이라도 요청하는 경우 UULE 쿠키를 생성한다. 구글 맵 최초 접속 시 UULE 쿠키가 존재하면 해당 쿠키의 위치 정보를 이용하고, 만약 존재하지 않다면 Wi-Fi 신호 기반 위치 정보를 사용한다. 구글 맵에서 현재 위치 정보를 요청하면 Chrome 브라우저에서 위치 정보 사용 권한을 요청한다. 이를 허용하는 경우 위치 정보를 쿠키에 저장하는데 이 때 Windows 10 위치 서비스 기능의 활성화 여부에 따라 저장되는 정보의 종류가 달라진다. 위치 서비스 기능이 활성화 된 경우 해당하는 GPS 신호 기반 위치 정보를 저장하고, 비활성화인 경우 Wi-Fi 신호 기반의 위치 정보를 저장한다.

반면 네이버 맵, 다음 맵의 경우 Chrome 브라우저에서 위치 정보 사용 권한을 따로 요청하지 않고 Wi-Fi 신호 기반의 위치 정보만 사용한다.

배달의 민족, 여기어때의 경우 위치 정보가 쿠키 파일에만 존재한다. Chrome 브라우저에서 요청한 권한을 허용하면 Windows 10 위치 서비스 기능 활성화 여부에 따라 저장되는 위치 정보의 종류가 구글 맵의 경우와 동일하다.

분석 결과 PC에서 웹 기반 LBS를 사용할 경우 위치 정보는 대부분 캐시 파일로 저장되고, 일부는 쿠키 파일로 존재함을 확인했다.

IV. 웹 기반 LBS의 효율성 검증

2.3의 내용대로 PC환경에서는 Wi-Fi 신호 기반 위치 정보와 위성 신호 기반 위치 정보를 사용한다. 또한 3장에서 살펴본 대로 운영체제의 위치 서비스 기능의 활성화 유무에 따라 웹 캐시에 저장되는 위치 정보의 종류가 달라진다.

이러한 웹 기반 LBS 분석 결과의 효율성을 검증하기 위하여 악의적인 사용자가 VPN을 이용해 자신의 위치를 조작하였다는 가정 하에 테스트를 수행했다.

실제 디바이스 사용 위치는 “동대문구 이문2동 쌍용아파트 107동”이며 설정한 VPN의 위치는 “경상북도 안동시 도산면”이다. 3장에서 선정한 웹 기반 LBS를 실행했을 때 웹캐시에 저장된 위치정보는 다음과 같다.

LBS	위치 활성화	위치 비활성화
구글	동대문구 이문2동	경상북도 안동시 도산면
구글 맵	동대문구 이문2동 쌍용아파트 107동	경상북도 안동시
네이버 맵	경상북도 안동시 용상동	안동시 용상동
다음 맵	경상북도 안동시 도산면	안동시 도산면
배달의 민족	동대문구 이문동	경상북도 안동시 도산면
여기어때	동대문구 이문동	안동시 안동

위치 서비스 기능이 비활성화인 경우 모든 위치 정보가 Wi-Fi 신호 기반 위치 정보로 실제 위치가 아닌 VPN의 위치를 표시해주는 한계가 존재한다. 그러나 기능을 활성화하면 구글, 구글 맵, 배달의 민족, 여기어때에 남는 위치 정보는 GPS 위성 신호 기반 위치 정보를 우선으로 받기 때문에 실제 디바이스의 위치를 표시한다. 이를 통해 기존 PC의 네트워크를 이용한 위치 특정의 한계점이 운영체제의 위치 서비스 기능으로 해결되어 더 정확한 위치 정보를 얻을 수 있다는 점을 확인하였다.

V. 결론 및 추후 연구 방향

Windows 10 운영체제의 출시 이후 PC의 GPS 위성 신호를 기반으로 한 위치 정보를 활용하는 웹 기반 LBS가 증가하고 있다. 또한 휴대용 노트북 PC의 사용률이 증가하여 PC에 남는 위치 정보가 유의미해지고 있으며 대부분 캐시의 형태로 저장되고, 일부 쿠키도 활용되고 있다. 본 논문에서는 웹 기반 LBS 사용 시에 남는 웹캐시 파일을 분석하고 어떤 종류의 위치 정보가 남는지 분류하였다. 이를 통해 악의적으로 Wi-Fi 신호 기반 위치를 조작한 경우에도 실제 디바이스의 위치를 파악할 수 있음을 확인하였다. 따라서 사용자의 위치 정보를 파악해야 하는 수사의 경우 PC의 웹캐시 파일에 남는 위치 정보 획득이 가능해졌으며 이를 통해 기존보다 더 정확한 위치 정보를 활용할 수 있게 되었다.

웹 기반 LBS는 본문에서 다룬 LBS 이외에도 지속적으로 증가하고 있기 때문에 추가적인 연구가 이루어진다면 더 많은 위치 정보를 추출할 수 있게 될 것이다. 또한 웹캐시 파일은 스마트폰의 웹 브라우저 애플리케이션을 사용했을 시에도 생성된다. 따라서 스마트폰을 추후 연구 대상에 포함하여 웹 브라우저 애플리케이션 별로 생성되는 캐시 파일을 분석하여 활용하는 방안을 마련하는 것도 유의미할 것이다.

[참고문헌]

[1] 방송통신위원회, 위치기반서비스 신고 사업자 현황, 2016.12.31.

[2] 방송통신위원회, 국내·외 위치정보 산업 동향 보고서, 2016.12

[3] Dohyun Kim, Jewan Bang and Sangjin Lee, Analysis of Smartphone-Based Location Information, Springer Science+Business MEdia B.V. 2012, December 2011.

[4] 연규철, 김문호, 김도현, 이상진, 스마트 기기에 설치된 내비게이션 어플리케이션의 위치 정보 흔적 연구, Journal of The Korea Institute of Information Security & Cryptology, Feb, 2016.

[5] 김제민, 백혜정, 박영택, 스마트폰 사용자의 GPS 기반 개인 경로 모델 학습 및 이동 경로 예측, 정보과학회논문지 : 소프트웨어 및 응용 제 39 권 제 1 호, 2012. 1.

[6] 한국인터넷진흥원(KISA), 2016 국내 LBS 산업 실태조사 결과 보고서, 2016. 12.

[7] 주안국, 문서 형태에 따른 분할 웹 캐시 구조와 효율적인 캐시 제거 기법, 서강대학교 정보통신대학원, 2003.

[8] 한국IDC, 국내 PC 시장 연구 분석, http://www.kr.idc.asia/press/pressreleasearticle.aspx?prid=519, 2017.

본 논문은 KITRI BoB 프로그램의 지원을 받아 작성되었습니다.

IoT 디바이스의 네트워크 정보 이력관리를 통한 사고분석 활용 방안

장대일, 김태은, 김환국*

*한국인터네진흥원 보안기술R&D2팀

A Study on the Application Methods of Incident Analysis through Network Information History Management

Daeil Jang, Taeeun Kim, Hwankuk Kim*

*Security Technology R&D Team 2, Korea Internet Security Agency

요 약

IoT 환경에서 포렌식은 네트워크 통신 데이터 및 디바이스 내에 존재하는 메모리와 로그를 통해 사고분석을 수행하는 것이 일반적인 방법이다. 하지만 네트워크 트래픽과 메모리는 휘발성 데이터이고, IoT 디바이스의 로그는 기기 및 환경적 제약으로 PC 환경에서와 같이 많은 정보를 제공받기는 어려운 것이 현실이다. 이를 위해 네트워크 정보의 이력관리를 이용한 사고분석 시 활용방안에 대해 논하고자 한다. IP, 펌웨어 버전, 포트번호, 프로토콜, 서비스 버전과 이와 연결된 취약점 정보 등 13가지 정보에 대해 이력관리를 수행하고, 변경 가능한 데이터에 대한 유클리드 거리를 이용하여 침해사고 시점 및 대상을 선정할 수 있음을 보였다.

I. 서론

사고 분석 시 시계열 데이터 분석은 매우 중요하고, 사고 시점을 확정하는 것은 매우 어려운 일이다. 이를 위해 기본적으로 휘발성 데이터 및 파일시스템, 로그 등 다양한 포렌식 아티팩트에 대한 분석을 수행한다. 하지만 IoT 디바이스에 대한 포렌식의 경우 환경적 제약으로 인해 정보의 획득 및 분석이 어렵다.

이에 본 논문에서는 IoT 환경에서 네트워크 정보 이력관리를 통한 침해사고 시 활용 방안에 대해 논하고자 한다. 네트워크 트래픽 정보에 대한 분석 및 이를 활용하기 위한 많은 연구가 진행되었으나, 이는 실시간 분석 및 탐지에 중점을 둔다. 하지만 이를 우회하여 공격에 성공한 경우 시스템에서 발생한 행위 및 로그에 의존할 수밖에 없다. 하지만 앞서 언급한 것처럼 IoT 환경의 경우 제공하는 정보의 질과 양이 높지 않기 때문에 분석을 수행하는데 문제가 발생할 소지가 많다. 이에 주기적으로 네트워크의 상황을 파악하고 이를 시계열 데이터로 변환하여 관리함으로써 침해사고 발생 시 네트워크 정보의 변화를 추적함으로써 침해사고에 활용할 수 있음을 보이고자 한다.

2장에서는 네트워크 정보 수집 및 시계열데이터 분석에 관련된 기존 연구를 살펴보고, 3장에서는 네트워크 정보 중 이력관리 대상을 선정하고, 선정된 정보 중 변경 가능성 및 유클리드 거리를 이용하여 네트워크 정보의 변화에 따른 그래프 변화를 살펴볼 것이다. 마지막으로 5장에서 본 논문의 결론 및 향후 연구에 대해 언급할 것이다.

II. 관련연구

2.1 네트워크 정보 수집 도구

인터넷 혹은 네트워크에 연결된 디바이스의 정보를 수집하기 위한 다양한 도구가 존재한다.

와이어샤크[1]처럼 네트워크 트래픽을 정적으로 수집하기 위한 도구와 Nmap[2], Shodan[3], Censys[4] 등 동적으로 원하는 정보를 수집하기 위한 도구가 존재한다. 정적 수집은 네트워크 스위치 및 라우터에서 트래픽을 수집 및 가공하여 원하는 정보를 추출하기 위한 방법이고, 동적 수집은 특정 대상 혹은 대역에 스캔 트래픽 발송 후 받은 응답값을 기반으로 원하는 정보를 추출하기 위한 방법으로 분류할 수 있다.

2.2 시계열데이터 분석

네트워크 정보에 대한 시계열 분석을 위해서는 시계열 데이터에서 패턴을 찾고 검색을 수행해야 한다. 이를 위한 일반적인 패턴 정의는 시계열 그래프가 나타내는 모양을 보고 그 특징에 따라 패턴으로 표현한다[6,7,8]. 또는 시계열 데이터를 일정 시간 간격으로 슬라이스하여 세그먼트를 만들고 각 세그먼트에 명칭적 표현을 적용하여 나타는 내는 방법[9]을 사용하거나 보여지는 패턴을 확률모델로 표현하여 사용하기도 한다[10,11].

III. 네트워크 정보 이력관리 대상 선정

본 장에서는 네트워크 정보 수집시스템으로부터 수집 가능한 데이터를 알아보고 포렌식을 위한 이력관리 대상을 선정하고자 한다. 앞서 언급한바와 같이 와이어샤크 등을 통해 수집 가능한 네트워크 트래픽데이터는 제외하고 네트워크 스캔을 통해 수집 가능한 정보를 대상으로 한다.

3.1 네트워크 정보 수집

네트워크 스캔을 통해 수집 가능한 정보에는 다양한 종류가 존재한다. 기본적인 대상의 IP 및 스캔 대상 포트, 해당 포트에서 사용하는 프로토콜, 어플리케이션 및 버전 정보 등이 존재한다. 수집 가능한 정보의 리스트는 다음과 같다.

- IP, Port, Protocol, Application Name, Application Version, OS Name, OS version, Firmware version, Product Name, Product Version, Protocol Header, Protocol Contents,

이러한 정보를 추출하기 위해 ZMap 기반의 커스터마이징된 도구를 사용한다. ZMap의 경우 12개 프로토콜을 대상으로 16개 서브프로토콜의 정보를 수집하고 이를 JSON 포맷으로 변환하여 관리한다. 위의 정보 중 ZMap으로 추출 가능한 정보는 다음과 같다.

- IP, Port, Protocol, Application Name, Application Version, Protocol Header, Protocol Contents,

여기에 OS를 식별하기 위한 TCP/IP 헤더 정보를 이용하는 룰 기반의 핑거프린트와 펌웨어 식별을 위한 각 IoT 디바이스 별 펌웨어 식별 기술을 이용하여 나머지 정보를 추출한다.

- OS Name, OS version, Firmware version, Product Name, Product Version

3.2 취약점 매칭 정보

이렇게 수집된 정보는 해당 디바이스의 취약점 발생 가능성을 측정하기 위한 정보로 활용된다. 이때 CVE를 기존으로 CVE와 매칭이 되는 어플리케이션 및 버전정보, OS 및 버전정보, Firmware 및 버전정보를 기반으로 취약점 발생 가능성을 판단하게 된다. CVE 취약점 리스트 중 스캔된 결과에 해당하는 어플리케이션등의 버전이 포함되어 있으면 해당 정보에 CVE-ID를 매칭시켜 관리하게 된다. 이를 위해 CVE 크롤러와 정형 취약점 정보 분석 및 파싱기술, 키워드 매칭 기술을 활용한다.

3.3 대상 정보에 대한 이력관리 셋 구조

수집된 정보와 취약점 정보가 매칭된 결과가 생성되면 이력관리를 위한 셋을 구성한다. 기본적인 구조는 다음과 같다. 하나의 IP를 스캔한 날짜를 먼저 기록하고 식별된 디바이스의 프로덕트 네임과 버전, 펌웨어 버전 및 OS를 식별한다. 그리고 포트의 경우 하나의 디바이스에 여러 포트가 열려있을 수 있어 각 포트 별 프로토콜 및 어플리케이션에 대한 정보를 수집하고 이와 매칭이 되는 CVE 정보를 기술한다. 이렇게 생성된 이력관리 셋은 IP 별 날짜를 기준으로 수집되는 데이터를 관리하게 된다.

- IP - Date - Product Name(Version) - Firmware Version - OS
  - Port1 - Protocol - Application - CVE-IDs
  - Port2 - Protocol - Application - CVE-IDs
  - Port3 ...

이렇게 관리되는 데이터는 시계열 데이터로 다양한 의미를 지니게 된다. 우선 각 IP 별 연결되는 디바이스의 현황을 파악하게 된다. 일반적으로 보안관제를 비롯하여 Shodan 및 Censys에서도 현재의 상황만을 보여주고 시간의 흐름은 표현되지 않는다. 이는 침해사고가 발생할 당시 감염 가능성이 있는 디바이스를 식별하기 위한 용도로 활용될 수 있다. 두 번째로 IP를 할당받은 디바이스의 네트워크 변화를 관찰할 수 있다. IoT 디바이스는 홈IoT를 제외한 환경적 특성상 여러 개체의 동일한 디바이스를 사용하게 된다. 이때 감염 가능성이 있는 디바이스 제품군에서 발생하는 동일한 네트워크 변화는 침해사고 분석에 있어 침해사고 시점을 판단하기 위한 중요한 아티팩트가 될 수 있다.

3.4 유클리드 거리를 이용한 시계열 데이터 분석

시계열 데이터 분석을 위해 일반적으로 시계열 정보에 대한 패턴을 정의하고 학습 모델을 생성하여 패턴 별 학습 모델을 작성한다. 그리고 생성된 학습 모델과 새로운 질의 시퀀스 사이의 유사도를 측정하여 새로운 질의 시퀀스가 어떤 패턴과 유사한 특성을 나타내는 가를 결정하게 된다. 하지만 IoT 환경의 경우 수집 가능한 네트워크 정보의 변화가 거의 없기 때문에 이러한 방법의 경우 시스템 리소스 소모량에 비해 그 효과가 적다. 예를 들어 IP 별 열린 포트 수의 변화, 제품-펌웨어 버전 정보 변화, 접근 페이지 변화 등이 네트워크 정보 중 변경될 여지가 있지만, 정보의 변화가 빈번하게 발생하지는 않는다. 이에 열린포트와 젬품-펌웨어 버전 정보 및 접근페이지 분석 결과 간 유클리드 거리를 이용하여 시계열 데이터에 대한 상관분석을 수행한다. 3차원 내 두 점 사이의 유클리드 거리 계산식은 다음과 같다.

$$d(p,q) = \frac{1}{1-\sqrt{(x_p - x_q)^2 + (y_p - y_q)^2 + (z_p - z_q)^2}}$$

여기에서 p는 현재 데이터, q는 직전데이터이고, x는 포트 수, y는 펌웨어 버전, z는 접근페이지 분석 결과를 의미한다. 이를 IoT 환경에 대입해보면 거의 대부분의 구간에서 현재 데이터와 직전 데이터 간 유클리드 거리 값 1이 나오는 것을 확인할 수 있다. 이때 펌웨어 업데이트, 신규 서비스 생성 및 침해사고 등으로 정보의 변경 시 유클리드 거리는 1 이외의 값이 나오게 된다. 또한 새로운 IP에서 접근 시에도 1 이외의 값이 나오게 된다.

그림 1의 경우는 단일 디바이스에서 인덱스 페이지의 정보가 변경된 후 다시 복구했을 경우 유클리드 거리의 변화를 나타내고 있다, 그림 2는 단일 디바이스에서 펌웨어 업데이트로 네트워크 정보가 변경되었을 경우 유클리드 거리를 나타내고 있고, 그림 3은 동일 디바이스군에서 취약점에 의해 침해 사고과 확산되었을 경우 각 디바이스 별 유클리드 거리에 대한 표준편차를 이용하여 네트워크 정보의 변화를 보여주고 있다. 마지막으로 그림 4는 동일 디바이스군에 새로운 디바이스가 추가되었을 경우 디바이스군에 속한 디바이스들의 유클리드거리에 대한 표준편차를 보여준다.

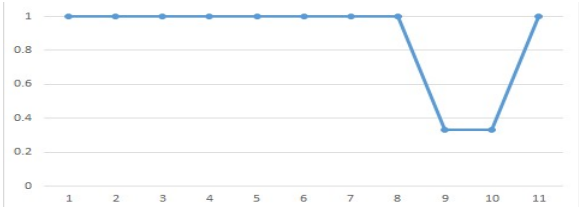


그림 1. 디바이스 정보 변경 후 복구

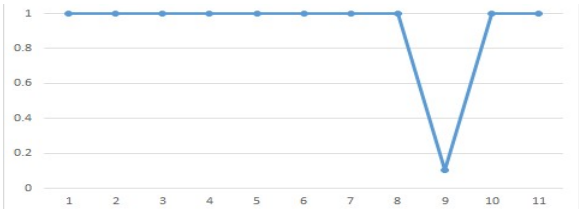


그림 2. 디바이스 정보 변경 후 지속 유지

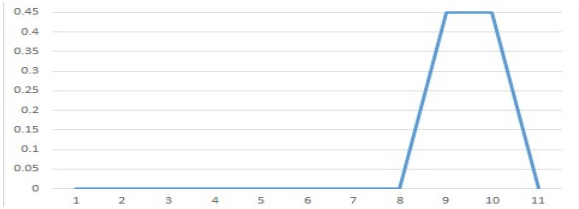


그림 3. 동일 디바이스 군에서 침해 확산

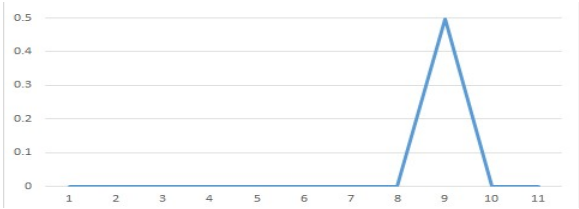


그림 4. 디바이스 군에 디바이스 추가

IV. 결론

IoT 환경에서 침해사고 분석 수행은 매우 제한된 정보로 인해 많은 시간과 노력을 필요로 한다. 특히 네트워크 트래픽 등 휘발성 데이터가 많이 빠른 대응을 필요로 한다. 이에 IoT 환경에서 사고분석을 위해 네트워크의 정보를 저장하는 것은 매우 의미가 있다.

본 논문에서는 IoT 환경에서 네트워크 정보에 대한 이력관리를 수행함으로써 사고분석 시 활용될 수 있음을 보였다. 네트워크 정보 중 변화 가능성이 있는 정보를 대상으로 수집 시간에 별 유클리드 거리에 따른 유사도 분석을 통해 네트워크 정보의 변화를 탐지하고, 변화 시점의 정보를 제공함으로써 침해사고 시점을 선정하는데 활용될 수 있음을 보였다. 또한 네트워크 정보와 취약점 정보의 연계를 통해 발생한 침해사고 시 활용될 수 있는 취약점 정보 및 침해사고 발생 시점에 취약점을 가지고 있는 디바이스 리스트를 제공함으로써 분석 대상 선정에 활용될 수 있음을 확인하였다.

현재 이력관리의 경우 디바이스 자체에 대해 동적인 스캔을 수행하여 수집되는 정보를 통해 이력관리를 수행하고, 이를 활용하기 위한 방안을 제시하였다. 이와 더불어 휘발성 데이터인 디바이스간 통신 내역 및 인바운드, 아웃바운드 트래픽의 필요정보에 대한 이력관리에 대한 연구가 필요하다.

[Acknowledgement]

이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2016-0-00193, IoT 보안 취약점 검색·공유 및 시험 기술 개발)

[참고문헌]

[1] Wireshark, Wireshark Foundation, https://www.wireshark.org/

[2] Nmap Security Scanner, https://nmap.org/

[3] Shodan, https://www.shodan.io/

[4] Censys, University of Michigan, https://censys.io/

[5] Z. Durumeric, E. Wustrow, and J. A. Halderman. “ZMap: Fast Internet-Wide Scanning and its Security Applications”. In Proc.USENIX Security Symposium, Aug. 2013.

[6] G. Xianping, "Pattern Matching in Financial Time series Data," 1998.

[7] E. Keogh, "A fast and robust method for pattern matching in time-series databases," In Proc. of the 9th Int. Conf. on Tools with Artificial Intelligence, pp.578-584, 1997.

[8] E. Keogh and P. Smyth, "A probabilistic Approach to fast pattern matching in time series databases," In the third conference on Knowledge discovery in Database and Data mining, pp.24-30, 1997.

[9] W. Wang and J. Yang, P. S. Yu, "Mining Patterns in Long Sequential Data with noise," ACM SIGKDD Exploration, pp.28-33, 2001.

[10] X. Ge and P. Smyth, “Deformable Markov model templates for time-series pattern matching,” In proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Boston, MA, Vol.20, No,23, pp.81-90, 2000.

[11] A. Malegaonkar, A. Ariyaeeinia, P. Sivakumaran, and J. Fortuna, "Unsupervised Speaker Change Detection Using Probabilistic Pattern Matching," IEEE Signal Processing Letters, Vol.13, No.8, 2006.

2017 한국정보보호학회 동계학술대회

D5 인증 및 키 관리

Simon Woo (SUNY Korea)

- **멀티캐스트 CoAP 보안을 위한 키 설정 기법**
  조정모, 한상우, 박창섭 (단국대학교)
- **OQS 프로젝트 중 격자 기반 키 교환 방식의 타이밍 등 공격 분석**
  안형철, 한성호, 최낙준, 김광조 (KAIST)
- **IoT 환경을 위한 효율적인 그룹 인증 및 키 교환 기법**
  이대휘, 이임영 (순천향대학교)
- **최신 PKI기반 V2X보안 기술 동향분석**
  이정호 (한국정보인증)

멀티캐스트 CoAP 보안을 위한 키 설정 기법*

조정모, 한상우, 박창섭

단국대학교

Key establishment scheme for multicast CoAP security

Jung-Mo Jo, Sang-Woo Han, Chang-Seop Park

Dankook University

요 약

본 논문에서는 멀티캐스트 CoAP 보안 키 설정 프로토콜을 제안한다. 멀티캐스트 CoAP에서 CoAP 클라이언트가 송신한 CoAP Request 메시지는 멀티캐스트로 CoAP 서버 그룹에게 전송되지만, 그에 대한 응답으로 각각의 CoAP 서버가 송신한 CoAP Response 메시지는 유니캐스트로 CoAP 클라이언트에게 전송된다. 따라서 CoAP Request 메시지는 그룹키로, 각각의 CoAP Response 메시지는 서버와 클라이언트 간의 개별키로 보호된다. 제안 프로토콜은 클라이언트와 서버가 최초 CoAP 메시지 교환 과정에서 ECDH를 이용하여 그룹키와 개별키를 설정하는 것이다. 제안 프로토콜은 DTLS Handshake를 대체할 수 있어 통신효율이 높고 확장성이 있으며, 개별키를 설정하기 때문에 종단간 보안을 지원할 수 있다.

I. 서 론

최근 LLN (Low-power Lossy Networks) 내에서 작동하는 다양한 센서 디바이스들이 IoT (Internet of Things)에 연동되기 시작하였다. 이를 가능하게 하는 기술로써 LLN 및 센서 디바이스의 자원제약적인 측면을 고려한 6LoWPAN [1] 및 RPL [2] 등의 다양한 프로토콜들이 제안되고 있다. IETF Core 워킹그룹에 의해 제안된 CoAP(Constrained Application Protocol) [3]의 등장과 함께 LLN 환경에서 작동하는 센서 디바이스들의 제어를 가능하게 하는 다양한 응용들이 나타나기 시작하였다. 그룹통신(Group Communication)은 다양한 IoT 시나리오에서 동질적인 IoT 디바이스들의 그룹에 대한 제어를 가능하게 하는 도구이기 때문에 최근에는 CoAP에서 멀티캐스트를 지원되는 표준 명세 [4]가 제안되었다. 특정 메시지가 한 개 이상의 센서 디바이스들에게 동시에 전달되고, 또한 다수의 센서 디바이스들로부터 센싱정보가 획득될 수 있다면 통신 효율의 개선이 가능하게 된다.

일반적인 유니캐스트 CoAP 보안을 위해서는 DTLS (Datagram TLS) Handshake 프로토콜 [5]이 적용되어 상호인증 및 세션키가 도출된다. DTLS Handshake 프로토콜은 한 쌍의 CoAP 클라이언트와 서버 간에만 기동이 되기 때문에 멀티캐스트 CoAP 보안을 위해서는 그것을 사용할 수가 없게 된다. 따라서 별도의 보안 메커니즘이 설계되어야 한다. 이에 대한 기존연구는 현재까지 [6, 7]이외에는 존재하지 않는다. 하지만 그 방식 역시 보안적인 측면에서 종단간 보안을 만족하지 않는다는 취약점을 내재하고 있다.

본 논문에서는 종단간 보안을 만족하는 멀티캐스트 CoAP 보안을 위한 보안 아키텍처를 RD(Resource Directory) [8]를 활용하여 설계하고, 이를 기반으로 그룹키 및 개별키 설정을 가능하게 하는 보안 메커니즘을 제안한다. 2장에서는 CoAP과 DTLS에 대해서 소개하고, 3장에서는 네트워크 모델과 설계 가정을 설명하고 프로토콜을 제안한다. 4장에서는 이에 대한 보안 분석을 하고 5장에서는 제안한 프로토콜의 재전송에 관련된 성능 분석을 한다. 마지막으로 6장에서는 결론을 맺는다.

II. CoAP과 DTLS

2.1 CoAP

CoAP은 IoT 환경에서의 표준 웹 전송 프로토콜로

* 본 연구는 미래창조과학부 및 한국 인터넷 진흥원의 "고용계약형 정보보호 석사과정 지원사업"의 연구결과로 수행되었음. (과제번호 H2101-17-1001)

자원 제약적인 디바이스와의 정보교환을 위해 경량화되었다. CoAP 종단간에 CoAP Request 및 CoAP Response 메시지의 교환을 통해서 CoAP 클라이언트는 특정 명령어를 센서 디바이스에게 전송하거나 또는 센싱정보를 CoAP 서버로부터 전달받게 된다. 특히 LLN 환경의 특성을 고려하여 CoAP에서는 CoAP 메시지의 신뢰성 있는 전송을 위해 Exponential-Backoff 방식에 기반을 둔 Stop-and-Wait 전송 프로토콜이 사용된다. CoAP은 일반적인 유니캐스트 환경에서뿐만 아니라, 다수의 CoAP 서버를 대상으로 하는 멀티캐스트 환경에서도 동작된다.

2.2 DTLS

DTLS는 IoT 클라이언트-서버 모델에서의 안전한 CoAP 기반 응용 보안 메커니즘의 사실 상의 표준이다. CoAP-DTLS에서 권고되는 보안모드(Security Mode)는 PSK(Pre-Shared Key), RPK(Raw Public Key) 및 ECDSA 기반 Certificate가 있다. DTLS 보안모드는 CoAP 종단간의 상호인증 및 세션키 도출을 위해서 Fig. 1.에서와 같이 3단계로 구성되는 DTLS Handshake 프로토콜의 세부동작을 규정한다. 보안협상(Security Negotiation) 단계를 통해서 보안모드 그리고 사용할 Cipher Suit 및 Nonce들이 결정된다. 키 교환(Key Exchange) 단계에서는 보안모드에 따른 인증키(Authentication Key, AK) 및 이를 기반으로 세션키(Session Key, SK)가 도출되며, 마지막으로 키 확인(Key Confirmation) 단계에서 상호인증이 수행된다. 모든 과정이 성공적으로 종료되면 도출된 세션키를 이용하여 CoAP 메시지에 대한 보호작업이 DTLS Record 프로토콜을 통해 실행된다.

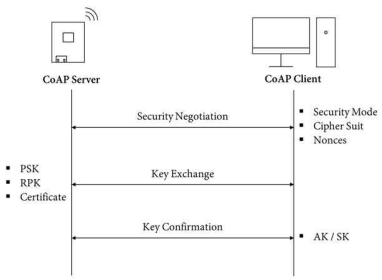


Fig. 1. DTLS Handshake Protocol

III. 제안 프로토콜

3.1 네트워크 모델과 설계 가정

본 논문에서 제안하는 프로토콜이 동작하는 네트워크의 모델은 Fig. 2.와 같고, 사용되는 기호는 Table 1.과 같다. 자원제약이 없는 CoAP 클라이언트는 인터넷상에 위치하며, 자원제약이 있는 CoAP 서버는 IEEE 802.15.4 기반 멀티 홉 6LoWPAN 네트워크상에 위치한다. CoAP 클라이언트 C는 CoAP Request 메시지를 멀티캐스트로 총 s개인 CoAP 서버들(S_1, S_2,⋯,S_j,⋯,S_s)에게 전송하고, 이를 받은 각각의 CoAP 서버 S_j는 CoAP Response 메시지를 유니캐스트로 CoAP 클라이언트 C에게 전송한다. RD(Resource Directory) 서버는 모든 CoAP 서버들이 어떠한 자원을 가지고 있는지에 대한 정보들을 저장하고 있으며, GC(Group Controller)의 기능도 수행한다. GC는 CoAP 서버들로 이루어진 멀티캐스트 그룹에게 IPv6 멀티캐스트 주소를 할당하고, 멀티캐스트 그룹을 관리한다. 다음과 같은 2개의 설계 가정을 한다.

설계 가정 1. GC는 그룹 ECDH 개인키·공개키 쌍 $(q_G, \boldsymbol{Q}_G)$를 생성하고 별도의 등록 프로토콜을 통하여 CoAP 서버 S_j에게 $\boldsymbol{Q}_G$와 q_G를, CoAP 클라이언트 C에게 $\boldsymbol{Q}_G$를 안전하게 분배하였다.

설계 가정 2. CoAP 클라이언트 C와 CoAP 서버 S_j는 각각 자신의 ECDH 개인키·공개키 쌍 $(q_C, \boldsymbol{Q}_C)$와 $(q_j, \boldsymbol{Q}_j)$를 가지고 있다.

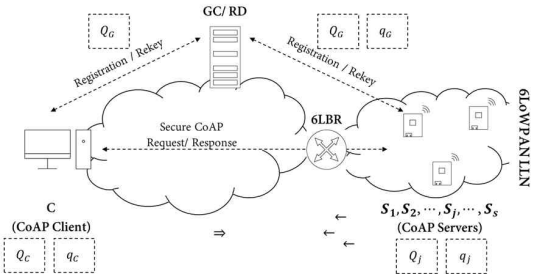


Fig. 2. Proposed Network Model

Table 1. Table of Notations

Notation	Description
C, S_j	CoAP client and CoAP servers, j = 1, 2, 3, ⋯,s
GC / RD	group controller / resource directory
$IPv6_X$	IPv6 address of $X(=C,\ j$ for $S_j)$
$IPv6_{Multicast}$	IPv6 multicast address
GK	group key
UK_j	unicast key shared between C and S_j
Q_X, q_X	ECDH public and private keys of $X(=C,\ GC,\ j)$
$MIC_X(K)$	Message-integrity code computed by $X(=C,\ j)$ with a symmetric key K covering all preceding message field
⇒, ←	multicast and unicast transmission

3.2 멀티캐스트 CoAP 키 설정 프로토콜

유한체 F_p(p는 소수)상의 타원 곡선 $E(F_p)$는 $y^2 \equiv x^3 + ax + b (\mathrm{mod} p)$를 만족하는 점들과 무한대 점 $\boldsymbol{O}$의 집합이다. 타원 곡선 매개 변수는 $(p,a,b,\boldsymbol{G},n)$로 나타내며, n은 타원 곡선 $E(F_p)$의 위수이고, $\boldsymbol{G}$은 n을 위수로 갖는 베이스 포인트 생성자이다. ECDH 개인키와 공개키 쌍은 $(q_X,\ Q_X)$으로 나타낸다. ECDH 개인키 q_X는 $[1, n-1]$범위에서 선택된 난수이고, ECDH 공개키 Q_X는 $Q_X = q_X \cdot \boldsymbol{G}$로 계산 된다. ECDH 공개키 Q_X로부터 ECDH 개인키 q_X는 도출될 수 없다.

제안 프로토콜에서 그룹키와 개별키는 Fig. 3.과

같이 계산된다. CoAP 클라이언트 C는 그룹키 $GK = ECDH(Q_G, q_C)$를 그룹 ECDH 공개키 $\boldsymbol{Q}_G$와 자신의 ECDH 개인키 q_C를 이용하여 계산하고, CoAP 서버 S_j와의 유니캐스트 키 $UK_j = ECDH(\boldsymbol{Q}_j, q_C)$를 CoAP 서버 S_j의 ECDH 공개키 $\boldsymbol{Q}_j$와 자신의 ECDH 개인키 q_C를 이용하여 계산한다. CoAP 서버 S_j는 그룹키 $GK = ECDH(Q_C, q_G)$를 클라이언트 C의 ECDH 공개키 $\boldsymbol{Q}_C$와 그룹 ECDH 개인키 q_G를 이용하여 계산하고, CoAP 클라이언트와의 유니캐스트 키 $UK_j = ECDH(\boldsymbol{Q}_C, q_j)$를 클라이언트 C의 ECDH 공개키 $\boldsymbol{Q}_C$와 자신의 ECDH 개인키 q_j를 이용하여 계산한다.

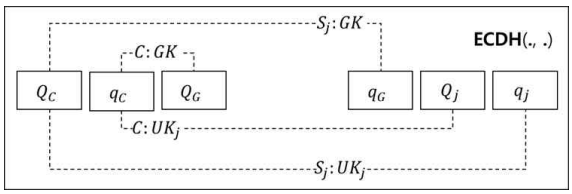


Fig. 3. Group key and Unicast key

그룹키와 유니캐스트 키는 CoAP 클라이언트와 CoAP 서버간에 최초 CoAP 메시지를 교환하는 과정인 프로토콜 1. *Security Bootstrapping*에서 설정된다. CoAP Request 메시지는 그룹키로, 각각의 CoAP Response 메시지는 CoAP 서버와 CoAP 클라이언트 간의 개별키(유니캐스트키)로 보호된다. $\{Epoch, SEQ_C \text{ (or } SEQ_j), Q_C \text{(or } \boldsymbol{Q}_j)\}$는 DTLS Record header의 security-related fields이고, $\{MIC_C(GK) \text{ or } MIC_j(UK_j)\}$는 DTLS Record trailer이다. $Epoch$는 별도의 프로토콜을 이용하여 그룹키 갱신을 위해 사용된다.

프로토콜 1. *Security Bootstrapping*

① $C(IPv6_C) \Rightarrow S_j(IPv6_{Multicast})$:

$\{Epoch, SEQ_C, Q_C\}, CoAP-Request, \{MIC_C(GK)\}$

② $C(IPv6_C) \leftarrow S_j(IPv6_j)$:

$\{Epoch, SEQ_j, Q_j\}, CoAP-Response, \{MIC_j(UK_j)\}$

$IPv6_C$를 IPv6 주소로 가진 CoAP 클라이언트 C는 그룹키 $GK = ECDH(\boldsymbol{Q}_G, q_C)$를 계산한다. $Epoch$를 0으로 설정하고, 또한 SEQ_C도 0으로 설정한다. $\{Epoch, SEQ_C, Q_C\}$, $CoAP-Request$와 그룹키 GK를 이용하여 메시지 무결성 코드 $MIC_C(GK)$를 계산한다. 그리고 $IPv6_{Multicast}$를 IPv6 멀티캐스트 주소로 가진 CoAP 서버 그룹에게 멀티캐스트로 $\{Epoch, SEQ_C, Q_C\}, CoAP-Request, \{MIC_C(GK)\}$를 보낸다(①). 이를 수신한 각 CoAP 서버 S_j는 그룹키 $GK = ECDH(\boldsymbol{Q}_C, q_G)$를 계산한 후, $MIC_C(GK)$를 검증한다.

$IPv6_j$를 IPv6 주소로 가진 각 CoAP 서버 S_j는 유니캐스트 키 $UK_j = ECDH(\boldsymbol{Q}_C, q_j)$를 계산한다. $Epoch$를 0으로 설정하고 SEQ_j도 0으로 설정한다. $\{Epoch, SEQ_j, Q_j\}, CoAP-Response$와 유니캐스트 키 UK_j를 이용하여 메시지 무결성 코드 $MIC_j(UK_j)$를 계산한다. 각 CoAP 서버 S_j는 $IPv6_C$를 IPv6 주소로 가진 CoAP 클라이언트 C에게 $\{Epoch, SEQ_j, Q_j\}, CoAP-Response, \{MIC_j(UK_j)\}$를 보낸다(②). 이를 수신한 CoAP 클라이언트 C는 자신의 유니캐스트 ECDH 개인키 q_C를 사용하여 유니캐스트 키 $UK_j = ECDH(\boldsymbol{Q}_j, q_C)$를 계산한 뒤에 $MIC_j(UK_j)$를 검증한다.

*Security Bootstrapping*의 결과로써 CoAP 클라이언트 C와 각 CoAP 서버 S_j 사이에 그룹키 GK와 유니캐스트 키 UK_j가 설정된다. *Secure CoAP Application* 세션은 프로토콜 2와 같이 시작된다.

프로토콜 2. Secure CoAP Application

③ $C(IPv6_C) \Rightarrow S_j(IPv6_{multicast})$:

$\{Epoch, SEQ_C\}, CoAP-Request, \{MIC_C(GK)\}$

④ $C(IPv6_C) \leftarrow S_j(IPv6_j)$:

$\{Epoch, SEQ_j\}, CoAP-Response, \{MIC_j(UK_j)\}$

안전한 CoAP 응용을 위해 앞으로 교환할 CoAP Request 및 CoAP Response 메시지 (③ 및 ④)는 그룹키 GK와 유니캐스트 키 UK_j로 보호된다. SEQ_C와 SEQ_j는 메시지가 처리될 때마다 1씩 증가한다. 만약 기밀 정보를 주고받을 경우에 $CoAP-Request$는 GK로, $CoAP-Response$는 UK_j로 암호화하여 기밀성을 보장할 수 있다.

IV. 보안 분석

4.1 종단간 보안 잇슈

GC는 그룹 ECDH 개인키 q_G를 생성하였기 때문에 그룹키 $GK = ECDH(Q_C, q_G)$를 계산할 수 있으므로, CoAP 클라이언트 C가 CoAP 서버들에게 멀티캐스트로 전송하는 CoAP Request 메시지를 해독할 수 있다. 그러나 유니캐스트 키 UK_j는 $UK_j = ECDH(\boldsymbol{Q}_j, q_C) = ECDH(\boldsymbol{Q}_C, q_j)$로 계산되므로 CoAP 클라이언트 C와 CoAP 서버 S_j를 제외한 다른 CoAP 서버 $S_i (i \neq j)$, GC를 비롯한 제 3자는 유니캐스트 키 UK_j를 계산할 수 없다. 따라서 제안 프로토콜은 유니캐스트 키에 의해서 종단간 보안을 제공할 수 있다.

4.2 UDP상의 신뢰할 수 있는 전송

3.2절의 *Security Bootstrapping*은 안전한 멀티캐스트 CoAP 응용을 위해 안정적으로 수행되어야한다. 그러나 멀티캐스트 CoAP Request 메시지와 유니캐스트 CoAP Response 메시지는 Non-Confirmable 메시지이므로 신뢰성이 없다는 문제가 있다. 따라서 *Security Bootstrapping*에서 CoAP 클라이언트는 멀

티캐스트 그룹 내의 모든 CoAP 서버로부터 CoAP Response 메시지를 수신할 때까지 동일한 CoAP Request 메시지를 멀티캐스트로 계속 재전송해야한다. 이에 대한 성능 분석은 5장에서 이루어진다.

V. 성능 분석

재전송을 요구하는 *Security bootstrapping*의 성능을 평가하기 위해 재전송 횟수는 CoAP 패킷 손실률과 멀티캐스트 그룹의 CoAP 서버의 수를 기반으로 하여 계산한다. ρ는 CoAP 패킷 손실률이고, M_j는 CoAP 클라이언트가 CoAP 서버 S_j로부터 CoAP Response 패킷을 수신하기 위해 CoAP Request 패킷을 보내는 전송 횟수이다. $M_j = i (i \geq 1)$의 확률은 $Pr\ [M_j = i] = \rho^{i-1}(1-\rho)$이다. CoAP 클라이언트가 CoAP Response 패킷을 정확하게 수신하기 위한 평균 전송 횟수$E(M_j)$는 다음과 같다.

$$E(M_j) = \sum_{i=1}^{\infty} i \cdot Pr[M_j = i]$$

$$= \sum_{i=0}^{\infty} (1 - Pr[M_j \leq i]) = 1/(1-\rho)$$

CoAP 클라이언트가 CoAP 서버로부터 온 CoAP Response 패킷을 올바르게 수신할 수 있도록 전송수를 M으로 정의한다. 서로 다른 CoAP 서버에서 손실된 패킷들의 이벤트가 독립적이라고 가정하면, $Pr\ [M \leq i] = \prod_{j=1}^{S} Pr[M_j \leq i] = (1-\rho^i)^s$이다. 따라서 3.2절의 *Security bootstrapping*에서 CoAP 클라이언트가 그룹 내의 모든 CoAP 서버로부터 CoAP Response 패킷을 받기 위한 CoAP Request 패킷의 평균 전송 수는 다음과 같다.

$$E(M) = \sum_{i=1}^{\infty} i \cdot Pr[M = i] = \sum_{i=0}^{\infty} (1 - Pr[M \leq i])$$

$$= \sum_{i=0}^{\infty} (1 - (1-\rho^i)^s)$$

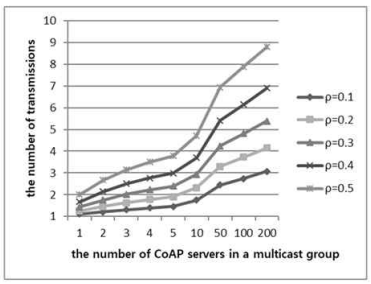


Fig. 4. Number of transmissions for the size of the multicast group

Fig. 4.는 다양한 패킷 손실률(ρ=0.1, 0.2, 0.3, 0.4, 0.5)과 멀티캐스트 그룹의 크기가 증가함에 따라 CoAP 클라이언트의 평균 전송 수를 보여준다. 예상대로 CoAP 서버 수가 증가하거나 패킷 손실률이 증가함에 따라 전송 횟수가 증가한다. CoAP 서버 수가 100개이고 손실률이 0.5이나 됨에도 불구하고 평균 전송수는 7.88이다. 따라서 재전송을 요구하는 *Security bootstrapping*은 전반적인 성능을 저하시키지 않는다.

VI. 결 론

본 논문에서는 멀티캐스트 CoAP 보안을 위한 보안 아키텍처 및 관련 보안 프로토콜들을 제안하였다. 멀티캐스트 CoAP 보안을 위한 가장 중요한 잇슈는 그룹키 및 개별키를 DTLS Handshake 프로토콜을 이용하지 않고 설정하는 것이다. 종단간 보안을 확보하면서 별도의 프로토콜을 도입하지 않고 기존 CoAP Request 및 CoAP Response 프로토콜 내에 상호인증 및 개별키 설정 프로토콜을 내장시키는 접근방법을 채택하였다. 보안분석 및 성능평가를 통해서 제안된 프로토콜의 유효성을 확인할 수 있었고, 향후 DTLS Handshake 프로토콜 자체를 멀티캐스트를 지원할 수 있게 수정 설계하는 방안을 검토해 볼 예정이다.

[참고문헌]

[1] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, Transmission of IPv6 Packets over IEEE 802.15.4 Networks, IETF RFC 4944, Sep. 2007.

[2] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, JP. Vasseur and R. Alexander, RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks, IETF RFC 6550, Mar. 2012.

[3] Z. Shelby, K. Hartke and C. Bormann, The Constrained Application Protocol (CoAP), IETF RFC 7252, Jun. 2014.

[4] A. Rahman and E. Dijk, Group Communication for the Constrained Application Protocol (CoAP), IETF RFC 7390, Oct. 2014.

[5] E. Rescorla and N. Modadugu, Datagram Transport Layer Security, IETF RFC 6347, Jan. 2012.

[6] S. Keoh, S. Kumar, O. Garcia-Morchon, E. Dijk, and A. Rahman, "DTLS-based Multicast Security in Constrained Environments", Work in Progress, IETF draft-keoh-dicemulticast-security-08, July 2014.

[7] M. Tiloca, K. Nikitin, and S. Raza, Axiom - DTLS-based Secure IoT Group Communication, ACM Tr. on Embedded Computing Systems, vol. 16, Article No. 66, May 2017.

[8] Z. Shelby, M. Koster, C. Bormann, P. van der Stok, and C. Amsuess, CoRE Resource Directory, IETF draft-ietf-core-resource-directory-11, Jul. 2017.

OQS 프로젝트 중 격자 기반 키 교환 방식의 타이밍 등 공격 분석1)

안형철* 한성호* 최낙준* 김광조*

*카이스트 정보보호대학원

Timing and Fault Attacks on Lattice-based Key Exchange Protocols in Open Quantum Safe Project

Hyeongcheol An* Seongho Han* Nakjun Choi* Kwangjo Kim**

*Graduate School of Information Security, KAIST.

요 약

Open Quantum Safe(OQS)는 현재 Stebila 등에 의하여 포스트 양자 암호(Post Quantum Cryptography)의 키 교환 프로토콜을 OpenSSL에 결합하는 오픈소스 프로젝트로 그 중 래티스(lattice) 기반 키 교환 프로토콜은 총 5 가지의 Frodo, Kyber, BCNS, MSrln, NewHope가 있다. 본 논문에서는 OQS의 래티스 기반 키 교환 프로토콜을 구현 SW상에서 검증이 용이한 타이밍공격(timing attack)과 오류 주입공격(fault attack)의 부채널공격 관점에서 안전성을 검증하고 이러한 부채널공격에 대하여 안전하게 구현할 수 있는 방법을 제시한다.

I. 서론

현재 키 교환 과정에서 널리 사용하고 있는 DH(Diffie-Hellman), ECDH(Elliptic Curve Diffie-Hellman) 키 교환 프로토콜은 이산 대수 문제 및 소인수 분해 문제의 어려움에 기반하기 때문에 Shor 알고리즘[1]에 따라 양자컴퓨터의 공격에 다항식 시간에 해독이 가능하다. 이에 NIST는 2017년 11월까지 양자 컴퓨터를 이용한 공격에도 안전한 포스트 양자 암호(Post Quantum Cryptography, PQC)를 공모하고 있으며, 그 중 이미 래티스(lattice) 기반 암호 프리미티브가 많이 있다.

1) 이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원(No.2017-0-00555, 양자 컴퓨터 환경에서 래티스 문제를 이용한 다자간 인증키교환 프로토콜 연구)과 2017년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. NRF-2015R1A2A2A01006812)

본 논문에서는 Open Quantum Safe(OQS)[3, 4] 프로젝트에 대하여 설명하고 구현 SW 상에서 검증이 용이한 타이밍공격(timing attack)과 오류 주입공격(fault attack) 관점에서 분석하고자 한다. OQS 프로젝트에서 래티스 기반 키 교환 프로토콜은 총 5 종류인 Frodo[5], Kyber[6], BCNS[7], MSrln[8], NewHope[9]가 있으며 각각에 대하여 부채널 공격 여부와 개선책을 논의하고 한다.

1.1 논문의 구성

본 논문의 구성으로 Ⅱ장에서는 널리 알려진 LWE(Learning-with-Errors), Ring-LWE, Module-LWE 문제에 대하여 간략히 설명하고, Ⅲ장은 OQS 프로젝트에 대하여 소개하고, Ⅳ장에서는 OQS 프로젝트 중 5개의 래티스 기반 키 교환 프로토콜에 대한 타이밍공격 및 오류주입공격과 개선책에 대하여 논하고, Ⅴ장에서는 결론과 본 논문 성과를 확장하여 메모리 취약점을 통한 부채널공격에 대한 가능성을 제시한다.

II. 배경지식

2.1 LWE 문제

LWE는 기계학습에서 시작된 문제로, 패리티(parity) 학습 문제의 일반화된 형태이다. LWE는 평균적인 래티스 문제가 가장 최악의 경우(worst case)만큼 어렵기 때문에 공개키 암호 등의 설계에 많이 사용된다. LWE 분포 $A_{s,\chi}$는 벡터 $s \in Z_q^n$이 주어졌을 때, 임의의 $a \in Z_q^n$를 균일하게 선택하고 에러 e는 분포 χ에서 추출하고, $(a, b = < s, a > + e \bmod q)$를 계산한다. 래티스 기반 암호에서 주로 쓰이는 결정문제(decision problem)는 $A_{s,\chi}$로부터 선택한 m개의 주어진 샘플 (a_i, b_i)과 $Z_q^n \times Z_q$로부터 선택한 m개의 임의의 샘플 (a_i, b_i)간에 구별이 어려움을 이용한다.

2.2 Ring-LWE 문제

Ring-LWE는 LWE 문제를 유한체(finite field) 상의 다항식 환(ring) 위에서 정의한 문제이다. Ring-LWE 문제의 해는 래티스의 NP-Hard 문제인 SVP(Shortest Vector Problem)으로 환원 될 수 있다. Ring-LWE 분포 $A_{s,\chi}$는 $s \in Z_q^n$이 주어져 있을 때, 임의의 $a \in Z_q^n$를 균일하게 선택한다. 에러 e를 분포 χ에서 추출하고, $Z_q[x]/\Phi(x)$ 위에서 $(a_i, b_i = a_i(x)s(x) + e_i(x))$ 계산한다. Ring-LWE의 결정문제는 $A_{s,\chi}$로부터 선택한 m개의 독립 샘플 (a_i, b_i)과 $Z_q[x]/\Phi(x)$로부터 임의의 m개의 샘플 (a_i, b_i)을 구별하는 것이다.

2.3 Module-LWE 문제

Module-LWE[10]는 LWE와 Ring-LWE를 일반화한 것이다. LWE 분포에서 a_i는 $(Z_q[x]/\Phi(x))^k$로부터 균일하게 추출하고, s는 β_η^k로부터 추출한다. e_i는 β_η로부터 새롭게 추출한다. 여기서 β_η는 이항분포이다. 결정문제는 $A_{s,\chi}$로부터 선택한 m개의 독립 샘플 (a_i, b_i)과 $(Z_q[x]/\Phi(x))^k \times \beta_\eta$로부터 임의의 m개의 샘플 (a_i, b_i)을 구별하는 것이다.

III. Open Quantum Safe 프로젝트

OQS는 포스트 양자 암호 중 키 교환 프로토콜에 대하여 오픈소스로 진행하고 있는 프로젝트이다. 현재 Stebila 등에 의하여 OpenSSL에 PQC 프로토콜을 추가할 예정이다. OQS 프로젝트에 총 9 종류의 키 교환 프로토콜이 있으며, 이 중 코드 기반 1 종류, 래티스 기반 5 종류, SIDH (Supersingular Isogeny Diffie-Hellman) 기반 2 종류, NTRU로 구성되어 있다. 본 논문에서는 5 종류의 래티스 기반 키 교환 프로토콜에 대하여 분석하였다.

3.1 Frodo[5]

Frodo는 Bos 등이 설계하고 구현한 프로토콜로 LWE 문제를 기반으로 설계하였다. 파라미터는 $n = 752$, $q = 2^{15}$이다. 타 프로토콜과 비교하여 NTT(Number Theoretic Transform)나 FFT(Fast Fourier Transform)를 쓰지 않은 것이 특징이다.

3.2 BCNS[6]

Bos 등에 의해 2015년 발표된 프로토콜로 Ring-LWE 문제를 기반으로 한다. 파라미터 $q = 2^{32} - 1$로 비교적 큰 편이다. 연산과정에서 FFT 알고리즘을 사용한다.

3.3 NewHope[7]

NewHope 프로토콜은 Alkim 등에 의해 BCNS 프로토콜의 단점을 보완하여 설계 및 구현하였다. $n = 1024$, $q = 12289 < 2^{14}$까지 파라미터의 크기를 줄인 것이 특징이다. NTT 연산에서 AVX2(Advanced Vector Extensions)을 사용하기 때문에 프로토콜의 속도가 증가하였다. NewHope 프로토콜에서는 BCNS의 에러 샘플링 분포를 이산 가우시안 분포에서 이항분포로 변경하였다.

3.4 MSrln[8]

Longa 등에 의해 제안된 Ring-LWE 기반 프로토콜이며, NTT 연산을 사용하여 연산속도가 증가하였다. 또한 새로운 모듈러 감쇄 알고리즘을 제안하였다. 수정된 NTT 알고리즘은 전체적

[표 1] OQS 오픈소스 프로젝트의 비교

항목	Frodo	BCNS	MSrln	NewHope	Kyber
기반 문제	LWE	Ring-LWE	Ring-LWE	Ring-LWE	Module-LWE
연산 알고리즘	-	FFT[1]	NTT[2]	NTT[2]	NTT[2]
에러 샘플링	연속 가우시안 분포	이산 가우시안 분포	이산 가우시안 분포	이항 분포	이항 분포

1) FFT(Fast Fourier Transform)
2) NTT(Number Theoretic Transform)

인 NTT의 연산 속도를 증가하여 프로토콜의 전체적인 부분에서 속도를 향상 시킬 수 있다. 연산속도가 동일한 C 환경에서 약 1.44배, AVX2 환경에서는 약 1.21배로 증가한다.

3.5 Kyber[9]

Kyber는 Bos 등에 의해 2017년 발표된 Module-LWE를 기반으로 한 KEM(Key Encapsulation Mechanism)에 프로토콜로 CCA(Chosen Plaintext Attack)에 대해 안전하며 NewHope 프로토콜과 같이 NTT 알고리즘과 AVX2 사용하기 때문에 빠른 NTT 연산 속도를 보여준다. 다만, 파라미터 n과 q의 값이 각각 $n=256$, $q=7681$로 구현상의 효율성을 위해 API를 통합적으로 사용한다.

IV. 타이밍과 오류주입공격

이번 장에서는 OQS 프로젝트의 래티스 기반 키 교환 프로토콜의 부채널공격 측면에서 분석한다. [표 1]에서 본 논문에서 분석한 OQS 프로젝트 프로토콜을 간단히 비교하였다.

4.1 연산 알고리즘

각 프로토콜은 안전한 연산과 빠른 연산 속도를 위하여 효율적인 연산 알고리즘을 사용한다. BCNS는 FFT를 사용하고 MSrln, NewHope. Kyber는 NTT를 사용한다. FFT 및 NTT에서의 연산은 입력값과 출력값이 달라짐에 따라 연산 시간의 차이가 생기지 않기 때문에 타이밍공격이 어렵다. LWE 및 Ring-LWE를 기반으로 하는 키 교환 프로토콜 연산과정 중, $b=as+e$ 에서 s와 e가 달라져도 전체 연산시간은 동일하기 때문에 타이밍공격을 시도하기 어렵다.

한편, 부채널공격 중 오류주입공격에는 모든 프로토콜이 취약하다. MSrln 프로토콜의 경우, $b=as+e$를 계산하는 과정이 as를 계산한 이후에 e를 더한다. 이 과정에서 e를 더하는 과정을 건너뛰거나 공격자가 주입하는 e를 임의로 바꾼다면 개인키 s를 가우스 소거법(Gaussian elimination)에 의하여 쉽게 구할 수 있다. 그러나 SW 상으로 구현되었을 경우 연산 과정이 분리되어 있을지라도 오류를 주입하는 것은 어려우나, HW로 구현된 경우에는 앞서 설명한 바와 같이 오류를 주입하여 개인키를 쉽게 얻을 수 있다.

$b=as+e$과정에서 [그림 1]과 같이 곱셈과 덧셈으로 이루어진 연산을 하기 위하여 2가지의 FFT API를 사용한다. [그림 2]에서는 이것을 하나로 통합하여 1개의 API로 구현하였다. 제안하는 API를 사용한다면, 오류주입공격에 대하여 안전하다.

NTT에서도 위와 같은 통합 API 구현으로 오류주입공격에도 안전한 래티스 기반 키 교환 프로토콜을 구현할 수 있다.

```
void oqs_kex_rlwe_bcns15_fft_mul(uint32_t z[1024], const uint32_t x[1024],
const uint32_t y[1024], struct oqs_kex_rlwe_bcns15_fft_ctx *ctx) {
        nussbaumer_fft(z, x, y, ctx);
}

void oqs_kex_rlwe_bcns15_fft_add(uint32_t z[1024], const uint32_t x[1024],
const uint32_t y[1024]) {
        int i;
        for (i = 0; i < 1024; i++) {
                add(z[i], x[i], y[i]);
        }
}
```

[그림 1] BCNS의 FFT 곱셈 및 덧셈 API

```
void oqs_kex_rlwe_bcns15_fft_muladd(uint32_t z[1024], const uint32_t x
[1024], const uint32_t y[1024], const uint32_t w[1024], struct
oqs_kex_rlwe_bcns15_fft_ctx *ctx) {
        int i;
        const uint32_t c[1024];

        nussbaumer_fft(c, x, y, ctx);

        for (i = 0; i < 1024; i++) {
                add(z[i], c[i], w[i]);
        }
}
```

[그림 2] 개선된 통합 FFT 연산 API

4.2 에러 샘플링

LWE 및 Ring-LWE 기반 키 교환 프로토콜은 충분히 랜덤한 에러를 추출하여 더하기 때문에 안전성이 확보된다. Frodo는 연속 가우시안 분포, BCNS, MSrln은 이산 가우시안 분포, NewHope와 Kyber는 이항분포를 사용한다.

에러 추출시 난수는 ChaCha20[11], AES의 CTR모드를 사용하여 생성한다. 이는 현재 OpenSSL에서 사용하고 있는 난수 생성과정과 일치한다. 5 종류의 프로토콜을 확인해 본 결과, 모든 프로토콜은 위와 동일하게 난수 생성을 하고 있으며, 실제 구현상의 문제도 존재하지 않았다. 따라서 에러 샘플링 과정은 안전하다는 것을 확인할 수 있었다.

V. 결론 및 향후 연구

본 논문에서는 NIST의 PQC 공모사업의 일환으로 오픈소스로 구현된 OQS 프로젝트에 대하여 소개하였다. OQS 프로젝트는 Frodo, BCNS, MSrln, NewHope, Kyber의 래티스 기반 키 교환 프로토콜이 있으며, 연산 알고리즘과 에러 샘플링 과정에서의 부채널공격에 대하여 분석하였다. 결과적으로 FFT와 NTT 과정에서 오류주입 공격이 가능한 것을 확인하였다. 이에 대한 개선책으로 FFT 연산시 덧셈과 곱셈으로 분리된 API를 통합된 API로 구현하였다.

향후 연구 과제로는 OQS 프로젝트에 포함되지 않은 Lizard[12]에 대한 부채널공격 가능성과 메모리관련 취약점도 확인하여 오류주입 없이 개인키를 추출하는 공격 등을 검증할 예정이다.

[참고문헌]

[1] P. W. Shor, “Algorithms for Quantum Computation: Discrete Logarithms and Factoring,” in Proceedings of the Foundations of Computer Science, 35th Annual Symposium on IEEE, 1994, pp. 124-134.

[2] https://csrc.nist.gov/Projects/Post - Quantum-Cryptography

[3] D. Stebila and M. Mosca, “Post-Quantum Key Exchange for the Internet and the Open Quantum Safe Project”, Cryptology ePrint Archive, Report 2016/1017, 2016.

[4] https://github.com/open-quantum-safe/

[5] J. W. Bos, C. Costello, L. Ducas, I. Mironov, M. Naehrig, and V. Nikolaenko. “Frodo: Take off the ring! Practical, quantum-secure key exchange from LWE.” In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 1006-1018.

[6] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, M. Schanck, and D. Stehlé, “CRYSTALS - Kyber: a CCA- secure module-lattice-based KEM”, Cryptology ePrint Archive, Report 2017/634, 2017.

[7] J. W. Bos, C. Costello, M. Naehrig, and D. Stebila. “Post-quantum key exchange for the TLS protocol from the ring learning with errors problem.” Security and Privacy (S&P), 2015 IEEE Symposium on. IEEE, 2016, pp.553-570.

[8] P. Longa and M. Naehrig. "Speeding up the number theoretic transform for faster ideal lattice-based cryptography", in International Conference on Cryptology and Network Security 2016, pp. 124-139.

[9] E. Alkim, L. Ducas, T. Pöppelmann and P. Schwabe, “Post-quantum Key Exchange - A New Hope”, In USENIX Security Symposium 2016, pp. 327-343.

[10] M. Albrecht and A. Deo, “Large Modulus Ring-LWE ≥ Module-LWE”, Cryptology ePrint Archive, Report 2017/612, 2017.

[11] Y. Nir and A. Langley. “ChaCha20 and Poly1305 for IETF Protocols”, No. RFC 7539. 2015.

[12] J. Cheon, D. Kim, J. Lee, and Y. Song, “Lizard: Cut off the Tail! Practical Post-Quantum Public-Key Encryption from LWE and LWR”, Cryptology ePrint Archive, Report 2016/1126, 2016.

IoT 환경을 위한 효율적인 그룹 인증 및 키 교환 기법[1)]

이대휘*, 이임영**

순천향대학교 컴퓨터학과*

순천향대학교 컴퓨터소프트웨어공학과**

Lightweight 3PAKE Scheme based on Lattice for IoT Environments

Dae-Hwi Lee*, Im-Yeong Lee**

Dept of Computer Science and Engineering, Soonchunhyang University*
Dept of Computer Software and Engineering, Soonchunhyang University**

요 약

IoT 환경에서는 많은 디바이스들이 서로 연결되어 데이터를 공유한다. 이 디바이스들은 그룹을 생성하여 그룹 내부에서 통신을 하거나, 외부의 다른 그룹과 통신을 할 수 있다. 그룹에서는 그룹 리더가 그룹 참여자들과 통신하게 되는데, 하나의 그룹에 참여자가 매우 많아진다면 안전한 통신을 위해 그룹 리더는 매우 큰 연산 오버헤드를 가지게 될 것이다. 여러 디바이스가 한 번에 통신을 요청하게 되면, 인증 및 키 교환 과정에서 세션의 수만큼 인증이 이루어져야 하기 때문이다. 본 논문에서는 IoT 환경에서 효율적인 그룹 통신을 위한 그룹 인증 및 키 교환 기법을 제안한다.

I. 서론

IoT 환경에서는 많은 디바이스들이 서로 연결되어 데이터를 공유한다. IoT 환경에서 통신에 참여하는 디바이스들 중 게이트웨이급 또는 서버급 디바이스들은 연산 능력이 좋을 수 있지만, 대부분의 센서 디바이스는 연산 능력이 낮다. 따라서 내부에서 동작하는 인증과 같은 보안 프로토콜은 낮은 연산능력을 가진 디바이스로도 충분히 작동을 할 수 있도록 경량화하여 설계되어야한다.

특히, WBAN(Wireless Sensor Network)와 같은 환경에서는 사용자의 신체에 부착된 센서 또는 스마트 워치 등의 디바이스들을 통해 데이터를 수집하여 스마트폰이라는 게이트웨이 역할을 수행하는 디바이스로 전송한다. 스마트폰에서는 수집한 정보를 기반하여 사용자의 건강 정보와 같은 서비스를 제공할 수 있다. 이와 같은 환경에서는 스마트폰과 같은 데이터를 수집하고 처리하는 디바이스가 연결된 센서 디바이스들을 연계하여 하나의 소규모 디바이스 그룹을 형성한다. 이 소규모 그룹은 서비스 제공자와 통신을 하여 사용자 맞춤형 서비스를 제공받거나, 다른 저장장치나 디바이스의 그룹과 통신할 수 있다. 스마트 홈에서도 마찬가지로, 하나의 스마트 홈은 하나의 그룹이 될 수 있으며, 서비스 제공자와 통신하여 그룹에 대한 업데이트를 수행하거나 다른 그룹인 이웃집과 통신할 수 있고, WBAN 환경에서 하나의 사용자 디바이스 그룹과 통신을 수행할 수도 있다. 하지만, 그룹간 통신을 수행할 때 하나의 그룹에 참여하고 있는 디바이스의 수가 많아지거나, 여러 디바이스가 동시에 통신을 수행해야 한다면, 안전한 통신을 위해서는 각 디바이스간 세션의

1) 본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 대학ICT연구센터육성 지원사업의 연구결과로 수행되었음
(IITP-2017-2015-0-00403)

수만큼 인증을 수행한 후 키 교환 과정을 거쳐 통신해야 한다. 이러한 환경에서 사용될 수 있는 기술은 그룹 인증 기술이다. 그룹 인증 기술은 그룹 단위 통신에서 매우 효율적인 인증 방법을 제시한다.

II. 관련연구

2.1 Group Authentication

IoT 네트워크에서는 많은 사용자와 디바이스들이 집중화되지 않고 분산되어있으며, 분산된 환경에서의 인증, 접근제어, 식별관리에 대한 문제에 직면하였다. 보안 솔루션, 이기종 통신, 리소스 제약과 분산 환경이라는 많은 과제들이 남아있다. 특히, 인증과 식별 관리는 IoT에서는 매우 중요한 과제이며, 이것은 간단하고 안전하고 빠르게 수행되어야 한다.

또한, 분산된 환경에서 여러 개의 디바이스들이 하나의 그룹 단위가 되었 으며, 안전한 그룹 통신을 위한 기술들도 제안되고 있다. 그룹 인증 기술은 그룹 리더가 그룹 내의 디바이스들과 1:N로 통신하는 환경에서 그룹 참여자인 사용자 또는 디바이스가 동일한 그룹에 속하는지 여부를 판별하고 인증을 수행할 수 있으며, 그룹 간 통신인 N:N의 환경에서는 그룹 간에 한 번의 인증만을 수행하여 안전한 통신을 수행할 수 있도록 도와주는 기술이다. 각 방식은 Intra-group Authentication, Group Authentication으로 불리운다. 일반적으로 하나의 그룹에는 하나의 그룹 리더가 결정되어 있으며, 그룹 내부의 기기가 통신 요청을 받거나 요청할 경우 인증 절차를 수행하게 된다.

기존에 제안된 그룹 인증 기법들은 비밀 분산 기술인 threshold 암호를 기반으로 그룹을 인증하였고, 이를 통해 그룹 키 동의까지 수행하였다. 그룹 인증은 단순히 그룹에 속한 참여자라는 것에 대한 인증을 할 뿐만 아니라, 계층형 구조를 가지는 그룹에서는 그룹 키를 이용하여 그룹 간 인증을 수행하는 방법도 제안되었다.

III. 보안요구사항

- 인증 : 그룹 참여자는 그룹 리더를 통해 그룹에 참여하고 있는지에 대한 여부와 정당한 사용자인지를 인증해야 한다.

- 재전송 공격 방지 : 인증과 키 교환에 사용되는 중간 값이 공격자에게 노출되어도 공격자는 값을 그룹 리더에게 재전송하여 정당한 사용자로 인증받을 수 없어야 한다.

- 효율성 : 그룹의 규모가 커질수록 그룹 리더에 대한 연산 오버헤드가 커진다. 따라서, 그룹 인증 및 키 교환 과정에 있어서 그룹 리더 측면에서의 효율성을 제공해야 한다.

IV. 제안방식

4.1 가입 단계

Step 1. 그룹 리더는 비밀키 x를 생성한다.

Step 2. 그룹 참여자인 노드 i는 그룹 참여를 위해 자신의 식별자인 ID_i를 그룹 리더에게 전송한다.

Step 3. 그룹 리더는 노드 i가 그룹에 이미 참여한지 확인을 위해 노드 식별자 기반의 비밀키 $b_i = h(x \| ID_i)$를 계산하고, b_i가 블룸필터 B에 포함되어있는지 확인한다. 만약 b_i가 이미 블룸필터 B에 포함되었다면, 요청 메시지를 폐기한다.

Step 4. i가 신규 노드일 시 그룹 비밀 값 $s = s_o \| s_1 \| \cdots \| s_{n_p - 1}$를 생성한다.

$$(s \in 0,1^{d(n_p-1)}, s_0 = 0^d)$$

Step 5. 그룹 리더는 각 노드들의 분산 비밀 값 생성을 위한 2차원 난수 배열 $r_j^i = rand(0,1^d)$를 생성한다.

$$(0 \le i \le k-2, 0 \le j \le n_p - 1)$$

Step 6. 그룹 리더는 난수 배열 r_j^i를 이용하여 $w_{(i,j)} = (\oplus_{h=0}^{k-2} r_h^{h \cdot i+j}) \oplus s_{j-i}$를 계산한다.

Step 7. 그룹 리더는 각 노드 i에 대한 분산 비밀값인 $w_i = w_{(i,0)} \| \cdots \| w_{(i,n_p-2)}$ 를 계산한다. w_i가 임계값인 k개가 모이면 그룹 리더는 그룹 비밀 값인 s를 복구할 수 있다.

Step 8. 그룹 리더는 그룹에 참여하고 있는 각 노드들에게 $(w_i \| b_i)$를 전송하고, 노드들은 각 값을 저장한다.

4.2 인증 요청 단계

Step 1. 노드 i는 세션 그룹키 생성을 위한 요청 값인 임의의 r를 생성한다.

Step 2. 노드 i는 가입 단계에서 저장한 노드의 비밀키인 b_i를 이용하여 $E_{b_i}(w_i \| ID_i \| r)$을 생성하고 그룹 리더에게 전송한다.

Step 3. 그룹 리더는 노드 i의 그룹 인증 요청 신호를 수신하고, 수신한 값을 복호화하여 $b_i = h(x \| ID_i)$를 생성한다. 그리고, b_i가 블룸필터 B에 포함되었는지 확인한 후, 그룹 인증에 참여할 노드들에게 참여 대기 신호 및 r을 송신한다.

Step 4. 각 노드들은 자신의 비밀키로 암호화한 $E_{b_n}(w_n \| ID_n \| r), ID_n$을 그룹 리더에게 전송한다. 이 때, 비밀값 s를 복원하기 위해서는 임계값 k개 이상의 노드가 참여해야 한다.

4.3 그룹 인증 및 키 교환 단계

Step 1. 그룹 리더는 수신한 노드들의 비밀키 $b_i' = h(x \| ID_i)$를 계산하고, 각 b_i'에 대하여 $b_i' \in B$ 여부를 확인한다. 이후, 그룹에 포함된 노드의 정보라면 암호화된 값을 복호화하고 r 값을 확인하여 유효성을 검증한다.

Step 2. 그룹 리더는 k개 노드들로부터 수신한 $w_{t_0}, w_{t_1}, \ldots, w_{t_{k-1}}$ 집합을 이용해 $w_{t_i} = w_{(t_i,0)} \| \cdots \| w_{(t_i,n_p-2)}$ 배열을 계산한다.

Step 3. 그룹 리더는 비밀 값 복구를 위한 배열 w를 다음과 같이 계산한다.

$$w = (w_{(t_0,0)}, \cdots, w_{(t_0,n_p-2)}, \cdots, w_{(t_{k-1},0)}, \cdots, w_{t_{k-1},n_p-2)})^T$$

Step 4. 그룹 리더는 다음과 같이 행렬 M을 계산한 후 $M \bullet w$를 계산하여 나온 행렬을 통해 비밀값 s를 복구한다. 위 계산은 Kurihara 기법[5]의 연산을 사용한다.

$$M = MAT(t_0, \ldots, t_{k-1})$$
$$M \bullet w = (s_1, \ldots, s_{n_p-1})^T$$
$$s = (s_1 \| \cdots \| s_{n_p-1})$$

Step 5. 그룹 리더는 그룹 세션키 $SK = h(s \| r)$를 생성한다.

Step 6. 그룹 리더는 생성한 세션키의 분배를 위해 n개 노드들에게 $E_{w_n}(SK \| r)$를 전송하고, 수신받은 노드들은 분산 비밀값을 통해 복호화하여 r을 확인 후 인증하여 최종적으로 세션키 SK를 획득한다.

V. 제안방식 분석

본 제안방식은 3장에서 도출한 보안요구사항을 다음과 같이 해결한다.

- 인증 : 그룹 참여자는 그룹 리더에게 식별자와 대칭키를 통한 정보를 전송하여 카운팅 블룸필터에 포함되었는지 여부를 판단하여 인증을 수행한다. 이를 통해 사전에 가입 단계를 거친 사용자임을 확인할 수 있다.

- 재전송 공격 방지 : 공격자는 인증 단계에서 비밀키 를 소유하고있지 않으므로 공격자가 생성한 을 이용해 전송 요청하는 등의 공격을 수행할 수 없다. 또한, 인증 요청 단계와 그룹 인증 및 키 교환 단계에서 전송되는 중간 값을 공격자가 가로채어 재전송하여도, 공격자는 분산 비밀값 를 소유하고 있지 않으므로 세션키 를 획득할 수 없다.

- 효율성 : 그룹 인증 및 키 교환시 그룹 참여자들의 정보를 받아 그룹 비밀값 를 연산하는 과정에서 XOR 연산을 기반으로 계산하기 때문에, 기존 공개키 비밀분산을 이용한 비밀 복원 과정들에 비해 그룹 리더 측면에서 효율성을 제공할 수 있다.

VI. 결론

본 제안방식에서는 그룹 리더에서 키 동의를 위해 대칭키 방식의 인증을 수행하고, XOR 연산 기반의 그룹 키를 생성하여 그룹 인증 및 키 동의를 수행하는 과정을 제안하였다. 그룹에 참여하고 있는 디바이스의 수가 많아질수록 그룹 리더에서 발생하는 연산 오버헤드는 매우 커지므로, 제안방식은 기존의 Threshold 기반의 그룹 인증 기법에 비해 그룹 리더에 대한 연산 오버헤드가 낮다.

그룹 인증은 IoT 환경의 발달로 최근 많은 연구가 진행되고 있는 분야로, 좀 더 다양한 환경에 맞도록 지속적인 연구가 진행되어야 한다. 또한, 이후에는 그룹 리더의 메모리에 대한 오버헤드를 줄이고, 디바이스 측면에서의 연산량을 줄이는 방향으로 연구가 필요할 것으로 판단된다.

[참고문헌]

[1] P. N. Mahalle, N. R. Prasad, and R. Prasad, "Threshold cryptography-based group authentication (TCGA) scheme for the internet of things (IoT)," Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems, 2014 4th International Conference on. IEEE, pp. 1-5, 2014.

[2] W. T. Su, W. M. Wong, and W. C. Chen, "A survey of performance improvement by group-based authentication in IoT," Applied System Innovation (ICASI), 2016 International Conference on. IEEE, pp. 1-4, 2016.

[3] L. Harn, "Group authentication," IEEE Transactions on computers, Vol. 62, No. 9, pp. 1893-1898, 2013.

[4] J. J. Huang, W. S. Juang, C. I. Fan, Y. F. Tseng, and H. Kikuchi, "Lightweight authentication scheme with dynamic group members in IoT environments," Adjunct Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing Networking and Services, pp. 88-93, 2016.

[5] J. Kurihara, S. Kiyomoto, K. Fukushima, and T. Tanaka, "A new (k, n)-threshold secret sharing scheme and its extension," Information Security, pp. 455-470, 2008.

[6] S. Bellovin, and M. Merritt, "Encrypted key exchange: password-based protocols secure against dictionary attacks," IEEE Symposium on Research in Security and Privacy, pp. 72-84, 1992.

[7] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," EUROCRYPT 2000, Lecture Notes in Computer Science, Vol. 1807, pp. 139-155, 2000.

최신 PKI기반 V2X보안 기술 동향분석[1)]

이정호*

*한국정보인증

Case study of PKI based V2X Security

Jung-Ho Lee*

*Korea Information Certificate Authority Inc.

요 약

초연결(Hyperconnectivity)을 통해 다양한 산업분야의 기술이 융합되는 '4차 산업혁명'이 이슈가 되고 이를 통해 여러 산업분야에 걸쳐 영향을 끼치고 있다. 그리고 자동차 산업시장은 IT와의 융합을 통해 자율주행이라는 새로운 시장을 만들고 있다. 과거 자동차는 단순 이동수단으로써의 기능을 하였지만 현재 자동차는 다양한 IT기술과의 융합을 통해 운전자에게 운전자보조시스템(Advanced Driver Assistance System, ADSA)과 같은 편의기능을 제공해주게 되었다. 또한 외부 네트워크와의 통신으로 교통정보와 같은 운전자가 필요로 하는 정보를 자동차에서 얻을 수 있게 되었다. 하지만 과거 폐쇄적 환경만 고려된 자동차 IT기술은 외부 네트워크와의 연결로 기존 IT환경의 보안위협에 노출되어 운전자의 안전이 위협받게 되었다. 이러한 보안 위협에 대응하기 위한 방법의 하나로 자동차와 통신기기와의 통신기술인 V2X(Vehicle To Everything)는 미국, 유럽 등 다양한 국가에서 보안 대책이 연구되고 있다. 본 논문에서는 V2X보안기술 중 PKI 인증체계를 이용한 V2X 보안기술 동향에 대해 알아본다.

I. 서론

수많은 전자기기들은 인터넷에 연결되어 다양한 정보를 공유하고 새로운 정보를 만들어 사용자에게 편의를 제공하고 있다. 자동차 산업에서도 운전자에게 다양한 편의를 제공하기 위해 자동차의 전자화와 V2X기술을 최근 출시되는 자동차에 적용하고 있다. 더욱이 미래에 나오게 될 자율주행차량의 경우 운전자의 조작이 없이 자동차에 탑재된 전자제어 부품들과 통신기술을 통해 운행되게 될 것이다.

하지만 자동차의 전자화와 외부와의 통신으로 인하여 운전자는 기존 IT환경이 가지는 다양한 보안위협에 노출되게 되었다. 특히 자동차에서의 보안위협은 운전자의 생명에 위협이 되기 때문에 기존 IT환경이 가지는 보안 위협보다 파급력은 커질 수 있다. 만약 자율주행차량이 출시가 되고 외부의 정보를 받아보는 V2X기술도 같이 적용되면 차량을 대상으로 하는 해킹은 지금보다 많아지게 될 것이고, 운전자의 안전을 위협하는 행위도 증가하게 될 것이다.

운전자의 안전을 위한 자동차보안 연구는 다양한 크게 2가지 관점으로 연구되고 있다. 첫째로 자동차를 전자적 제어를 보조하기 위한 ECU(Electronic Control Unit)와 ECU 간 통신네트워크(CAN, FlexRay, MOST, Lin 등)에서의 악의적 행위를 차단하기 위한 차량 내부 네트워크보안과 둘째로 차량외부의 정보를 송수신하기 위한 차량 외부 네트워크 보안으로 분류할 수 있다.

본 논문에서는 V2X를 통한 차량의 외부 네트워크의 통신 신뢰성을 확보하기 위한 자동차 외부네트워크 보안방법 중 하나인 PKI를 통한 V2X보안 기술동향에 대해 확인하고자 한다.

1) 이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No. B0717-16-0097.자율주행차량을 위한 V2X 서비스 통합 보안 기술 개발)

II. 배경지식

2.1 PKI 인증체계

PKI(Public Key Infrastructure) 인증체계는 공개키를 이용하여 사용자간의 통신 신뢰성을 확보하는 하는 기술이다. 국내의 경우 PKI 인증체계를 이용한 공인인증체계는 인증서를 이용하여 인터넷상에서 인터넷 뱅킹과 같은 서비스 이용 시 사용자 인증에 사용이 된다. 국내에서 사용되는 공인인증체계의 구조는 다음 그림 1과 같다.

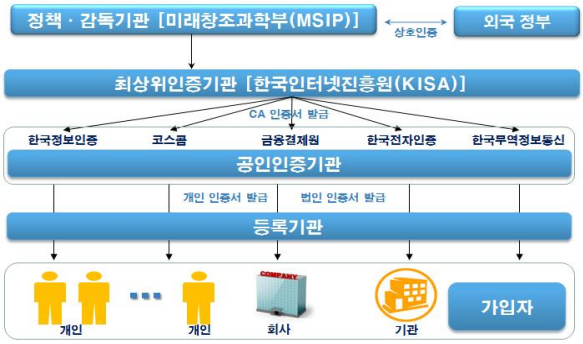


그림 1. 공인전자서명인증체계 구조

PKI 인증체계 구조는 계층적 구조로 이루어져 있으며 최상위 인증기관인 Root CA는 하위의 인증기관의 인증서를 발급한다. 인증기관에서는 인증기관의 인증서로 사용자가 사용하는 인증서에 서명하여 발급한다. 이렇게 계층적인 인증체계를 구성하는 이유는 신뢰하는 인증기관에서 서명한 인증서를 발급하여 다른 사용자에게 신뢰성을 확보하기 위해서이다. PKI 체계를 이용한 통신은 통신에 참여하는 객체들의 메시지의 신뢰성을 확보할 수 있다.

2.2 V2X(Vehicle To Everything)

V2X는 V2V(Vehicle to Vehicle), V2N(Vehicle to Nomadic Device), V2I(Vehicle to Infrastructure) 등 자동차와 다양한 통신기기간 통신을 의미한다. 최근 V2X는 자동차에서 다양한 정보를 수집하고 운전자에게 주행에 필요한 서비스를 제공해준다. 예를 들어 내비게이션 같이 도로의 주행정보를 수집하고 최단거리, 최단시간 등 최적의 주행정보를 운전자에게 제공해 준다.

향후 자율주행을 위한 V2X의 경우 자동차에서 1초에 10건 이상의 메시지를 생성하여 주변의 자동차와 통신을 하면서 주행에 필요한 정보를 파악할 것이다. 하지만 V2X의 통신 신뢰성이 확보되지 않으면 해커와 같은 악성사용자에 의해 공격되기 쉽다. 하지만 현재 V2X환경에서의 악성 행위자 및 부정메시지에 대한 정의는 연구 중이며, V2X환경에서 지켜져야할 자산을 식별하고 보호하는 기법에 대해 연구가 진행 되어야 한다.

III. Vehicular PKI 동향

3.1 IEEE 1609.2

IEEE 1609.2는 자동차가 다른 통신기기와 무선통신에서 사용해야할 보안 규격이다. V2X통신에서 사용되는 차량단말(OBU, On-Board Unit), 노변기기(RSU, Road Side Unite), IoT기기 등 무선통신을 할 수 있는 표준규격이 정의 되어 있다.

IEEE 1609.2는 PKI 인증체계를 기본으로 구성되어 있으며, 인증체계의 구조는 그림 3과 같다.

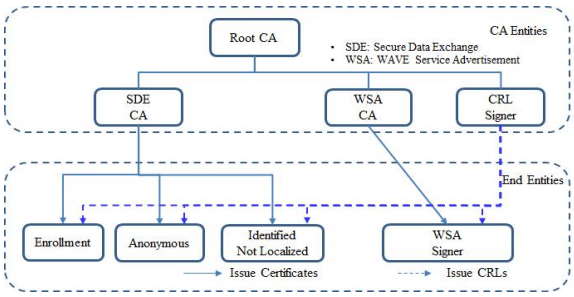


그림 3. IEEE 1609.2 인증체계 구조

IEEE 1609.2는 Root CA, SDE CA(Scure Data Exchange CA), WSA CA(WAVE Service Advritisment CA), CRL Signer로 구성되어 있다. Root CA는 CA의 인증서를 발급하고 CA에서는 종단 객체인 차량단말, 노변기기에 사용되는 서비스에 따라 사용되는 인증서를 발급한다.

3.2 SCMS(Security Credential Management System)

SMCS는 세계 유명 자동차 제조사가 V2X통신 보안을 위한 연구와 표준제정을 위해 만들어진 컨소시엄이다. SCMS에서는 인증서를 통한 V2X통신에 대한 메시지의 신뢰성을 확보하고 있다. 또한 익명인증서를 통해 운전자의 위치와 같은 프라이버시를 보호한다.

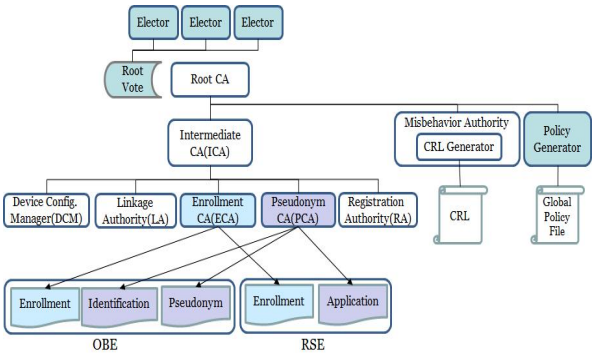


그림 4. SCMS 인증체계 구조

기본적인 PKI인증체계와 동일한 계층적인 구조를 가지고 있다. 자동차에서는 Enrollment Certificate(등록인증서)로 불리는 자동차 식별용 인증서를 발급받고 이를 이용하여 자동차 간 통신에서 사용할 인증서인 pseudonym Certificate(익명인증서)를 발급을 요청한다. 자동차에서는 익명인증서를 다량으로 발급받아 주기적으로 교체하면서 운전자의 위치를 추적하기 어렵게 만들어 프라이버시를 보호한다.

미국도로교통안전국인 NHTSA(National Highway Trafic Safety Administration)에서는 SCMS CV Safety Pilot Project라 불리는 프로젝트를 통해 탬파, 뉴욕, 와이오밍 3개의 주에서 V2X에 대한 적합성 및 현장에서의 테스트를 통해 관련기술들을 검증하는 과정을 이루고 있다.

3.3 Euro Certificate Policy

European Commission에서는 C-ITS (Cooperative Intelligent Transport Systems)에서 인증서를 이용한 보안통신을 위한 인증 정책을 PKI 인증체계 기반으로 유럽의 도로에서 사용되는 C-ITS 모델을 정의하고 있다. 또한 유럽에서 인증서를 발급하는 기관과 이를 사용하는 통신기기를 위한 관련 법률과 기술의 요구사항을 정의하고 있다.

유럽은 다수의 국가가 연계되는 Multipel Root CA 기반으로 이루어져 있으며 각 국가의 Root CA는 주기적으로 Root CA의 인증서를 공유한다. European Commission에서 정의한 V2X인증체계는 그림 5와 같다.

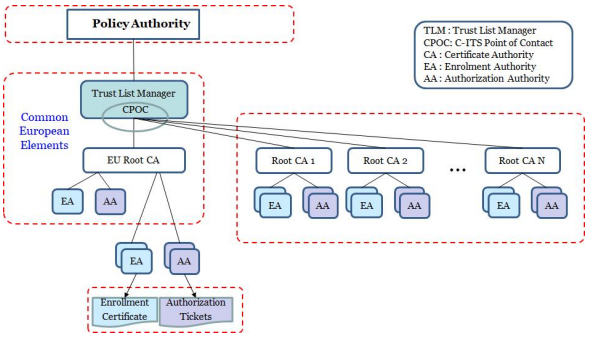


그림 5. European Commission의 인증체계 구조

유럽 C-ITS 신뢰모델을 위한 기관과 그에 대한 역할은 다음과 같다.

- 정책기관(Policy Autority)은 C-ITS 신뢰모델에 참여하는 공공 및 민간 관계자들로 구성되며 인증서에 대한 정책을 관리하며 PKI시스템의 전체적인 관리를 담당한다.
- TLM(Trust List Manager)은 정책기관이 지정한 고유한 객체이며, C-ITS 신뢰모델의 전반적인 보안 운영을 위한 유효한 인증정책에 따른 운영과 정책기관에 관한 정기적인 보고를 담당한다.
- C-ITS Point of Contact(CPOC)는 정책기관이 지정한 객체이며 C-ITS 신뢰모델의 모든 객체간의 보안통신과 다른 신뢰모델 객체가 제출한 절차 변경 요청 및 권고사항을 검토한다.

IV.결론

최근 자율주행에 대한 관심이 높아지고 자동차에서 다양한 정보들이 오고가고 있다. 하지만 다른 IT의 기술에 비해 자동차 보안에 대한 인식과 기술에 대한 연구는 부족한 현실이다. 자동차에서의 보안 위협은 재산상의 피해뿐만 아니라 인명피해를 동반한다. 그렇기 때문에 자동차 보안에 대한 연구와 표준이

끊임없이 이루어 져야 한다.

다양한 통신기기가 자동차와 통신하게 될 것이고 통신에 대한 신뢰성 확보는 무엇보다 중요하게 될 것이다. V2X는 미국, 유럽은 물론 현재 국내에서도 인프라를 구축하여, 국가적인 사업으로 이끌면서 자율주행차량을 위한 준비를 이루어지고 있다. 이와 같이 보안에 대한 연구도 동시에 이루어지고 있다. 본 논문을 통해 다음연구에서는 V2X에서 신뢰성이 확보된 통신기술에 대한 연구를 진행할 것이다.

[참고문헌]

[1] IEEE 1609.2-2013, IEEE Standard for Wireless Access in Vehcilular Enviroments Security Services for Application and Management Messages, 26, April, 2013.

[2] IEEE 1609.2-2016, IEEE Standard for Wireless Access in Vehcilular Enviroments Security Services for Application and Management Messages, 1, March, 2016.

[3] Department of Transportation(DoT) Highway Traffic Safety Administration 49 CFR Part 571 [Docket No. NHTSA-2016-0126] RIN 2127-AL55 Federal Motor Vehicle Safety Standards; V2V Communications, Notice of Proposed Rulemaking (NPRM).

[4] Security Credential Management System Proof of Concept Implementation, EE Requirements and Specifications Supporting SCMS Software Releas 1.1, 04, may, 2016.

[5] Certificate Policy for Deployment and Operation of European Coopeative Intelligent Transport Systems(C-ITS), European Commsion, 1, June, 2017

2017 한국정보보호학회 동계학술대회

E1 모바일 보안(I)

김 형 종 (서울여대)

- **공개된 채널환경에서의 NTRU 기반의 RFID 상호인증에 관한 연구**
  황용운, 이임영 (순천향대학교)
- **자마린에서 C#으로 개발된 안드로이드 앱에 대한 정적 역공학 분석**
  김규식, 심재우, 임경환, 조성제 (단국대학교)
- **Fuzzy Hash와 Dex 파일 분석 로그를 이용한 안드로이드 악성 애플리케이션 유사도 비교**
  조하은, 천호진, 정수환 (숭실대학교)
- **EU GDPR 사례를 통한 바이오정보보호 개선 방안**
  정부금, 권헌영, 임종인 (고려대학교)

공개된 채널환경에서의 NTRU 기반의 RFID 상호인증에 관한 연구+

황용운*, 이임영**

순천향대학교 컴퓨터학과

A Study on NTRU-based RFID Mutual Authentication for Insecure Channel

Yong-Woon Hwang†, Im-Yeong Lee**

Dept of Computer Science and Engineering, Soonchunhyang University*,**

요 약

RFID 기술은 다양한 분야 사용되고 있으며, 우리들 주변에서도 흔히 볼 수 있다. 그러나 RFID 기술에는 서버, 리더 및 태그 간 데이터 통신 시 공격자에 의한 스푸핑 공격, 태그 위조 공격 등 다양한 보안 위협이 발생한다. 이를 해결하기 위해 다양한 보안 기술이 존재하며 그 중 하나가 RFID 상호인증 기술이다. 현재까지 대칭키, 공개키 등 다양한 암호시스템을 활용하여 RFID 태그와 리더 사이의 보안 위협을 해결하였으며, 기존에 제시된 RFID 상호인증 대부분이 서버와 리더간의 통신이 안전하다는 가정하에 리더와 태그간의 상호인증에 중점을 두었다. 하지만 최근에 리더를 스마트폰이나 PDA와 같은 모바일 장치에 장착함으로써 서버와 리더간에 통신채널이 안전하지 않게 되었다. 이에 서버와 리더 태그가 공개된 채널에서의 RFID 상호인증을 다루는 방식들이 연구되고 있다. 본 논문은 공개키 암호를 사용하여 공개된 채널에서 안전하고 효율적인 RFID 상호인증 프로토콜을 제안할 것이다.

I. 서론

최근 정보통신기술의 발전으로 우리는 IoT(Internet of Things)라 불리는 시대에 살고 있으며, 이 시대에는 사물들이 가지는 센서로 인터넷을 통해 실시간으로 정보를 주고받아 대화를 나눌 수 있어 인간의 삶을 편리하게 한다. 수 많은 IoT 센서들 중 RFID 센서는 리더기(RF Module)를 통해 물품에 부착되어 있는 태그(Tag)를 식별할 수 있는 기술이다. RFID는 태그(Tag)와 리더(Reader), 서버(Back-end Server)로 구성 되는데, 리더를 통해 태그에 저장된 데이터를 가져와, 서버에 보내어 제어하고 처리한다. 최근 RFID 기술은 유통, 제조업, 물류, 국방, 교통 등 다양한 산업분야에서 적용되어 사용된다. 그러나 RFID 기술은 태그와 리더, 서버간의 통신시 데이터를 주고받는 과정에서 공격자에 의한 도청이나 스푸핑공격, 재전송공격, 서비스 거부 공격 등 다양한 공격이 발생할 수 있다. 이를 해결하기 위해 안전한 보안기술이 필요한데. 그 중 하나가 RFID 시스템에서 각 구성요소간의 상호인증기술이다. 현재까지 다양한 기법의 RFID 상호인증이 제안되었으며, 최근에는 ECC 암호와 같은 경량공개키 암호시스템 기반의 RFID 인증기술에 관한 연구가 진행되고 있다. 또한 기존에 제시된 RFID 상호인증 대부분이 리더와 태그간의 상호인증에 중점을 두었다면, 최근에 스마트폰이나 PDA와 같은 모바일 장치에 리더의 역할을 부여함으로써 서버와 리더간에 통신채널이 안전하지 않게 되었다. 이에 서버와 리더 태그가 공개된 채널에서의 RFID 상호인증을 다루는 방식들이 연구되고 있다.

본 논문에서는 공개된 채널에서 NTRU 암호

+ 이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2017-0-00156, 블록체인의 정보보호 프레임워크 및 평가 방법 개발)

를 사용하여 다양한 보안위협에 안전하고 기존에 제안된 RFID 상호인증 방식들과 비교해 효율적인 경량 RFID 상호인증 프로토콜을 제안한다.

II. 관련연구

본 장에서는 NTRU 암호기술에 대해 설명할 것이며, 기존에 제시된 RFID 상호인증 방식들에 대해 분석한다.

2.1 NTRU

NTRU 암호는 1996년 Jeffrey Hoffstein 등에 의해 제안된 다항식 환에서의 공개키 암호방식으로 키 생성이 쉬우며 속도가 빠르고 적은 양의 메모리를 요구한다. 기존 공개키 암호 RSA, ECC(Elliptic Curve Cryptography) 등과 비교하여 동일한 안전성을 제공하면서 암·복호화 속도가 빠르다는 이점을 갖는다. 또한 격자(Lattice) 이론을 기반으로 하기 때문에 양자컴퓨팅이 발전하더라도 기존의 RSA나 ECC와 같은 계산상의 복잡성에 기반을 둔 공개키 암호시스템에 비해 안전하기에 최근에도 NTRU 기반 상용기술에 대해 연구가 진행 중이다. NTRU 암호시스템은 다음과 같다[1].

· 키 생성 단계 : 초기 키 생성 단계는 먼저 작은 계수를 갖는 $N-1$차의 두 개의 다항식 $f, g \in R$를 선택한다. f는 $\bmod p$와 $\bmod q$상에서 역원이 존재해야 하며 이 역원은 f_p^{-1}, f_q^{-1}이며, f, g, f_p^{-1}, f_q^{-1}는 비밀키 공개키는 h이다.
  - $f_p^{-1} * f \equiv 1 (\bmod p)$
  - $f_q^{-1} * f \equiv 1 (\bmod q)$
  - $h = f_p^{-1} * g (\bmod q)$
· 암호화 : 메시지 $m \in R$에 대해 임의의 다항식 $\Phi \in R$를 선택하여 메시지 m을 암호화 한다.
  - $e = p * h * \Phi + m (\bmod q)$
· 복호화 단계 : 암호화 e로부터 메시지를 복호화하기 위해 개인키 f 를 곱하여 다항식 α를 구한다. α의 계수들은 $[-q/2, q/2]$ 범위 안에 있어야 하며, 개인키 f_p^{-1}를 다항식 α에 곱함으로써 메시지 m를 복호화 한다.
  - $\alpha \equiv f * e (\bmod q)$
  - $\equiv f * (p * h * \Phi + m) (\bmod q)$
  - $\equiv f * p * h * \Phi + f * m (\bmod q)$
  - $\equiv f * p * (f_q^{-1} * g) * \Phi + m (\bmod q)$
  - $\equiv p * \Phi * g + f * m) (\bmod q)$
  - $m' \equiv \alpha * f_p^{-1} (\bmod p)$

2.2 NTRU 기반의 RFID 상호인증 방식

현재까지 NTRU, ECC 암호기반의 경량 공개키 암호시스템 기반의 RFID 상호인증 기술이 계속해서 제안되었지만, 다양한 보안위협에 취약한 방식들이 존재하였다. 특히 NTRU 기반의 논문들 중 Shi 등은 2013, 2014년에 NTRU 기반의 RFID 상호인증방식을 제안하였는데, RFID 통신 중에 공격자가 도청을 통해 식별자를 추측할 수 있었으며, 이로 인해 다양한 보안위협이 발생하였다. 이를 Behzad 등이 프로토콜을 개선하여 보안요구사항을 만족시켰지만, 연산량 효율성이 Shi 방식에 비해 떨어졌다. 이후에 Samad 등이 NTRU 기반의 RFID 상호인증방식을 제안하여, 기존에 제안된 RFID 경량 상호인증방식에 비해 안전하다고 주장하였다.

III. 제안방식

본 장에서 NTRU 암호를 사용하여 다양한 보안위협에 안전하고 효율적인 RFID 상호인증 방식을 제안한다. 본 제안방식은 태그와 리더, 서버간의 통신이 안전하지 않는 공개된 채널에서 통신하는 것을 가정하였으며, 공개된 채널에서 각 구성요소간의 안전한 상호인증이 주요 목표이다.

3.1 시스템 계수

본 제안방식에서 사용되는 시스템 계수는 다음과 같다.

· $*$: 참여 객체(T: *Tag*, R: *Reader*, S: *Server*)
· K, K_n : 태그와 서버만이 공유하고 있는 이전 및 새로운 비밀키
· R_n, T_n : 리더와 태그에 의해 생성된 난수
· IDS, IDS_n : 태그의 이전 및 새로운 의사 아이디(index-pesudonym)
· h_S, f_S : 서버의 공개키와 개인키
· Φ : 암호화를 위한 난수

· ID_T, ID_S : 태그와 서버의 고유 식별자
· $Rot(A,B)$: A를 $w(B)$비트 왼쪽 회전, $w(B)$는 B의 해밍웨이트(Hamming weight)임

3.2 가정단계

이 단계는 RFID 각 구성요소가 인증하기전 필요한 정보를 공유하는 단계로서 태그는 h_S, ID_T, IDS, K를 서버는 f_R, ID_S, IDS, K를 가진다. 여기서 인증키 K와 태그의 식별자 ID_S, ID_T는 의사아이디 IDS에 대응되는 값이다. 태그와 서버는 가지고 있는 정보를 통해 공개된 채널에서 안전하게 상호인증을 수행한다.

3.3 인증단계

인증단계는 각 구성요소들이 인증정보를 주고받아 비교 검증함으로서 상대방이 정당함을 인증하는 단계이다.

Step 1. 리더는 난수 R_n를 생성하여 $Query$(질의 메시지)와 함께 태그에게 보낸다.

Step 2. $Query$에 반응한 태그는 난수 T_n를 생성하여 태그인증정보 m_1를 연산하고 NTRU 암호기술로 암호화하여 리더로부터 받은 R_n과 함께 리더로 보내준다.

- $m_1 = K \oplus T_n$
- $e(IDS \| m_1 \| K) = p * h_R * \Phi + m \pmod q$

Step 3. 리더는 태그로부터 받은 R_n를 통해 범위 내에 있는 태그임을 인지하고 태그로부터 받은 인증정보 서버의 데이터베이스에서 태그로부터 받은 e를 서버에게 보내준다.

Step 4. 서버로부터 받은 정보 e를 자신의 개인키 f_R로 복호화하여 IDS, m_1, K를 획득하고 자신의 DB에서 IDS에 대응하는 인증키 K와 식별자 ID_S를 선택한다. 그리고 태그로부터 받은 인증키 K와 비교하여 일치하면 정당한 태그임을 인증하고, 이후에 태그 인증정보 m_1를 통해 T_n

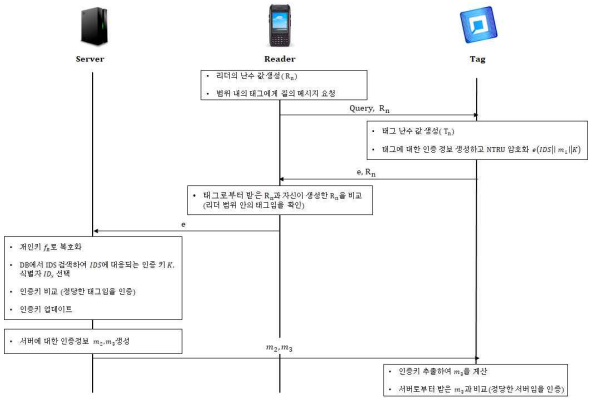


그림 1. 제안방식 상호인증 시나리오

를 추출한다. 또한 기존의 인증키 K를 새로 업데이트 한다.

- $f_R = e \pmod q = f_R * [e(IDS \| m_1 \| m_2)]$
- $Verify\, K = K$ (정당한 태그임을 인증)
- $T_n = m_1 \oplus K$
- $K_n = K$ (새로 생성된 키로 업데이트)

Step 5. 서버는 서버인증정보 m_2, m_3을 생성하여 리더를 통해 서버에게 보내준다.

- $m_2 = K_n \oplus T_n$
- $m_3 = Rot(ID_S + K_n, T_n)$

Step 6. 서버는 태그로부터 받은 m_2를 통해 새로운 인증키 K_n를 추출하고, m_3'를 연산하여 서버로부터 받은 m_3과 비교함으로써 정당한 서버임을 인증한다. 인증키 K_n은 다음 통신 때 계속 사용된다.

- $K_n = m_1 \oplus T_n$
- $m_3' = Rot(ID_T + K_n, T_n)$
- $Verify\, m_3 = m_3'$ (정당한 서버임을 인증)

IV. 제안방식 분석

본 장에서는 제안한 NTRU 기반의 RFID 상호인증방식에 대해 안전성과 효율성을 분석한다. 다른 제안방식과 비교해 본 제안방식은 서버와 리더가 안전하지 않는 공개된 채널에서

표 1. 기존의 제시된 RFID 상호인증 프로토콜 안전성 비교

Items \ Scheme	Shi Scheme 1 [2]	Shi Scheme 2 [3]	Behzad Scheme [4]	Samad Scheme [5]	The Proposed Protocol
위치추적공격	×	×	○	○	○
도청	×	×	○	○	○
스푸핑 공격, 재전송 공격	×	×	○	○	○
전방향 안전성	×	×	○	○	○
서비스 거부 공격	×	×	×	×	○
비동기화	×	×	×	×	○
연산량	2E+2D+8X	1E+1D+7X	2E+2D+10X	2E+2D	1E+1D+10X+2Rot
상호인증	리더와 태그간의 상호인증 (서버와 리더는 안전한 채널을 가정)				서버와 태그간의 상호인증 (공개된 채널)

○: 다양한 보안위협으로부터 안전함; ×: 다양한 보안위협으로부터 안전하지 못함; E: 공개키 암호화; D: 공개키 복호화; X: XOR 연산; Rot(): Rot연산;

통신시 다양한 보안위협으로부터 데이터를 안전하게 보호하기 위해 제안된 상호인증 방식이다. 본 제안방식은 태그가 서버에게 인증정보를 보낼 때 NTRU 암호를 사용하여 보내기 때문에 공격자가 중간에 데이터를 도청해도 다양한 공격을 시도 할 수 없다. 또한 서버가 태그에게 인증정보를 보낼 때는 간단한 수식으로 데이터를 변형해 전송하는데, 공격자가 이 데이터를 가로채도 태그의 값을 추측할 수 없으며, 다양한 보안위협에 안전하다. 기존에 제안되었던 NTRU 기반의 RFID 상호방식들에 비교해 다양한 보안위협에 대한 안전성을 만족시키며 연산량 효율을 높였다.

V. 결론

기존의 RFID 상호인증 방식은 서버와 리더간의 통신이 안전하다는 가정하에 리더와 태그간의 상호인증에 중점을 두었다. 하지만 최근에 리더를 스마트폰이나 PDA와 같은 모바일장치에 장착함으로써 서버와 리더간에 통신 채널이 안전하지 않으며, 이에 서버와 리더 태그간의 통신에서 발생할 수 있는 취약점을 보안하는 논문들도 제안되고 있다. 본 논문은 NTRU 암호를 사용하여 서버와 리더, 태그가 공개된 채널환경에서 안전하고 효율적인 RFID 상호인증 방식을 제안하였다. 본 제안방식은 기존의제시된 NTRU 암호기반의 RFID 상호인증방식들과 비교해 다양한 보안위협으로부터 안전성을 만족하며, 연산량을 감소시켜 효율성을 높이는데 중점을 두었다.

향후 연구로는 본 논문에서 제시한 상호인증 방식을 실제 다양한 RFID 환경에 적용하기 위해 BAN logic나 실제 검증을 통한 안전성 증명이 필요할 것으로 사료된다.

[참고문헌]

[1] Park, Sung Wook, and Im Yeong Lee. "Authentication Scheme based on NTRU for the Protection of Payment Information in NFC Mobile Environment." KIPS Transactions on Computer and Communication Systems 2.3, 2013. pp. 133-142.

[2] Shi, Zhi Cai, et al. “A Novel RFID Authentication Protocol Based on NTRU”. Applied Mechanics and Materials. Trans Tech Publications, 2013. pp. 16-20.

[3] Shi, Zhicai, Yongxiang Xia, and Chaogang Yu. "A strong RFID mutual authentication protocol based on a lightweight public-key cryptosystem." Indonesian Journal of Electrical Engineering and Computer Science 12.3, 2013. pp. 2320-2326.

[4] Abdolmaleki, Behzad, et al. "Attacks and improvements on two new-found rfid authentication protocols." Telecommunications (IST), 2014 7th International Symposium on. IEEE, 2014. pp. 895-900.

[5] Rostampour, Samad, and Mojtaba Eslamnezhad Namin. "A New Efficient and Secure Mutual Authentication Protocol For RFID Systems." IEEE, 2015.

자마린에서 C#으로 개발된 안드로이드 앱에 대한 정적 역공학 분석*)

김규식, 심재우, 임경환, 조성제*
단국대학교 컴퓨터학과

Static Reverse Engineering of Android Apps in C# on Xamarin

Gyoosik Kim, Jaewoo Shim*, Kyeonghwan Lim*, Seong-je Cho*
Dept. of Computer Science & Engineering, Dankook University

요 약

자마린(Xamarin)은 크로스 플랫폼 도구로 모바일 앱을 C#으로 작성하게 해준다. 자마린을 사용하여, 개발자들은 기존의 C#코드를 재사용하면서 모바일 앱을 쉽게 개발할 수 있는 장점이 있는 반면, 공격자들 또한 악성 앱을 쉽게 제작하여 더 많은 플랫폼에 전파할 수 있게 되었다. 하지만, C#으로 작성된 앱에 대한 정적 역공학에 대한 체계적 연구는 수행된 적이 없다. 본 논문에서는 자마린에서 C#으로 개발된 안드로이드 앱의 구조를 분석하고 난독화된 C# 앱을 정적으로 역공학하는 방법을 제안한다. 하나의 앱 내에, 자바 바이트코드, C# 기반의 DLL 라이브러리, C/C++ 기반의 네이티브 라이브러리가 모두 포함된 경우를 대상으로 실험하여 제안 방법을 검증한다.

I. 서론

최근 모바일 크로스 플랫폼 도구(cross-platform tool, CPT)의 사용이 증가하고 있다[1,2,3,4]. 이러한 도구는 모바일 앱의 개발 생명주기 동안 효율성을 증대시켜 준다. 현재 여러 모바일 플랫폼들이 경쟁하고 있는데, 특정 플랫폼을 위한 네이티브 앱(native app)을 각각 개발하는 것은 개발 비용을 크게 증가시킨다. 네이티브 앱은 특정 모바일 플랫폼의 기능을 충분히 활용할 수 있는 반면, 서로 다른 플랫폼 간에 코드를 공유하기 어렵다. 특정 플랫폼은 Java나 Objective-C와 같은 전용 프로그래밍 언어와 전용 도구를 필요로 하며, 업데이트나 버그 수정과 같은 유지보수 비용도 매우 비싸다. 이러한 상황에서 특정 플랫폼에 독립적인 모바일 크로스 플랫폼 도구가 주목받고 있다.

자마린(Xamarin)은 모바일 크로스 플랫폼 도구/프레임워크로, 안드로이드, iOS, 또는 윈도우폰(Widows Phone)을 위한 모바일 앱을 C#으로 작성하게 해준다[1,2,3,4]. C# 프로그래머들은 자마린을 이용하여 새로운 프로그래밍 언어를 익힐 필요 없이 모바일 앱(자마린 앱)을 즉시 개발할 수 있으며, 기존의 C# 코드를 재사용하고 플랫폼 간에 상당한 코드를 공유할 수 있다. 2015년 기준 자마린 웹 사이트에 의하면, 90만 명의 개발자들이 크로스 플랫폼 앱들을 개발하기 위해 자마린 도구를 사용하고 있다[1].

한편, 공격자들도 멀웨어 제작시간을 최소화하면서 최대의 공격 효과를 내려고 자마린을 악용한다[5]. C#으로 악성 앱들을 더 빨리 제작하여 더 많은 플랫폼에 전파할 수 있기 때문이다. 백신 우회용 악성 실행파일을 제작하는 공격도구들의 집합인 Veil-Framework에서는, 페이로드가 C#으로 개발되기도 했다. 실제 인터넷을 검색하여 보면 C#으로 키로거를 작성하는 방법, C# Crypto-Ransomware 작성 및 DDoS 멀웨어 관련 유투브 영상, C#으로 콘솔 기반의 트로이 목마를 작성하는 튜토리얼, C# Nemesis.Worm 등을 접할 수 있다. Mylonas 등은 실험을 통해 학생들이 C#을 이용하여 Windows Mobile에서 이틀 만에, Windows Phone에서는 하루 만에 악성 앱을 제작할 수 있음을 보였다[6, 7].

이처럼 C# 기반의 악성 앱들이 도입되고 있지만, C#으로 제작된 악성 앱에 대한 효과적 분석 기법은 연구된 바가 없다. 이에 본 논문에서는 자마린으로 개발된 앱을 정적으로 역공학하는 방법을 제안한다. 실험에서는, 난독화 기법

* 이 논문은 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단 기초연구사업의 지원을 받아 수행된 연구임(No. 2015R1A2A1A15053738) 또한 본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 SW중심대학지원사업의 연구결과로 수행되었음(2017-0-00091).

을 포함시켜 C# 앱을 재패키징한 후, 이를 역공학해 보임으로써 제안 방법의 실효성을 보인다.

II. 관련연구

Mylonas 등은 스마트폰 플랫폼의 보안 수준을 평가하기 위해, 프로그래머들이 스마트폰 멀웨어를 얼마나 쉽게 제작할 수 있는지를 실험하였다[6, 7]. 실험에서 프로그래머들은 각 플랫폼 도구와 라이브러리를 사용하여, 안드로이드에서는 Java로 12시간 만에 멀웨어를 제작하였고, iOS에서는 Objective-C로 7일 만에 멀웨어를 제작하였다. Windows Mobile 6에서는 C#으로 이틀 만에 멀웨어를 제작하였고, Windows Phone 7에서는 C# 프로그래밍 경험을 가진 학부생이 하루 만에 멀웨어를 제작하였다. 자마린이 지원하는 모노(Mono)는 크로스 플랫폼 앱을 개발하도록 해 주는 .NET 프레임워크로, 리눅스, 윈도우 등에 설치될 수 있다[5]. 모노에서 개발된 프로젝트는 MoMA(Mono Migration Analyzer)을 사용하여 한 플랫폼에서 다른 플랫폼으로 이식될 수 있어, MoMA로 인해 악성 앱들의 생산성이 증대될 수 있다고 한다.

Boushehrinejadmoradi 등은 X-Checker라는 테스팅 도구를 개발하여, 홈 플랫폼 API를 동일 행위를 제공하면서 타깃 플랫폼 API로 변환할 때 발생할 수 있는 일관성 오류를 분석하였다[2]. 실험결과, 자마린에서 Windows Phone과 안드로이드 API의 구문 간에 변환 시에 47개의 버그를 탐지하였다. Martinez 등은 크로스 플랫폼 모바일 앱 개발 프레임워크를 사용하여 개발된 모바일 앱들의 품질을 분석하였다. 즉, 자마린과 같은 크로스 컴파일 프레임워크를 사용해 작성된 앱들의 개발 및 유지보수 과정을, 버그 탐지 및 정정에 초점을 두고 평가했다[3]. 여러 논문들이 악성 앱 개발 용이성과 버그 탐지·정정에 대해 연구하였지만, 자마린에서 C#으로 개발된 안드로이드 앱에 대한 역공학 분석 연구는 현재까지 수행된 바가 없다.

III. 실행파일 포맷 및 모노 런타임

자마린으로 개발된 앱을 역공학하기 위해서는 컴파일 과정과 실행파일 구조, 런타임 동작 원리를 파악해야 한다. 자마린 컴파일러는 C# 소스를 번역하여 공통 중간언어(Common Intermediate Language, CIL)를 생성한다. CIL은 Java의 바이트코드에 대응되며, CLR (Common Language Runtime)의 JIT(just-in-time) 컴파일을 통해 기계어로 변환된다[5]...NET PE 파일의 포맷이 그림 1에 나타나 있다.

안드로이드 플랫폼에서, C# 소스코드는 중간언어(IL) 코드로 컴파일되어 모노 런타임(Mono Runtime)과 함께 패키지화된다. 이 IL은 모노 런타임에 의해 실행되며, 모노 런타임은 그림 2에서 Mono CLR 부분에 해당 된다[8, 9]. 자마린 앱에 포함된 모노 런타임은 라이브러리(Xamarin.Android)로 Mono Assembly들을 사용한다. Xamarin.Android에서 모노 아키텍처는 JNI (Java Native Interface)와 peer objects 구성된다. JNI는 C# 또는 C++과 같은 코드가 Java 코드를 호출할 수 있게 해 준다. Peer objects는 Mono CLR (모노 런타임)과 Dalvik VM(자바 런타임)을 말한다[9]. MCW(Managed Callable Wrapper)는 모노 런타임이 자바 코드를 호출할 때 사용되며, ACW(Android Callable Wrapper)는 자바 런타임이 .NET 코드를 호출할 때 사용된다.

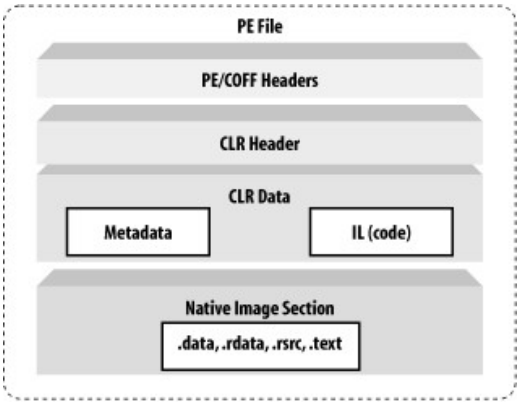


그림 1. .NET PE 파일의 포맷 [5]

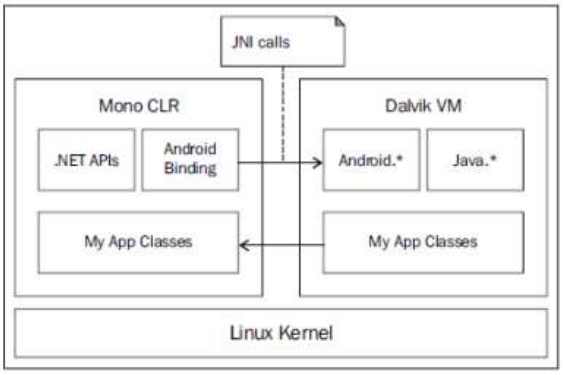


그림 2. 안드로이드에서 모노 아키텍처 [9]

IV. 자마린 앱에 대한 정적역공학 방법

4.1 C#으로 개발된 안드로이드 앱의 구조

Java로 작성된 안드로이드 앱에는 없지만, 자마린에서 개발된 안드로이드 앱에는 assemblies 폴더와 lib 폴더가 추가로 구성된다. assemblies 폴더에는 C# 코드가 CIL로 변환된 dll

파일들이 포함된다. 이 assemblies 폴더는 안드로이드 MainActivity, Java에서 C# 코드를 호출할 수 있게 해 주는 JNI, 안드로이드 API에 대한 C# 바인딩, 암호화 관련 API, 파일 입출력 기능 등을 제공하며, lib는 모노 런타임 환경과 JIT 컴파일 기능 등을 제공한다. lib 폴더에는 C/C++ 기반의 네이티브 라이브러리인 so 파일들, 즉 모노 런타임이 존재한다.

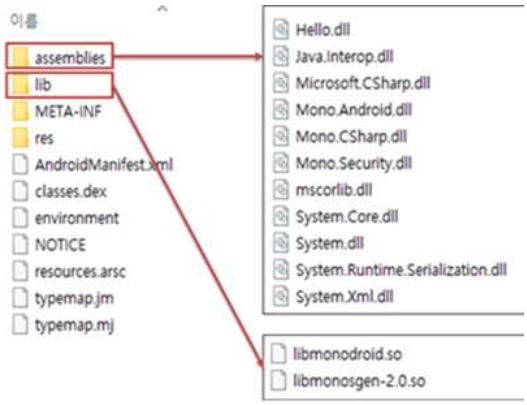


그림 3 C#으로 작성된 안드로이드 앱의 구조

4.2 자마린 앱에 대한 기본적인 정적 역공학

역공학은 실행코드를 역어셈블 (disassemble)하거나 역컴파일(decompile)하는 과정을 포함한다. 자마린에서 개발된 안드로이드 앱의 경우, 실행파일은 자바 바이트코드, .NET 프레임워크의 공통 중간언어로 된 dll, 네이티브 라이브러리인 so 모듈들로 구성된다. 이들 모듈을 역공학하기 위해서 해당 실행코드에 적합한 도구를 사용한다. 즉, 악성 앱의 바이트코드를 자바소스로 역컴파일하기 위해 Jadx나 Jeb를 사용한다. 공통 중간언어(CIL)로 된 dll을 .NET 기반 C#으로 역컴파일하기 위해서는 ILspy나 .NET Reflector을 사용한다. 라이브러리인 so 바이너리를 어셈블리어나 C 기반 코드로 변환하기 위해서 IDA pro를 사용하였다.

자마린 앱의 assemblies 폴더에 개발자가 작성한 DLL이 존재한다. DEX 파일 내 MonoPackageManager_Resources 클래스의 assemblies 배열에 개발자가 작성한 DLL파일과 서드파티 라이브러리 리스트들이 저장되어 있으며, 첫 번째 배열에 지정된 어셈블리인 XamarinEvolve.Droid.dll이 주요 기능을 나타내는 파일이름이다. 또한, 어셈블리는 Mono CLR 상에서 가장 먼저 실행되는 진입점(보통은 특정 액티비티)을 포함하고 있다. 진입점을 SplashActivity라고 할 때, 그림 4와 같이 SplashActivity는 Runtime.register( )라는 (C/C++로 작성된) 네이티브 함수를 호출하는데, 이 함수는 네이티브 코드에서 Java_mono_android_Runtime_register( )로 표현되어 있다. 이 함수 내 monodroid_runtime_invoke( )는 인자 값으로 어셈블리 이름, 실행할 클래스 및 메소드 이름, 파라미터를 요구하며 인자 값에 맞는 어셈블리의 코드를 실행한다. 역공학 시, 앱 진입점을 기준으로 분석하였다.

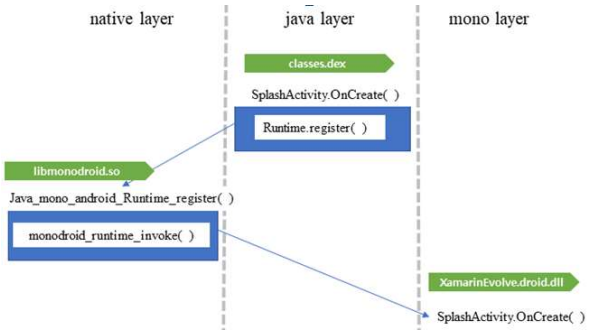


그림 4. 자마린 앱 실행 흐름 순서

4.3 난독화된 앱에 대한 정적 역공학

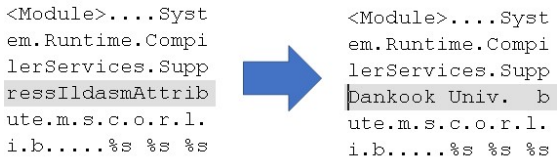
대표적 역공학 방해 기법으로 중간언어 기반의 코드 난독화가 있다. 난독화란 역공학을 방해/지연시키는 기법들로, 안티 덤프, 제어흐름 보호, 문자열 변환, 리소스 암호화 등의 기법 등을 포함한다. 본 논문에서는 기존 닷넷 기반 난독화 도구인 ConfuserEx v1.0.0을 이용하여 'Xamarin Evolve' 앱의 어셈블리(dll) 파일을 난독화하여 보았다. 실험대상인 Xamarin Evolve 앱은 자마린 플랫폼을 소개하는 앱이다.

Xamarin Evolve 앱의 XamarinEvolve.Droid.dll 파일에 난독화 기법을 적용하고, APKtool을 이용해 재패키징하여 앱을 실행하여 보았다. 실험 결과, ILdasm 사용 방지 기법(Anti ILdasm Protection), 제어흐름 보호(Control flow Protection) 기법, 심볼이름 변경(Rename Protection) 기법은 적용이 가능했으나, 이외에 다른 난독화 기법은 재패키징 후 정상 실행되지 않았다. 이는 ConfuserEx에 의해 internal class <Module>에 추가된 C/C++ 코드들이 추가되어 있어, 안드로이드 환경에서는 해당 C/C++ 코드를 JIT 컴파일을 제대로 수행하지 못해 실행이 불가능한 것으로 파악된다.

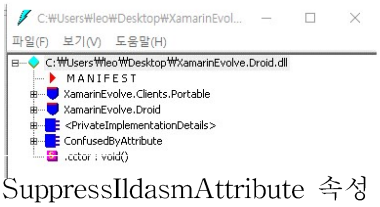
4.3.1. ILdasm 사용 방지 기법의 극복

ILdasm 실행파일은 어셈블리(dll) 파일 내에 SuppressIldasm 속성이 존재할 경우 해당 어셈블리 파일을 분해하지 않는다. SuppressIldasm 속성이 존재하는 어셈블리를 역어셈블하기 위하여, C# dll에서 ILdasm 방지 속성인 Suppres

sIldasmAttribute를 무시하고, ILdasm이 수행되도록 ILdasm 실행파일을 수정하였다. ILdasm는 어셈블리 파일을 입력값으로 하며, 해당 어셈블리를 IL 코드로 변환해 주는데 어셈블리 내에 SuppressIldasmAttribute가 존재할 경우 IL변환 기능을 수행하지 않는다. 이를 해결하기 위해 ILdasm 실행파일 내에 존재하는 SuppressIldasmAttribute의 텍스트를 검색하고 이를 임의의 다른 텍스트로 수정하여 어셈블리 내의 SuppressIldasm 속성이 존재하더라도 어셈블리 파일을 IL 코드로 변환하였다. 그림 5는 텍스트를 변경한 ILdasm 실행파일을 이용하여 SuppressIldasm 속성이 존재하는 어셈블리 파일을 IL 코드로 변환한 결과다.



(a) ILdasm 실행파일 변조



(b) SuppressIldasmAttribute 속성 우회

그림 5. ILdasm 실행파일 변조 후, SuppressIldasm 속성 우회

4.3.2 제어흐름 및 심볼명칭 난독화의 극복

제어흐름 난독화는, 원래 함수에 더미 코드 삽입, if문/switch문/for문 추가 등으로 코드의 실행흐름을 변경한다. 제어흐름이 변경되면 코드의 양이 많아지며 의미 없는 제어흐름이 추가되어 분석가로 하여금 역공학을 어렵게 한다. 실험에서 Xamarin Evolve 앱의 어셈블리에 제어흐름 난독화를 적용하여, 제어 흐름이 변경되고 대상 코드의 양이 원본에 비해 많아진 것을 확인하였다. 심볼 난독화는 타입, 메소드, 특성, 이벤트, 필드, 네임 스페이스 및 메소드 매개변수 등의 이름을 변경하는 것이다. 구체적으로 오버로딩을 이용한 메소드 이름 변경 기법, 멤버 이름을 유니코드 문자열로 변환 등을 적용한다. 실험결과, 이러한 난독화는 코드 이해 및 분석을 어렵게 할 뿐, 역공학 가능하였다.

V. 결론

이 논문에서는 안드로이드 환경에서 C#으로 개발된 자마린 앱의 구조 및 동작 원리를 분석하였다. 또한, 자마린 앱에 3가지 난독화 기법을 적용하여 성공적으로 재패키징하였고, 이 난독화 및 재패키징된 앱을 정적으로 역공학하는 방법에 대해 제시하였다. 이는 우리가 아는 한 C#으로 제작된 모바일 앱을 역공학하는 최초의 연구로, C#으로 작성된 안드로이드 악성 앱을 체계적으로 분석하는데 활용될 수 있다.

[참고문헌]

[1] Willocx, Michiel, Jan Vossaert, and Vincent Naessens. "A quantitative assessment of performance in mobile app development tools." IEEE International Conference on Mobile Services (MS), 2015

[2] Boushehrinejadmoradi, Nader, et al. "Testing cross-platform mobile app development frameworks." The 30th IEEE/ACM International Conference on Automated Software Engineering(ASE), 2015.

[3] Martinez, Matias, and Sylvain Lecomte. "Towards the quality improvement of cross-platform mobile applications." Proceedings of the 4th International Conference on Mobile Software Engineering and Systems. 2017.

[4] Popa, Marius. "Considerations Regarding the Cross-Platform Mobile Application Development Process." Academy of Economic Studies. Economy Informatics 13(1), 40, 2013.

[5] Pontiroli, Santiago M., and F. Roberto Martinez. "The Tao of .NET and PowerShell Malware Analysis." Virus Bulletin Conference. 2015.

[6] Mylonas, Alexios, et al. "On the feasibility of malware attacks in smartphone platforms." International Conference on E-Business and Telecommunications. Springer Berlin Heidelberg, 2011.

[7] Mylonas, Alexios, et al. "Smartphone security evaluation The malware attack case." Proceedings of the International Conference on Security and Cryptography (SECRYPT), 2011.

[8] Anchugov, Denis. "Cross-platform mobile software development: technology overview and a practical example." 2016.

[9] Haberl, DI FH Norbert. "Cross Platform Development Possibilities and drawbacks of the Xamarin platform.", FH Joanneum, University of Applied Sciences. 2015.

Fuzzy Hash와 Dex 파일 분석 로그를 이용한 안드로이드 악성 애플리케이션 유사도 비교

조하은*, 천호진*, 정수환**

*숭실대학교 정보통신공학과, **숭실대학교 전자정보공학부

Comparing similarity of Android Malware through analyzing Dex file with Fuzzy Hash

Haeun Cho*. Hojin Chun*, Souhwan Jung**

*Information & Telecommunication Engineering, Soongsil University.

**School of Information and telecommunication Engineering, Soongsil University

요 약

매해 새로운 안드로이드 악성애플리케이션들이 생성되고 있지만 새로운 악성 애플리케이션 패밀리의 수는 매우 적다. 이는 매일 새로 생성되는 악성 애플리케이션들이 새로운 악성 패밀리가 아닌 기존의 악성 애플리케이션의 변조이기 때문이다. 따라서 파일 간의 유사도를 잘 계산할 수 있다면 새로 생성되는 악성 애플리케이션에 대해 보다 쉽게 대응할 수 있다. 본 논문은 Dex 파일을 통하여 애플리케이션을 분석해 로그를 만들고 ssdeep을 이용하여 악성 애플리케이션을 분류할 수 있도록 유사도를 구한다. 그리고 이러한 유사도를 이용하면 리패키징된 악성 애플리케이션을 분류해 낼 수 있다.

I. 서론

전 세계적으로 안드로이드 보급률이 증가하면서 안드로이드 악성 애플리케이션도 매해 급격하게 증가하였다. 2015년에는 새로운 악성 애플리케이션이 매일 약 6400개 생성되고 있는 것을 확인할 수 있다[1]. 이에 악성코드 분석가들은 대량의 악성 애플리케이션을 자동으로 분석하는 시스템을 도입하고, 애플리케이션 분석을 자동화시키기 시작했다. 안드로이드 악성 코드 제작자들은 이러한 분석 시스템에 탐지 되지 않기 위해 코드 난독화나 동적 로딩 기법 등 다양한 방법을 사용하여 악성코드들을 지능화 시켰다[2].

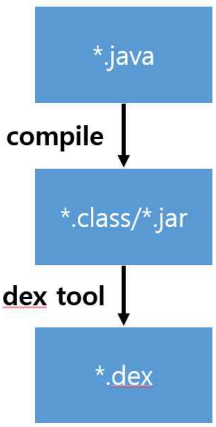
매일 수천 개의 새로운 악성코드가 생성되고 매해 새로 생성되는 악성코드의 수가 증가하고 있지만 새로운 악성 애플리케이션의 패밀리의 경우에는 2011년 30개, 2012년 24개, 2013년 18개 생성되었다[2]. 이는 매일 새롭게 생성되는 악성 애플리케이션들이 새로운 패밀리에 속하는 악성 애플리케이션이 아닌 기존 악성 애플리케이션을 새로운 악성코드처럼 만들기 위해 일부 기능을 추가하거나 안티 바이러스에 의해 탐지되는 것을 우회하기 위한 기능을 넣은 것이 다수이기 때문이다. 따라서 새롭게 만들어진 악성 애플리케이션을 효과적으로 분류할 수 있다면 대량의 악성코드 샘플을 효과적으로 대응할 수 있을 것이다. 본 연구에서는 Java코드를 최적화한 classes.dex 파일을 통하여 Java 클래스 간의 호출관계를 정리한 로그 파일을 만들고, 그 로그 파일을 이용하여 기존의 악성 애플리케이션들 긴의 유사도를 구한다.

서론에 이어 본 논문은 2장에서 배경지식으로 안드로이드 Dex와 ssdeep에 대해 소개한다. 3장에서는 본 논문에서 제안한 기법을 소개하고 동작 과정에 대해 설명한다. 마지막으로 4장에서는 본 논문의 결론을 제시하고 향후 연구 및 방향에 대해 서술한다.

II. 배경 지식

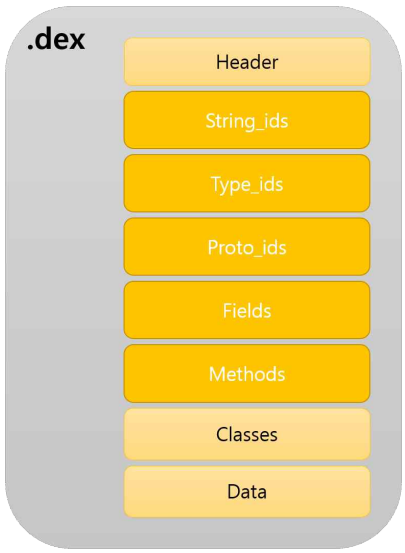
2.1 Dex

classes.dex 파일은 안드로이드 애플리케이션의 Java 코드들을 DVM(Dalvik Virtual Machine)에서 효율적으로 실행될 수 있도록 최적화한 파일이다



[그림 1] dex 파일이 만들어지는 과정

Dex 파일의 header에는 파일의 magic값과 checksum, signature값이 있고, string, type, proto, field, method, class_defs 영역의 사이즈와 해당 영역으로의 offset을 나타내는 정보를 갖고 있다. Dex 파일에서는 사용된 문자열을 String_ids 영역을 이용하여 별도로 관리하고 해당 레이아웃은 전체 문자열의 수와 전체 문자열의 시작 위치를 가리키고 있다. Type_ids는 소스코드 상의 기록된 함수의 반환 값, 인자 값에 사용된 변수 변환 형식이 나타나있고 해당 영역에 의해 별도로 관리된다. Proto_ids는 메소드의 원형에 대한 정보를 갖고 있다. Field_ids는 소스코드 내에서의 패키지 이름, 클래스 이름 등의 정보를 갖고 있고 해당 영역에서 별도로 관리가 되어진다. Method_ids는 메소드 정보의 개수와 시작 위치에 대한 정보를 갖고 있고 소스 코드 내의 모든 클래스에 대한 정보는 Classes에서 갖고 있다. 마지막으로 Data영역에서는 해당 소스코드내의 모든 데이터 정보를 갖고 있다[3].



[그림 2] dex 파일 구조

2.2 ssdeep

ssdeep은 Fuzzy Hash(Context Triggered Piecewise Hashing)를 사용하여 파일의 유사도를 측정할 수 있는 대표적인 프로그램이다. 이 프로그램에 사용된 Fuzzy Hash 알고리즘은 파일 내에서 context를 찾아서 블록화 한 후 각 블록의 해시 값을 구하고 구해진 해시 값을 이용하여 유사도를 구한다. 이러한 방식을 통해서 구해진 해시 값은 동일한 데이터를 가진 블록에선 동일한 해시 값을 다른 데이터를 가진 블록에서는 다른 해시 값을 가지므로 유사도를 계산할 수 있다[4].

III. 제안 기법

분석된 dex 파일의 로그를 가지고 ssdeep을 이용하여 안드로이드 애플리케이션 간의 유사도를 자동으로 계산하는 프로그램을 구현한다.

3.1 Dex 파일 분석

Dex 파일의 헤더를 분석하여 애플리케이션의 class, method, 호출 흐름 등을 판단할 수 있다. 따라서 Dex 파일을 분석할 수 있도록 코드를 작성하여 Dex 파일을 분석하고 애플리케이션의 class, method 호출 흐름이 담긴 로그를 자동으로 생성한다. 본 연구에서는 [그림 2]와 같이 호

출 관계를 알 수 있도록 로그 파일을 작성하였다. 이렇게 로그파일 내에 Java 클래스 간의 호출 흐름을 넣음으로써 로그 파일의 크기를 늘릴 수 있다. 이러한 로그 작성법은 작은 크기의 데이터(파일)에서의 분석이 어렵다는 ssdeep의 단점을 해결할 수 있고, 단순하게 정적 로그를 추출하는 것 보다 정확한 유사도를 얻을 수 있다.

```
:om.shell.SuperApplication copyLib >> java.io.File mkdirs
:om.shell.SuperApplication copyLib >> java.util.zip.ZipFile getInputStream java.util.
:om.shell.SuperApplication copyLib >> java.io.FileOutputStream <init> java.io.File
:om.shell.SuperApplication copyLib >> java.io.InputStream read B[]
:om.shell.SuperApplication copyLib >> java.io.InputStream close
:om.shell.SuperApplication copyLib >> java.io.FileOutputStream close
:om.shell.SuperApplication getCRC32 >> java.io.File exists
:om.shell.SuperApplication attachBaseContext >> android.app.Application attachBaseCon
:om.shell.SuperApplication copyLib >> java.lang.String equals java.lang.Object
:om.shell.SuperApplication copyLib >> java.io.File <init> java.lang.String
:om.shell.SuperApplication copyLib >> java.io.File <init> java.lang.String
:om.shell.SuperApplication copyLib >> java.util.zip.ZipFile <init> java.lang.String
:om.shell.SuperApplication copyLib >> java.util.zip.ZipFile getEntry java.lang.String
:om.shell.SuperApplication copyLib >> java.util.zip.ZipEntry getCrc
:om.shell.SuperApplication copyLib >> java.util.zip.ZipFile getEntry java.lang.String
:om.shell.SuperApplication copyLib >> java.util.zip.ZipEntry getCrc
:om.shell.SuperApplication copyLib >> java.util.zip.ZipFile close
:om.shell.SuperApplication loadLibs >> java.lang.Runtime getRuntime
:om.shell.SuperApplication loadLibs >> java.lang.Runtime exec java.lang.String
:om.shell.SuperApplication loadLibs >> java.lang.Process getInputStream
:om.shell.SuperApplication loadLibs >> java.io.InputStreamReader <init> java.io.Input
:om.shell.SuperApplication loadLibs >> java.io.BufferedReader <init> java.io.Reader
:om.shell.SuperApplication loadLibs >> java.io.BufferedReader readLine
:om.shell.SuperApplication loadLibs >> java.lang.String contains java.lang.CharSequen
:om.shell.SuperApplication onCreate >> android.app.Application onCreate
```

[그림 3] Dex 파일 분석 결과

3.2 제안하는 방법

다음과 같은 순서로 새로운 악성 애플리케이션의 유사도를 계산하는 프로그램을 구현한다.

(1) 새 애플리케이션에서 classes.dex 파일을 추출한다.

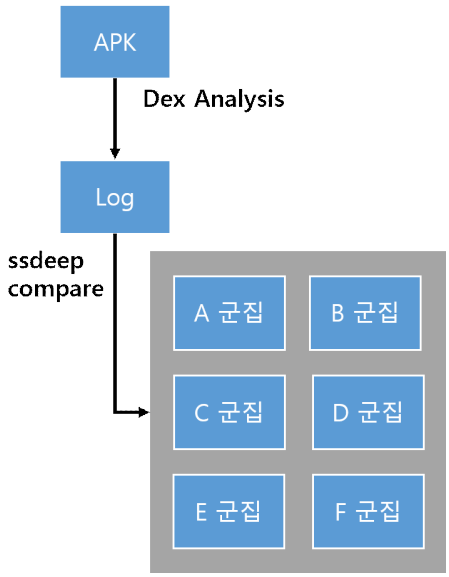
(2) 추출된 classes.dex 파일을 분석하여 Java 클래스 간의 호출 흐름이 담긴 로그 파일을 생성한다.

(3) ssdeep을 이용하여 로그 파일과 기존에 만들어진 애플리케이션 군집과의 유사도를 구한다.

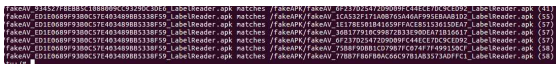
IV. 분석

4.1 평가

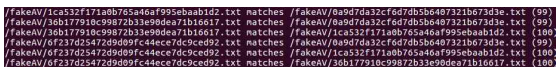
Dex 분석 로그를 이용하여 유사도 계산한 결과를 평가하기 위해 바이너리 파일을 이용하여 유사도 계산을 하였을 때와 비교하였다. 다음 [그림 5]와 [그림 6] 은 같은 군집 내에서의 바이너리 파일을 사용하여 유사도를 계산한 결과와 Dex 분석로그를 이용하여 파일간의 유사도를 계산한 결과이다.



[그림 4] 유사도 계산 방법



[그림 5] 바이너리 APK들의 유사도



[그림 6] Dex 파일 로그들의 유사도

해당 그림을 보면 알 수 있듯이 Dex 분석 로그 파일이 APK 바이너리 보다 정확한 유사도를 갖는다. 정확한 유사도를 계산할 수 있게 되면 리패키징이 된 악성 애플리케이션의 탐지, 머신러닝의 데이터로 적용할 수 있기 때문에 대량의 악성 애플리케이션 분석에 용이하다.

4.2 한계점

ssdeep은 파일 구조적으로 바이너리 전체가 바뀌는 경우 탐지하지 못하고, 파일의 크기자 작을 때 전혀 다른 파일과 매칭이 되는 등 정확성이 떨어진 다는 문제가 있다[5].

Dex 파일은 Java 코드가 DVM(Dalvik Virtual Machine)에서 잘 돌아갈 수 있도록 최

적화한 파일이기 때문에 Native 코드를 포함하고 있지 않다. 최근의 악성코드들은 정적 분석을 우회하기 위해 Native 코드에 악성 코드들을 숨겨두기 때문에 모든 코드를 비교한 것이 아니라는 문제가 있다.

V. 결론

본 논문에서는 Dex 파일 분석을 통하여 획득한 Java의 클래스 호출 관계를 이용하여 악성 애플리케이션의 유사도를 계산하였다. 바이너리 기반 유사도 분석보다 개선되었지만 Dex 파일이 안드로이드 SDK에 한정되므로 Native 영역에 대한 분석이 불가능하다는 문제가 있다. 또, ssdeep이 작은 사이즈의 파일은 유사도 분석이 불가능하고, 구조적으로 바이너리 전체가 바뀌게 되는 겨우 전혀 탐지하지 못한다는 문제가 있다. 그러므로 앞으로 동적 분석과 네이티브 분석에 대한 연구를 통하여 새로운 로그 파일을 만들고 개선된 유사도 분석 결과를 얻을 수 있도록 한다.

Acknowledgment

본 연구는 산업통상자원부 및 한국산업기술평가관리원의 우수기술연구센터(ATC)사업과 [10045904, 클라우드 컴퓨팅 환경하에서 정보보안 서비스를 제공하기 위한 SecaaS 프레임워크 원천기술 개발과 이를 이용한 1Gbps급 모바일 정보유출방지 서비스 구축] 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2017-0-01853, 머신러닝 기반 지능형 악성코드 분석 통합 플랫폼)

[참고문헌]

[1] Data G, "Mobile malware report." September, 2015.

[2] Parvez Faruki, Ammar Bharmal, Vijay Laxmi, Vijay Ganmoor, Manoj Singh Gaur, Mauro Conti, Muttukrishnan Rajarajan, "Android Security: A Survey of Issues, Malware Penetration, and Defenses," IEEE communications surveys & tutorials, vol.17, no.2, pp.998-1022, 2015.

[4] Kornblum Jesse, "Identifying almost identical files using context triggered piecewise hashing," Digital investigation 3, pp.91-97, 2006.

[3] Ehringer David, "The dalvik virtual machine architecture," Techn. report, vol.4, no.8, 2010.

[5] Kornblum Jesse, "Identifying almost identical files using context triggered piecewise hashing," Digital investigation 3, pp.91-97, 2006.

EU GDPR 사례를 통한 바이오정보보호 개선 방안

정부금*, 권헌영*, 임종인*
*고려대학교 정보보호대학원 정보보호학과

Improvement of Bio Data Protection through EU GDPR

Boo-Geum Jung*, Hun-Yeong Kwon*, Jong-In Lim*
*Division of information security, Korea University.

요 약

센서 기반의 IoT 기술의 발달로 바이오정보 인식은 더욱 정확성을 띄게 되었고, 지문 뿐만 아니라 얼굴, 홍채 인식이 스마트폰에 쓰이기 시작하고 있고, 인터넷 전문은행, 모바일 결재 등 비대면 금융 서비스의 활성화 등으로 바이오 정보는 본인 인증을 위한 수단으로서 다양한 분야에서 활용되고 확산되고 있다. 그러나 현재의 개인정보보호법제는 이러한 급진적인 기술적, 사회적 발전을 수용하기에는 미흡한 부분이 있는 실정이다. 이에 본 논문에서는 최근 바이오정보를 개인정보에 명시적으로 추가하여 제정된 EU 개인정보보호일반법인 GDPR을 참조하여 바이오 정보의 활용과 보호를 위한 개선사항들을 도출하여, 안전한 바이오 인증 서비스 제공을 위한 법제 및 정책 수립 시 도움이 되고자 한다.

I. 서론

온라인에서의 다양한 활동을 기반으로 우리의 일상이 이루어지고 있어, 원격 및 비대면 상황에서 본인임을 입증하는 개인 인증은 온라인 활동에 필수적인 관문이며, 오프라인에서 열쇠로 문을 열고 들어가는 것과 동일한 의미를 갖는다.

개인 인증을 위한 수단으로는 패스워드, 휴대전화, 아이핀, 주민등록번호, 신용카드가 있으나 항상 소지하고 있거나 암기하고 있어야 하는 불편함이 있고, 또한 일련의 대형 개인정보 유출 사고들로 인해 더 이상 안전성을 담보할 수 없는 상황이 되어, 최근 바이오 정보를 활용한 개인 인증이 급진적으로 성장하고 있다[1].

바이오 정보의 보호에 대한 이슈는 이미 오래전부터 제기되어 왔지만, 이에 대응하기 위한 법제도의 개선은 미흡한 상황이다. 바이오정보 보호 가이드라인[1]이 2005년 및 2007년에 발표되었고, DNA 정보 수집을 목적으로 한 법률[2], CCTV 규제에 대한 내용이 「개인정보 보호법」에 들어가 있으며[2], 바이오 정보를 전체적으로 포괄하여 법적으로 규제되고 있지 않은 상황이다.

이에 본 연구에서는 최근 유럽에서 바이오정보를 명시적으로 다루고 있는 GDPR[3]을 참조하여 바이오 정보의 활용과 보호를 위한 개선사항들을 도출하여, 시급히 필요한 안전한 바이오 인증 서비스를 위한 법제 및 정책 수립 시 도움이 되고자 한다.

II. 바이오정보 보호의 필요성

개인정보는 살아있는 개인에 관한 정보[3]로

1) http://it.chosun.com/news/article.html?no=2814586

2) 개인정보보호법 제25조 영상정보처리기의 설치·운영제한

3) “개인정보”란 살아 있는 개인에 관한 정보로서 성명, 주민등록번호 및 영상 등을 통하여 개인을 알아

개인정보보호법[4]에서는 민감정보로 건강정보를 포함하고 있고, '바이오정보보호 가이드라인'에서는 바이오 정보를 "지문·얼굴·홍채·정맥·음성·서명 등 개인을 식별할 수 있는 신체적 또는 행동적 특징에 관한 정보를 말하며, 가공되지 않은 원본정보와 그로부터 추출되어 생성된 특징정보를 포함한다"라고 정의하고 있다. 즉, 바이오정보는 살아있는 개인에 관한 정보로 개인정보로 포함되어 보호되어야 한다.

바이오정보를 활용하는 다양한 서비스들이 제공되고 있지만, 구체적인 법적 장치가 없어 이에 대한 대비가 시급한데, 바이오 정보는 유출돼도 바꿀 수 없어 철저한 보안책 마련이 필요하기 때문이다.

바이오 정보 관련 보안 사고가 국내외에서 발생하고 있다. 3D 프린터를 이용하여 실리콘으로 위조지문을 제작하여 증명서 등 서류를 발급받아 수십억대의 부동산을 불법으로 명의이전한 사건(2014년 10월), 소방공무원들이 실리콘으로 위조지문을 만들어 단말기에 인식시켜 초과근무수당을 챙긴 사건이 있었다(2016년 2월). 또한, 독일 해커단체인 CCC는 구글에서 찾은 푸틴 러시아 대통령의 고해상도로 출력한 사진을 이용해서 홍채 인식에 성공하는 모습을 공개했다(2014년). 미국에선 보안 서버가 해킹당해 저장된 생체정보 데이터가 대거 유출된 사례도 있다. 이러한 사고 사례는 개인 고유의 특징인 생체정보가 위조되고 공개된 정보를 이용하여 악용될 수 있음을 경고한 것이다.

최근 EU를 비롯하여 주요 국가와 국제기구들에서는 개인정보보호제도를 새롭게 개혁하고 있는데, 이는 인터넷이 시작된 이후 급속하게 발전한 정보통신기술과 개인정보보호와의 합리적인 균형을 이루기 위한 노력으로 파악된다. 특히, 유럽연합에서 제정한 GDPR은 30년 동안 사용되어온 '1995 개인정보보호지침[5]을 대체하였다.

국내 개인정보보호법은 2011년에 제정되어 개인정보보호에 관한 일반법의 역할을 하고 있으나, 바이오정보의 활용 및 보호에 관한 내용을 충분히 담아내지 못하고 있다. 이에 추가적으로 바이오정보 가이드라인을 정의하고 각 분야별로 각각의 법 및 가이드라인들에서 바이오정보를 다루고 있으나 불명확성, 중복성 등으로 인해 효율적 바이오정보 기반 서비스의 발전이 사실상 어렵다. 따라서, 바이오 정보를 활용하여 다양한 서비스가 이미 우리의 삶 속에 들어와 있는 현실에서 개인의 프라이버시를 보호하면서 활용을 위한 개선방안에 대한 검토는 시의적절하고 필수적인 것으로 판단된다.

III. EU GDPR에서의 바이오정보

GDPR에서는 새로이 개인정보로서 바이오정보(bio data)를 유전정보, 생체인식 정보, 건강관련 정보로 명시하고 있다(제9조 특정 범주의 개인정보 처리(Processing of special categories of personal data).

일반적인 개인정보에 대해서도 정보주체의 보호는 중요하지만 특히 민감정보인 바이오정보에 대한 정보주체의 보호는 더욱 중요한 사안이다. 이에, GDPR은 정보주체를 보호하기 위해 여러 가지 제도를 추가적으로 규정하였다. 그 중 제25조에서는 데이터 보호 중심 설계 및 데이터 보호 설정[4]을 규정하고 있다. 개인정보처리자는 개인정보의 처리 수단을 결정한 시점과 처리 당시 시점에서, 정보 최소화 등 개인정보보호의 원칙을 이행하고 정보주체의 권리를 보호하기 위해, 처리에 필요한 안전조치를 포함하기 위해 고안된 가명처리 등, 적절한 기술 및 관리 조치를 이행해야 한다(by Design). 또한 개인정보 처리자는 기본설정을 통해, 처리의 개별 특정 목적에 필요한 정도에 한하여 개인정보가 처리될 수 있도록 보장하기 위한 적절한 기술 및 관리 조치를 이행해야 한다(by Default). 이런 조항들은 개인정보에 대한 정보주체의 권리를 실질화하기 위하여 상품이나 서비스의 구상·기획 단계에서부터 적용할 것을 제안받고 있는 프라이버시 중심설정(Privacy by

볼 수 있는 정보(해당정보만으로는 특정 개인을 알아볼 수 없더라도 다른 정보와 쉽게 결합하여 알아볼 수 있는 것을 포함한다)를 말한다.(개인정보보호법 제2조 제1호)

4) Data protection by design and by default

Default), 프라이버시 중심설계(Privacy by Design)의 원칙[5] 및 유럽네트워크정보보호원(ENISA)의 프라이버시 및 개인정보보호 중심 설계의 공학적인 설계 방법[6]을 법으로 정의한 것이다.

GDPR에서는 개인정보 영향평가도 적극적으로 도입하였다. 특히 제35조와 해설전문 91에서는 바이오인식 정보 등 민감정보의 대규모 처리에 대해서는 개인정보 영향평가가 적용되어야 한다는 점을 명시하고 있다.

GDPR은 '프로파일링'(profiling)[7]을 규정하고 이를 제한하는 조항들을 도입하였다. 자동처리에 따라 정보주체에게 피해가 가지 않도록 규정한다. 유럽연합 역외의 개인정보처리자라 하더라도 정보주체에 대한 결정을 할 때나, 정보주체의 개인적 선호, 행동과 태도를 분석하거나 예상하는 프로파일링 기법 같은 개인정보처리 기술을 잠재적·계속적으로 사용하여 개인을 인터넷에서 추적하는 경우에는 이 법의 적용을 받는다(해설전문 24). 정보주체는 프로파일링 유무와 해당 프로파일링의 결과에 대해 고지받아야 한다(해설전문 60).

GDPR에서는 데이터의 보호에 가명처리의 개념을 새로이 도입하여 익명정보(anonymisation)와 가명처리(pseudonymisation)를 구분하여 다루고 있다. '가명처리'는 추가 정보를 사용하지 않고는 특정 정보주체를 식별할 수 없도록 개인정보를 처리하는 것을 의미하며, 이 경우, 해당 추가정보는 별도로 보관되며 개인정보로 식별되거나 식별될 가능성이 있는 개인에게 해당되지 않도록 보장하기 위한 기술 및 관리조치가 적용된다(제4조(5)). 이때 GDPR의 가명처리, 익명정보의 구분을 적용해 보면, 공익을 위한 유지보존의 목적, 과학이나 역사 연구의 목적 또는 통계 목적에서의 개인정보 처리라 하더라도 가명처리는 개인정보로서 정보주체의 동의 등 절차에 따라 처리해야 하고, 익명정보는 개인정보가 아니므로 이 법의 적용이 배제된다.

IV. 국내 바이오정보 관련 법제 검토

바이오정보를 개인정보로 볼 때 국내 개인정보에 관한 일반법인 개인정보보호법을 따르게 된다.

바이오 정보가 정의되어 있는 부분은 민감정보 관련 조항이다. 사상·신념, 노동조합·정당의 가입·탈퇴, 정치적 견해, 건강, 성생활 등에 관한 정보나 그 밖에 정보주체의 사생활을 현저히 침해할 우려가 있는 정보로서 개인정보보호법 시행령에 의해 규정된 유전정보, 범죄경력자료 등이 민감정보에 해당되어 그 처리가 제한된다.

그러나 「개인정보 보호법」은 모든 바이오인식 정보를 민감정보로서 포함하고 있지 않다. 지문은 개인의 고유성, 동일성을 나타내고 정보주체를 타인으로부터 식별가능하게 하는 가장 보편적이고 중요한 개인정보이지만, 현행 개인정보 보호법에서는 민감정보로 규율되고 있지 않다. 또 민족 또는 인종적 기원을 드러낼 수 있는 유전정보는 민감정보로서 보호받고 있지만, 국제 규범에서 통상 민감정보로서 보호하고 있는 민족 또는 인종적 기원에 대한 보호를 우리 법률은 명시하고 있지 않다.

개인정보보호법 제3조에서는 바이오정보가 포함된 개인정보의 주체를 보호하기 위하여 정보 처리 목적을 명확하게 하고 그 목적에 필요한 범위에서 최소한의 정보만을 처리해야함을 원칙으로 규정하고 있다.

개인정보의 수집과 제공에 관해서는 제15조

---

5) 프라이버시 중심설계란, 캐나다 온타리오 개인정보보호감독관인 카부키안 박사가 1990년대 "미래의 프라이버시는 단지 법률이나 규제 체제 준수만으로 보장할 수 없다. 프라이버시 보장은 기관들이 시행과정에서 기본설정으로 채택해야만 한다"는 취지로 고안한 개념이다. (Ann Cavoukian, 2013).

6) ENISA 보고서에서 제시한 8가지 설계 전략은 ① 최소화(Minimize) ② 숨기기(Hide) ③ 분리(Separate) ④ 총계(Aggregate) ⑤ 통지(Inform) ⑥ 통제(Control) ⑦ 강화(Enforce) ⑧ 입증(Demonstrate) (ENISA, 2014)

7) 우선 '프로파일링'이란, "개인에 관한 특정한 개인적 측면을 평가하기 위해, 특히 개인의 업무능력, 경제 상황, 건강, 개인의 성향이나 관심사, 신뢰도, 행동, 위치, 이동에 관한 측면을 분석 및 예측하기 위해 개인정보를 사용하는 모든 개인정보의 자동처리 형태를 의미한다"(제4조(4))

와 제17조의 범위를 넘어서는 이용 및 제공은 원칙적으로 금지되지만, '통계작성 및 학술연구 등의 목적을 위하여 필요한 경우로서 특정 개인을 알아볼 수 없는 형태로 개인정보를 제공' 하는 경우 예외가 허용된다. 즉, 이 경우 개인정보 처리자는 정보주체 또는 제3자의 이익을 부당하게 침해할 우려가 있을 때를 제외하고는 개인정보를 목적 외의 용도로 이용하거나 이를 제3자에게 제공할 수 있다. 이렇게 개인정보보호법은 목적 외 이용·제공에 대한 예외사유를 한정적으로 명시한다. 또한 개인정보보호법에서는 '통계 및 학술 연구 등'에 따른 목적 외 이용·제공을 규정(제18조 제2항 제4호)하고 있다.

개인정보보호법은 제3조 제7항에서 "개인정보처리자는 개인정보의 익명처리가 가능한 경우에는 익명에 의하여 처리될 수 있도록 하여야 한다"라고 규정하고 있다. 그러나 익명처리에 대해서 구체적 내용이나 방법에 대하여 정의하지 않는다.

또한, 개인정보보호법은 GDPR이 중점적으로 새로이 규정한 가명처리에 대해 규정되어 있지 않는 것으로 보인다.

V. 바이오정보보호 개선 방안

EU GDPR을 중심으로 살펴볼 때, EU는 GDPR 로 28개국의 개인정보보호를 일관되게 적용하고 있는 반면, 바이오 정보의 보호 및 활용에 관련한 국내 법제는 여러 법제에 흩어져 있어 복잡한 경향을 띈다.

또한, 여러 법제들에서 조금씩 다른 의미로 정의하고 있는 '바이오인식 정보'를 일관되게 정리하여 유전정보 및 건강관련 정보 등을 포함하는 바이오 정보에 대해 전체적으로 통일적인 법제를 적용할 수 있어야 한다.

바이오 정보에 대한 보호를 위하여 현행 「개인정보 보호법」상의 민감정보에 대한 보호 조치를 제고할 필요가 있다. 특히 개인정보 처리자가 민감 정보의 처리를 위해 정보주체로부터 동의를 구할 때 그 동의 요건을 보다 명시적으로 규정할 필요가 있다.

바이오 정보 보호의 민감성에 비추어 가장 중요한 것은 사고의 사전 예방이다. 그런 의미에서 현재 정의되고 있지 못한 개인 바이오정보의 영향평가 제도를 실시할 수 있어야 한다. 더불어 프라이버시 중심 설계 및 프라이버시 중심 설정(Privacy by default)를 도입할 필요가 있다.

또한, 프로파일링과 가명처리에 대한 명확한 기준과 처리를 추가하여 바이오정보의 활용 및 보호에 있어 문제점이 없도록 하여야 할 것이다.

VI. 결론

온라인에서의 경제생활이 일상화되어 간편한 바이오정보를 활용한 금융, 인터넷, 건강 등의 서비스 활성화는 피하거나 제한할 수 없는 것이지만, 동시에 개인의 정보보호도 결코 양보될 수 없다. 이 점에서 바이오정보의 활용과 개인 바이오정보보호는 함께 추구하여야 할 것이고, 이러한 개인바이오정보의 활용은 개인정보보호를 내재하면서 허용되어야 할 것이다.

EU GDPR을 중심으로 살펴보았을 때, 바이오정보에 대한 명확하고 일관된 정의가 필요하며, 바이오정보의 본인 인증 목적 외 사용에 대해서는 구체적인 규정이 필요하다. 또한 바이오정보에 대한 영향 평가및 프로파일링 정의, 가명화에 대한 구체적인 정의가 추가되어야 한다고 판단된다. 이러한 개선방안들을 도출함으로써, 바이오 정보의 활용과 보호의 균형 확보를 위한 법제 및 정책 수립을 위한 자료가 될 수 있을 것이다.

[참고문헌]

[1] 바이오정보이용 가이드라인 (시행 2007)

[2] 디엔에이신원확인정보의 이용 및 보호에 관한 법률 (시행 2017.7.7.)

[3] GDPR(General Data Protection Regulation) (Regulation (EU) 2016/679)

[4] 개인정보보호법 (시행 2017.7.26.)

[5] EU Data Protection Directive (Directive 95/46/EC)

2017 한국정보보호학회 동계학술대회

E2 모바일 보안(II)

노 건 태 (서울사이버대)

- **NFC 스마트카드 해킹을 통한 부당이득 취득 사례 및 공격 기법에 관한 연구**
  이상욱 (전남대학교), 나동빈 (한국교원대학교), 박한울 (경기대학교), 이태일 (한국교통대학교), 강종구 (유원대학교)
- **엔트로피를 기반으로 한 전자거래 바이오인증 강도 산정 방법 제시 및 비교 분석**
  김지혜, 사경진, 염흥열 (순천향대학교)
- **악성코드 분석을 위한 안드로이드 DEX 바이너리 기반 Java API 추출 방법**
  안진웅, 박정수, 정수환 (숭실대학교)
- **지속 인증을 위한 사용자 행위 정보 기반특징 요소 별 성능 분석 연구**
  구예은, 이호연, 권태경 (연세대학교)

NFC 스마트카드 해킹을 통한 부당이득 취득 사례 및 공격 기법에 관한 연구

이상욱[a], 나동빈[b], 박한울[c], 이태일[d], 강종구[e]

[a]전남대학교 컴퓨터정보통신공학과, [b]한국교원대학교 컴퓨터교육과, [c]경기대학교 컴퓨터과학과,

[d]한국교통대학교 컴퓨터정보공학과, [e]유원대학교 정보통신보안학과

A Study on the Attack Technique and Unjust Enrichments through NFC Smart Card Hacking

Sang-Wook Lee[a], Dong-Bin Na[b], Han-Ul Park[c],

Tae-Ill Lee[d], Jong-Goo Kang[e]

[a]Division of Computer Information and Communication Engineering, Chonnam National University,

[b]Division of Computer Education, Korea National University of Education,

[c]Division of Computer Science, Kyonggi University,

[d]Division of Computer Information Engineering, Korea National University of Transportation,

[e]Division of Information and Communication Security, U1 University

요 약

MIFARE Classic은 NFC(Near Field Communication) 스마트카드의 일종으로 결제, 사용자 인증 등의 다양한 목적으로 사용된다. 다만 MIFARE Classic이 가지는 구조적 취약점으로 인해 다년간 카드 복제 및 Data 변조를 통한 해킹 사고가 빈번히 발생해왔다. 현재는 보안이 강화된 스마트카드 모델들이 나왔음에도 여전히 일상생활에서 대부분의 스마트카드는 MIFARE Classic 모델을 사용하여 불법적인 위·변조에 노출이 되어있는 실태이다. 본 논문에서는 현재까지 언론에 보도되지 않았던 사업상의 막대한 금전적 손실을 가져올 수 있는 스마트카드 해킹 기법 및 사례에 관한 연구를 진행할 것이다.

I. 서론

1.1 MIFARE 기술의 개요

MIFARE는 NXP사에서 개발한 스마트카드 기술로 MIFARE Classic 1K/4K 모델이 1994년 첫 번째로 출시되었다. 이후 MIFARE Classic은 세계 전역의 교통카드 기술로 채택되어 사용되었고 다양한 결제·인증카드가 MIFARE Classic을 활용해 개발되었다. 단순히 잔액 Data만을 담기에는 1K의 메모리가 크다는 점에서 MIFARE Classic MINI 등의 적은 용량의 모델도 개발되었다.

최근 국내에서 사용되는 대부분의 교통카드는 MIFARE Plus 2K/4K 모델을 탑재하여 나오고 있다. 128비트 암호화를 지원해 기존의 MIFARE Classic 모델에 비해 상대적으로 높은 보안성을 갖추고 있다. 다만 카드의 단가 문제와 보안의식의 결여로 인해 아직까지도 산업 전반에 사용되는 대부분의 스마트카드는 MIFARE Classic 모델이다. 결과적으로 금전적 손실이 발생할 수 있는 가능성이 현재 사회 곳곳에 만연한 상황이다.

1.2 국내 NFC 스마트카드 해킹 사례

악의적인 NFC 스마트카드 이용자들은 금전적 이익을 취하기 위해 카드의 구조적 결함을 이용한 해킹을 시도한다. 2014년 국내에서 MIFARE Classic 기술을 사용한 교통카드를 해킹하여 1억 8천 여 만원의 부당이득을 취한 일당이 검거된 사건이 언론에 보도되었다. 또한 최근까지 대부분의 공기업에서 출입통제 카드로 취약한 MIFARE Classic을 사용하고 있다는 사실이 언론을 통해 지적된 바 있다. 이와 같은 사례들은 스마트카드 해킹이 사회적으로 큰 파장을 일으킬 수 있으며

취약한 NFC 스마트카드 기술이 현재까지도 사용되고 있다는 것을 반증한다. 이러한 취약점들은 잠재적인 위험요소이므로 스마트카드를 이용하는 관련 사업체들은 경각심을 가질 필요가 있다.

II. 관련 연구

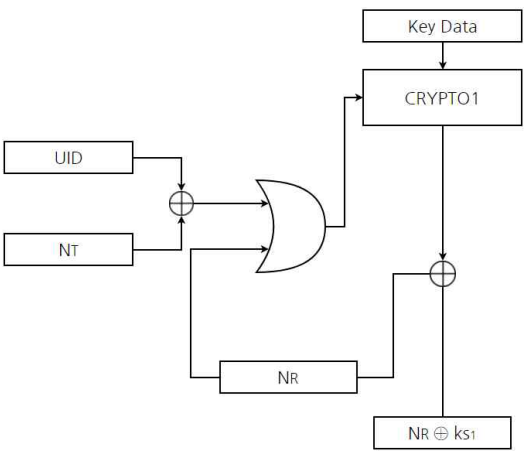
2.1 MIFARE Classic의 보안

MIFARE Classic 태그에는 기본적인 보안 기능이 존재한다. 가장 많이 사용되는 MIFARE Classic 1K에는 총 16개의 Sector가 존재하며 각 Sector는 4개의 Block으로 구성된다. 개별 Sector에 접근하기 위해서는 4번째 Block의 Key Data가 필요하다. 하나의 Key는 48Bit로 구성되며 정확한 Key Data로 인증을 수행하지 않는 경우 해당 Sector에 기록된 전체 Data를 읽을 수 없다.

	Byte Number															
Block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0																
1																
2																
3	Key A						Access Bits				Key B					

(그림2-1) MIFARE Classic Sector 메모리구조

MIFARE Classic 태그는 논리적·구조적인 취약점을 가지고 있다. 이러한 취약점은 MIFARE Classic이 CRYPTO1 암호화 기술을 사용한다는 점에서 기인한다. CRYPTO1는 Seed 값을 이용해 Key Stream을 생성하여 통신 및 인증을 수행한다. 이 때 Seed 값으로는 Sector의 Key Data가 이용되며 MIFARE Classic 스마트카드에 내장된 전체 암호화 스키마는 다음의 그림과 같다.



(그림2-2) MIFARE Classic 암호화 스키마

CRYPTO1 암호화 스키마를 적용한 MIFARE 스마트카드에는 다양한 취약점이 존재한다. 대표적으로 스마트카드에 내장된 Chip의 연산 기능은 성능 측면에서 제한이 존재해 일정한 시간마다 동일한 난수를 생성한다. 주기적으로 난수가 일치하는 한계점을 악용해 공격자는 LFSR Rollback Attack으로 Key Data를 알아낼 수 있다. 난수 일치 문제를 제외하고도 CRYPTO1 내부에서 사용되는 suc() 함수의 연산 취약점, N_R Data 노출 취약점, 알려진 명령어 취약점 등이 존재한다.

이와 같이 다양한 기술적 취약점이 존재한다는 점에서 공격자는 사전에 Key Data를 가지고 있지 않아도 다양한 해킹 기법을 이용해 MIFARE Classic 스마트카드에 저장된 모든 Sector의 Key Data를 단기간에 찾아낼 수 있다. 앞서 언급한 취약점은 Chip 단의 구조적인 취약점이며 해당 취약점을 실제 공격자 측면에서 악용할 수 있는 해킹 기법은 다음 절에서 제시할 것이다.

2.2 MIFARE Classic 해킹 기법

1) Default Key Search Attack

MIFARE Classic Chip 제조사는 카드를 제조할 때 모든 Sector의 Key Data를 Default Key로 설정하여 판매한다. 이후 실제로 고객에게 서비스를 제공하는 회사가 Default Key를 자신의 시스템에 맞는 Key로 변경하도록 권고된다. 일반적으로 사용되는 Default Key는 다음 그림과 같다.

0xFFFFFFFFFFFF	0xA0B0C0D0E0F0
0xA1B1C1D1E1F1	0xA0A1A2A3A45

(그림2-3) 일반적으로 사용되는 Default Key 목록

하지만 서비스를 제공하는 상당수의 회사는 이러한 Default Key를 변경하지 않고 그대로 사용한다. 또한 Key를 변경하더라도 주요 Sector에 한해서만 변경하므로 Default Key에 대한 전수조사 공격을 통해 상당수 카드를 해킹할 수 있다.

2) Nested Authentication Attack

공격자가 단 하나의 Sector에 대한 Key만 알

고 있어도 CRYPTO1의 구조적 취약점을 이용해 나머지 모든 Key를 알아낼 수 있는 공격 기법이다. 실제 공격을 수행할 때는 하나의 Key Data를 이용해 중첩된 인증을 수행한다. MFOC(Mifare Classic Offline Cracker)를 이용하면 컴퓨터 한 대와 RFID 리더기만 가지고도 공격을 수행할 수 있다. 사용할 수 있는 RFID 리더기는 Proxmark 3, ACR 122U 등이다. 국내 연구에서는 Proxmark 3를 사용한 연구 사례만 존재하나 이 논문에서는 Proxmark 3보다 3배 이상 저렴한 ACR 122U을 활용한 효율적 해킹 기법의 위험성을 알리고 이에 대한 경각심을 요구하고자 한다.

3) Snooping Attack

스마트카드와 리더기 간의 통신을 스니핑하여 Key Data를 찾아내는 공격으로 Proxmark 3을 이용해 스니핑 명령을 입력하여 수행할 수 있는 공격이다. 물론 스니핑 이후에 Sector의 Key를 정확히 찾을 수 있는지의 여부는 Key 분석 프로그램에 달려있다. 다만 일반적으로 대부분의 카드에서 실현 가능한 시간 내에 Key Data를 찾을 수 있다는 점에서 위협적인 공격 기법이다.

III. 국내에 현존하는 NFC 취약점

3.1 효율적인 공격 벡터에 대한 분석

앞서 MIFARE Classic의 세 가지 해킹 기법에 대해서 다루었다. 다만 기존의 선행 연구 및 해킹 사례들은 $320 정도에 해외에서 판매되고 있는 Proxmark3을 이용한 공격방법을 제시한 사례가 많다. 하지만 Proxmark3 Kit처럼 고가의 장비가 아니더라도 $30 미만의 PN532 모듈을 탑재한 ACR 122U 리더기를 이용해 MFOC 공격을 수행할 수 있다. 이 때 총 가격은 $80 이내이다.

결과적으로 악의적 해커들은 적은 금액을 투자하여 산업 전반에 사용되는 스마트 결제 카드를 해킹하여 막대한 양의 부당이득을 얻을 수 있는 것이다. 특히 2014년 언론에서 교통카드 해킹 사건을 다룰 때 제시되었던 약 50만 원에 유통되던 교통카드 해킹 장비와 비교했을 때 금액상으로 훨씬 효율적이다. 이는 현재는 스마트 폰이나 Proxmark 3와 같은 고가의 추가적인 장비 없이도 해킹이 가능하다는 것을 반증하며 구체적인 해킹 환경은 다음 그림에서 보여주는 바와 같다.



(그림2-4) ACR 122U를 이용한 스마트카드 해킹 환경

본 연구에서는 위 그림과 같이 구축된 해킹 환경에서 ACR 122U를 이용해 MFOC 기법으로 공격을 수행했다. 이 때 대부분의 스마트카드는 모든 Sector의 Key Data를 찾아내는 데 통상적으로 십 분 정도가 소요되었다. 이는 앞으로 제시할 국내 산업 전반의 보안상 취약한 스마트카드가 매우 짧은 시간에 해킹당할 수 있다는 의미다.

3.2 충전식 세탁카드 취약점

국내 상당수 대학교에서는 세탁기 이용을 학생들에게 유료로 제공하고 있으며 별도의 MIFARE Classic 카드를 구매하도록 하여 이에 금액을 충전하고 이용하도록 운영하고 있다. 본 연구에서는 대표적으로 C사에서 제공하는 대학교의 충전식 세탁카드를 조사해보았다. C사의 세탁카드는 교내 기숙사 로비에서 세탁카드를 구매하도록 한 뒤에 1,000원, 2,000원, 5,000원 단위로 충전하도록 하여 세탁카드를 이용하도록 하고 있다. 또한 세탁카드의 뒷면에는 자신의 계좌 정보를 기입해 수거함에 제출하면 남은 잔액을 환불 받을 수 있는 정책이 마련되어있는 상태다.

본 논문은 연구 과정에서 ACR 122U 리더기를 이용해 Nested Authentication Attack으로 숨겨진 모든 Sector의 Key를 확인하였다. 1,000원이 충전된 세탁카드의 Data는 다음의 그림과 같다.

	Byte Number															
Block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	88	00	01	00	00	FF	FF	07	80	69	FF	FF	FF	FF	FF	FF
1	88	00	01	00	00	FF	FF	07	80	69	FF	FF	FF	FF	FF	FF
2	88	00	01	00	00	FF	FF	07	80	69	FF	FF	FF	FF	FF	FF
3	66	77	63	6F	69	6E	FF	07	80	69	FF	FF	FF	FF	FF	FF

(그림2-5) 1,000원이 충전된 세탁카드 Data

이후에 금액을 1,000원 씩 충전해보며 Data Regularity를 찾은 결과 Sector 3의 Block 0, 1, 2에 남아있는 잔액에 대한 정보가 빅 엔디안(Big-Endian) 방식으로 입력된다는 사실을 발견했다. 총 잔액 99,990원까지 입금할 수 있으며 이 때의 공격 더미(Dummy)는 다음의 그림과 같다.

	Byte Number															
Block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	88	09	09	09	09	FF	FF	07	80	69	FF	FF	FF	FF	FF	FF
1	88	09	09	09	09	FF	FF	07	80	69	FF	FF	FF	FF	FF	FF
2	88	09	09	09	09	FF	FF	07	80	69	FF	FF	FF	FF	FF	FF
3	66	77	63	6F	69	6E	FF	07	80	69	FF	FF	FF	FF	FF	FF

(그림2-6) 세탁카드 부당 충전 더미 Data

더불어 실제 MIFARE Classic 카드는 3,000원 가량에 유통되고 있으나 C사의 세탁카드는 1,000원에 발급받을 수 있도록 제공하고 있다. 따라서 악의적인 해커가 해당 세탁 카드를 1,000원에 대량으로 구매한 뒤에 ACR 122U를 이용해 99,990원을 충전해 환불 받는다면 막대한 양의 부당이득을 얻을 수 있을 것이다. 특히 대학교 뿐 아니라 국내 대부분의 세탁 시설에서도 동일한 취약점을 갖는 MIFARE Classic 태그를 이용하고 있어 사업상 큰 위험에 처해있는 상황이다.

3.3 찜질방 후불제 결제카드 취약점

상당수의 찜질방에서는 편리한 결제를 위해 고객에게 MIFARE Classic Chip을 내장한 키 밴드(Key Band)를 착용하도록 한다. 다만 퇴실할 때 구매했던 물건을 한 번에 결제하는 시스템이라는 점에서 금액 변조로 인한 부당이득의 위험에 노출되어있는 상황이다. 해커는 사전에 ACR 122U 리더기를 활용해 찜질방 키 밴드에 내장된 태그의 Sector Key Data를 얻어낸 뒤 찜질방 안에서는 Android 스마트 폰으로 MCT(Mifare Classic Tool)를 이용해 사용 금액을 변조할 수 있다.

3.4 충전식 교통카드 취약점

정부, 학교, 단체 중 상당수는 소속원을 위해 다양한 교통 서비스를 제공하고 있다. 본 연구에서는 K 대학교의 등하교 목적으로 운영되는 셔틀버스의 취약점에 대해서 조사하였다. 조사 결과 해당 셔틀버스는 MIFARE Classic 스마트카드를 이용해 학생들이 선불로 금액을 충전하여 사용할 수 있도록 교통 서비스를 제공하고 있었다.

	Byte Number															
Block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	88	13	00	00	00	00	00	00	00	00	00	00	00	00	00	9B
1	88	13	00	00	00	00	00	00	00	00	00	00	00	00	00	9B
2	88	13	00	00	00	00	00	00	00	00	00	00	00	00	00	9B
3	47	4E	F4	55	94	7F	F0	07	80	69	FF	FF	FF	FF	FF	FF

(그림2-7) 5,000원이 충전된 교통카드 Data

본 연구에서는 우선적으로 5,000원의 금액이 충전된 상태의 교통카드를 대상으로 ACR 122U 리더기를 이용해 Nested Authentication Attack을 수행했다. 그 결과 얻은 Sector 2의 전체 Block에 대한 Data가 위와 같았다. 앞서 세탁카드 해킹 시연에서 수행했던 것과 마찬가지로 1,000원 씩 충전하며 Data Regularity를 확인한 결과 Sector 2의 Block 0, 1, 2의 앞의 두 Byte가 남아있는 잔액을 의미하며 마지막 Byte가 Check Sum으로 사용된다는 것을 파악할 수 있었다.

	Byte Number															
Block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	50	C3	00	00	00	00	00	00	00	00	00	00	00	00	00	13
1	50	C3	00	00	00	00	00	00	00	00	00	00	00	00	00	13
2	50	C3	00	00	00	00	00	00	00	00	00	00	00	00	00	13
3	47	4E	F4	55	94	7F	F0	07	80	69	FF	FF	FF	FF	FF	FF

(그림2-8) 교통카드 부당 충전 더미 Data

분석 결과 모든 Data는 16진수 형태의 리틀 엔디안(Little-Endian) 방식으로 기록되는 것을 확인했다. 이를 활용해 제작한 50,000원 잔액의 부당 충전 교통카드 더미 Data는 위와 같다. 실제 교통카드 리더기는 $C350_{16}$을 잔액으로 읽어 $50{,}000_{10}$로 해석하며 마지막 Byte는 다른 전체 Byte를 더한 뒤 100_{16}으로 나눈 나머지와 같다. 위 예시에서는 $(50 + C3)_{16} \% 100_{16} = 13_{16}$이므로 Check Sum으로 13이 마지막 Byte에 기록된다.

3.5 후불제 교통카드 취약점

국내 금융 회사인 K사는 MIFARE Classic 모델을 적용한 후불제 교통카드를 판매하고 있다. 이러한 후불제 교통카드로 대중교통 서비스를 이용하면 대중교통 운행을 이후에 통상 3일 내외의 기간을 단위로 정산이 이루어진다. 이러한 특징 때문에 대중교통을 이용한 고객은 실시간으로 금액을 부과 받지 않는다. 결과적으로 악의적 공격자는 UID Data 변경이 가능한 MIFARE Classic 카드를 이용해 타인의 후불제 교통카드를 복제한 뒤에 해당 카드로 대중교통 서비스를 이용할 수 있다. 이 경우 피해자가 피해 사실을 파악하기까지 상당한 시간이 소요되며 추적이 어렵다.

2017년 이전까지 각종 언론 및 논문에서 MIFARE 규격을 사용하는 선불제 교통카드의 취약점에 대해 언급이 된 바가 있다. 그 결과 많은 스마트카드 제조사는 이러한 취약점을 인식하고 보안이 향상된 MIFARE DESFire 등의 스마트카드를 이용하는 추세다. 하지만 아직까지 후불제 교통카드를 포함해 서버와 연동된 스마트카드 시스템에선 이처럼 취약한 NFC 모델인 MIFARE Classic을 그대로 사용하고 있어 사회 전반에 다양한 공격 벡터가 만연해 있는 상태이다. 따라서 악의적인 공격자가 다양한 공격 기법을 통해 부당 이득을 얻을 수 있어 각별한 주의가 필요하다.

3.6 주요시설 출입카드 취약점

국내에서 사용되는 상당수의 주거시설 및 공기업 출입카드의 리더기는 MIFARE Classic 태그를 지원한다. 본 논문에서는 L사의 사원 출입카드와 A 아파트의 인증카드를 대상으로 연구를 진행했다. 연구 결과 L사의 출입카드는 MIFARE Classic 태그를 사용하고 있으며 주거시설에서도 리더기가 스마트카드의 UID 및 Sector의 Data만을 확인해 주거인을 인증하고 있어, 단기간에 출입카드가 복제될 수 있는 위험에 처해있다.

본 연구에서는 사전에 인증된 카드를 MCT로 덤프 Data를 추출한 뒤에 MIFARE Classic 1K 공 카드(Blank Card)에 UID를 포함한 모든 Block Data를 복제하는 해킹 시연에 성공했다. 즉 공격자의 Android 스마트 폰을 타인의 스마트 카드에 가져다 대는 것만으로 복제 출입카드를 만들 수 있다. 이는 금전적 손실을 유발할 수 있는 2차 범죄와 직결되므로 보안상 큰 위험요소다.

IV. 결론

본론에서 제시한 바와 같이 현재 국내 산업 전반에서는 다양한 목적으로 MIFARE Classic을 적용한 스마트카드가 사용되고 있으며 이는 심각한 보안 취약점을 가지고 있다. 특히 최근에는 Proxmark 3와 같은 고가의 장비가 없이도 ACR 122U 등의 저렴한 장비로 공격을 수행할 수 있으므로 공격자에게 몹시 유리한 상황이다.

본 연구에서는 프로젝트 진행에 있어 한국정보기술연구원 차세대 보안리더 양성 프로그램(Best of the Best)의 지원을 받아 충전식 세탁카드, 찜질방 후불제 결제카드, 충전식 교통카드, 후불제 교통카드, 주요시설 출입카드를 대상으로 하여 NFC 스마트카드의 Block 내부 Data 및 UID 변조를 통한 부당이득 취득 사례 연구와 해킹 시연을 진행하였다. 제시된 모든 사례는 실제 국내의 개별 산업을 위협할 만큼 막대한 금전적 손실을 유발할 수 있다.

상당수 산업 시설에서는 최대한의 경제적 이득을 얻기 위해 MIFARE 스마트카드 모델 중에서 저렴한 Classic을 사용하고 있다. 하지만 추후의 해킹 피해를 방지하기 위해서는 MIFARE Classic 모델보다 높은 보안 수준을 갖추고 있는 MIFARE DESFire 혹은 Security Level 3을 적용한 MIFARE Plus 모델을 사용해야 한다. MIFARE Classic 모델을 계속 사용해야 하는 상황이라고 하더라도 최소한 알려진 키를 사용하지 않고, 금액 정보를 포함하는 Block에는 Encoding 및 Check Sum을 적용할 것을 권장한다. 결론적으로 현재까지의 NFC 스마트카드 기술은 그 편리함의 이면에 다양한 보안 취약점이 존재하기 때문에 본 연구를 통해 NFC 관련 서비스 개발사 및 사용 업체들의 경각심을 촉구하는 바이다.

[참고문헌]

[1] ISO/IEC 14443-3:2016, Identification cards - Contactless integrated circuit cards -

Proximity cards - Part 3: Initialization and anticollision

[2] Flavio D. Garcia, Gerhad de Koning Gans, Ruben Muijrers, Peter van Rossum, Roel Verdult, Ronny Wichers Schreur, and Bart Jacobs. “Dismantling MIFARE Classic”. In Computer Security - ESORICS 2008, v olume 5283/2008 of Lecture Notes in Computer Science, pages 97 - 11 4. Springer Berlin / Heidelberg, 2008.

[3] Nicolas T. Courtois. “The dark side of security by obscurity.” In International Conference on Security and Cryptography. Springer, 2009.

[4] 이예림. "Study on the Enhanced Analysis Method for MIFARE Classic Access Control ID Card." 석사, 국민대학교 일반대학원, n.d.

[5] 해킹으로 교통카드 잔액 조작…억대 ‘꿀꺽’, KBS News, 2014/06/13

[6] 해킹에 취약한 신분증 사용...공기업 출입통제 ‘구멍’, <보안뉴스>, 2014/09/17, http://www.boannews.com/media/view.asp?idx=42967

[7] MFOC GitHub Project: https://github.com/nfc-tools/mfoc

엔트로피를 기반으로 한 전자거래 바이오인증 강도 산정 방법 제시 및 비교 분석*

김지혜*, 사경진*, 염흥열**

*순천향대학교 융합서비스보안학과 **정보보호학과

Electronic transaction of Biometric Authentication Strength Level Estimation Based on Entropy

Ji-Hye Kim*, Kyeong-Jin Sa*, Heung-Youl Youm**

*Department of Convergence Service Security, **Department of Information Security Engineering, SoonChunHyang University.

요 약

요즘 전자거래의 사용이 점점 증가하면서 다양한 인증수단이 이용되고 있다. 그 중 사용자의 바이오 정보를 활용한 인증수단이 여러 방면에 활용되고 있다. 바이오 인증의 여러 요소들의 특징을 알아본다. 본 논문에서는 바이오 인증 강도를 산정 시 엔트로피를 이용하도록 제시한다. 또한 강도 값을 통해 현 금융권 각 은행이 사용하는 바이오 인증이 적절한 경우로 사용되는지 적정성을 비교해 보았다.

I. 서론

IT 기술의 성장으로 정보주체의 신분을 증명하는 기술인 바이오인증 기술은 서비스 환경 변화와 센서 모듈의 보급으로 전자거래의 이용이 급격하게 증가하였고 중요성이 대두되고 있다.

하지만, 아직까지 국내에서 바이오 인증수단의 보안 강도를 나누는 방법이 명확하지 않다. 이에 본 논문에서는 엔트로피를 활용하여 산정한 수치로 바이오 인증 수단의 보안강도를 측정하는 방법을 제시하고자 한다. 또한 강도를 바탕으로 현 금융권에서의 인증 수단 및 서비스가 적정하게 사용되는지 비교해보고자 한다.

본 논문의 구성으로는 2장은 바이오 인증 기술을 신체 기반과 행동 기반으로 다루고 이를 토대로 3장에서 엔트로피를 기반으로 한 바이오 인증 강도 산정 방법이 기술되고 현 금융서비스의 적정성을 검토한다. 그 후, 4장에서 결론을 맺는다.

* 본 연구는 과학기술정보통신부와 IITP의 정보통신·방송 연구개발사업의 일환으로 수행하였음.[R0190-17-2054]

II. 바이오 인증 기술

바이오인증기술은 인간의 고유한 생체정보를 여러 장치로 개인을 식별하거나 인증하는 기술이다. 생체정보는 정맥, 홍채, 얼굴, 지문의 신체적 정보 특징과 음성, 서명, 걸음걸이, 자판입력과 같은 행동적 정보 특성으로 구성된다.[1] [표 1]을 보며 각 기술 별 특징을 비교해 볼 수 있다.

Ⅲ. 엔트로피를 기반으로 한 바이오인증 강도 산정 방법

엔트로피란 실제 키값의 예측 가능성으로 정의된다. 인증 팩터 별 인증 강도를 나눠서 위의

[표 1] 바이오 인증 기술 별 특징 비교[2]

바이오인증		특징 & 장점	단점
생체 기반 인증	지문	지문인식 전용센서(광학, 정전, 초음파 등)를 활용하여 지문의 영상을 획득·인식하는 기술로 오작동 비율이 낮고, 빠른 응답속도 보유	간혹 지문인식이 잘 되지 않는 경우가 있으며, 위조, 복사 등 악용 가능성이 존재
	홍채	홍채의 고해상도 이미지 패턴을 적외선을 이용해 촬영하고, 디지털 변환 및 수학적 처리를 통해 개인의 고유한 특징을 추출	홍채 손상이 되지 않는 경우, 한 평생 이용가능하고, 오인율, 복제 가능성이 매우 낮으나, 사용자 거부감, 인식거리, 야외 주변 조도의 영향을 받는 한계가 있음.
	정맥	적외선을 통해서 혈관 투사 잔영으로 신원을 확인하는 방식	변형가능성이 적고 복제가 거의 불가능 하지만, 정맥분포 추출이 쉽지 않아 시스템 소형화가 어려우며 구축비용이 높음
	얼굴	얼굴전체보다는 50여 곳의 코, 입, 턱, 눈썹, 골격 등을 분석하여 인식하는 기술로 가장 자연스럽고 생체 정보 제공 거부감이 낮음	주변 환경 등에 따라 오작동 확률이 높고, 변조 및 위조를 통한 악용 가능성이 타 기술에 비해 높은 단점이 있음
행동 기반 인증	음성	친숙한 정보 전달 수단이라 거부감이 낮고 손발이 자유롭지 않은 상황에서도 정보 입력이 가능한 장점이 있음	주변 잡음, 사용자에 따른 인식률 차이가 있고, 음성 녹음 등 악용 우려 등 단점이 있음
	서명	서명과정 중 펜의 움직임, 속도, 압력과 서명의 모양 특징을 분석하여 인식하며 분실, 도난 위험이 없다는 장점이 있음	손의 사용이 불편한 경우 서명을 할 수 없고, 상황에 따른 인식률 차이가 있으며 서명 복제나 위조가 가능하다는 단점이 있음

보증 등급을 구현하였고 다음의 방법으로 엔트로피를 구함을 알 수 있다. 엔트로피는 지식기반, 생체기반의 방법으로 구하는 방법이 있다.

3.1 바이오 인증의 타인일치율(FMR)

생체기반 인증 팩터의 엔트로피 계산은 FMR(False Match Rate)를 사용하여 수치를 구할 수 있다. 여기서 FMR은 타인일치율로 오일치율이라고도 부른다. 승인되지 않는 타인증자의 바이오인식 정보가 인가자의 정보와 일치하는 경우로 잘못 판정되어서 승인이 된 비율이다.[3]

3.2 엔트로피 산정 방법

생체기반 인증 팩터의 엔트로피 계산은 FMR과 식 (2)를 참고하여 엔트로피 계산에 필요한 키 스페이스를 구하고, 식 (3)를 활용하여 엔트로피를 산정 할 수 있다.

$$p(false\ match) = FMR(1) \qquad \text{식 (1)}$$

$$k_b = 1/FMR(1) \qquad \text{식 (2)}$$

$$H_{\max} = \log_2 k_b \qquad \text{식 (3)}$$

다음의 [표 2]은 바이오 인증 방식을 FMR과 엔트로피를 계산한 수치를 표로 정리하였다. 바이오 인증 중 지문의 FMR은 [표 6]에 나타난 것처럼 0.001%의 값을 가진다. 식(1) - 식(3)을 이용하면 아래와 같이 식 (4)가 된다.

$$k_b = \frac{1}{0.001} \Rightarrow H_{\max} = \log_2\left(\frac{1}{0.001}\right) = 9.9658 \text{식 (4)}$$

다른 바이오 인증 요소 별 FMR을 식 (4)와 마찬가지로 식(1) - (3)까지를 활용하여 산정하면 아래와 같이 엔트로피가 나오게 된다.

[표 2] 바이오 인증의 엔트로피

방식 설명	FMR(%)	엔트로피
바이오 인증-얼굴	0.001	9.9658
바이오 인증-홍채	0.0001	13.2877
바이오 인증-지문	0.001	9.9658
바이오 인증-음성	0.01	6.6439
바이오 인증-정맥	0.0001	13.2877
바이오 인증-키스트로크	0.01	6.6439
바이오 인증-마우스움직임	0.43	1.2176

위의 [표 2] 나타난 엔트로피 값을 통해 바이오 인증 강도를 나누었다. 수치 값 10을 기준으로 그 이하는 강도 1로, 값이 10 이상의 경우는 강도 2로 보았다. 강도가 높을수록 더 안정적이다. 다음의 [표 3]에 바이오 인증 별 강도를 제

시한다. 전자거래에서 강도 값이 안정적인 전자거래의 지표가 될 수 있을 것이다.

[표 3] 바이오 인증 강도

방식 설명	엔트로피	인증강도
바이오 인증-얼굴	9.9658	1
바이오 인증-홍채	13.2877	2
바이오 인증-지문	9.9658	1
바이오 인증-음성	6.6439	1
바이오 인증-정맥	13.2877	2
바이오 인증-키스트로크	6.6439	1
바이오 인증-마우스움직임	1.2176	1

3.3 금융권에서 활용되는 바이오 인증 서비스의 강도 비교 분석

금융권에서도 다양한 바이오 인증 기술이 사용되고 있다. 이에 금융권에서 제공하는 서비스에 따른 인증 기술이 적절한 인증 강도를 가지는지 본 논문에서 제시한 엔트로피를 활용하여 비교해보고자 한다.

3.3.1 국내의 바이오 인증을 사용하는 금융서비스

아래의 표로 바이오 인증 기술로 국내 금융권에서 사용되고 있는 사례를 나타내었다. [표 4]은 사용자가 직접 ATM나 방문해야하는 거래와 모바일을 이용한 전자거래이다.

[표 4] 국내 주요 금융기관들의 바이오인식정보를 이용한 금융서비스[5]

오프라인 금융서비스			
구분	신한은행	기업은행	
서비스	디지털 키오스크	홍채인증 ATM	
생체정보	손바닥 정맥	홍채	
주요서비스	계좌조회, 계좌 신규발급, 금융상품가입, OTP 발금	ATM 입금, 출금	
기본거래여부	불필요	필요	
생체정보등록 시	비대면	대면	
인증강도	정맥(2)	홍채(2)	
모바일 금융서비스			
구분	기업은행	우리은행	농협은행
서비스	바이오정보기반 공인인증서비스	우리 삼성페이	NH스마트 금융센터
생체정보	지문	지문	지문
주요서비스	공인인증서 비밀번호 대체	ATM 입금, 출금	금융상품 가입
기본거래여부	필요	필요	필요
생체정보등록 시	비대면	비대면	비대면
인증강도	지문(1)	지문(1)	지문(1)

3.3.2 인증 수단 및 서비스 적정성 비교

[표 4]에 나타난 금융사별 인증 수단과 제공 서비스가 적절한지 비교해보았다. [표 4]에서 사용되는 정맥과 홍채의 엔트로피 값은 13.2877으로 인증 강도가 2이다. [표 4]에서 사용되는 인증 수단인 지문은 9.9658의 엔트로피 값을 가지고 인증 강도가 1이다. 제공하는 서비스의 위험 수준은 아래와 같다.

또한, [표 5]은 ITU-T X.1254와 NIST SP800-63을 참고하여 각 금융서비스의 위험 등급을 분류한 것이다.

[표 5] 금융서비스의 위험 수준[6, 7]

위험 등급	서비스
1등급	공인인증서 발급, OTP발급
2등급	계좌 이체(일정금액 이상), 계좌 발급, 공인인증서 갱신, 공인인증서 폐지, 공인인증서 비밀번호 대체, ATM 출금, 금융상품 가입
3등급	계좌 이체(일정금액 이하), ATM 입금, 계좌조회

또, 각 위험 등급에 따라 필요하다고 판단되는 인증 수준은 아래 [표 6]과 같다.

[표 6] 금융서비스의 위험 수준에 따른 인증 수준

위험 등급	인증 수준
1등급	바이오 인증 강도 2이상과 추가적인 다른 인증수단 또는 면대면 인증
2등급	바이오 인증 강도 2이상의 인증 수단
3등급	바이오 인증 강도 1정도의 인증 수단

본 논문에 나타나 [표 4]에서의 금융서비스만을 [표 5]과 [표 6]을 이용해 적절한 인증 수단을 제공하는지 비교 분석 하였다.

[표 7]의 오프라인 금융서비스에서는 ATM 입금과 계좌 조회는 위험 등급 3이고, 금융 상품 가입과 ATM 출금은 위험 수준 2등급이다. 각 서비스에 필요 인증 수준은 인증 강도 1과 2이상의 인증 수단인데, 사용하는 바이오 정보는 정맥과 홍채로 강도 2의 인증수단이다. 따라서, [표 7]의 오프라인 금융서비스는 적절한 인증 수준을 사용한다고 생각한다.

본 논문에 나타난 모바일 금융서비스에서는 생체 정보로 지문을 사용한다. 지문은 인증 강도 1로 산정하였고, 우리은행의 ATM 입금은 위험 등급 3등급으로 인증수단으로 지문이 적절하지만, 그 외의 ATM 출금, 금융상품 가입, 공인인증서 비밀번호 대체는 위험 등급 2등급으로 지문보다 높은 인증 강도를 가진 인증 수단이 필요하다고 판단된다.

[표 7] 금융서비스 인증 수단 적정성 유무

오프라인 금융서비스				
은행	**서비스**	**인증수단(강도)**	**위험 등급**	**적정성 유무**
신한	계좌조회	정맥 (2)	3	O
	계좌발급		2	O
	금융상품가입		2	O
	OTP 발급		1	X
기업	ATM 입금	홍채	1	O
	ATM 출금		2	O
모바일 금융서비스				
은행	**서비스**	**인증수단 (강도)**	**위험 등급**	**적정성 유무**
기업	공인인증서 비밀번호 대체	지문 (1)	2	X
우리	ATM 입금	지문 (1)	3	O
	ATM 출금		2	X
농협	금융상품 가입	지문(1)	2	X

Ⅳ. 결론

본 논문에서는 바이오 인증 기술들의 특징에 대해 분류하여 각각의 인증 기술을 알아보았다. 그리고 엔트로피의 산정 방식을 통해 각 바이오 인증 요소 별 강도를 나눴다. 이 결과 값을 적용한 실제의 전자거래 사례로 적정성을 비교하였다.

점점 바이오인증 기술의 사용이 증가하고 있지만 각 기술 별 명확한 강도 산정을 보기 어려웠다. 본 논문은 엔트로피의 산정 방식을 사용하여 바이오 인증 기술 별 보안 강도를 제시하고 이 강도에 따라 어떤 전자거래에 사용되는지를 알아보았다. 이를 통해 바이오 인증의 적절한 거래의 보안성을 높일 수 있다고 기대해 본다.

향후, 엔트로피의 산정 방식을 토대로 바이오 인증뿐만 아니라 다른 인증 수단에 적용하여 전자거래의 적절한 인증을 제공할 수 있게 되기를 바란다.

[참고문헌]

[1] 윤재호, 홍진실, 금융결제국 전자금융부, 바이오인증기술 최신 동향 및 정책과제, pp.1, August, 2016.

[2] 생체인식 기술 및 업계동향, IITP ICT SPOT Issue 2016-09호, 2016.10]

[3] TTA, TTAK.KO-12.0125, "지문인식 알고리즘 성능 시험 가이드라인", pp.7, 2009.12

[4] Blase Ur 외 11명, “Measuring Real-World Accuracies and Biases in Modeling Password Guessability”, USENIX Security Symposium, 2015.8

[5] 한승진, 금융권 바이오인식 기술 운영 현황, TTA Journal 165호, 2016.5]

[6] ITU-T X.1254, Entity authentication assurance framework, 2013.

[7] DRAFT NIST Special Publication 800-63-3, Digital Identity Guidelines, 2017.6.

악성코드 분석을 위한 안드로이드 DEX 바이너리 기반 Java API 추출 방법

안진웅*, 박정수**, 정수환***

*숭실대학교 정보통신공학과, **숭실대학교 융합소프트웨어학과, ***숭실대학교 전자정보공학부

API call extraction from Android Dex binary for malware analysis

Jinung Ahn*, Jungsoo Park**, Souhwan Jung***

*School of Information and Telecommunication Engineering, Soongsil University.
**Department of Software Convergence, Soongsil University.
***School of Information and Telecommunication Engineering, Soongsil University.

요 약

안드로이드 어플리케이션은 APK(Android Package Kit) 파일 안의 classes.dex로 대표되는 DEX(Dalvik EXcutable) 파일을 통해 실행된다. 따라서 DEX 파일을 분석하여 안드로이드 어플리케이션이 SDK(Software Development Kit)단에서 사용하는 모든 문자열과 클래스, 메소드에 대한 정보 등을 획득할 수 있으며, 어플리케이션에서 호출하는 모든 안드로이드 API에 대한 정보 또한 획득할 수 있다. 본 논문에서는 안드로이드 어플리케이션의 악성행위 분석을 위하여 DEX 바이너리를 정적 분석하여 어플리케이션에서 호출하는 안드로이드 API를 획득하는 시스템을 설계 및 구현한다.

I. 서론

스마트폰을 이용하는 전세계의 사용자 수는 지난 2016년 11월, 39억명을 달성하여 전세계 인구의 절반을 넘어섰고, 스마트폰과 태블릿 PC를 이용한 인터넷 이용률이 51.3%로 데스크탑을 이용한 인터넷 이용률인 48.7%를 추월하였다[1]. 전세계적으로 스마트폰 보급률이 높아짐에 따라, 안드로이드 악성 어플리케이션의 수도 증가하여 지난 2017년 1월에 발견된 모바일 악성코드의 수가 약 30만 건에 달한다[2]. 이에 따라 안드로이드 어플리케이션에 대한 분석 수요도 높아지고 있다.

본 논문에서는 안드로이드 어플리케이션의 DEX 파일 구조와 안드로이드 어플리케이션 정적 분석에 사용하는 기존 리버스 엔지니어링 툴에 대해 연구하고, DEX 파일의 정적 바이너리 분석을 통해 어플리케이션이 호출하는 API를 추출할 수 있는 시스템을 제안하고 구현하였다.

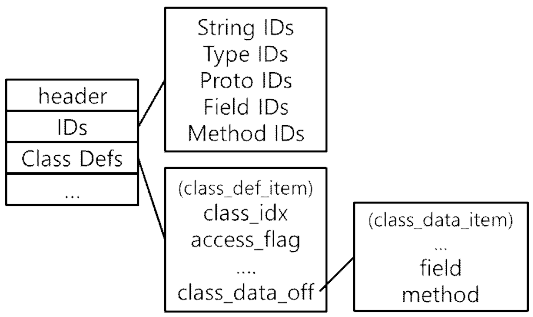
서론에 이어 본 논문의 2장에서는 관련 연구로 안드로이드 어플리케이션의 DEX 파일 구조와 어플리케이션을 정적 분석하는 기존 상용 툴에 대해 기술한다. 다음으로 3장에서는 본 논문에서 제안하는 안드로이드 API 호출 추출에 대해 기술한다. 마지막으로 4장에서는 결론 및 향후 연구 방향을 제시한다.

II. 관련 연구

2.1 DEX 파일 구조

안드로이드 어플리케이션은 classes.dex라는 DEX 파일이 존재하며, 어플리케이션 개발자가 작성한 코드가 bytecode 형태로 저장되어 있다. DEX 파일의 구조는 안드로이드 개발자 홈페이지에 공개되어있다[3]. DEX 파일은 [그림 1]과 같이 필드별로 구분하여 어플리케이션 내에서 사용하는 문자열과 타입, 클래스, 메소드에 대한 정보로 구성되어있다. DEX 파일의 헤더는

항상 70바이트로 고정되어 있으며 checksum, 파일 크기, 해시값 등의 정보와 함께 각 필드에 대한 offset과 크기를 지정하고 있다. 이 중 IDs와 Class Defs의 각 필드별 의미를 [표 1]에 정리하였다.



[그림 1] Dex 파일 구조

[표 1] DEX 파일 필드별 설명

필드명	설명
Dex Header	각 필드별 offset과 크기
String IDs	앱 내 모든 문자열
Type IDs	앱 내 사용하는 모든 타입
Proto IDs	함수의 파라미터와 리턴 타입
Field IDs	클래스 내 변수 정보
Method IDs	클래스 내 메소드 정보
Class DefS	클래스 선언부

특정 클래스의 메소드의 정보를 획득하기 위해선 다음과 같은 과정을 거친다. 먼저, DEX 헤더에서 class_defs 필드의 offset을 획득 후 class_defs 필드에 접근한다. class_defs 필드에서 해당 클래스의 class_data_off를 통해 class_data_item 포맷으로 저장된 클래스의 변수와 메소드 정보를 획득한다. 각 메소드는 method_ids 필드에 대한 offset을 가지고 있으며, method_ids 필드에서 proto_ids 필드에 대한 offset을 통해 메소드 파라미터와 리턴 타입을 획득한다. 파라미터와 리턴 타입은 type_ids 필드에 대한 offset으로 획득하고, 클래스명과 메소드명, 타입에 대한 명칭은 string_ids에서 획득한다.

2.2 안드로이드 어플리케이션 정적 분석 툴

안드로이드 어플리케이션을 분석하기 위한 방법으로 정적 분석과 동적 분석이 존재한다. 동적 분석은 실제로 어플리케이션을 실행시키며 동작 정보를 수집하여 분석하는 반면, 정적 분석은 어플리케이션을 실행시키지 않고 파일만으로 분석하는 방식이다. 안드로이드 어플리케이션을 정적 분석할 수 있는 상용 툴이 존재하며, 대표적으로 dex2jar와 apktool이 있다. dex2jar는 classes.dex 파일을 jar 파일로 변환하는 리버스 엔지니어링 툴이다[4]. 일반적으로 jar 파일을 GUI 형태로 소스코드를 볼 수 있도록 해주는 jd-gui 툴[5]과 같이 사용하여 안드로이드 어플리케이션을 분석할 수 있다.

apktool은 안드로이드 어플리케이션 파일인 apk 파일에서 디컴파일을 수행하는 리버스 엔지니어링 툴이다[6]. apktool은 apk 파일을 디컴파일하여 smali 코드로 변환하며, smali 코드를 java 코드로 변환하는 baksmali 툴과 같이 사용하여 안드로이드 어플리케이션을 분석할 수 있다.

```
C:₩Users₩Cracker₩Downloads>java -jar apktool_2.3.0.jar d unicode_apk.apk
I: Using Apktool 2.3.0 on unicode_apk.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
S: WARNING: Could not write to (C:₩Users₩Cracker₩AppData₩Local₩apktool₩framework), using C:₩Users₩Cracker₩AppData₩Local₩Temp₩ instead...
S: Please be aware this is a volatile directory and frameworks could go missing, please utilize --frame-path if the default storage directory is unavailable
I: Loading resource table from file: C:₩Users₩Cracker₩AppData₩Local₩Temp₩1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
Exception in thread "main" brut.androlib.AndrolibException: brut.directory.DirectoryException: Error copying file:  ↔
        at brut.androlib.Androlib.decodeUnknownFiles(Androlib.java:224)
        at brut.androlib.ApkDecoder.decode(ApkDecoder.java:165)
        at brut.apktool.Main.cmdDecode(Main.java:160)
        at brut.apktool.Main.main(Main.java:72)
Caused by: brut.directory.DirectoryException: Error copying file:  ↔
        at brut.directory.DirUtil.copyToDir(DirUtil.java:88)
        at brut.directory.AbstractDirectory.copyToDir(AbstractDirectory.java:208)
        at brut.androlib.Androlib.decodeUnknownFiles(Androlib.java:217)
        ... 3 more
Caused by: java.io.FileNotFoundException: unicode_apk₩unknown₩ ↔        (파일 이름, 디렉터리 이름 또는 볼륨 레이블 구문이 잘못되었습니다)
        at java.io.FileOutputStream.open0(Native Method)
        at java.io.FileOutputStream.open(Unknown Source)
        at java.io.FileOutputStream.<init>(Unknown Source)
        at java.io.FileOutputStream.<init>(Unknown Source)
        at brut.directory.DirUtil.copyToDir(DirUtil.java:84)
        ... 5 more

C:₩Users₩Cracker₩Downloads>
```

[그림 2] 디컴파일 방해로 apktool 수행 실패

dex2jar와 apktool 모두 디컴파일 방해기법이 적용된 어플리케이션의 경우 [그림 2]와 같이 정상적인 디컴파일이 수행되지 않는다는 단점이 있다.

III. 제안 기법

3.1 API 호출 목록 추출 시스템

앞서 2장에서 서술한 바와 같이, DEX 파일은 필드별로 클래스와 메소드 정보를 담고 있고, DEX 헤더에서 필드별 offset과 크기를 지정하고 있다. 따라서 DEX 파일의 헤더 정보를 바이너리 값으로 읽어들인 후 분석하면 어플리케이션 내부에서 사용되는 각 클래스와 메소드 내에서 동작하는 실행코드를 확인할 수 있다. 메소드를 호출하는 특정 Dalvik 명령어와 인자값을 통하여 해당 앱에서 호출되는 api들을 획득할 수 있다. 메소드를 호출하는 Dalvik 명령어는 총 열 개로, 모두 invoke-* 로 시작한다는 특징을 가진다[7]. 해당 Dalvik 명령어와 명령어별 opcode를 [표 2]에 정리하였다.

[표 2] 함수를 호출하는 Dalvik 명령어

명령어	opcode(hex)
invoke-virtual	6E
invoke-super	6F
invoke-direct	70
invoke-static	71
invoke-interface	72
invoke-virtual/range	74
invoke-super/range	75
invoke-direct/range	76
invoke-static/range	77
invoke-interface/range	78

본 논문에서 제안한 안드로이드 API 호출 목록 추출 시스템을 구현하여 [그림 2]와 같이 apk 파일 내에 존재하는 클래스와 메소드에서 호출하는 안드로이드 API 호출을 추출하였다.

본 논문에서 구현한 시스템은 어플리케이션의 DEX 파일을 바이너리 값으로 읽어들인 후 파싱하기 때문에 디컴파일 방해 기법이 적용되어 dex2jar 혹은 apktool이 디컴파일하지 못하는 어플리케이션도 분석이 가능하다는 장점을 가진다.

```
tester@ubuntu:~/tester/getAPI$ python getAllAPI.py
java.lang.StringBuilder ------ <init>
java.lang.StringBuilder ------ append
java.lang.StringBuilder ------ append
java.lang.StringBuilder ------ toString
java.io.File ------ exists
java.io.File ------ isFile
java.io.File ------ <init>
java.io.File ------ <init>
java.io.File ------ exists
java.io.File ------ exists
java.io.File ------ exists
java.io.File ------ exists
java.io.File ------ exists
java.lang.Runtime ------ getRuntime
java.lang.Runtime ------ exec
```

[그림 3] API 호출 목록 추출 일부

3.2 API를 통한 어플리케이션 악성행위 분석

앞서 추출한 안드로이드 API를 통해, 해당 안드로이드 어플리케이션의 동작을 파악할 수 있다. 특히 악성 어플리케이션의 경우 민감한 정보를 유출하는 유형이 많아 민감한 정보를 수집하는 API와 정보를 유출하는 API가 동시에 존재할 경우 어플리케이션이 악성 행위를 수행한다고 의심할 수 있다. 예를 들어, android.telephony.TelephonyManager 클래스의 getDeviceId 메소드를 호출한 경우 민감한 정보인 안드로이드 디바이스의 식별 번호를 수집하려 하는 의도를 파악할 수 있다. 혹은, android.media.projection.MediaProjection 클래스의 createVirtualDisplay 메소드를 호출한다면, 가상 화면을 통해 사용자에게 최초 1회를 제외하고 알리지 않고 화면을 캡처할 수 있기 때문에 개인정보가 유출될 수 있는 위험성을 지닌다. 정보 유출 관련 API로는 네트워크를 통하여 정보를 전송할 수 있는 API들이 있다. 많이 사용되는 예로, HTTP 통신에 주로 사용되는 java.net.URLConnection 클래스의 openConnection 메소드, 소켓 통신에 사용되는 java.net.Socket 클래스의 생성자인 <init> 메소드를 예로 들 수 있다. 어플리케이션에서 추출한 API가 이러한 정보 추출 API와 정보 유출 API가 동시에 존재할 경우 악성행위를 의심할 수 있는 근거가 된다.

3.3 한계점

본 논문에서 구현한 시스템은 안드로이드 어플리케이션의 DEX 파일을 정적 분석하여 안드로이드 API 호출을 추출하기 때문에, 자바로 구현된 안드로이드 SDK 영역만 분석할 수 있다. 따라서 JNI(Java Native Interface)를 사용한 Native 영역의 분석이 불가능해 SO 파일에서 호출하는 시스템콜에 대한 정보는 추출할 수 없다는 한계점을 가진다. 이를 보완하기 위해서 추가적인 연구를 통해 Native 영역에서의 시스템콜 호출에 대한 정보를 추출하는 모듈을 추가하여야 한다.

IV.결론

최근 지능화된 안드로이드 어플리케이션은 정적 분석되는 것을 막기 위하여 정적 분석 방지 난독화 기법을 적용하고 있다. 하지만 어플리케이션이 난독화되더라도 classes.dex 파일을 추출하여 정적으로 바이너리 분석을 시행하면 어플리케이션에서 사용하는 SDK단의 클래스와 메소드, 안드로이드 API 호출 정보를 획득할 수 있다.

본 논문에서는 정적 분석 방지 난독화가 적용된 안드로이드 어플리케이션에서도 어플리케이션 실행파일인 APK 파일 안의 DEX 파일을 정적 바이너리 분석하여 안드로이드 SDK에서 호출하는 API를 추출할 수 있는 시스템을 설계 및 구현하였다. 이를 활용하여 추출되는 안드로이드 API를 통해 안드로이드 어플리케이션의 동작 의도를 파악하고, 나아가 어플리케이션의 악성행위 유무를 판단하기 위한 기초자료로 활용될 수 있을 것이다.

Acknowledgment

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 대학ICT연구센터육성 지원사업(IITP-2017-2012-0-00646)과 산업통상자원부 및 한국산업기술평가관리원의 우수기술연구센터(ATC)사업의 일환으로 수행하였음. [10045904, 클라우드 컴퓨팅 환경하에서 정보보안 서비스를 제공하기 위한 SecaaS 프레임워크 원천기술 개발과 이를 이용한 1Gbps급 모바일 정보유출방지 서비스 구축]

[참고문헌]

[1] “2017년 모바일 트렌드 전망,” KT경제경영연구소, January, 2017.

[2] “ASEC REPORT,” AhnLab, vol. 85, January, 2017.

[3] David Ehringer, “The dalvik virtual machine architecture,” Techn. report, vol.4, no.8, March, 2010.

[4] “안드로이드 애플리케이션에 대한 역공학 방지 기법,” 보안공학연구논문지, vol. 1, no.1, pp.41-50, February, 2013.

[5] Desnos, Anthony, and Geoffroy Gueguen. "Android: From reversing to decompilation," Proc. of Black Hat Abu Dhabi, pp.77-101, 2011.

[6] “안드로이드 기반 모바일 메신저 취약점 분석 및 시큐어 코딩 적용방안,” 한국컴퓨터정보학회 동계학술대회논문집, vol.22, no.1, January, 2014.

지속 인증을 위한 사용자 행위 정보 기반 특징 요소 별 성능 분석 연구*

구예은, 이호연, 권태경
연세대학교 정보보호연구실

Performance analysis of features based on user behavior information for continuous authentication

Yeeun Ku, Hoyeon Lee, Taekyoung Kwon
Information Security Lab., GSI, Yonsei University.

요 약

최근 사용자의 자각을 최소화하는 동시에 높은 보안성을 제공하기 위한 지속 인증 기술 연구가 활발하게 진행되고 있다. 하지만, 기존 연구들은 지속 인증의 정확도를 향상시키는 것에 초점이 맞춰져 있기 때문에 인증에 활용되는 행위 특징 요소들에 대한 분석은 상대적으로 미비하다고 할 수 있다. 따라서 본 논문에서는 특징 요소의 성능을 개별적으로 분석함으로써 사용자 식별력이 우수한 특징 요소와 해당 특징 요소에 가장 적합한 머신러닝 알고리즘을 객관적으로 제시하고자 한다.

I. 서론

스마트기기의 다양한 기능과 서비스 제공으로 인해 사용자의 민감한 정보가 기기에 저장되었고, 스마트기기의 안전한 사용을 위한 사용자 인증 기술의 중요성이 점차 부각되고 있다. 때문에 기존 인증 기술인 패스워드, PIN 뿐만 아니라 지문, 얼굴, 홍채 인식과 같은 생체 기반 인증 기술이 모바일 환경에 함께 적용되고 있다. 이러한 스마트기기를 대상으로 한 다양한 인증 기술의 적용은 더 안전한 모바일 환경을 제공하기 위한 노력이라고 할 수 있다.

하지만, 기존 인증 방식은 새로운 세션이 시작되는 시점마다 인증을 요구하는 방식을 사용한다. 이 방식은 세션이 끝나기 전 악의적인 사용자들에 의해 세션이 가로채질 수 있는 취약점이 존재한다. 이를 해결하기 위해 세션이 유지되는 상태에서 사용자의 자각을 요구하지 않으면서도 지속적으로 안전한 환경을 제공할 수 있는 지속 인증 방식에 대한 연구가 꾸준히 진행되고 있다.

지속 인증 관련 연구는 사용자의 논리적 행위 정보를 활용한 인증 기술 연구와 스마트기기에 부착된 센서를 활용한 지속 인증 연구가 꾸준히 진행되고 있다[1-10]. 하지만, 기존 연구들은 특징 요소들을 조합하여 사용자를 식별하기 위한 연구에 초점이 맞추어져 있을 뿐, 각 특징 요소(feature)들의 성능을 판단할 수 있는 객관적 근거를 제시하는 연구는 부족하다고 할 수 있다.

이에 대해 본 논문은 스마트기기 사용자의 행위 정보로 활용될 수 있는 특징 요소에 대한 성능을 분석하고자 한다. 본 논문의 목적은 머신러닝 기반의 특징 요소 별 성능 분석을 통해

* “본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 대학 ICT연구센터육성지원사업의 연구결과로 수행되었음”(IITP-2017-2016-0-00304) 또한, 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2017-0-00380, 차세대 인증 기술 개발)

사용자 식별력이 우수한 특징 요소를 선별하는 데 있다. 또한 실험을 통한 성능 분석 결과는 지속 인증 기술을 개선시키는데 활용될 수 있을 것이다.

본 논문의 구성은 다음과 같다. 2장에서 사용자 행위 정보를 통해 얻어지는 특징 요소 별 성능 분석 방법을 설명하고, 3장에서 특징 요소들의 성능을 분석한다. 4장에서는 관련 연구를 소개하며, 마지막 5장에서 결론을 맺는다.

II. 특징 요소 별 성능 분석 방법

본 장에서는 머신러닝 알고리즘을 이용하여 특징 요소의 성능을 개별적으로 분석하는 방법에 대해 다룬다. 성능 분석은 '데이터 수집', '특징 요소 추출', '학습 및 성능 분석'의 3단계를 거쳐 진행된다.

2.1 데이터 수집

데이터 수집 단계에서는 사용자가 스마트기기 화면에 패턴을 그리는 동안 발생하는 터치 이벤트, 가속도계 및 자기장 센서 데이터를 수집한다. 데이터는 사용자의 패턴 입력이 시작되는 시점부터 끝나는 시점까지 수집되며, 사용자의 스마트기기 사용에 방해가 되지 않도록 백그라운드 상에서 데이터 수집이 이뤄진다.

2.2 특징 요소 추출

특징 요소 추출 단계는 수집된 데이터로부터 사용자의 특징 요소를 추출하는 단계이다. 패턴을 구성하는 임의의 한 점이 입력된 시점부터 다른 한 점이 입력되기 직전까지를 유닛(unit)

이벤트 타입	내용
터치 이벤트	터치 이벤트 발생 수
	터치 이벤트 압력/면적의 평균/표준 편차/최댓값/최솟값
	슬라이드 속도의 평균/표준 편차/최댓값/최솟값
	이동 속도/시간/슬라이드 각도
가속도계/자기장 센서	기기의 x, y, z축에 작용하는 가속도/기울기 값 평균
	기기의 x, y, z축에 작용하는 양(+)의 가속도/기울기 값 개수

표 1. 특징 벡터의 구성 요소

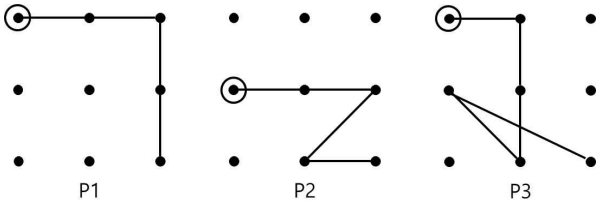


그림 1. 실험 대상 패턴

으로 정의하며, 각 유닛 별로 사용자 특징 요소를 추출한다. 특징 요소는 터치 이벤트 수집 데이터에서 16개, 가속도계와 자기장 센서 데이터에서 12개의 특징 요소가 추출되어 총 28차원의 특징 벡터로 구성된다. 추출되는 특징 요소는 표 1과 같다.

2.3 학습 및 성능 측정

학습 및 성능 측정 단계에서는 특징 요소의 식별 정확도를 측정하며, 측정 결과를 토대로 각 특징 요소들의 성능을 분석한다. 특징 요소 추출 단계에서 얻어진 각 특징 요소들은 성능 분석을 위한 데이터로 활용되며, 머신러닝 알고리즘을 통해 특징 요소들의 성능을 분석하였다.

III. 실험 및 결과

실험을 통해 사용자 식별력이 우수한 특징 요소와 해당 특징 요소에 가장 적합한 머신러닝 알고리즘을 도출한다. 본 장에서는 성능 검증을 위한 실험 환경 및 검증 방법에 대해 서술한다.

3.1 실험 환경

실험을 위해 어플리케이션 형태의 패턴 락 인증 시스템을 구현하였으며, 랩 스터디 형식으로 총 8명을 대상으로 실험을 진행하였다. 실험에 사용되는 패턴은 최소 4개 이상의 점과 1개 이상의 각도를 가져야 한다는 조건을 설정하였으며, 실험에는 이를 만족하면서 각도의 개수가 각기 다른 그림 1의 3가지 패턴을 사용하였다.

한 사람당 3가지 패턴을 모두 입력하도록 요구하였으며, 패턴 별로 25회씩 입력을 진행하여 총 600개의 데이터를 수집하였다. 이때 최초 입력된 5개의 데이터는 사용자의 행위 정보를 정확하게 반영하지 못하므로 실제 실험에는 이를 제외한 총 480개의 데이터를 사용하였다.

데이터는 사용자가 패턴 입력을 수행할 때

동시 수집되며 모든 사용자들에게 자연스럽게 앉은 상태에서 패턴 입력을 수행하도록 요구하였다. 데이터 수집 기기로 Nexus 5를 사용하였으며, 안드로이드 6.0.1 버전에서 실험을 진행하였다.

3.2 실험 및 검증 방법

실험 데이터는 특징 요소 추출 단계를 거쳐 특징 요소로 가공되며, 가공된 특징 요소는 머신러닝 알고리즘 구동을 위한 입력 데이터로 사용된다. 각 특징 요소 별로 하나의 머신러닝 알고리즘에 대해 학습 및 분류를 진행하였다.

실험에는 총 6가지의 머신러닝 알고리즘이 사용되었다. 지도 학습 방식의 k-Nearest Neighbors(k-NN), Gaussian Naive Bayes(GNB), Support Vector Machine(SVM), Decision Tree(DT), Neural Network(NN), Random Forest(RF) 분류 알고리즘이 사용되었으며, python 기반의 머신러닝 라이브러리인 Scikit-learn을 이용하였다.

검증은 특징 요소의 사용자 식별력을 객관적으로 검증하기 위해 k-fold cross validation을 이용하였다. k-fold cross validation이란 데이터 집합을 총 k개로 나누어 k-1개의 집합을 학습 데이터로, 나머지 1개의 집합을 검증 데이터로 사용하여 머신러닝 모델의 성능을 측정하는 방법이다. 실험에는 20-fold cross validation을 이용하였으며, 분류 결과를 기반으로 특징 요소의 사용자 식별력을 검증하였다.

검증 척도로는 EER 값을 이용하여 사용자 식별력을 측정하였다. EER 값은 FAR과 FRR의 값의 평균값으로 생체 인식 기술의 성능을 검증하고자 할 때에 이용되는 척도이다. EER 값이 낮을수록 성능이 좋은 특징 요소로 판별된다.

$$EER(\%) = \frac{FAR + FRR}{2}$$

3.3 실험 결과 분석

추출된 특징 요소 별로 머신러닝 알고리즘을 구동시킨 후 FAR과 FRR을 산출한다. 사용자 식별력이 우수한 특징 요소의 상위 그룹을 판별하기 위해 EER이 낮은 하위 5개의 특징 요소에 대해서 실험 결과를 분석하였다.

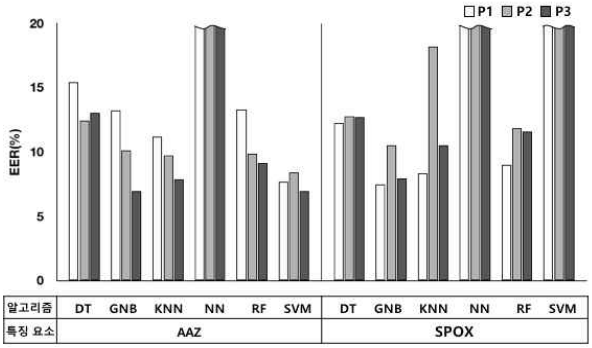


그림 2. 특징 요소 별 알고리즘 성능 비교

특징 요소의 성능을 분석한 결과, 28개의 특징 요소 중 기기의 z축에 작용하는 가속도 값 평균(AAZ)과 기기의 x축에 작용하는 양의 기울기 값 개수(SPOX)가 실험 패턴에 상관없이 우수한 성능을 보였다(그림 2). 또한 패턴에 종속되어 우수한 성능을 보이는 특징 요소들이 존재하였으며, 이에 대한 목록은 표 2와 같다.

실험 패턴에 관계없이 특징 요소 AAZ에 SVM 알고리즘을 적용한 경우 P1, P2, P3에서 각각 7.68%, 8.39%, 7.68%의 EER 값을 보였으며, 특징 요소 SPOX에 GNB 알고리즘을 적용한 경우 P1, P2, P3에서 각각 7.46%, 10.58%, 7.95%의 EER 값을 보였다.

실험 결과를 토대로 패턴에 관계없이 우수한 성능을 보인 AAZ, SPOX은 지속 인증에 쓰일

특징 요소	P1	P2	P3
터치 이벤트 발생 수(NTM)			7.01
이동 속도(TSS)	9.51	11.38	
이동 시간(TD)			7.01
기기의 y축에 작용하는 가속도(AAY)	9.96	11.29	
기기의 y축에 작용하는 양의 기울기 값 개수(SPAY)			7.01
기기의 z축에 작용하는 양의 기울기 값 개수(SPAZ)	10.27		7.01
기기의 x축에 작용하는 기울기(AOX)		5.85	7.54
기기의 y축에 작용하는 기울기(AOY)		9.78	

표 2. 패턴에 종속적인 특징 요소 별 EER(%)

수 있는 특징 요소인 것으로 보인다. 특히 최대 성능을 보인 알고리즘 또한 패턴에 종속되지 않는 결과를 보인 것을 볼 때, 지속 인증에 우선적으로 고려될 수 있는 특징 요소인 것으로 판단된다.

IV. 관련 연구

안전한 모바일 환경을 위해 다양한 인증 기술이 스마트기기에 적용되고 있지만, 세션이 시작될 때에만 사용자 인증을 요구하며 세션이 끊기면 매번 인증을 수행해야하기 때문에 충분한 안전성과 편의성을 제공하지 못한다. 이러한 문제점을 해결하기 위한 노력으로 행위 기반 지속 인증 기술에 대한 관심이 높아지고 있다.

Li 등은 사용자의 동작을 프로파일링 하고, 이를 활용해 사용자를 지속적으로 인증하는 행위 기반 인증 기술 연구를 수행하였다[1]. Feng 등은 터치스크린 상에서 제스처를 할 때 나타나는 인증 요소를 활용하는 연구를 수행하였다[2]. 이외에도 터치 이벤트 기반의 여러 인증 요소들을 활용하기 위한 연구가 진행되었다[3, 4, 5, 6]. 또한 가속도계, 자이로, 자기장 센서 등의 센서 를 활용하는 연구 또한 활발히 진행되어 왔으며[7, 8, 9], Buriro 등은 센서 데이터와 함께 추가적으로 터치 행위를 인증 요소로 활용해 정확도를 개선하였다[10].

V. 결론 및 향후 연구

본 논문에서는 안드로이드 패턴 락 시스템에서 획득한 사용자의 행위 정보를 통해 특징 요소를 추출하였다. 그리고 머신러닝 알고리즘을 활용해 각 특징 요소 별 개별적 성능을 분석하는 실험을 진행하였다. 논문 결과는 지속 인증의 성능 개선을 위해 높은 식별력을 보이는 특징 요소를 선별하는 지표로 활용될 수 있을 것으로 기대된다.

하지만 본 논문에서는 실험 패턴 개수의 한계로 인해 패턴의 다양성에 상관없이 특징 요소의 식별력에 일관성을 보이는지 충분히 증명하지 못했다. 패턴에 종속되지 않으면서 최적의 성능을 보일 수 있는 특징 요소를 선별할 필요가 있을 것으로 보이며, 이에 대한 향후 연구를 진행할 예정이다.

[참고문헌]

[1] Li, Fudong, et al. "Behaviour profiling for transparent authentication for mobile devices." European Conference on Cyber Warfare and Security. Academic Conferences International Limited, 2011.

[2] Feng, Tao, et al. "Continuous mobile authentication using touchscreen gestures." Homeland Security (HST), 2012 IEEE Conference on Technologies for. IEEE, 2012.

[3] Li, Lingjun, Xinxin Zhao, and Guoliang Xue. "Unobservable Re-authentication for Smartphones." NDSS. 2013.

[4] Draffin, Benjamin, Jiang Zhu, and Joy Ying Zhang. "KeySens: Passive User Authentication through Micro-behavior Modeling of Soft Keyboard Interaction." MobiCASE. 2013.

[5] Frank, Mario, et al. "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication." IEEE transactions on information forensics and security 8.1 (2013): 136-148.

[6] Xu, Hui, Yangfan Zhou, and Michael R. Lyu. "Towards continuous and passive authentication via touch biometrics: An experimental study on smartphones." Symposium On Usable Privacy and Security, SOUPS. Vol. 14. 2014.

[7] Shi, Weidong, et al. "Senguard: Passive user identification on smartphones using multiple sensors." Wireless and Mobile Computing, Networking and Communications (WiMob), 2011 IEEE 7th International Conference on. IEEE, 2011.

[8] Zheng, Nan, et al. "You are how you touch: User verification on smartphones via tapping behaviors." Network Protocols (ICNP), 2014 IEEE 22nd International Conference on. IEEE, 2014.

[9] Sitová, Zdeňka, et al. "HMOG: New behavioral biometric features for continuous authentication of smartphone users." IEEE Transactions on Information Forensics and Security 11.5 (2016): 877-892.

[10] Buriro, Attaullah, et al. "Hold and Sign: A Novel Behavioral Biometrics for Smartphone User Authentication." Security and Privacy Workshops (SPW), 2016 IEEE. IEEE, 2016.

2017 한국정보보호학회 동계학술대회

E3 모바일 보안(III)

이 문 규 (인하대)

- **안드로이드 보안 솔루션 검증 결과 변조를 통한 기능 우회 취약점의 대응방안 연구**
  김규헌, 한미정, 유재욱, 장준영, 진호용, 지한별 (KITRI BoB), 김경곤 (고려대학교), 신정훈 (세종대학교)
- **안드로이드 어플리케이션 자동 분석 보고서 제작 프로그램 설계 및 구현**
  천호진, 강성은, 정수환 (숭실대학교)
- **소프트웨어 권한 분리에 대한 연구**
  방인영, 이영한, 안선우, 백세현, 백윤흥 (서울대학교)
- **온라인 게임 보안을 위한 매크로의 우회 기법과 대응방안 연구**
  박영서, 정담지, 김지은, 김나경, 김연주, 전정훈 (동덕여자대학교)

안드로이드 보안 솔루션 검증 결과 변조를 통한 기능 우회 취약점의 대응방안 연구*

김규헌*, 한미정*, 유재욱*, 장준영*, 진호용*, 지한별*, 김경곤**, 신정훈***

*KITRI 차세대 보안리더 양성 프로그램(Best of the Best),
**고려대학교 정보보호대학원, ***세종대학교 컴퓨터공학과

A Study on Countermeasure for Bypassing Android Security Solution through Manipulating Return Value

Kyuheon Kim*, Mijeong Han*, Jaewook You*, Junyoung Jang*,
Hoyong Jin*, Hanbyeol Ji*, Kyounggon Kim**, Jeonghoon Shin***
*Best of the Best, Korea Information Technology Research Institute,
**Graduate School of Information Security, Korea University,
***Department of Computer Engineering, Sejong University

요 약

안드로이드 애플리케이션을 사용하는 기업, 개인을 표적으로 다양한 유형의 사이버 공격이 활발히 진행되고 있다. 이 중 금융 서비스의 경우 금전적인 피해를 끼칠 수 있기 때문에 더욱 높은 수준의 보안성이 확보되어야 한다. 이러한 금융 애플리케이션은 보안성 확보를 위해 여러 검증기능을 수행하는 보안 솔루션을 탑재하였으며 그 기능에 대한 다양한 연구가 선행되었다. 그러나 아무리 정교한 검증을 수행하더라도 애플리케이션 개발이 완료된 후 보안 솔루션이 탑재되기 때문에, 애플리케이션과 솔루션이 독립적으로 동작하여 솔루션의 검증 결과값 위·변조만으로 검증기능의 우회가 가능하다. 따라서 본 논문에서는 이에 대한 대응방안으로 토큰 발급을 통한 클라이언트 요청 값 검증을 제시하여, 검증 결과값 위·변조를 통한 우회 취약점을 방지하며 보안 솔루션을 사용 중인 안드로이드 애플리케이션의 보안성 향상에 기여하고자 한다.

I. 서론

한국은행에 따르면, 2017년 6월말, 전체 인터넷 뱅킹 등록 고객 중 63.8%, 즉 절반 이상의 고객이 스마트폰 뱅킹을 이용하고 있다.[1] 뿐만 아니라 2014년 말부터 금융 규제가 완화되며 생겨난 간편결제 서비스는 새로운 결제 수단으로 각광 받으며, 모바일 환경을 기반으로 급격히 성장하였다.

이러한 모바일 기반 금융 서비스들은 높은 편리성과 접근성을 장점으로 많은 사람들이 일상생활에서 빈번히 이용하고 있다. 전자금융 서비스는 개인 및 기업의 자산과 직접적으로 연관되어 있기 때문에 더욱 높은 수준의 보안성이 요구되며, 만약 사이버 공격이 이루어졌을 시엔 금융정보 탈취나 부정결제 등 매우 심각한 피해가 우려된다.[2] 특히 전 세계 스마트폰 OS 이용률의 80% 이상을 점유하고 있는 안드로이드 OS의 애플리케이션은, 역공학이 쉬운 JAVA 언어로 구현되어 악성코드 삽입과 리패키징 공격 등 수많은 위협에 노출되어 있다.[3]

모바일 보안 솔루션 개발사 ARXAN의 조사결과, 안드로이드 애플리케이션의 전 분야에서 90% 이상, 특히 금융분야 애플리케이션의 95%가 해킹당한 이력이 있어 금융 애플리케이션이 해커들의 주 공격표적이 되고 있음을 확인할

* 본 연구는 KITRI 차세대 보안리더 양성 프로그램(Best of the Best)의 연구 지원으로 수행되었습니다.

수 있었다. 이를 방지하기 위해 대부분의 금융 애플리케이션이 무결성 검증, 디버깅 방지, 악성코드 검사 등의 보안 기능을 제공하는 보안 솔루션을 탑재하였다. 또한, 해당 보안솔루션에서 더 체계적이고 안전한 검증 수행을 위해 동적 로딩을 이용하는 등 다양한 연구가 선행되었다.[4][5]

그러나 해당 연구들에서 제시한 방법으로 검증을 수행할지라도, 애플리케이션 서비스의 모든 기능이 개발된 후 보안솔루션이 탑재되기 때문에, 애플리케이션과 보안 솔루션의 독립적인 구조상 검증 결과값을 위·변조하는 것만으로도 보안 솔루션을 우회할 수 있다.

본 논문에서는 이러한 방법을 통해 실제로 우회가 가능함을 검증하기 위해 금융 애플리케이션 중 이용률이 높은 간편결제 애플리케이션 7개를 선정하여 분석 및 우회실험을 진행하였다. 뿐만 아니라 이러한 독립적 구조상 발생하는 취약점을 방지할 수 있는 대응방안으로 '토큰 발급을 통한 클라이언트 요청 값 검증 방법'을 제안한다.

본 논문의 2장에선 애플리케이션과 보안 솔루션의 동작 과정에 대해 설명하고, 결과값 위·변조를 통한 보안 솔루션 우회를 실험을 통해 검증한다. 3장에서 대응 방안을 제시하고, 끝으로 4장에서 결론과 향후 연구 방향에 대해 서술한다.

II. 보안 솔루션 검증기능 우회 취약점

본 장에서는 애플리케이션과 보안 솔루션의 동작 과정에 대해 설명하고, 결과값 위·변조를 통한 검증기능 우회 취약점을 실험을 통하여 확인한다.

2.1 안드로이드 애플리케이션의 보안 솔루션 동작 과정

Fig. 1.과 같이 애플리케이션에서 보안솔루션을 호출하면 보안 솔루션에서 DEX(Dalvik Executable)와 네이티브 라이브러리 코드 및 서버와의 상호작용을 통해 애플리케이션 자체와 구동 환경의 보안성을 검증한다. 검증 후 결과값이 반환되면 보안 솔루션은 해당 응답 값을 바탕으로 애플리케이션을 정상 실행하거나 종료하며 동작을 제어한다.

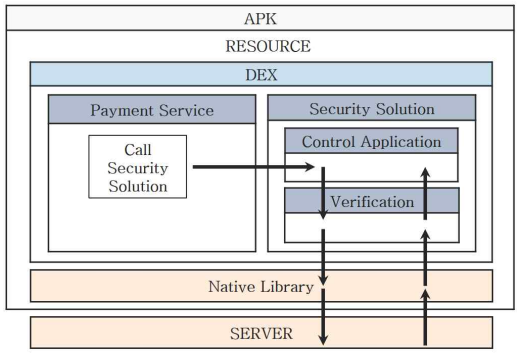


Fig. 1. Flow of Android Application's Security Solution

보안 솔루션은 이와 같은 과정을 통해 무결성의 훼손, 변조된 OS(루팅), 혹은 디버깅 환경에서의 애플리케이션 실행 등, 보안성을 저해할 수 있는 상황을 식별하여 애플리케이션을 제어함으로써 위협을 차단하는 구조로 동작한다.

2.2 검증 결과값 위·변조를 통한 보안 솔루션 우회 취약점

일반적으로 개발업체에서는 서비스의 주 기능 개발을 완료한 후, 보안성의 확보를 위해 보안 솔루션을 그 위에 탑재한다. 개발업체가 원본 apk파일을 업로드 하면 보안 솔루션 업체에서 apk파일에 보안 기능을 포함하고 이를 호출하는 코드를 추가해 보안 솔루션을 적용한다.

이와 같은 과정으로 애플리케이션의 주요 서비스가 개발된 이후 보안 솔루션이 그 위에 탑재되기 때문에, 검증의 통과 여부와 관계없이 애플리케이션이 독립적으로 동작할 수 있다. 따라서 기존에 선행되었던 정교한 검증 수행을 위한 연구들이 제시한 방법들을 사용하더라도, 검증 루틴과는 무관하게 검증 결과값이 애플리케이션을 제어하는 루틴으로 전달되는 과정에서 해당 값을 위·변조하는 간단한 방법만으로 보안 솔루션의 검증기능을 우회할 수 있는 취약점이 발생한다.

Security Solution	Integrity Check	Rooting Check	Anti-Debugging
A	O	O	O
B	O	N/A	N/A
C	O	O	O
D	N/A	O	O
E	N/A	O	N/A
F	O	O	N/A
G	X	X	X

* O : Bypass Success, X : Bypass Fail, N/A : Not Applicable

Table 1. Status of Bypassing Payment Application's Security Solution Through Forgery of Verification Result Value

실제로 보안 솔루션이 적용되어 있는 국내 간편결제 애플리케이션 중 이용률이 높은 7개를 선정하고, 애플리케이션을 역분석하여 보안 솔루션 동작 루틴을 분석, 검증 결과값 위·변조를 통한 보안 솔루션 우회 실험을 진행했다. Table 1.은 해당 7개의 애플리케이션에 탑재된 보안 솔루션 현황 및 주요 검증기능 우회 결과를 정리한 표이다. 이러한 보안 솔루션의 검증 결과값 변조를 통한 우회로 7개의 애플리케이션 중 'G' 보안 솔루션을 탑재한 애플리케이션을 제외한 나머지 모든 6개의 애플리케이션의 검증기능이 우회가 가능함을 검증하였다.

III. 토큰 발급을 통한 검증기능 강화

애플리케이션의 주요 기능들은 보안솔루션과는 관계없이 독립적으로 개발되었기 때문에, 보안 솔루션이 비정상적인 검증 결과값을 전달하더라도 정상적으로 동작할 수 있다.

본 논문의 검증 결과값 위·변조를 통한 검증기능 우회 취약점이 발생하는 핵심적인 이유는 이러한 애플리케이션과 솔루션의 독립적 구조 때문이다.

따라서 이에 대한 대응방안으로, 보안 솔루션 검증기능을 통과했을 때만 유효한 토큰을 부여하고, 이 토큰이 있어야만 애플리케이션의 주요 기능이 정상적으로 동작할 수 있도록 함으로써 보안 솔루션과 애플리케이션의 주 기능이 서로 종속적으로 동작할 수 있도록 하는 방안을 제시하고자 한다.

구체적인 프로세스는 Fig. 2.와 같다. 보안 솔루션에 의한 검증에 성공했을 경우 서버로부터 유효한 토큰을 받는다. 이후 애플리케이션의 주요 기능을 수행하기 위해 서버와 통신을 할 때는 유효한 토큰을 함께 전달해야만 정상적인 서비스 이용이 가능한 구조로 보안 솔루션을 설계한다. 그렇다면, 단순히 검증 결과값 위·변조를 통해 애플리케이션 제어 루틴을 속여 애플리케이션이 종료되지 않도록 하는 것만으로는 애플리케이션을 정상적으로 이용할 수 없다. 따라서 본 논문에서 제시된, 검증 결과값을 위·변조하는 간단한 방법으로 보안 솔루션의 검증기능이 우회되는 취약점의 발생을 막을 수 있다.

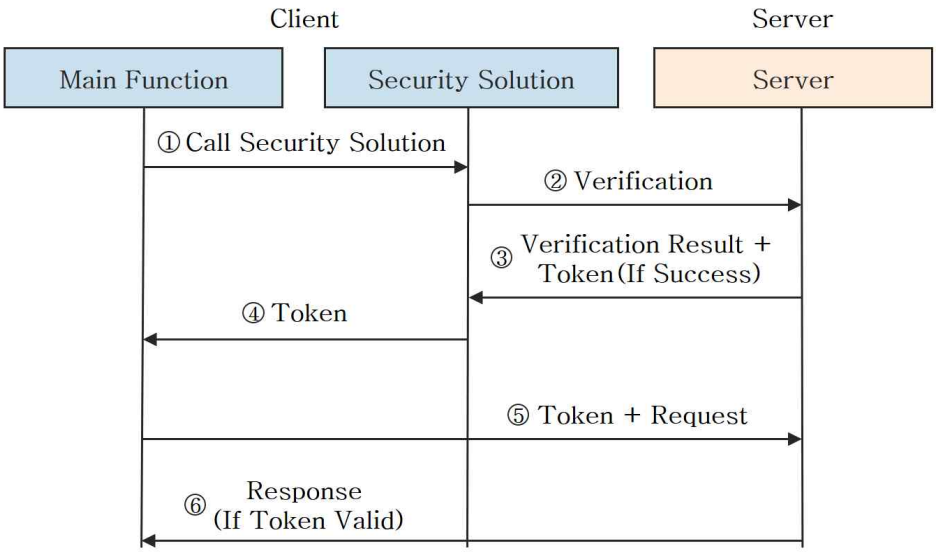


Fig. 2. Process of Token Issuance

이와 같은 대응방안을 실제로 적용하기 위해서는 애플리케이션에 보안 솔루션을 적용시키는 프로세스의 변화가 필요하다. 기존에는 주요 기능의 개발이 완료된 애플리케이션 위에 보안 솔루션을 적용시키고, 이미 개발된 애플리케이션의 주요 기능에 대해서는 수정을 최소화하는 방향으로 보안 솔루션이 탑재되었다. 하지만 본 논문에서 제시한 대응방안을 적용하기 위해서는 애플리케이션의 보안 솔루션으로부터 유효한 토큰을 받아오고 애플리케이션의 주요 기능 수행 시 서버로 토큰을 전송해 보안 솔루션의 검증기능이 성공적으로 수행되었음을 증명하는 등의 수정이 필요하다.

IV. 결론

본 논문에서는 보안 솔루션의 검증 결과값을 위·변조하여 검증기능을 우회할 수 있는 취약점에 대해 알아보고, 이에 대한 대응방안으로 '토큰 발급을 통한 클라이언트 요청 값 검증 방법'을 제시하였다.

보안성의 확보가 중요한 간편결제 애플리케이션을 통해 취약점 검증 실험을 진행하였지만, 금융 애플리케이션뿐만 아닌 보안 솔루션을 탑재한 거의 모든 안드로이드 애플리케이션에서 해당 방법으로 보안솔루션 우회가 가능할 것이다. 따라서 이에 대한 대응방안의 마련이 시급하다. 본 논문에서 제시한 대응방안을 적용하기 위해선 애플리케이션과 보안 솔루션이 종속적으로 동작하는 것이 핵심이기 때문에, 애플리케이션 설계 단계에서부터 보안에 대한 고려가 필요하다.

하지만 본 논문에서 제시한 대응방안을 적용 하더라도, 검증기능의 루틴을 분석하여 정상 검증 결과값을 서버에 전송하면 서버는 유효한 토큰을 클라이언트에 전달해줄 것이다. 이러한 한계를 극복하기 위해선, 애플리케이션의 무결성을 확보할 수 있는 방안에 대한 연구와, 역분석과 위·변조를 어렵게하기 위한 보안 기능 간의 연계에 대한 연구가 필요하다고 판단된다.

[참고문헌]

[1] "Domestic Internet banking service in Q2/2017", The Bank of Korea, pp. 1, Aug. 2017

[2] Woojin Lee, Kyungo Lee, "A Study on the Vulnerability of Using Intermediate Language in Android: Bypassing Security Check Point in Android-Based Banking Applications," Journal of the Korea Institute of Information Security & Cryptology, 27(3), pp. 549-562, Jun. 2017

[3] Soonil Kim, Sunghoon Kim, Dong Hoon Lee, "A study on the vulnerability of integrity verification functions of android-based smartphone banking applications," Journal of the Korea Institute of Information Security & Cryptology, 23(4), pp. 743-755, Aug. 2013

[4] Jeong-min Kim, "A study on the improvement of android banking app integrity using dynamic key value," Master's Thesis, Hannam University, Feb. 2017.

[5] Hyunjo Kim, Jin-Young Choi, "Research on Secure Coding and Weakness for Implementation of Android-based Dynamic Class Loading," Journal of Korea Multimedia Society, vol. 19, no. 10, pp. 1792-1807, Oct. 2016

안드로이드 어플리케이션 자동 분석 보고서 제작 프로그램 설계 및 구현

천호진*, 강성은*, 정수환***

*숭실대학교 정보통신공학과, **숭실대학교 전자정보공학과

Design and implementation of the Android application Automated Analysis Report Program

HoJin Chun*, Seongeun Kang*, Souhwan Jung***

*School of Information and Telecommunication Engineering, Soongsil Univiersity. **School of Electronic Engineering, Soongsil University.

요 약

악성 코드가 점차 지능화됨에 따라서 Android Application 분석가들은 웹 또는 프로그램 형식의 자동 분석 리포트를 활용하여 악성 코드를 분석하고, 대응 방안을 고안하고 있다. 본 논문에서는 JSON 타입으로 분석된 악성코드 데이터를 해석하여 워드 파일 형식의 분석 리포트로 출력하는 프로그램을 설계 및 구현했다. 분석가들이 제안한 프로그램을 사용하여 출력한 분석 보고서를 이용해 좀 더 빠르고 명확한 악성코드 파악을 돕는 것을 목적으로 한다.

I. 서론

현재 이루어지고 있는 사이버 위협 중 악성코드는 사이버 상에서 가장 심각한 위협으로 분류되고 있는데, 특히 AV-TEST 연구소에서 발행한 통계자료에 따르면 최근 10년 동안 신규 악성코드의 양이 지속적으로 증가하는 것을 확인 할 수 있다. 또한 그 양이 폭발적으로 증가 하는 것 또한 확인 할 수 있다[1]. 또한 신규 악성코드는 안티 바이러스 프로그램의 탐지를 회피하기 위해 코드의 내용이 복잡하고 정교해지고 있어, 실제 50% 이상의 악성코드가 탐지되지 않아 위협적인 상태로 유지되고 있다[2,3].

이런 지능형 악성코드를 좀 더 효과적으로 대응하기 위해 CNSL은 Android Application의 동적 및 정적분석을 통해 좀 더 정밀하고 꼼꼼한 탐지를 진행하고 있으며, 이를 JSON 타입의 파일로 정리했다. 하지만 JSON 파일 또한 분석가가 내용을 확인하기에는 내용이 분산되어 있어 시간이 다소 걸린다는 단점이 있다. 따라서 본 보고서에서 다루는 자동 분석 결과 보고서 제작 프로그램을 활용하여 좀 더 빠르고 명확한 악성코드 분석을 목적으로 한다.

2장에서는 타 Web/Program에서 제공하는 악성코드 툴의 분석 보고서 내용을 확인하고, 분석한 내용을 종합하여 이들의 한계나 보완해야 할 점을 도출한다. 3장에서는 앞서 종합한 내용을 바탕으로 좀 더 체계적이고 간결한 보고서를 만드는 프로그램을 구현한다. 4장에서는 앞으로 Android Application 자동 분석 결과 보고서 제작 프로그램이 제작 되어야 할 방향과 추가로 구현하면 좋을 점에 대해 제시한다.

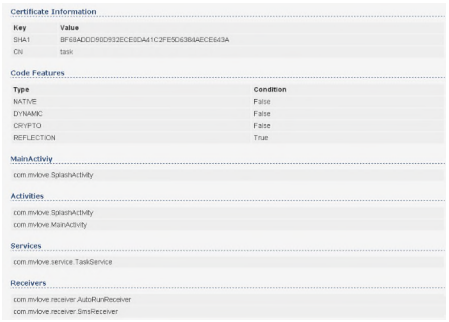
II. 관련 연구

2.1 모바일 악성코드

모바일 악성코드란 기존 PC 환경에서 발생하는 악성코드와 유사하게 모바일 단말기를 대상으로 개인정보 유출, 시스템 파괴, 원격지 접속

등의 악의적인 행위를 수행하기 위해 제작된 악성 프로그램을 의미한다[4]. 모바일 악성코드를 분류하면 대표적으로 전파를 목적으로 다른 파일을 감염시키는 바이러스, 유용한 프로그램을 가장한 Trojan, 자신의 복사본을 생성하여 전파하는 Warm이 있으며, 이중에서도 사회공학기법을 이용하여 사용자의 설치를 유도하는 Trojan이 가장 많은 비율을 차지하고 있다[5, 6, 7, 8].

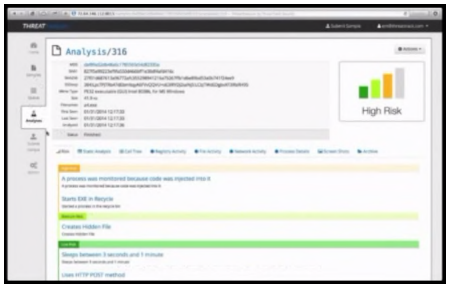
2.2 Web Analysis Report 분석



[그림 1] SandDroid

SandDroid의 분석 리포트에는 정적 분석과 동적 분석이 모두 포함되어 있다. 정적 분석은 권한 분석, 성분 분석, 악성 코드 탐지, 분류 분석등이 포함되어 있다. 정적 분석은 파일 작업을 포함하여 Application이 실행 중 동적 행동, 네트워크 행동 (GET/POST 요청, TCP/UDP 연결 등), 개인 정보 누출 등을 모니터링 한다. [9]

2.3 Program Analysis Report 분석

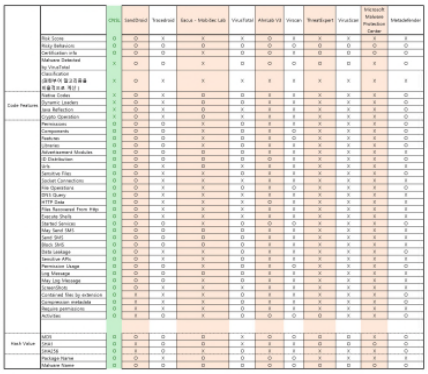


[그림 2] ThreatAnalyzer

미국 정부의 75%가 사용하고 국내의 여러 주요기관에서 사용 중인 ThreatAnalyzer은 예전에 샌드박스로 알려져 있던 프로그램이다.

ThreatAnalyzer은 동적 SSDeep (퍼지) 해시 매칭 기능, 통합 위협 요소 체계 기능, IOC기능을 이용해 APK의 행동이나 네트워크 관련 정보를 분석 리포트에 기록 한다.

2.4 Analysis Report 분석 비교



[그림 3] Analysis Report의 분석비교

[그림 3]의 왼쪽 녹색 부분은 본 자동 보고서의 데이터 탑재 정보여부이다. 다른 자동 분석 리포트 엔진보다 많은 양을 기록하고 있음을 확인 할 수 있다. 해당 자돈 분석 리포트를 활용하고 추가 발생되는 위험에 맞추어 APK의 분석요소를 추가하면서 끊임없이 악성코드에 대응하고 분석하는 자세를 갖추어야 할 것이다.

III. 분석 프로그램 개발

3.1 Data 읽어오기

3.1.1 참조 및 class 선언



[그림 4] JSON 파일

JSON 형식의 Data 파일에는 APK의 분석 정보가 담겨있다. Static에는 정적 분석을 통한 Service, Permission, Activity 등등이 포함되어 있으며, Dynamic에는 동적 분석을 통한 자료들이 포함되어 있다.

.NET Framwork를 기반으로 분석 프로그램을 제작한다. JSON.NET은 JSON파일을 객체로 작업하기 위한 API를 사용하기 위해 필요하다.

Android Application 자동 분석 리포트의 구성은 5개의 .cs파일로 이루어진다.

① AndroidPermissions.cs : android permission이 각각 해당하는 정보를 선언

② ClassDefines.cs : 파싱하는 JSON파일의 크기와 타입을 가져옴

③ JsonInfo.cs : 파싱할 각종 변수 선언

④ JsonParser.cs : JObject인 Json을 선언하고 각각 파싱하는 값의 크기를 가져옴

⑤ Form1.cs : 메인 소스 코드

먼저 dig라는 이름의 객체를 생성하고 .JSON 확장자의 파일을 연다. 불러온 .JSON파일의 경로를 URL로 설정하고 m_parser이라는 이름의 JsonParser를 선언하고 URL경로의 즉 불러온 파일의 이름, 크기, 타입을 불러온다. 마지막으로 JSON파일을 불러와 파싱준비를 마친다.

[표 1] 변수의 크기 읽기

```
if (null != m_parser.Json)
{
        JArray array2 = JArray.Parse(m_parser.Json["apk"]["Files"].ToString());
        foreach (JObject itemObj in array2) //apk-Files 파싱
        {
                ( 각각의 읽어올 정보의 크기를 읽어오는 부분 )
        }
}
```

[표 1]은 파싱중에 가져올 APK Files 또는, Permission, Activity, Service등 APK마다 각각의 개수가 다른 것을 고려하여 그 개수를 측정하는 부분이다. 크기를 측정하는 이유는 이후 작성되는 자동 보고서에 표의 크기를 결정하기 위함이다.

3.1.3 Button2 : Write_Data

문서를 생성하기에 앞서 변수를 선언한 뒤 word 문서 객체를 생성한다. 이후 머리말과 머릿표를 생성하고, [그림 2]의 APK 기본정보가 들어있던 파트를 파싱한다.

[표 2] APK Info 파싱

```
m_parser.FileName;  // File name
apkBasicInfo.GetFileSize(m_parser.FileSize, FileSizeFormat.DontCare); // File Size
apkBasicInfo.File_Type(m_parser.FileType).ToUpper();  // File type
m_parser.Json["apk"]["FileMagic"].ToString();  // File Magic
m_parser.Json["apk"]["Filename"].ToString();  // APK Name
m_parser.Json["apk"]["FileSize"].ToString();  // APK Size
m_parser.Json["apk"]["SHA256"].ToString();  // SHA256
m_parser.Json["apk"]["MD5"].ToString();  // MD5
```

[표 2]에서는 각각 파싱하는 파일의 이름, 크기, 매직, JSON파일에 포함된 APK의 이름, 크기, SHA256, MD5값을 불러온다. 이후에는 Contain ed Files에서는 해당 APK에 포함된 모든 파일의 이름, 확장자, 크기, 타입, 경로를 가지고 오며, 확장자의 중 중요한 파일인 XML, DEX, PNG 등은 개수를 카운트 한다.

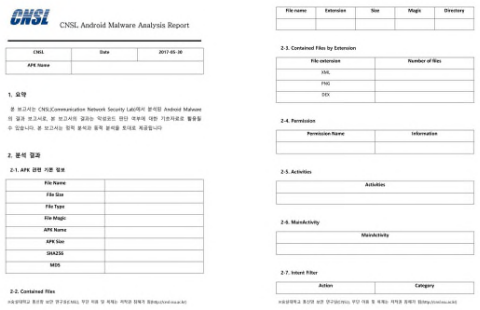
[표 3] Permission 정보 읽어오기

```
string[] Temp = value4[i].ToString().Split(',');
Temp[2];  // 배열 2번값 즉permmsion 값을삽입
(string)androidPermissions[Temp[2]];  // permission 값을androidPermission에서 값을 찾아와가져옴
```

Permission은 android.permission.@@@의 형태로 구성되므로 .을 기준으로 Temp[2]배열의 값만 분리한다. 그리고 앞서 선언한 androidPermissions를 이용해 Temp[2]의 값을 보내어 정보를 반환한다. 이와 동일한 방법으로 Activities와 MainActivity, Action, Category, Service, dynamic 파싱을 진행한다. Dynamic의 경우엔 각각의 값이 존재하는지에 대한 유효성 검사가 필요하다.

IV. 출력 리포트 분석

4.1 기본 리포트 출력 형식



[그림 5] 기본 분석 리포트 1



[그림 6] 기본 분석 리포트 2

[그림 5]와 [그림 6]은 앞서 제안한 자동 분석 프로그램을 이용해 작성한 기본 형식의 분석 리포트이다.

4.2 분석 후 출력 형식


CNSL Android Malware Analysis Report

[그림 7] 분석 리포트 1

[그림 7]는 APK의 기본 정보가 파싱되어 기록 되었다. 파싱하는 날짜와 .JSON 파일의 크기, 이름, 확장자와 .JSON이 담고있는 APK의 크기, 이름, 확장자를 다시 한 번 리턴한다. 그리고 [그림 5]과 [그림 6]의 표에 각각의 값을 파싱하여 입력한다.

4.3 분석 보고서의 한계와 보완할 점

CNSL 분석 보고서는 앞서 분석했던 [그림 3]의 타사 자동 분석 보고서와는 확연하게 '간결하고 핵심적인 부분을 파악할 수 있다'는 차이점을 갖는다. 하지만 JSON파일의 형식에 이상이 있을 경우 파싱이 어렵다는 결정적인 문제점이 발생하고 있다. 이를 보완하기 위해서는 JSON파일을 해당 자동 분석 프로그램에서 작동시키기 전에 검사하는 과정이 필요하다.

두 번째, CNSL의 분석 보고서에는 타 엔진에서 해당 APK를 분석한 자료가 빠져 있다. 이는 분석가로 하여금 해당 분석 보고서가 얼마나 신빙성 있는지 뒷받침 하는 척도가 될 수 있다. 따라서, Virustotal이나 VerScan, Metadefender 같은 기능이 보완되면 좋을 것이다.

V. 결론

본 보고서에서는 타사의 Android Application 자동 분석 보고서에서의 개선할 점을 파악하고 이들을 추가하여 새로운 형식의 Android Application 분석 보고서 제작 프로그램을 제안하였다. 대부분의 분석 보고서는 APK의 기본정보와 동적 분석의 일부만을 기록했기 때문에, 자체 개발한 Android Application 자동 분석 리포트는 추가로 정적 분석을 한다는 것에 타사의 분석 보고서에 비하여 경쟁력이 있다고 할 수 있겠다. 악성코드 분석 전문가가 본 논문에서 제안한 분석 리포트를 보았을 때, 악성이라고 의심 할 수 있는 정보에 대한 해석이 용이하다는 장점은 백신을 개발하고 대응방안을 작성하는데 시간적으로 큰 도움이 된다고 할 수 있다.

Android Application 자동 분석 보고서 프로그램은 앞으로 진화하는 악성 코드의 트렌드에 맞추어 분석하는 내용 또는 방식을 변화 시키는 것이 향후 연구의 목적이 되겠다.

[참고문헌]

[1] European Union Agency for Network and Information Security (ENISA): ENISA Threat Landscape 2015, https://www.enisa.europa.eu/publications/etl2015, Dec 201

[2] 정진혁, 이형찬, 김익수, 이정현, "모바일 악성코드 현황 및 대응 방안", 한국정보보호학회 동계학술대회, 2009.

[3] K. Dunham, S. Abu-Nimeh, M. Becher, S.Fogie, B. Hernacki, J. A. Morales, and C. Wright, Mobile Malware Attacks and Defense, Syngress, 2008.

[4] W. Jansen and K. Scarfone, "Guidelines on Cell Phone and PDA Security," NIST, 2008.

[5] A. Schmidt and S. Albayrak, "Malicious Software for Smartphones," Tech. Rep., DAILabor, 2008.

[6] M. Khapra and N. Uchat, "Mobile Worms, Viruses and Threats," Indian Institute of Technology.

[7] A. Gostev, "Mobile Malware Evolution: An Overview," 2006.

[8] A. Gostev, "Kaspersky Security Bulletin 2006: Mobile Malware," 2007.

소프트웨어 권한 분리에 대한 연구

방인영, 이영한, 안선우, 백세현, 백윤흥*

*서울대학교 전기정보공학부

A Study on Software Privilege Separation

Inyoung Bang, Younghan Lee, Sunwoo Ahn, Sehyun Baek, Yunheung Paek*

*Department of Electrical and Computer Engineering Seoul National University

요 약

최소 권한의 원리는 프로그램의 보안성을 제고하기 위한 핵심적인 소프트웨어 디자인 원칙이다. 애플리케이션 소프트웨어 또는 시스템 소프트웨어가 최소 권한의 원리에 위배되는 모놀리식한 구조를 띠게 되면 잠재적 보안 위협에 대해 안전할 수 없다. 이에 대한 해결책으로 애플리케이션 소프트웨어와 시스템 소프트웨어에 대한 권한 분리를 논의하고, 구체적으로 어떠한 방법들이 선행연구에서 논의되었는지를 서술한다.

I. 서론

Saltzer와 Schroeder의 1975년 논문인 “The Protection of Information in Computer Systems”에서 제안된 최소 권한의 원리(principle of least privilege)[1]는 프로그램의 보안성을 제고하기 위한 핵심적인 소프트웨어 디자인 원칙이다. 많은 소프트웨어들은 단일한 권한을 가진 채 코드를 수행하는 모놀리식(monolithic) 디자인의 형태를 가지고 있다. 모놀리식 소프트웨어의 근본적인 문제는 어떠한 방식으로든 취약점이 공격받게 될 경우, 취약점 외의 다른 영역마저도 위협에 처하게 되며, 공격자가 손쉽게 프로그램이 이상행동을 하게끔 혹은 정상적으로 작동할 수 없게끔 만들 수 있다는 것이다. 공격에 대한 방어기법이 아무리 고도화될지라도, 모든 공격을 무효화할 수 있는 시스템은 실현불가능하다. 따라서 미처 대처하지 못한 취약점이 공격받더라도 위협을 완화하는 소프트웨어적 구조가 반드시 필요하다. 이에 대한 본질적인 해결을 위해 최소 권한의 원리를 구현하고자하는 연구가 활발히 진행되어 왔으며, 본 논문에서는 최소 권한의 원리에 위배되는 모놀리식한 소프트웨어에 대한 해결책에 대해 논의하고자 한다.

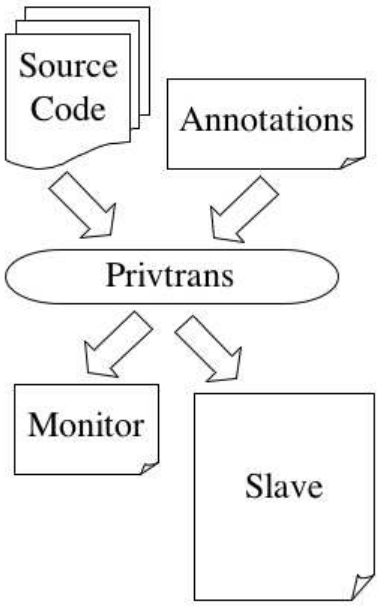
II. 본론

소프트웨어를 크게 애플리케이션 소프트웨어와 시스템 소프트웨어로 나누어, 최소 권한의 원리를 이용하여 보안성을 향상시키는 방법에 대해 서술하려 한다.

2.1 애플리케이션 소프트웨어 권한 분리

높은 권한을 필요로 하는 애플리케이션 프로그램(application program)은 공격자가 주어진 권한을 악의적으로 이용하여 원하는 행동을 할

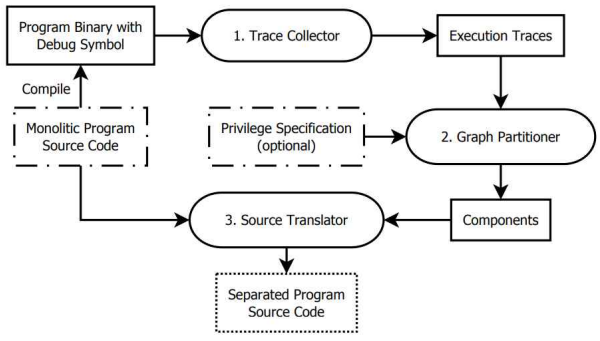
수 있기 때문에 주된 공격 대상이 된다. 공격자는 공격이 성공할 경우 프로그램 자체에 주어진 권한을 이용하여 임의적인 명령을 내릴 수 있다. 이를 막기 위해서 모놀리식 프로그램을 권한이 필요한 프로그램인 모니터(monitor)와 권한이 필요 없는 슬레이브(slave)로 분할하여, 프로그램에 주어지는 권한상승(privilege elevation)을 최소화하는 방법이 가능하다. Brumley와 Song은 프로그램을 모니터와 슬레이브로 자동으로 분할하는 방법인 Privtrans를 제시한다.[2] 이러한 과정은 프로그래머가 제공한 주석(annotation)을 이용하여 자동으로 이루어지기 때문에, 프로그래머가 직접 많은 노동력을 투입하여 코드를 수정할 필요가 없다. 이에 대한 모식도는 [그림 1]과 같다.



[그림 1] Privtrans 모식도

Privtrans의 기작은 다음과 같다. 프로그래머는 코드분할의 자동화를 위해 권한이 필요한 변수(variable)나 함수(function)에 대해 주석을 제공한다. 다음으로 Privtrans는 소스코드에 대해 프로시저간 정적 코드 분석을 수행한다. 분석결과를 바탕으로 C로 쓰여진 소스코드를 C로 쓰여진 마스터와 슬레이브로 분할한다. 수작업으로 코드분할을 수행하게 되면, 오류가 발생하기도 쉽고, 프로그래머의 노력이 많이 필요하다. 하지만 일련의 과정을 통해 프로그램에 주석을 입력하는 것만으로 수작업 기반과 유사한 수준의 보안성 증대를 달성할 수 있다.

이에 대한 후속연구로써, Y.Wu 외 3명은 ProgramCutter를 제안한다. ProgramCutter는 모놀리식한 프로그램을 필요한 권한만이 주어지게끔 자동적으로 자르는 방법이다. 이를 위해 소스코드로부터 데이터 의존성 그래프(data dependency graph)를 구성하고, 최소한의 권한만이 주어지게끔 부분그래프(subgraph)로 분할한다. ProgramCutter가 Privtrans와 차별화되는 가장 중요한 점은 정적 분석을 하는 Privtrans와 달리 동적 분석을 사용하여 코드를 분할한다는 것이다. 정적 분석에는 세 가지의 문제점이 있다. 첫째로, 타입 캐스팅(type casting), 함수 포인터, 포인터 연산과 같은 프로그램의 동적 거동을 분석할 수 없다. 둘째로, 버퍼의 크기와 같은 런타임 정보를 알 수 없기 때문에, 요소간 통신 비용을 산정할 수 없다. 권한이 다른 요소간 통신을 위해서는 RPC(remote procedure call)가 필요한데, ProgramCutter는 최적화를 통해 요소간 통신 비용을 최소화함으로써 성능 향상을 기대할 수 있다. 마지막으로, 정적 분석을 통해 권환 최소화를 달성하려면 주석이 제공되어야 하고, 이를 위해서 프로그래머에게 해당 모놀리식 프로그램에 대한 전문적인 지식이 요구되므로 그 사용이 제한적이다.



[그림 2] ProgramCutter 모식도

ProgramCutter의 모식도는 [그림 2]와 같다. 먼저, 모놀리식 프로그램은 디버그 심볼(debug symbol)과 함께 컴파일 된다. 프로그램은 trace collector에서 실행되면서 동적 분석을 위한 실행 추적(execution trace) 정보를 수집한다.

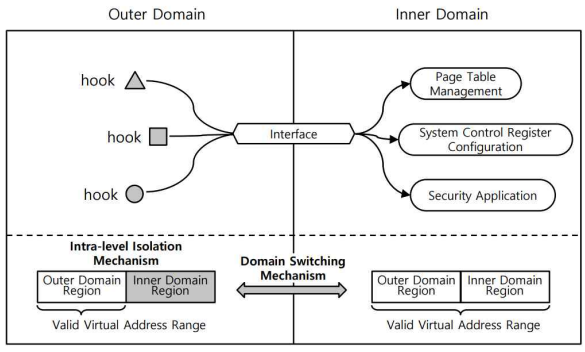
Graph partitioner는 실행 추적 정보들을 분석하여 최적의 코드분할을 찾고, 모놀리식 프로그램을 여러 요소로 분리한다. 마지막으로 분리된 요소들을 source translator가 권한이 분리된 프로그램으로 작성한다. 일련의 과정을 통해서 프로그램에 대한 전문적인 지식이 없이도 성능이 향상된 코드 분할을 자동적으로 수행할 수 있다.

1.2 시스템소프트웨어 권한분리

시스템 소프트웨어(system software)는 애플리케이션 소프트웨어에 상대되는 개념으로써, OS나 하이퍼바이저(hypervisor)와 같이 다른 소프트웨어의 플랫폼이 되는 소프트웨어를 일컫는다. 애플리케이션 소프트웨어에 대한 권한 분리의 중요성은 1.1에서 전술한 바 있다. 그렇다면 애플리케이션의 플랫폼이 되는 시스템 소프트웨어가 핵심적인 코드들이 단일한 메모리 영역에 저장되어 있는 모놀리식 디자인을 취한다면 보안성을 보장할 수 없다는 것은 자명하다. 코드 중 일부가 에러가 있거나, 보안적으로 취약할 경우 다른 영역을 위협하게 되며, 심각할 경우 시스템 전체가 전복될 위험이 있기 때문이다.

하지만 시스템 소프트웨어에 최소 권한의 원리를 적용하는 데에는 몇 가지 어려움이 있다. 첫째로, 시스템 소프트웨어는 이미 프로세서의 가장 높은 권한 레벨에서 수행되고 있으므로 더 높은 등급의 권한이 필요하다. 둘째로, 다른 소프트웨어의 신뢰 컴퓨팅 기반(trusted computing base)인 시스템 소프트웨어는 성능 문제에 훨씬 더 민감하고, 더 높은 등급의 권한을 가진 소프트웨어의 운용은 필연적으로 성능 저하를 수반하기 때문에 바람직하지 못하다. 따라서 하드웨어를 이용하여 기존의 권한을 세분화하는 동일권한 내(intra-level) 권한 분리가 합리적이다. 이를 위해 시스템 소프트웨어의 코드영역을 내부영역(inner domain)과 외부영역(outer domain)으로 쪼개며, 보안적으로 중요한 코드들은 내부영역에 격리한다. 예를 들어, 페이지 테이블(page table)과 시스템 제어 레지스터(system control register)와 관련한 명령들은 내부영역에 위치하게 된다. 내부영역과 외부영역의 코드는 물리적으로 동일한 권한을 가진 채 수행되지만, 논리적으로 내부영역이 더 높은 권한을 가지는 것으로 이해할 수 있다. 따라서 외부영역의 보안취약점이 파훼되어 공격자에게 제어권이 넘어가게 되더라도, 내부영역의 안전은 보장될 수 있다.

동일권한 내 권한 분리를 위해서는 두 가지 기능의 구현이 필요하다. 첫째는 동일권한 내 격리 기작(intra-level isolation mechanism)이고, 둘째는 영역 전환 기작(domain switching mechanism)이다. 동일권한 내 격리 기작을 구현하기 위해서는, 즉 외부영역에서 내부영역으로의 접근을 제한하기 위해서, MMU(memory management unit)라는 하드웨어의 도움을 받게 된다. 하지만 메모리 보호를 위해 설계된 전통적인 MMU로는 프로세서의 권한 등급에 따른 메모리 세분화만이 가능하기 때문에 동일권한 내 격리 기작을 구현하는 데 부족함이 있으며, 동일권한 내 권한분리는 하드웨어 시스템에 의존적이다. 영역 전환 기작은 내부 영역과 외부 영역 사이의 안전한 제어권 전환을 보장하기 위한 기능요소이다. 시스템 소프트웨어에 동일권한 내 권한 분리를 도입하기 위해서는 두 가지 기능이 모두 갖춰져야 한다. 둘을 모두 갖춘 시스템 소프트웨어의 모식도는 [그림 3]과 같다.



[그림 3] 동일권한 내 권한 분리 모식도

HyperSafe[4]나 Nested Kernel[5]의 경우 x86 프로세서의 WP(write-protection)라는 특정 메모리 영역의 쓰기 접근을 제어하는 특별한 레지스터 비트를 이용하여 이를 구현하였다. WP 비트를 이용하여 페이지 테이블을 조작함으로

써, 프로세서의 동일권한 내에서 내부영역이 외부영역에 대해 독립적인 쓰기 권한을 가지게 된다. WP 비트가 0일 경우 내부 영역 내에서의 메모리 변경 명령이 가능해지며, 1일 경우 내부 영역의 쓰기 접근이 거부되므로 내부영역의 보안성이 보장된다.

HILPS[6]의 경우 ARM-64 아키텍쳐의 TCR(translation control register)이라는 시스템 제어 레지스터를 이용하여 동일권한 내 권한 분리를 구현하였다. TCR 내의 특정한 필드는 시스템의 가상주소 영역의 범위를 결정하는데, 이 필드를 조작함으로써 가상주소 영역의 범위를 제어할 수 있으며 내부영역과 외부영역의 분리와 전환이 가능하다.

III. 결론

본 논문에서는 보안의 핵심원리인 최소 권한의 원리를 통해 소프트웨어의 보안성을 향상하는 방법에 대하여 서술하였다. 특히, 권한 분리의 대상이 크게 애플리케이션 소프트웨어일 경우와 시스템 소프트웨어일 경우에 대해 논하였다. 또한 선행연구에서 제시된 Privtrans, ProgramCutter, HyperSafe, Nested Kernel, HILPS가 권한 분리를 위해 구체적으로 어떤 방법을 사용하였는지를 제시하였다.

IV. ACKNOWLEDGEMENT

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 대학 ICT연구센터 육성지원 사업의 연구결과로 수행되었으며 (IITP-2017-2015-0-00403), 2017년도 정부(미래창조과학부)의 재원으로 정보 통신 기술 진흥센터의 지원을 받아 수행된 연구 (No.2016-0-00078, 2017년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구(NRF-2017R1A2A1A17069478), 맞춤형 보안 서비스 제공을 위한 클라우드 기반 지능형 보안기술 개발) 및 2017년도 두뇌 한국 21 플러스 사업에 의하여 지원되었음

[참고문헌]

[1] J.H.Saltzer and M.D.Schroeder, The Protection of Information in Computer Systems, Proceedings of the IEEE, 1975.

[2] D.Brumley and D.Song, Privtrans: Automatically Partitioning Programs for Privilege Separation, USENIX Security, 2004.

[3] Y.Wu, J.Sun, Y,Liu and J.S.Dong, Automatically Partition Software into Least Privilege Components Using Dynamic Data Dependency Analysis, IEEE/ACM International Conference on Automated Software Engineering, 2013.

[4] Z.Wang and X.Jiang, Hypersafe: A Lightweight Approach to Provide Lifetime Hypervisor Control-Flow Integrity, IEEE Symposium on Security and Privacy, 2010.

[5] N.Dautenhahn, T.Kasampalis, W.Dietz, J.Criswell and V.Adve, Nested Kernel: An Operating System Architecture for Intra-Kernel Privilege Separation, International Conference on Architectural Support for Programming Languages and Operating Systems, 2015.

[6] Y.Cho, D.Kwon, H.Yi and Y.Paek, Dynamic Virtual Address Range Adjustment for Intra-Level Privilege Separation on ARM, Network and Distributed System Security Symposium, 2017.

온라인 게임 보안을 위한 매크로의 우회 기법과 대응 방안 연구

박영서*, 정담지*, 김지은*, 김나경*, 김연주*, 전정훈*
* 동덕여자대학교 컴퓨터학과

A Study on Detour Scheme of Macro and its Counterplan for Online Game Security

Yeong-Seo Park*, Dam-Zi Jeong*, Na-Kyung Kim*,
Ji-Eun Kim*, Yeon-Ju Kim*. Jeong-Hoon Jeon*
* Computer Science Department, Dongduk Women's University

요 약

최근 온라인 게임 산업이 발전하면서 게임을 보다 편리하게 할 수 있는 매크로 기술 공유가 활발해지고 있다. 특히 악의적인 게임 유저들이 지속적으로 매크로 우회 코드를 공유하고 있고 그 중에는 불법 매크로 또한 다수 포함되어있다. 게임 시스템 내 불법 매크로에 대한 제재가 잘 이루어지지 않아 게임 보안 프로그램의 탐지율에 대한 일반 유저의 신뢰성이 저하되고 있다. 이러한 매크로들의 우회 기법들은 후에 모바일 게임 보안 프로그램에서도 참고하여 응용될 가능성이 있다. 따라서 게임 보안 프로그램의 탐지율을 향상시키기 위해 현재 제작되는 매크로에 대한 우회기법의 실험과 분석이 요구된다.

본 연구에서는 보안 프로그램이 탐지 할 수 없는 게임 매크로가 보안을 우회하는 기법들을 정의하고, 이에 따른 매크로 대응 방안을 연구하여 매크로 프로그램의 확산을 방지할 수 있는 방안을 제안한다.

I. 서 론

최근 컴퓨터, 스마트폰 등의 기기가 발전함에 따라 보다 규모가 커진 게임 산업이 주목받고 있다. 게임 산업을 장해하는 요소에는 게임 매크로 봇을 개발하여 부당한 이익을 얻는 사람들이 있다.[1] 불법 매크로 개발자들은 게임 내의 보안 프로그램을 우회하는 등의 행위로 게임 보안 프로그램에 대한 신뢰성을 저해한다.[2]

매크로 프로그램들 중에서 유저가 손쉽게 접할 수 있는 것이 Windows 응용 프로그램 매크로인 오토핫키(Autohotkey)이다. 다른 언어로 제작할 수 있는 매크로들도 존재하지만, 대표적으로 GUI의 자동화가 가능한 장점 때문에 사용되고 있다. 오토핫키로 제작한 매크로는 단순키 조작에 의한 매크로나 매크로를 방지하기 위해 띄우는 창 혹은 특정한 오브젝트 등을 인식하기 위한 이미지 서치 매크로 등이 대표적인 매크로로 제작된다. 매크로가 유행하는 대표적인 게임들은 많은 유저 수만큼 사설 서버를 두어 게임 작업장을 만드는 곳도 증가하고 있는 추세다.[3] 게임 개발 측에서는 우회 기법에 대한 대책을 많이 고안해왔지만 악성 유저들은 우회 기법을 파악해 매크로에 적용하여 게임 산업과 유저에 지속적인 피해를 입히고 있다.

본 연구에서는 보안 프로그램이 탐지 할 수 없는 게임 매크로가 보안을 우회하는 기법들을 정의하고, 이에 따른 매크로 대응 방안을 연구하여 매크로 프로그램의 확산을 방지할 수 있는 방안을 제안한다.

II. 매크로의 종류와 우회 기법 연구

2.1 오토핫키 매크로의 유형

오토핫키로 제작되는 대표적인 매크로는 핫키로 등록해서 사용하는 키 조작 매크로와 이미지를 지정해서 인식하는 이미지 서치 매크로(Image Search macro)가 있다.

가) 키 조작 매크로

게임에 사용되는 키들을 분류해서, 단순 반복되는 입력을 지정하여 자동화 시키는 키 조작 매크로는

실제 시간을 적용하여 이동 범위를 정함으로 일반 매크로의 불특정한 범위보다 안정적으로 매크로를 동작시킬 수 있다. 또한, 화면에 따라 움직이는 마우스의 좌표를 활용하여 이동 범위의 정확성을 높이는 방법으로도 이용된다.

나) 이미지 인식 매크로

이미지 인식 매크로는 게임 내의 특정 이미지를 토대로 인식하는 이미지 서치 기법을 활용한 매크로이다. 이 매크로는 특정 오브젝트를 지정하여 이미지를 저장한 후에 일치하는 오브젝트를 발견했을 시에 특정 행동을 하는 매크로이다.

또한, 인식을 제대로 못하는 문제가 있어서, 문제점을 보완하기 위해서 C# DLL로 C# 언어와 같이 사용하는 방법으로도 이용된다.

2.2 우회기법의 종류

우회기법에는 코드 내용을 수정하는 리네이밍(renaming) 기법, 실행파일을 우회하여 컴파일하는 우회 컴파일 기법과 로그 수집 경로를 이용한 오토핫키 메인 서버 우회가 있다.

2.2.1 리네이밍 기법

리네이밍(Renaming) 기법은 오픈 소스 소프트웨어인 오토핫키에서 키보드, 마우스의 후킹 코드가 적혀있는 hook.cpp의 일부 코드를 바꿔서 접근하는 기법이다. 코드에 사용되는 뮤텍스 및 클래스 명을 게임 내 보안 코드가 탐지할 수 없게 만들어서, 매크로가 후킹을 사용할 수 있게 하는 방법이다. 일반적으로 이 기법을 사용해서 우회 매크로를 제작한다.

2.2.2 우회 컴파일러

오토핫키 프로그램은 기본적으로 스크립트가 .ahk로 저장되며, 오픈소스 소프트웨어 설치 시에 저장된 컴파일러로 실행 파일로 만든다. 기본적으로 제공되는 컴파일러로 제작된 오토핫키 매크로 프로그램은 보안프로그램에 의해 탐지된다. 우회 컴파일러 기법은 보안 프로그램에 우회할 수 있는 우회 컴파일러를 제작하여, 오토핫키 매크로 파일을 우회 컴파일 시켜서 탐지를 방해한다.

2.2.3 메인 서버 우회

메인 서버 우회 기법은 메인 서버에서만 로그가 기록되어 실제로 매크로 프로그램을 사용하는 컴퓨터에는 로그가 남지 않는 것을 이용한 우회기법이다. 게임 회사의 보안 프로그램은 기본적으로 관리자 권한을 받아 게임 유저의 로그 기반으로 불법 매크로 프로그램을 탐지한다.

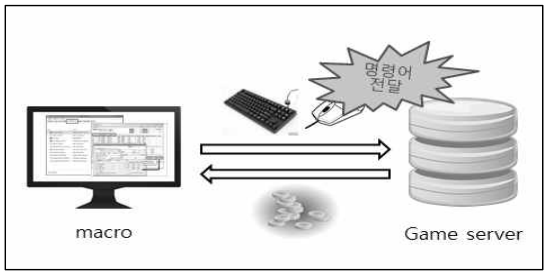
매크로에서는 웹 호스팅 기법을 사용하여 오토핫키 스크립트를 실행하거나, 컴파일된 실행파일에는 탐지가 제대로 이루어지지 않는다.

III. 매크로의 사례와 우회 기법 분석

3.1 매크로 실행 사례와 GUI

키 조작 매크로와 이미지 인식 매크로를 실행했을 당시 보다 더 정교하게 사용하기 위해서 옵션 수정 기능과 윈도우 메시지 박스를 활용한다. 또한, 매크로 프로그램을 거래하기 위해서 구매자가 사용하기 쉽게 GUI 자동화 툴로 매크로 프로그램을 제작한다. 다음은 키 조작 매크로와 이미지 서치 매크로의 활용 사례이다.

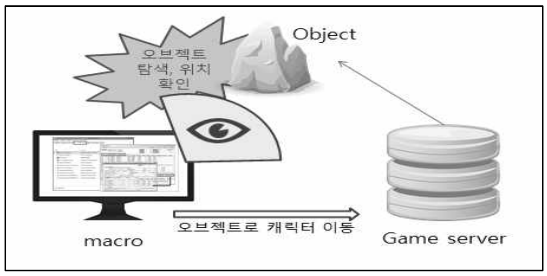
3.1.1 키 조작 매크로 활용 사례



〔그림 1〕 자동 사냥 매크로 사례

〔그림 1〕에서 매크로 사용자는 공격키와 이동키를 설정해 서버에 반복해서 명령어를 전달한다. 또한 GUI를 이용해 딜레이, 점프키 등을 선택하여 아이템을 손쉽게 습득할 수 있게 제작되는 매크로다.

3.1.2 이미지 서치 매크로 활용 사례



〔그림 2〕 이미지 서치 매크로 사례

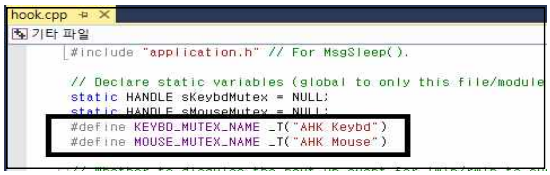
〔그림 2〕에서 이미지 서치 매크로는 게임 내 오브젝트를 탐색하고 위치를 저장 한 후에 그 오브젝트로 캐릭터가 이동해서 매크로에 작성된 특정 행동을 하고, 오브젝트에 관련한 아이템을 획득한다.

3.2 우회 기법 분석

리네이밍 기법과 우회 컴파일러 기법, 메인 서버 우회 기법을 직접 실행해보고 구조와 코드를 분석을 수행하였다.

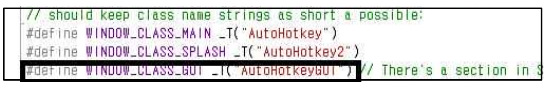
3.2.1 리네이밍 기법

리네이밍 기법에서는 Autohotkey에서 제공하는 오픈소스 소스코드 파일 내의 hook.cpp와 defines.h 코드 일부를 수정하여 보안 프로그램이 탐지하지 못하도록 한다.



〔그림 4〕 hook.cpp의 수정되어야 할 코드

〔그림 4〕의 hook.cpp에서는 코드 내의 키보드의 뮤텍스 이름과 마우스 뮤텍스의 이름인 "AHK Keybd" 문자열과 "AHK Mouse"문자열 대신에 의미 없는 문자열을 적어서 탐지하지 못하게 막는다.



〔그림 5〕 defines.h의 수정되어야 할 코드

또한, 〔그림 5〕의 defines.h에서는 windows GUI사용을 탐지하는 것을 막기 위해서 코드 내의 클래스에 "AutoHotkeyGUI" 문자열을 위와 같이 의미 없는 문자열로 수정하여 보안 프로그램이 탐지하지 못하게 만든다. 리네이밍 우회기법은 코드를 수정한 후에는 debug 모드 대신 release모드로 실행파일을 생성한다. release 모드를 사용함으로써, 관련된 디버그 정보를 없애며 탐지가 어렵게 만드는 우회 기법이다.

3.2.2 우회 컴파일러 기법

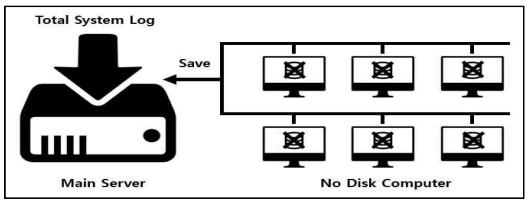


〔그림 7〕 우회 컴파일러의 함수

우회 컴파일러의 대표적인 방법은 오토핫키 설치 시에 제공되는 컴파일러(Ahk2.Exe)를 변형하는 형태로 제작되는 것이다.

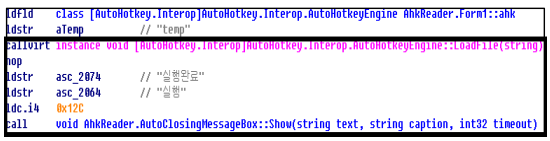
기존 컴파일러에서 제공되는 함수 라이브러리는 14개를 사용하지만, 〔그림 7〕의 우회 컴파일러의 함수를 분석한 결과 Crypt함수를 추가하는 등 여러 함수를 추가해 약 482개의 함수를 사용하고, Auto와 연관된 문자열을 바꾸는 등의 방식으로 제작된다.

3.2.3 메인 서버 우회 기법



〔그림 8〕 NDS(No Disk System)의 구조

〔그림 8〕의 노 디스크 시스템(No Disk System)[1]으로 운영되는 PC방이 늘어나면서, 노 디스크 시스템 구조를 악용한 매크로 작업장이 활성화되고 있다. 노 디스크 시스템은 클라이언트 단에 있는 컴퓨터에 하드가 없는 대신 로그를 메인 서버의 컴퓨터에 저장하는 시스템이다. 보안 탐지 프로그램은 클라이언트 단에 있는 컴퓨터의 매크로 로그가 없는 것을 보고, 매크로의 동작 여부를 판단하기에 매크로를 탐지하지 못한다. 따라서 우회 기법에 위의 노 디스크 시스템을 이용하기도 한다.



〔그림 9〕 AutoHotkey의 Interop 라이브러리 사용

최근에는 이러한 노 디스크 시스템의 원리를 비슷하게 이용하여 [2].NET 호스팅 기법을 사용한다. 〔그림 9〕의 AutoHotkey. Interop 라는 .NET 호스팅 환경과 통신이 가능한 모듈을 사용하여, 메인인 웹 호스팅 서비스 단에서 오토핫키 매크로 실행 파일을 동작시킨다. 따라서 보안 탐지 프로그램은 컴퓨터의 매크로 실행 로그가 없는 것을 보고, 매크로의 동작 여부를 판단하기에 탐지하지 못한다.

1) No Disk System : 하드가 없이 메인 서버의 컴퓨터에 연결된 컴퓨터의 로그를 전송하여 기록하는 시스템
2) .NET Hosting : 운영체제 및 프로그래밍 언어에 관계없이, 인터넷을 통해 응용 프로그램을 호스팅 할 수 있는 서비스

IV. 대응방안 연구

4.1 리네이밍 기법 대응방안

로그 탐지는 주로 문자열 위주로 이루어지기 때문에 문자열을 바꾸면 탐지가 되지 않는다. 대응 탐지를 위해 키보드와 마우스 이벤트에 있는 문자열을 위주로 기본 문자열과 달라졌는지 비교한다. 그리고 다른 이벤트 코드가 수정되었을 때도 탐지할 수 있도록 발생할 수 있는 문자열 패턴을 정리한다. 패턴들을 분석하여 Aircrack에서 사용되는 딕셔너리 리스트(Dictionary List)와 같은 파일을 만들어서 보안 프로그램이 탐지할 수 있도록 한다.

4.2 우회 컴파일러 기법 대응방안

기본 컴파일러에서 변형한 우회 컴파일러를 사용했다면 기본 컴파일러를 분석하여 사용하는 함수를 확인한다. 또한 기본 컴파일러에서 사용하는 함수를 기반으로 하여, 기본 컴파일러가 아닌 다른 함수를 실행하는 프로그램을 탐지할 수 있도록 한다.

탐지를 방해하기 위해 실행파일이 3)패킹(Packing)이 되어있을 때, 패킹 탐지를 위해서 오픈 패킹 탐지 툴인 PEID, MRC와 REMINDer 등을 이용하거나 4)엔트로피(Entropy) 기반의 탐지가 부가적으로 이루어져야 한다.[4]

4.3 메인 서버 우회 기법 대응방안

노 디스크 시스템(No Disk System)은 클라이언트 PC가 활성 상태인 경우에는 해당 시스템 사용자의 행위가 메인 서버의 swp 파일에 고스란히 남게 된다.[5] 따라서 메인 서버의 MAC 주소 정보를 사용하여, swp파일을 추출하는 방법이 있고, swp의 이용 흔적을 확인할 수 있게 하는 툴을 이용하는 방법이 있다. 동시에 탐지가 이루어지지는 않겠지만, 악성 매크로에 대한 제재는 이루어질 수 있어서 최적의 대응방안이라 생각한다.

또한 .NET 호스팅 기법의 Interop와 같은 오픈소스를 분석하여 웹 호스팅으로 연결하는 라이브러리 사용을 막아야 한다. 위처럼 컴파일하지 않아도 실행이 가능하기 때문에 라이브러리 사용을 탐지하는 것만이 대응방안이 될 수 있다.

3) 패킹(Packing) : 프로그램의 코드를 분석하기 어렵도록 암호화하거나 정크 코드(불필요한 쓰레기 코드)를 삽입하는 행위

4) 엔트로피(Entropy) : 한 메시지에 들어갈 수 있는 정보량의 비트 수

V. 결론

매크로의 우회기법 조사에서 코드 수정부터 시작하여, 컴파일러, 시스템 구조 등을 이용하여 매크로의 우회기법들이 계속해서 발전하고 있다. 따라서 해당 매크로의 기본적인 구조를 파악하고 단계별로 개선점을 찾아야 한다.

본 연구에서는 두 가지의 매크로의 종류와 세 가지의 우회 기법들을 정의하고, 대응방안을 제시하였다. 또한 앞서 기술한 우회 기법들은 현재에도 쓰이고 있는 것을 확인하였고, 프로그램들이 어떤 원리로 동작하는 지에 대해서 분석하였다.

하지만, 오토핫키 매크로에 한해서 논문을 작성했지만, C++이나 C#을 작성한 매크로는 분석하지 못했다. 또한, 안드로이드 모바일 게임 관련해서는 매크로 탐지 기법을 실험해보지 못했다.[6]

향후에는 게임 관련 패킷을 분석하여 위조 패킷을 보내는 매크로에 대한 분석과 대응방안에 대해 연구할 예정이다.

[참고문헌]

[1] K.M. Woo, H.M. Kwon, H.C. Kim, C.K.Kim, and H.K. Kim, "What Can FreeMoney Tell Us on the Virtual BlackMarket," ACM SIGCOMM, vol. 41, no. 5,pp. 392-393, 2011.

[2] 강아름, 우지영, 박주용, 김휘강, "온라인 게임 봇 탐지를 위한 사용자 행위 분석", 정보보호학회논문지, vol. 22, no.2,pp 225 - 238. 2012

[3] 최진혁, "온라인 게임 아이템 불법취득에 대한 Policing 관점에서의 위험성 고찰", 한국경찰연구학회, 2010, vol.9, no.2, pp. 185-222

[4] 한승원, 이상진, "Packed PE File Detection for Malware Forensics", 한국정보처리학회논문지, vol. 16C (5), no. 5. pp 555 - 562. 2009

[5] 김동화, 남궁재웅, 박정흠, 이상진, "No Disk System 환경에서의 사용자 행위 분석", 정보보호학회논문지, vol. 23 (6), no.3,pp 491 - 500. 2013

[6] 김정현, 이상진, "안드로이드 모바일 게임 환경에서의 터치 이벤트 정보를 이용한 매크로 탐지 기법 연구", 정보보호학회논문지, vol. 25, no.5,pp 1123 - 1129. 2015

2017 한국정보보호학회 동계학술대회

E4 암호 이론(I)

염 용 진 (국민대)

- **S-box기반 암호의 최적 차분경로 자동 탐색 방법(MILP 적용)**
  박연지, 이호창(고려대학교), 홍득조 (전북대학교), 홍석희 (고려대학교)
- **부호 기반 암호 연구동향 조사**
  박병규, 김수리, 김희석, 홍석희 (고려대학교)
- **암호데이터 중복제거 환경에서 데이터 블록간의 연계를 이용한 소유권 관리 기술**
  김원빈, 이임영 (순천향대학교)
- **SPN 구조를 갖는 Block Cipher에서 Key Schedule이 차분 성질에 미치는 영향**
  김현우, 김성겸, 강형철 (고려대학교), 홍득조(전북대학교),
  홍석희 (고려대학교)

S-box기반 암호의 최적 차분경로 자동 탐색 방법(MILP 적용)

박연지*, 이호창*, 홍득조**, 홍석희*

*고려대학교 정보보호대학원, **전북대학교 IT 정보공학과

Applying MILP Method to Searching Optimal Probability of Bit-based Differential Characteristics

YeonJi Park*, HoChang Lee*, Deukjo Hong**, Seokhie Hong*

*Graduate School of Information Security, Korea University.

**Department of Information Technology, Chonbuk National University.

요 약

본 논문은 MILP를 기반으로 차분 경로를 분석함에 있어 비트 단위 분석을 통해 차분의 전파 경로를 보다 정확히 예측하면서, 확률변수와 Espresso 알고리즘을 활용하여 보다 간결한 제약식과 과정을 통해 차분 경로 탐색을 가능하게 하는 방법을 소개한다. 해당 방법을 GIFT-64에 적용하여 실험한 결과 1개 라운드를 48개의 제약식으로 표현가능했으며 [7]에서 소개된 방법과 대비하여 활성 S-box갯수는 많으나 확률이 높은 차분 경로를 찾을 수 있었다. 이 방법을 활용해 향후 이러한 구조의 암호에 대해서 보다 유용한 분석 표준을 제시할 수 있고 선형 분석이나 불능차분 분석에도 활용 가능할 것으로 예상된다.

I. 서론

차분 분석은 Eli Biham과 Adi Shamir에 의해 제안되어[1] 현재까지 블록암호의 분석에 보편적으로 사용되는 방법이다. 차분 분석은 평문 쌍의 차분 값($\Delta P = P \oplus P'$)을 알면 그에 대응되는 암호문의 차분 값($\Delta C = C \oplus C'$)을 확률적으로 예측할 수 있다는 사실에 기반하며 이를 통해 차분 경로를 구성하여 key를 찾아낸다.

자동화 도구를 사용하면 차분 분석에 필요한 차분경로를 효율적으로 구성이 가능하다. 자동화 도구는 대칭키 암호의 분석과 디자인 설계를 효율적으로 하기 위해 사용되는 도구인데 그 중 Mixed Integer Linear Programming(MILP)는 Mouha, Wu, Wang에 의해 대칭키 분석에 사용될 수 있음이 알려진 분석도구이다[2]. MILP 문제는 선형부등식 형태의 제약식과 주어진 식의 최댓값 또는 최솟값을 구해주는 목적함수로 구성되며 MILP Solver는 목적함수를 만족하는 최적의 결과를 찾아준다.

기존 연구로는 Mouha가 Word 단위로 MILP를 사용해 차분 특성 및 선형 특성을 탐색하였으며[2] 이후 Sun에 의해 bit 단위로 차분 경로를 분석함으로써 차분의 확산 형태를 보다 정확하게 분석 가능함을 보였다[3]. Fu는 ARX 구조 암호에 대해 MILP를 이용한 모델링을 적용하여 차분 및 선형 특성을 탐색하였으며[4], Sasaki, Todo[5], Xiang[6]은 각각 SPN구조에서 불능차분을 찾는 방법 , Division property를 이용하여 integral distinguisher를 찾는 방법을 제안하였다. 그러나 이렇듯 자동화 탐색 방법이 발전해 왔음에도 MILP를 활용한 비트 기반 분석은 연산량이 지수승으로 증가하는 문제로 인해 보다 정확한 경로와 확률을 계산할 수 있음에도 널리 쓰이지 못하였다.

본 논문에서는 이를 개선하기 위해 Espresso 알고리즘이라는 논리식 최소화 알고리즘을 이용하여 제약식을 최소화하여 연산량을 대폭 축소하였으며, 제약식 내에 입/출력변수와 확률변수를 포함하여 동시에 처리함으로써 차분분포표(Differential

Distribution Table, DDT)로 알 수 있는 S-box의 차분성질을 이용해 최적화 된 확률을 구할 수 있었다. 제안하는 방법을 GIFT[7]에 적용하여 현재 활성 S-box의 하한선만 계산되어있는 GIFT의 차분특성에 대해 보다 정확한 확률과 차분 분포를 확인할 수 있었으며 그 결과 활성 S-box갯수는 기존 분석보다 많지만 확률은 보다 높은 경로를 확인할 수 있었다.

II. 관련 이론

1.1 표기법

본 논문에 사용된 표기법은 다음과 같다.

- o Δ: 차분 값($\Delta P = P \oplus P'$)
- o $S(P)$: S-box를 통과한 X의 결과 값
- o $\oplus$: 배타적 논리합(Exclusive-Or, XOR)
- o $\wedge$: 논리곱(logical Conjunction, AND)
- o $\vee$: 논리합(logical disjunction, OR)
- o x_i: x의 오른쪽에서 i번째 비트(left most)
- o per[i][j]: i번째 S-box의 j번째 출력을 전치 함수에 입력한 결과 값

1.2 차분 성질

차분은 여러 가지 형태를 가질 수 있으나 통상적으로 사용하는 차분은 XOR 차분이다. 암호는 비선형 구조와 선형 구조의 조합으로 이루어지는데 그 중 차분의 성질이 변하는 구간은 비선형 구조이다[1].

$$(P \oplus K) \oplus (P' \oplus K) = P \oplus P'$$

$$S(P \oplus K) \oplus S(P' \oplus K) \neq S(P \oplus P')$$

그러므로 S-box기반 암호에서 차분의 성질이 달라지는 구간은 S-box를 통과하는 구간이며, 특정 입력차분이 S-box에 입력될 경우 S-box의 성질에 따라 일정확률로 특정 출력차분으로 출력된다. 모든 입력차분에 대한 모든 출력차분을 표 형태로 만든 차분분포표를 작성하여 입력차분에 대한 출력차분의 분포를 확인할 수 있다.

표 1의 3비트 S-box의 예시를 살펴보면 입력차분값(ΔP)이 $011_{(2)}$로 S-box에 입력되면 출력차분값(ΔC)이 $001_{(2)}$이 될 확률은 2/8 = 1/4 이다.

표 1. Sample 3비트 S-box의 차분분포표

ΔP \ ΔC	000	001	010	011	100	101	110	111
000	8	0	0	0	0	0	0	0
001	0	0	4	0	0	4	0	0
010	0	2	0	2	2	0	2	0
011	0	2	0	2	2	0	2	0
100	0	0	4	0	0	0	0	4
101	0	0	0	0	0	4	0	4
110	0	2	0	2	2	0	2	0
111	0	2	0	2	2	0	2	0

차분경로에서의 확률 계산은 각 S-box에서의 연산이 독립이라는 가정 하에 각 라운드 별 모든 S-box의 예상 입력차분에서 예상 출력차분이 나올 확률을 곱하여 계산한다. 그렇기 때문에 높은 확률을 가진 입/출력차분을 연결하여 차분 경로를 구성하면 효율적으로 차분 분석을 할 수 있게 된다.

1.3 Word-based / bit-based 차분 특성 분석 방법

MILP로 차분 특성을 분석하려면 입력과 출력에 대한 제약식을 설정하여 불가능한 조합을 걸러내고 목적함수를 통해 가능한 조합을 최적화하도록 암호를 모델링해야한다. 모델링 방법은 Word-based와 bit-based 두 가지 방식으로 나뉜다.

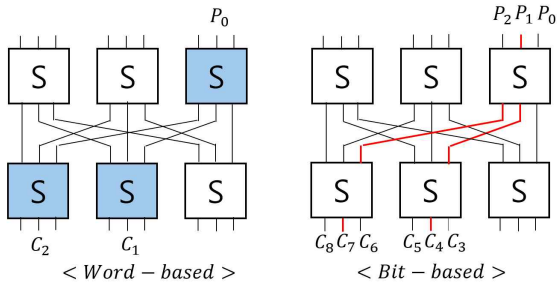


그림 1. Word-based 방식과 bit-based 방식 비교

Word-based 방식은 최소한의 활성 S-box를 만드는 차분경로가 가장 높은 확률을 갖는 차분경로라는 가정에서 시작한다. 여기서 활성 S-box란 입력차분이 존재하는 S-box를 의미하는데 차분경로에서의 활성 S-box가 많아질 경우 곱해지는 1보다 작은 확률 값이 많아지므로 대체로 전체 확률 값이 낮아지는 결과를 가져오기 때문이다. 활성 S-box를 표현하기 위해 S-box의의 입력과 출력을 Word 단위로 표현하고 차분분포표의 특성에 따라 S-box를 최소한으로 거치는 차분

경로를 구성한다. Word-based 방식은 입력과 출력을 단순화 하여 연산시간이 줄어든다는 장점이 있지만 차분의 정확한 확산을 표현할 수 없기 때문에 정확한 확률 계산 및 차분 특성의 분포 형태를 알 수 없다는 단점이 있다.

Bit-based 방식은 입력차분과 출력차분을 bit 단위로 표현하는 방식이다. 실제 차분분포표에 따라 차분의 입/출력 경로를 구성하며 이에 따라 차분의 정확한 확산 및 차분 경로의 확률을 구할 수 있다는 장점이 있는 반면 경로탐색에 필요한 연산시간이 비트의 지수승으로 증가한다는 단점이 있다. 본 논문에서는 보다 정확한 값을 표현하기 위해 Bit-based 방식을 활용하였으며 Boolean 함수와 합의 곱(Product-of-Sum)을 이용해 차분 성질을 표현했다.

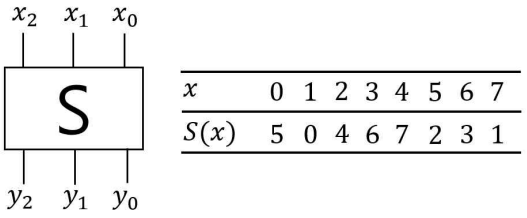
n비트 입력과 n비트 출력을 갖는 S-box를 2n비트의 입력과 1비트의 출력을 갖는 Boolean 함수 $f(x,y)$로 표현한다고 가정하자.

$$f(x,y)=\bigwedge_{c\in 0,1^{2n}}((\bigvee_{i=1}^{n}(x_i\oplus c_i))\bigvee(\bigvee_{i=1}^{n}(y_i\oplus c_i)))$$

차분분포표에 의해 가능한 입력차분과 출력차분은 출력을 1로 두고(c=1) 불가능한 출력은 0으로 두면(c=0), AND 연산에 의해 하나라도 만족하지 않는 입/출력 차분의 조합은 걸러지므로 $f(x,y)\geq 1$이라는 제약식을 통해 불가능한 출력은 쉽게 걸러낼 수 있게 된다. 이를 통해 n비트 입력과 출력을 갖는 S-box연산을 표현할 수 있는 제약식을 만들어 낼 수 있다.

III. S-box 기반 암호의 차분 경로 탐색 방법

1.1 Bit-based S-box의 차분 성질 표현



x	0	1	2	3	4	5	6	7
$S(x)$	5	0	4	6	7	2	3	1

그림 2. Sample 3비트 S-box

위 S-Box의 차분분포표를 작성하면 표1과 같은 결과가 도출되며 이에 따라 입력차분 3비트와 출력차분 3비트, 가능한 확률 2가지(1/4 또는 1/2)를 하나의 벡터로 표현하면 다음과 같은 형태로 표현할 수 있다.

$$v=(x_2,x_1,x_0,y_2,y_1,y_0,p^{1/2},p^{1/4})$$

Ⅱ장 1.3절에서 나타낸 것과 같이 8비트 벡터 v를 합의 곱으로 표현하면 표 1에서 0이 아닌 출력을 갖는 값 이외의 나머지 가능한 모든 8비트 벡터를 이용해 제약식을 만들 수 있다.

$$f(x,y,p)=(x_2\vee x_1\vee x_0\vee y_2\vee\overline{y_1}\vee y_0\vee\overline{p^{1/2}}\vee p^{1/4})\wedge\ldots$$
$$\ldots\wedge(\overline{x_2}\vee\overline{x_1}\vee\overline{x_0}\vee\overline{y_2}\vee\overline{y_1}\vee y_0\vee p^{1/2}\vee\overline{p^{1/4}})$$

다음 제약식은 벡터v의 각 자리와 1:1로 대응되어 $(x_2\vee x_1\vee x_0\vee y_2\vee\overline{y_1}\vee y_0\vee\overline{p^{1/2}}\vee p^{1/4})$는 '00001010'으로 표현 가능하며 이는 표1에서 나올 수 없는 조합 중 하나인 0x0→0x2(확률 1/2)을 나타낸다.

1.2 논리식 최소화 알고리즘

Ⅲ장 1.1절의 결과는 차분분포표상의 불가능한 조합이 모두 표현된다. 예시로 든 S-box에서 가능한 조합은 23개 이므로 제약식으로 출력되는 벡터 수는 2^8-23개다. 제약식의 수가 많아지면 비교 연산의 수가 많아지므로 탐색연산속도에 영향을 줄 수밖에 없다. 이 때문에 제약식을 단순화하기 위해 논리식 최소화 알고리즘을 활용되며 논리식 최소화 알고리즘을 거친 제약식은 다음과 같은 형태의 벡터로 출력된다.

10-01--0 1

이 벡터는 8비트 벡터v의 각 자리가 나타내는 변수값과 대응되며 0은 $+x_i$, 1은 $+(1-x_i)$을 나타낸다. '-'는 0이든 1이든 어떠한 값이 들어가도 상관없음을 의미하고 맨 끝의 1은 벡터의 결과가 1비트 결과값으로 출력됨을 의미한다.

1.3 MILP를 이용한 최적화

MILP 문제를 만들기 위해서는 목적함수 설정, 제약식 작성, 변수선언이 필요하다. 본 논문에서는 MILP Solver 프로그램으로는 Gurobi Optimizer를 활용하였으며 MILP문제는 LP파일로 만들어 실행하였다.

입력차분과 출력차분, 한 S-box에서 출력가능한 확률을 변수로 놓고 Ⅲ장 1.1절 및 Ⅲ장 1.2절의 과정을 거쳐 제약식을 작성, r라운드 모델을 만든다.

이후 해당 모델에 가능한 모든 비트조합을 대입하여 목적함수를 가장 최적화 해주는 비트조합을 찾아낸다. S-box 기반 암호의 차분 경로 탐색에서 목적함수는 차분경로에서 나타나는 확률들의 가중치(확률을 2^{-n}이라고 할 때, n값) 합이 최소가 되는 값이다.

목적함수에 대한 최적화가 끝나면 각 라운드별 입력차분과 출력차분, 확률을 나타내는 변수에 대입된 비트 값을 확인하여 최적화 된 차분 경로를 확인할 수 있다.

Algorithm 1. Efficient Method of Searching Optimal Differential Characteristics

Input: ΔX, ΔY, p
Output: Optimal Differential trail
1. Procedure find Differential trail(ΔX, ΔY, p) do
2. for $i \in \Delta X$ do
3. for $o \in \Delta Y$ do
4. for $s \in p$ do
5. creat r-ROUND model “DC.lp”
6. m = read(“DC.lp”)
7. m.optimize()
8. for i 0 to r-1 do
9. write ΔX, ΔY, p

IV. 탐색 방법 적용(GIFT-64)

1.1 GIFT[7]

GIFT는 2017년 Subhadeep banik 등에 의해 제안된 경량 블록 암호이다.

PRESENT와 같은 디자인 전략을 가져왔지만 선형공격에서 PRESENT의 단점을 보완하고 보다 빠르고 작게 구현 가능한 형태로 효율성을 높인 암호이다.

GIFT는 64, 128 두 종류의 블록 사이즈를 가지며 각각 28라운드, 40라운드를 갖는다. S-box는 공통적으로 4bit S-box를 사용하며 SubCell, PermBits, AddRoundKey 함수로 구성된다.

1.2 새로운 MILP 모델 적용

GIFT S-box는 4bit의 6-uniform S-box로 입/출력 차분에 대해 가능한 확률은 $2^{-1.415}$, 2^{-2},2^{-3} 3가지가 있다.

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$GS(x)$	1	a	4	c	6	f	3	9	2	d	b	7	5	0	8	e

그림 3. GIFT S-box(4비트)

위 S-box에 대하여 입력차분 4비트, 출력차분 4비트와 가능한 확률 3가지를 다음과 같은 형태의 벡터로 표현하였다.

$$v = (x_3, x_2, x_1, x_0, y_3, y_2, y_1, y_0, p^{3/8}, p^{1/4}, p^{1/8})$$

Ⅲ장 1.1절의 방법으로 불가능한 조합의 벡터를 모두 출력하고 그 결과를 논리식 최소화 알고리즘으로 활용되는 Espresso 알고리즘과 Quine-McCluskey 알고리즘으로 최소화 하여 다음과 같은 결과값을 얻을 수 있었다.

표 2. 논리식 최소화 알고리즘 적용 결과

구조	불가능한 조합	QM	Espresso
GIFT S-box	1950	105	48

이후 MILP를 이용한 최적화를 위해서 LP파일을 생성하였으며 이 때 입력변수는 $X[r-1][i]$, 출력변수는 $X[r][i]$이며($0 \le i < 64$) 총 16개의 4bit S-box에 대응되는 각각의 확률은 $p^i[r][j]$ ($i \in 3/8, 1/4, 1/8$, $0 \le j < 16$)으로 표현하였다.

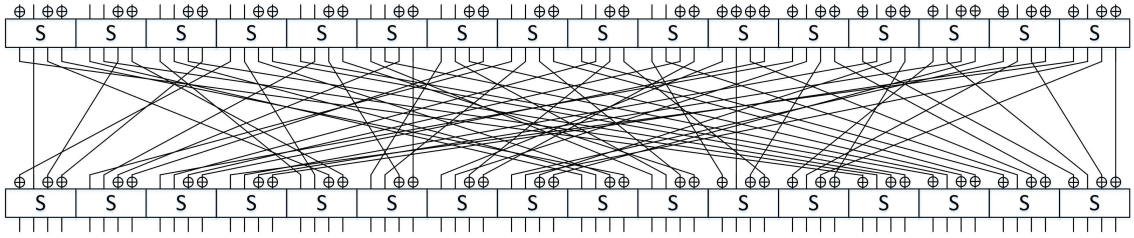


그림 4. GIFT 구조

Binary

$$X[0][0]\ldots X[0][63], p^{3/8}[0][0], \ldots p^{1/8}[0][15]$$

$$p^{3/8}[r-1][0] \ldots p^{1/8}[r-1][15], X[r][0]\ldots X[r][63]$$

목적함수는 확률변수에 가중치를 준 뒤에 가중치를 준 확률변수 값을 최소화하도록 하였다.

$$Obj = 1.415\,p^{3/8}[0][0] + 2\,p^{1/4}[0][0] + \ldots\ldots$$
$$+ 2\,p^{1/4}[r-1][15] + 3\,p^{1/8}[r-1][15]$$

제약식은 Espresso 알고리즘을 통해 도출된 48개의 제약식이 각 S-box마다 적용되므로 제약식의 총 개수는 48×16+1(입력차분이 0이 아님을 나타내는 식 포함)개 이다. 입력 변수는 $X[r][i]$ 출력 변수는 y대신 $X[r+1][per(i)]$를 대입하며 $+(1-x_i)$형태는 MILP의 특성상 상수를 모두 부등식의 우변으로 옮겨서 처리하였다.

$$-X[0][3]-X[0][2]-X[0][1]+X[0][0]-X[1][17] \geq -3$$
$$\vdots$$
$$X[r-1][63]-X[r-1][62]-X[r-1][61]+X[r-1][60]$$
$$+X[r][62]-X[r][28]+p_a[r-1][15] \geq -2$$

1.3 적용 결과

표 3. MILP적용 최적 차분경로 탐색 결과(GIFT-64)

ROUND	$-\log_2 p$	time	활성 S-box	활성 S-box[7]
1	1.415	0.07s	1	1
2	3.415	0.24s	2	2
3	7	3s	3	3
4	11.415	1.7m	6	5
5	17	4.5m	7	7
6	22.415	58m	10	10
7	28.415	4h	13	13

새로운 MILP 모델 적용 결과 총 7라운드에 대해 다음과 같은 라운드별 가장 높은 확률의 경로를 확인할 수 있었으며 해당 경로에 대한 입력 차분 및 출력차분의 정확한 위치도 확인할 수 있었다. GIFT에 대해 차분 경로와 확률을 정확히 분석한 이전 연구가 없기 때문에 비교 대상은 없으나 GIFT 제안 논문인 [7]의 결과와 비교하였을 때 활성 S-box 개수는 제안논문보다 많음에도 가장 높은 확률을 갖는 경로를 찾을 수 있었다.

V. 결론

본 논문에서는 MILP를 이용한 효율적 차분 경로 분석 방법을 소개하였다. 제안하는 방법을 통해 확률과 차분경로를 한 번에 연산하여 보다 간편하고 빠른 속도로 차분 경로를 분석할 수 있었다. 또한 GIFT-64에 대한 비트단위 분석으로 보다 정확한 확률 및 입/출력 차분값을 알 수 있었으며 이전 연구 결과 보다 많은 활성 S-box를 가지지만 최고의 확률을 갖는 새로운 경로도 찾을 수 있었다. 향후 해당 모델을 이용해 선형관계식이나 불능차분, Division Property등 다양한 성질을 분석하는데 적용할 수 있을 것으로 판단되며 다른 암호구조에 적용하여 결과를 도출 가능할 것으로 판단된다.

[참고문헌]

[1] Eli Biham, Adi shamir, Differential Cryptanalysis of DES-like Cryptosystem, Journal of Cryptology, 1990..

[2] Nicky Mouha, Qingju Wang, Dawu Gu, and Bart Preneel, Differential and Linear Cryptanalysis Using Mixed-Integer Linear Programming, Inscrypt 2011.

[3] Siwei Sun, Lei Hu, Peng Wang, Kexin Qiao, Xiaoshuang Ma, and Ling Song. Automatic Security Evaluation and (Related-key) Differential Characteristic Search: Application to SIMON, PRESENT, LBlock, DES(L) and Other Bit-Oriented Block Ciphers, ASIACRYPT 2014.

[4] Kai Fu, Meiqin Wang, Yinghua Guo, Siwei Sun, and Lei Hu. MILP-Based Automatic Search Algorithms for Differential and Linear Trails for Speck. FSE, 2016.

[5] Yu Sasaki and Yosuke Todo. New Impossible Differential Search Tool from Design and Cryptanalysis Aspects - Revealing Structural Properties of Several Ciphers. EUROCRYPT, 2017.

[6] Zejun Xiang, Wentao Zhang, Zhenzhen Bao, and Dongdai Lin, Applying MILP Method to Searching Integral

Distinguishers Based on Division Property for 6 Lightweight Block Ciphers, ASIACRYPT, 2016.

[7] Subhadeep Banik, Sumit Kumar Pandey, Thomas Peyrin, GIFT: A Small Present Towards Reaching the Limit of Lightweight Encryption, CHES, 2017.

[8] Siwei Sun, David Gerault, Pascal Lafourcade, Qianqian Yang, Yosuke Todo, Kexin Qiao, and Lei Hu. Analysis of AES, SKINNY, and Others with Constraint Programming. Symmetric Cryptol, 2017.

[9] Ling Sun, Wei Wang, Meiqin Wang Automatic Search of Bit-Based Division Property for ARX Ciphers and Word-Based Division Property. ASIACRYPT, 2017.

부호 기반 암호 연구동향 조사*

박병규*, 김수리**, 김희석***, 홍석희**

*고려대학교 일반대학원 사이버국방학과, **고려대학교 정보보호대학원, ***고려대학교 사이버보안학과

Survery of Code-based Cryptography research trend

Byeung-Gyu Park*, Suh-Ri Kim**, Hee-Seok Kim***, Seok-Hie Hong**

*Department of Cyber Defence, General Graduate School, Korea University, **Graduate School of Information Security, Korea University, ***Department of Cyber Security, Korea University

† 요 약

기존에 사용하는 RSA, ECC등의 암호는 양자컴퓨터가 나오게 되면 더 이상 안전하지 않게 된다. 따라서 래티스, 부호기반 암호 다변수함수 기반 암호 등이 유망한 대체 암호로 여겨진다. 그 중 부호기반암호는 빠른 속도로 인하여 연구가 활발하게 진행중이다. 본 논문에서는 부호기반암호의 원리와 기반이 되는 부호이론 및 암호체계에 대해서 소개하고 여러 변형들을 키 크기, 속도를 중점으로 조사하여 표로 도식화 해놓았다.

I. 서론

현재의 공인인증서나 전자서명에 사용되는 공개키 암호인 RSA와 ECC는 각각 인수분해와 타원곡선 이산대수 문제의 어려움에 기반을 두고 있다. 하지만 양자 컴퓨터가 개발되어 Shor알고리즘이 구현될 경우 RSA와 ECC는 더 이상 안전하지 않기 때문에 이에 대응하는 후양자 암호에 대한 연구가 활발히 진행하고 있다. 현재 후양자 암호의 후보로는 다변수 함수기반 암호, 래티스기반 암호, 부호기반 암호, 아이소제니기반 암호, 해시기반 전자서명 체계가 있다. 이들 중에서 부호기반 암호는 암호화 과정이 단순한 행렬 곱셈 연산으로 다른 암호 체계보다 속도가 빠른 편으로 후 양자 암호의 후보로 연구가 활발히 진행되고 있다. 이에 본 논문은 부호기반 암호의 연구 동향에 대해 서술하고자 한다.

본 논문의 구성은 다음과 같다. 우선 2장에서 부호이론이 어떻게 부호기반 암호로 이용되는지와 주요 부호이론을 소개하고, 3장에서는 현재까지 연구된 주요 부호기반 암호에 설명을 한다. 마지막으로 결론에서 각 부호기반 암호체계에 대해 속도, 키 크기 구현상의 특징에 대한 표를 보이고 향후 계획에 대해서 설명하도록 한다.

II. 부호이론의 원리

2.1 부호어의 복호화

부호기반 암호는 부호이론으로부터 생겨나게 되었다. 부호이론은 오류가 발생할 수 있는 통신로에서 통신을 할 때, 메시지에 오류가 추가되면 메시지를 받은 수신자가 오류를 벗겨내서

* 이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2017-0-00520, (ICT 기초연구실) SCR-Friendly 대칭키 암호 및 응용모드 개발)

†

올바른 메시지를 받을 수 있도록 만들어졌다. 기본적인 흐름은 메시지를 오류가 발생할 수 있는 통신로를 통해 그냥 보내지 않고 부호이론을 이용해 부호화(encoding) 해서 보내면, 이를 받은 수신자가 오류 정정 과정을 통해 오류를 검출 및 제거 하고, 복호화(decoding)과정을 거쳐 올바른 메시지로 복구 해내는 과정이다. 본 논문에서는 오류 정정 과정 및 복호화과정을 통 틀어 복호화를 한다고 표현하겠다. 부호기반 암호체계는 위와 같은 복호화과정을 트랩도어(trapdoor)로 이용해서 만들어졌다.

복호화를 하려는 사람이 부호어(codeword)생성에 필수 정보인 생성행렬(generator matrix) 또는 홀짝검사행렬(parity-check matrix)를 알 경우에는 오류가 추가된 부호어를 최적화된 복호화 알고리즘을 이용해서 복구 할 수 있지만, 위의 정보를 모를 경우에는 information set이나 statistical 복호화 알고리즘과 같은 일반적인 복호화 알고리즘을 사용해야 한다. 후자의 경우에도 복호화가 가능하지만, 최적화 알고리즘보다 속도가 훨씬 오래 걸리고 부호어의 크기를 크게 사용하면 위의 알고리즘을 이용해서 복호화 하는 것은 NP hard 문제가 된다. 부호기반 암호체계는 부호어 생성에 필요한 정보를 모르는 사용자는 복호화를 하기 어려움을 이용해서 공개키 암호로써 설계되었다.

2.2 선형 부호(linear code)

오류 정정 부호중에 현재 부호기반 암호체계에 사용되고 있는 주요 부호인 Goppa 부호, QC-MDPC 부호 등 모두 선형 부호를 기반으로 설계하였다. 본 장에서는 먼저 선형 부호에 대해 알아보겠다.

정의 2.2.1 (이진 선형 부호)

이진 선형 부호 - (n,k)는 k비트열 메시지를 n비트열 부호어로 바꾸어주는 일대일 함수이다. $(0,1)^k \rightarrow (0,1)^n$.

정의 2.2.2 (생성행렬과 홀짝검사행렬)

$k \times n$행렬 G의 열벡터공간이 F위에서의 C를 생성하고, 열벡터공간의 차원이 k라면, 행렬 G를 F위에서의 (n,k)-C의 생성행렬이라고 한다. $(n-k) \times n$ 행렬 H에 대해서 H^T의 열벡터들이 G의 영공간의 기저라면, H를 부호C의 홀짝검사행렬이라고 한다. H를 이용해서 n비트 벡터가 C에 포함되는지 검증 할 수 있다. $H \times c^T = 0 = m \times G \times H^T = m \times 0$,

정의 2.2.3 신드롬(Syndrome)

벡터 $\hat{c} = c + e \in F_p^n$의 신드롬은 F_q^{m-k}위의 원소 $S_{\hat{c}} = H \times \hat{c}^T = H \times (c^T + e^T) = H \times e^T$이다.

일반적인 선형 부호의 부호어 생성 방법은 메시지 m과 위에서 정의한 생성행렬 G를 곱하여 부호어 c로 부호화한다. 복호화 과정은 신드롬값이 0이라면 오류가 들어가지 않은 부호어이므로, 오류 정정 과정 없이 복호화 해주면 되고, 신드롬값이 0이아니라면 해당 부호의 최적화 오류 정정 알고리즘을 이용해서 오류 정정을 하고 복호화를 하여 메시지 m을 얻게 된다.

Ⅲ. McEliece 와 Niederreiter 암호체계

1978년에 개발된 McEliece암호체계는 암호화 과정에서 랜덤화를 사용한 첫 암호 체계이다. McEliece는 임의의 선형 부호를 복호화 하는것의 어려움을 기반으로 하여, 개인키는 생성행렬 G, 가역행렬S, 순열행렬P로 구성되었으며 공개키는 $\hat{G} = SGP$이다. 암호화는 메시지 m에 생성행렬 $\hat{G}$를 곱한 후 에러 벡터 e를 더해 $c = m\hat{G} + e$로 한다. 복호화는 먼저 암호문에 P^{-1}를 곱한 다음에 개인키를 이용해 디코딩한 후, S^{-1}를 곱해 메시지 m을 얻는다. 이 때 사용하는 G는 적절한 선형 부호를 이용해 생성할 수 있지만, 현재까지는 Goppa code를 사용하는 것이 안전하다고 알려져 있다.

Niederreiter암호는 1986년에 Harald Niederreiter에 의해 개발되었다. Niederreiter 암호체계는 McEliece와 유사한 방식인데 비밀 키(secret key)로 G대신에 H를 사용하여, 암호화 과정에 $\hat{G}$을

메시지 m에 곱하고 오류를 더하는 것이 아닌, 메시지 m을 n비트열로 부호화하고 $\hat{H}$와 곱해주는 방식으로 암호화가 진행 된다. 따라서 비밀 키가 H가 되고 공개 키(public key)가 $\hat{H}$이 된다. Niederreiter 암호체계는 암호화 과정에서 메시지를 부호화한 후에 H와 곱셈을 해주므로, McEliece의 암호화 방식의 경우에 발생할 수 있는 메시지에 오류가 더해지지 않은 부분의 정보유출을 방지 할 수 있다. 또한, 암호문으로써 오류가 더해진 부호어 대신에 신드롬을 사용하기 때문에, 오류 정정과정을 거치지 않고 속도가 빠른 복호화과정으로 바로 이동 가능하다.

Ⅳ. QC-MDPC 기반 암호체계

4.1 QC-MDPC 기반 암호체계의 장점

Goppa 부호기반 McEliece 암호체계 현재 까지도 안전하다고 여겨지는 암호체계이다. Goppa 부호를 기반부호로 사용하는 이유는 Patterson 알고리즘을 이용하면 빠른 다항시간 복호화를 할 수 있고, Goppa 부호를 구성하기는 쉬우나 생성행렬 G를 찾기는 어렵기 때문이다[1].

Goppa 기반 McEliece 암호는 2008년에 일반적인 선형 부호 복호화 알고리즘중 하나인 information set 복호화 알고리즘을 이용한 공격으로 깨졌다[2]. 논문[2]에서 저자는 암호체계가 80비트 안전성을 가지기 위해서 제시한 이진 부호는 Goppa 부호-[n=1632, k=1269, t=33, HW(e) = 34](t : Goppa 다항식 $g(x)$의 최고차항의 차수)이다. 따라서 공개키의 최소 크기가 $1269\times(1632-1269)=460647bits=57KB$ 로 매우 큰 키를 사용해야 한다. 이는 키 관리에 비효율적인 면이 커서 QC-MDPC 부호기반의 암호체계가 나오게 되는 배경이 된다. QC-MDPC 부호의 정의는 다음과 같다.

정의 4.1.1(QC-MDPC code)

(n,r,w)-QC-MDPC 부호 : QC-MDPC 행렬 H에 의해 정의되는 차원이 r인 n비트 부호어이다. $C=x\in F_2^n | H\times x^T=0.$

QC_MDPC 기반 McEliece 암호체계는 2012년에 Misoczki에 의해 제안되었다[4]. QC-MDPC 기반 McEliece 암호체계는 생성행렬이 주기성을 가진다는 사실을 이용해 기존처럼 행렬 전체를 저장하는 것이 아니라 한 줄만 저장하면 되어, Goppa기반 McEliece 암호체계보다 효율적인 키 크기를 가진다. 예를들어 80비트안전성을 가지기 위해 QC-MDPC부호를 사용하면 공개키 최소 크기가 4800비트로 기존 McEliece에 비해 약 95배나 줄어들었다[3].

4.2 QC-MDPC 기반 암호의 변형

QC-MDPC기반 McEliece는 확률적인 복호화 알고리즘을 사용하기 때문에 실패율이 존재하며, 이 확률은 10^{-7}이다. 이를 이용해 공격하는 논문이 제안되었으며, 이 외에도 연산시간이 동일 하지 않아 소요시간 분석에 취약하고, 2015년에는 복호화 알고리즘의 신드롬 연산을 타겟으로 하는 전력분석 공격 방법이 제안되었다[5, 6]. 따라서 이에 대응하기 위해 2016년에는 비트 슬라이싱 기법과 일정시간 구현이 적용된 QcBits가 제안되었다[8]. 또한, QcBits의 복호화 실패율은 10^{-8}로 이전 기반 암호체계보다 실패율이 줄어들었다. 하지만, QcBits또한 2017년에 복호화 알고리즘의 신드롬연산을 타겟으로 전력분석 공격을 통해 키를 찾는데 성공했다[9].

Ⅴ. 비교 및 결론

표 5.1에서는 Goppa 기반 McEliece, QC-MDPC 기반 McEliece, QC-MDPC기반 KEM/DEM , 그리고 QcBits 각각에 대해서 개발년도, 속도, 키 크기, 평문 크기, 구현특징, 그리고 복호화 실패율에 대해서 비교분석을 보여준다. Cortex-M4 플랫폼에서의 속도 측정값은 논문 [8]을 참고 하였다. 표 5.1에서 확인 할 수 있듯이 QC-MDPC 기반 암호체계가 키의 크기가 매우 작은 것이 확인 가능하나 RSA나 ECC 암호에 비해 키의 크기가 비교적 크다. 따라서 동일한 80비트안전성을 가지면서 키 크기를 더

욱 줄이고 여러 부채널 분석에도 안전한 부호 기반 암호체계을 설계하는 것을 연구 방향으로 가지고 연구할 계획이다.

표 5.1-(속도 : Cortex-M4 플랫폼으로 측정)

암호체계		McEliece (Goppa) [2]	McEliece(QC-MDPC) [10]	KEM/DEM(QC-MDPC) [7]	QcBits [8]
개발년도		2008	2014	2016	2016
속도 (cycle)	키 생성	-	148,576,008	63,185,108	140,372,822
	암호화	-	7,018,493	2,623,432	2,244,489
	복호화	-	42,129,589	18,416,012	14,679,937
공개키 크기		460647bits	4801bits	4801bits	4801bits
평문 크기		1269bits	4801bits	32bytes	59bytes
구현 특징		information set 복호화 알고리즘에 대해 안전	암호화/복호화에 일정시간구현	-	일정시간구현, 비트슬라이싱
복호화 실패율		-	10^{-7}	10^{-7}	10^{-8}

[참고문헌]

[1] Bernstein, Daniel J., Tanja Lange, Christiane Peters (2008), Attacking and Defending the McEliece Cryptosystem.

[2] Daniel J. Bernstein, Tanja Lange, Christiane Peters. :Attacking and Defanding the McEliece Cryptosystem. PQCrypto 2008 : Post-Quantum Cryptography.

[3] Stegan Heyse, Ingo von Maurich, and Tim Guneysu, “Smaller Keys for Code-Based Cryptography: QC-MDPC McEliece Implementations on Embedded Devices,” in CHES, ser. Lecture Notes in Computer Science, G. Bertoni and J.-S. Coron, Eds., vol. 8086. Springer, 2013, pp. 273 - 292.

[4] Rafael Misoczki, Jean-Pierre Tillich, Nicolas Sendrier, Paulo S. L. M. Barreto, MDPC-McEliece: New McEliece variants from moderate density parity-check codes, IEEE International Symposium on Information Theory (2013), 2069-2073.

[5] Qian Guo, Thomas Johansson, and Paul Stankovski. A key recovery attack on MDPC with CCA security using decoding errors. Cryptology ePrint Archive, Report 2016/858,2016.

[6] Cong Chen, Thomas Eisenbarth, Ingo von Maurich, and Rainer Steinwandt. Differential power analysis of a McEliece cryptosystem. In Tal Malkin, Vladimir Kolesnikov, Allison Bishop Lewko, and Michalis Polychronakis, editors, ACNS 15, volume 9092 of LNCS, pages 538-556. Springer, Heidelberg, June 2015.

[7] Ingo von Maurich, Lukas Heberle, and Tim G□uneysu. IND-CCA secure hybrid encryption from QC-MDPC Niederreiter. In Tsuyoshi Takagi, editor, Post-Quantum Cryptography-7th International Workshop, PQCrypto 2016, Fukuoka, Japan,

[8] Tung Chou. QcBits: Constant-time small-key code-based cryptography. In Benedikt Gierlichs and Axel Y. Poschmann, editors, CHES 2016, volume 9813 of LNCS, pages 280-300. Springer, Heidelberg, August 2016.

[9] Rossi, M., Hamburg, M., Hutter, M., and Marson, M. E. (2017). A side-channel as-sisted cryptanalytic attack against qcbits. Cryptology ePrint Archive, Report 2017/596.http://eprint.iacr.org/2017/596.

[10] Ingo von Maurich, Tim G□uneysu, Towards Side-Channel Resistant Imple-mentations of QC-MDPC McEliece Encryption on Constrained Devices, in PQCrypto 2014 [8] (2014), 266-282.

암호데이터 중복제거 환경에서 데이터 블록간의 연계를 이용한 소유권 관리 기술+

김원빈*, 이임영**
순천향대학교 컴퓨터학과

Ownership Management Scheme Using Linkage Between Data Blocks in an Secure Data De-duplication Environment

Won-Bin Kim*, Im-Yeong Lee**
Dept of Computer Science and Engineering, Soonchunhyang University*,**

요 약

데이터 중복제거 기술은 데이터 스토리지에 동일한 데이터가 반복되어 저장되는 것을 방지하여 스토리지 공간을 절약할 수 있게 해주는 기술이다. 데이터 중복제거 기술에는 다양한 방식이 존재하지만 일반적으로는 파일 원본을 일정 크기의 데이터 블록을 나눈 뒤, 각 블록을 비교하는 방식으로 하여 데이터 중복제거 효율을 높인다. 따라서 각 데이터 블록마다 별개의 소유권을 가지게 되며, 각 소유권은 데이터 블록의 원본보다 작은 크기로 구성된다. 이 경우 데이터의 블록을 획득하고자 하는 공격자는 각 데이터 블록에 들어갈 수 있을 것으로 예측되는 데이터를 추측하여 데이터 소유권을 획득하는 사전공격이 가능해지게 된다. 이는 각 데이터별로 별개의 소유권을 가지게 되며, 각 소유권을 다른 블록을 취득하는데 영향을 주지 않기 때문에 발생한다. 따라서 본 연구에서는 이러한 문제를 해결하기 위한 방법을 제안한다.

I. 서론

데이터 중복제거 기술은 새로운 데이터가 저장될 때, 해당 데이터가 이미 스토리지 내부에 존재하는지 여부에 따라 저장을 수행하는 기술이다. 따라서 동일한 데이터의 반복된 저장을 방지할 수 있다. 클라우드 서버에 보관된 데이터는 원본파일을 일정한 크기로 나눈 블록형태로 보관된다. 업로드 된 각 데이터 블록은 식별자 정보를 가지고 있으며, 이 식별자는 해당 데이터를 소유한 사용자의 소유권과 연결되어있다. 하지만 각 데이터 블록은 서로 독립적인 소유권 형태를 가지기 때문에 파일 F에서 파생된 데이터 블록 A,B,C,D,E를 모두 획득하지 못하면 원본 파일 F를 복구할 수 없다. 이 문제는 반대로 5개의 데이터 블록 중 하나만을 획득하는 것도 가능하다는 것이다. 일반적으로 데이터 블록은 원본 파일에 비해 매우 작은 크기를 가진다. 따라서 공격자가 각 블록의 데이터를 유추할 수 있으며, 공격자는 예상되는 데이터를 이용하여 식별자를 생성해보는 방법으로 데이터를 추측하는 사전공격이 가능하다. 따라서 본 연구에서는 이러한 문제를 해결하기 위해 원본 파일 F로부터 파생된 데이터 블록을 서로 연결하는 링크 데이터를 생성하고, 모든 링크 데이터를 소유하여야만 원본 파일 F에 속한 모든 데이터를 획득할 수 있는 방법을 제안한다.

+ 본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 대학ICT연구센터육성 지원사업의 연구결과로 수행되었음
(IITP-2017-2015-0-00403)

II. 관련 연구

2.1 CSD(Client-Side Deduplication)

데이터 중복제거 기술은 업로드 할 데이터가 스토리지에 이미 존재하는지 여부를 판단하는 과정을 수행한다. 이 과정에서 동일한 데이터가 존재하는지 판단하는 것은 서버의 몫이다. 하지만 중복제거를 수행하는 위치는 다양한 방식이

존재할 수 있다. 대표적으로 server-side deduplication(SSD)와 client-side deduplication (CSD)로 구분할 수 있다[1]. 먼저 SSD는 모든 데이터를 업로드하고, 서버측에서 중복된 데이터를 걸러내는 역할을 수행한다. 따라서 많은 데이터 전송 트래픽이 발생할 수 있다. 반면, CSD는 서버측에 데이터의 식별자 또는 태그를 전송하여 데이터의 존재 여부를 판단하고 사용자측에 결과를 알려준다. 따라서 사용자가 모든 데이터를 업로드 할 필요가 없기 때문에 데이터 전송 트래픽이 절감되는 효과가 있다. 하지만 서버측에 데이터가 존재하는지 확인하는 비용이 중복된 데이터를 서버로 전송하는 비용과 큰 차이가 없다면 CSD의 이용 가치가 없게 된다. 따라서 본 연구에서는 데이터 중복 확인에 드는 비용이 감소될 수 있도록 연구를 수행하였다.

2.2 CE(Convergent Encryption)

CE는 암호화된 데이터의 중복제거를 수행할 수 있도록 하는 기술이다[2]. 따라서 일반적인 암호화 방식을 이용하지 않는다. 일반적인 암호화 방식에서는 키 생성을 위해 키 스케쥴링 알고리즘을 이용한다. 이 방식을 이용할 경우 서로 다른 사용자로부터 생성된 암호데이터는 동일한 원본에서 파생되었어도 동일 여부를 비교할 수 없다. 따라서 암호화된 데이터의 비교가 가능하게 하기 위해 CE가 제안되었다. CE는 해시알고리즘을 이용하여 데이터 소스를 변환하고, 변환된 해시 데이터를 키로 이용한다. 따라서 특정 데이터 소스를 암호화 할 경우 항상 동일한 암호데이터가 생성된다. 이를 통해 암호화된 두 데이터가 동일한지 비교할 수 있기 때문에 암호화된 데이터의 중복제거가 가능해지게 된다.

2.3 MLE(Message-Locked Encryption)

Message Locked Encryption(MLE)은 2013년 Bellare 등에 의해 제안된 데이터 암호화 기술이다[3]. 이 기술은 동일한 데이터 소스에서 동일한 암호화된 데이터를 생성하기 위한 방법을 제시하였다. MLE의 일반적인 접근법은 데이터소스 M을 이용하여 K←H(M)을 계산하고, K를 키로 사용하여 C←E_K(M)을 생성하는 방법을 이용한다. MLE에는 총 4가지의 암호화 기술이 포함되며 이는 각각 Convergence Encryption(CE), Hash and CE without Tag Check (HCE1), Hash and CE with Tag Check (HCE2), and Randomized Convergent Encryption (RCE)이다. CE와 HCE1은 오로지 데이터의 암호화, 복호화만을 수행한다. HCE2는 태그를 이용한 무결성 검증과 함께 암호화 및 복호화를 수행한다. RCE는 HCE2를 기반으로 하며, 키 생성 과정에서 One Time Pad를 이용한 무작위 키를 생성하여 암호화를 수행하고, 무작위 키를 K로 암호화 하는 과정을 갖는다. 이로인해 RCE는 키 생성 엔트로피가 향상되는 장점을 갖는다. 본 연구에서는 RCE를 이용하여 프로토콜 설계를 하였다.

III. 보안 요구사항

본 장에서는 보다 안전하고 효율적인 연산 수행을 위해 다음과 같은 보안요소가 요구된다.

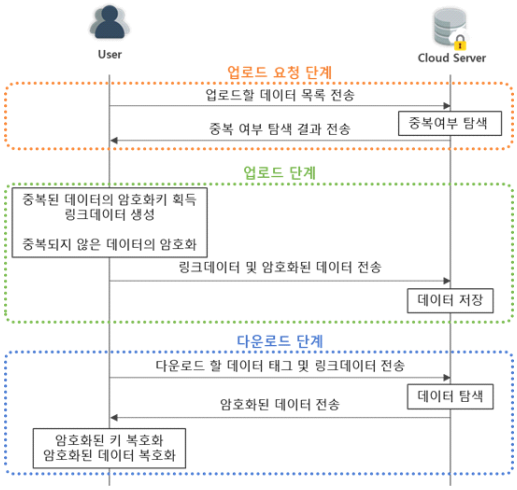
- 기밀성(Confidentiality) : 클라우드 서버에 업로드 된 데이터는 정당한 소유자 이외에는 원본을 확인할 수 없도록 해야 한다. 또한 사전 공격을 통한 키 추측공격에서 안전할 수 있어야 한다.

- 무결성(Integrity) : 클라우드 서버에 업로드 된 데이터는 내용이 변경되어서는 안되며, 만약 내용이 변경되었을 경우, 사용자는 데이터의 변경을 확인할 수 있어야 한다.

- 효율성(Efficiency) : 중복 제거 기술의 스토리지 공간 효율성을 보장하면서 중복 제거 처리 과정에서의 연산량 및 통신량의 효율성 및 안전성을 제공해야한다.

IV. 제안방식

본 논문의 제안방식 전체 형태는 (그림 1)과 같다. 본 제안방식은 MLE의 RCE 방식을 기반으로 하여 순서대로 연결되는 각 블록의 연계



(그림 1) 제안방식의 전체 흐름도

를 이용하여 파일 f에 속한 전체 블록의 소유권을 검증하는 방식을 이용한다.

4.1 시스템 계수

본 제안방식의 시스템 계수는 다음과 같다.

- $*$: 참여 객체 (u : 사용자, M : 메타데이터 서버 SS : 스토리지 서버)
- i : 전체 블록의 인덱스
- j : 중복되지 않은 블록의 인덱스($j \in i$)
- k : 중복된 블록의 인덱스($k \in i$)
- f : 원본 파일
- b_i : 원본 파일의 블록($f = b_1, b_2, \ldots, b_i$)
- sk : 사용자의 비밀키
- K_i : CE를 통해 생성된 키 암호화 키
- L_i : b_i의 암호화 키
- T_i : b_i의 태그
- C_{1_i} : 암호화된 b_i
- C_{2_i} : 암호화된 키 L_i
- O_j : L_{j-1}와 L_j의 사이의 연결 데이터
- Rq_{nD} : 중복되지 않은 블록의 목록
- Rq_D : 중복된 데이터의 소유권을 요청하는 데이터

4.2 데이터 업로드 요청 단계

데이터 업로드 요청 단계에서는 업로드 할 블록의 암호화 키, 태그를 생성하고, 이를 이용하여 클라우드 서버로 중복 여부의 확인을 요청하는 과정을 포함한다.

Step 1. 사용자는 업로드 할 파일의 블록을 이용하여 $K_i \Leftarrow H(b_i)$와 $T_i \Leftarrow H(K_i)$를 각각 생성한다.

Step 2. 사용자는 생성된 데이터 블록의 태그 T_i의 리스트를 클라우드 서버로 전송한다.

Step 3. 클라우드 서버는 사용자가 전송한 태그 목록 중 클라우드 스토리지에 존재하지 않는 데이터 블록의 태그 T_j를 Rq_{nD}에 추가한다. 그리고 클라우드 서버에 존재하는 데이터 블록의 암호화된 키 데이터 C_{2_k}를 Rq_D에 추가한다.

Step 4. 클라우드 서버는 사용자에게 Rq_{nD}와 Rq_D를 전송한다.

4.3 데이터 업로드 단계

데이터 업로드 단계에서는 클라우드 서버가 전송한 Rq_{nD}, Rq_D에 포함된 목록을 확인하여 이미 보관된 데이터의 암호화 키를 획득한다. 그리고 신규 업로드를 수행하는 데이터의 링크데이터와 암호화된 데이터를 생성하여 클라우드 서버로 전송하는 과정을 포함한다.

Step 1. 사용자는 이전 단계에서 클라우드 서버로부터 전송 받은 Rq_D를 확인하여 중복된 데이터의 C_{2_k}를 획득한다. 그리고 $L_k \Leftarrow C_{2_k} \oplus K_k'$를 통해 암호화 키 L_k를 획득한다.

Step 2. 사용자는 Rq_{nD}에 포함된 태그 T_j의 암호화 키 L_j를 무작위로 선택한다. 사용자는

전체 데이터 블록 중 클라우드 서버에 존재하지 않는 데이터의 암호화 키와 이전, 이후 데이터 블록의 암호화 키를 이용하여 $O_{j_1} \Leftarrow L_{j-1} \oplus L_j$와 $O_{j_2} \Leftarrow L_j \oplus L_{j+1}$을 계산한다.

Step 3. 사용자는 $C_{1_j} \Leftarrow E_{L_j}(b_j)$를 통해 암호문 C_{1_j}를 획득한다. 그리고 $C_{2_j} \Leftarrow L_j \oplus K_j$ 를 통해 암호화된 키 C_{2_j}를 획득한다.

Step 4. 사용자는 생성된 O_{j_1}, O_{j_2}의 리스트, C_{1_j}, C_{2_j}의 리스트를 클라우드 서버로 전송한다.

Step 5. 클라우드 서버는 C_{1_j}를 스토리지 서버에 저장하고, 메타데이터 서버에 $C_{1_{j-1}} \rightarrow C_{1_j} \rightarrow C_{1_{j+1}}$을 연결하는 데이터 O_{j_1}, O_{j_2}와 암호화된 키 데이터 C_{2_j}를 저장하여 업로드 단계를 마친다.

4.4 데이터 다운로드 단계

데이터 다운로드 단계에서는 사용자가 클라우드 서버에서 데이터를 다운로드 하는 과정을 나타낸다. 이 과정에서 사용자는 클라우드 서버에 링크 데이터를 전송하여 파일 f를 구성하는 전체 데이터 블록의 소유권을 확인하는 과정을 포함한다.

Step 1. 사용자는 다운로드할 파일 f를 구성하는 데이터 블록의 링크 데이터 O_i의 전체 목록을 T_1과 함께 서버로 전송한다.

Step 2. 클라우드 서버는 사용자가 전송한 T_1과 O_i의 목록을 이용하여 $\{C_{1_i}\}$와 $\{C_{2_i}\}$를 탐색하고, 사용자에게 전송한다.

Step 3. 사용자는 $L_i \Leftarrow C_{2_i} \oplus K_i$를 연산하여 암호화 키 L_i를 획득하고, $b_i \Leftarrow D_{L_i}(C_{1_j})$를 통해 데이터 블록 $\{b_i\}$를 생성한다. 그리고 $b_1 \| b_2 \| \ldots \| b_i$를 통해 원본 파일 f를 생성한다.

V. 제안방식 분석

• 기밀성(Confidentiality) : 데이터의 소유권이 없는 사용자는 클라우드 스토리지의 해당 데이터에 접근할 수 없다. 업로드 되는 모든 데이터는 암호화되어 기밀성이 보장된다. 업로드 된 데이터의 다운로드를 위해서는 전체 데이터 블록의 링크 데이터를 소유하고 있어야만 한다.

• 무결성(Integrity) : 사용자는 다운로드한 데이터와 데이터 태그를 비교하여 데이터의 변형을 확인할 수 있다.

• 효율성(effectiveness) : Client-Side Deduplication 방식을 이용하여 전체 전송되는 데이터의 양을 줄일 수 있다. 또한 블록 단위 중복제거 방식을 사용하여 파일 단위 중복제거 방식보다 중복제거 효율을 높일 수 있다.

VI. 결론

본 제안방식은 블록 단위 중복제거 방식에서 각 데이터 블록마다 별개의 소유권을 가지고 있기 때문에 발생할 수 있는 사전공격을 해결하기 위한 방법을 제안한다. 이 방법은 개별 데이터 블록의 소유권을 가지고 있더라도 파일 f를 구성하는 전체 블록의 링크데이터를 소유하고 있지 않으면 원본파일 f를 획득할 수 없기 때문에 사전공격을 방어할 수 있다.

[참고문헌]

[1] 김건우, 주영현, 엄영익, “클라우드 스토리지 데이터 중복제거 기술 동향”, 한국통신학회 2012 년도 동계종합학술발표회, 2012.

[2] Storer, Mark W., et al. "Secure data deduplication." Proceedings of the 4th ACM international workshop on Storage security and survivability. ACM, 2008.

[3] M. Bellare, et al. "DupLESS: server-aided encryption for deduplicated storage." Presented as part of the 22nd USENIX Security Symposium, 2013.

SPN 구조를 갖는 Block Cipher에서 Key Schedule이 차분 성질에 미치는 영향*

김현우*, 김성겸*, 강형철*, 홍득조**, 홍석희*

*고려대학교 정보보호대학원, **전북대학교 IT 정보공학과

The Effect of Key Schedule on the Differential Properties of Block Cipher with SPN Structure

HyunWoo Kim*, SeongGyeom Kim*, HyungChul Kang*, Deukjo Hong**, Seokhie Hong*

*Graduate School of Information Security, Korea University.

**Department of Information Technology, Chonbuk National University.

요 약

본 논문에서는 키 스케줄을 복잡도에 따라 세 개의 종류로 구성하여 각각의 키 스케줄에서 생성되는 라운드 키를 사용하여 10비트 SPN구조 암호의 차분 특성을 분석한 결과를 제시한다. 암호의 안전성에 대하여 기존의 연구는 반복 암호(iterated cipher)에서 키 스케줄이 미치는 영향을 분석하기 위해 8비트, 10비트, 12비트로 간단한 Feistel 암호를 구현하여 전수조사를 실시하였는데, SPN구조 암호에서도 동일한 결과가 나오는 지를 실험적으로 확인함으로써 차분 공격에 대한 암호의 저항성이 SPN구조를 가지는 암호에서도 키 스케줄의 영향을 받는다는 사실을 검증했다.

I. 서론

Lars R. Knudsen과 John E. Mathiassen[1]이 2004년에 동일한 라운드 함수를 반복적으로 사용하는 반복암호(iterated cipher)의 차분 분석(differential cryptanalysis)[2] 및 선형 분석(linear cryptanalysis)[3]에 대한 저항성에 대해 키 스케줄의 역할을 실험적으로 제시했다. 이러한 공격 이론에서 항상 라운드 키가 독립이라는 가정이 사용되는데, 실제로 라운드 키는 대부분 키 스케줄을 통해 하나의 마스터 키에서 외부에 다른 입력 값이 없이 확장된다. 독립적인 라운드 키를 가정함으로써 차분특성 및 선형근사식의 확률은 Markov chain에 의해 모델링 될 수 있으며 대부분의 반복암호에서 이들의 확률은 몇 번의 라운드를 거친 후 균일분포(uniform distribution)로 수렴된다는 것이 잘 알려져 있다[4]. [1]에서는 복잡한 키 스케줄을 가진 암호가 보다 단순한 키 스케줄을 가진 암호보다 차분 및 선형공격에 대한 저항성이 더 높다는 것을 실험을 통해 보여주며 잘 설계된 키 스케줄을 가진 암호가 더 적은 라운드로 균일 한 분포에 도달한다는 것을 보여준다. 실험은 8, 10, 12비트로 이루어진 간단한 Feistel 암호에 대해서 진행되었으며 결과는 키 스케줄에 따라 “Best Difference”, “Probability”, “Standard Deviation” 값들이 테이블로 제시되어 있다. 제시된 테이블에 의하면 Feistel 암호는 키 스케줄의 강도에 따라 차분 및 선형 공격에 대한 저항성이 올라간다. 본 논문에서는 간단한 SPN구조 암호를 설계하여 같은 실험을 진행하며, 제시된 결과가 SPN구조를 가지는 암호에 대해서도 동일하게 적용되는지를 확인함으로써 주어진 결과가 블록 암호의 구조와 상관없이 동일하게 나타나는지 검증하

* 이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2017-0-00520, (ICT 기초연구실) SCR-Friendly 대칭키 암호 및 응용모드 개발)

며, 이전 결과에 대한 한계점과 개선방향에 대해 논한다.

II. 실험

본 섹션에서는 실험을 위한 SPN구조 암호의 구조와 사용된 세 개의 키 스케줄에 대해 설명하고, 진행 된 실험 방법에 대해 설명한다.

Ⅱ.1 SPN구조 암호 및 키 스케줄

SPN구조 암호는 '혼돈(confusion)'의 효과를 주는 S-layer와 '확산(diffusion)' 효과를 주는 P-layer로 구성된 블록 암호이다. 각 라운드 키는 S-layer를 통과하기 전에 XOR연산 된다. 본 논문에서 사용할 암호는 10-bit 블록 암호로서, 두 개의 5-bit S-box를 S-layer로 하며 P-layer는 간단한 비트 치환(bit permutation)으로 구성하였다. 한 라운드 암호화는 다음과 같이 진행되며 모양은 그림 1과 같다.

$$C_{i+1} = P(S(C_i \oplus K_i)), i = 1,2,3,...,r$$

이때, $C_1 = P$로 정의한다. 사용된 S-BOX는 다음과 같으며, 각 S-BOX는 bijection을 만족하도록 임의로 생성하였다.

$$S_1 = \begin{Bmatrix} 14,30,8,9,27,28,0,26,12,4, \\ 6,18,31,22,19,1,16,17,29,24, \\ 15,3,7,2,20,13,25,5,23,21,11,10 \end{Bmatrix}$$

$$S_2 = \begin{Bmatrix} 24,15,18,0,7,16,9,22,23,11, \\ 1,5,4,20,12,26,14,28,21,8,29, \\ 10,27,13,25,2,19,31,17,6,3,30 \end{Bmatrix}$$

다음으로 사용된 키 스케줄에 대해 설명한다. 본 논문에서 사용된 키 스케줄은 키 스케줄의 강도에 따라 세 개로 구성되었으며 간단한 키 스케줄부터 모든 라운드 키가 독립적으로 생성되는 키 스케줄로 구성되어 있다. i 라운드의 라운드 키는 10-bit 이며 K_i로 표시한다.

Key schedule 1:

Input: $K = K^L \mid K^R$

For $i = 1$ **to** $r/2$ **do**

$$K_{2i-1} = K^L \mid K^L$$
$$K_{2i} = K^R \mid K^R$$

For $i = 1$ **to** r **do** $K_i = K_i$ **XOR** i

첫 번째 키 스케줄은 각 라운드 키가 마스터 키의 최상위 5-bit와 최하위 5-bit를 번갈아 가면서 사용하는 단순한 구조로 이루어져 있으며, 각 라운드 키는 라운드 상수 i를 XOR 하여 사용한다. 두 번째 키 스케줄은 첫 번째 키 스케줄보다 조금 복잡한 구조를 가진다.

Key schedule 2:

Input: $K = K^L \mid K^R$

$$K_1 = K^L \mid K^R$$
$$K_2 = K^R \mid K^L$$
$$K_3 = LS(K^L,2) + RS(K^R,3) \mid LS(K^R,2) + RS(K^L,3)$$
$$K_4 = LS(K^R,2) + RS(K^L,3) \mid LS(K^L,2) + RS(K^R,3)$$

For $i = 5$ **to** r **do** $K_i =$**Rotate**$(K_{i-3},1)$

For $i = 1$ **to** r **do** $K_i = K_i$ **XOR** i

LS는 LeftShift, RS는 RightShift를 의미하며 Rotate는 왼쪽으로 1-bit rotation 하는 연산을 의미한다. 마지막 세 번째 키 스케줄은 암호학적으로도 안전강도를 보장받을 수 있는 난수발생기(DRBG)를 이용하여 필요한 라운드 수만큼 라운드 키를 랜덤하게 생성하는 키 스케줄이다. 키 스케줄에 대한 라운드 키 간의 상관계수 절대값의 평균(5라운드 일 때, 20가지 키쌍 조합의 평균)은 다음 표와 같다. 상관계수는 -1과 1 사이의 값으로, 라운드 키 간의 독립성의 정도를 수치로 보여주며, 서로간의 선형 의존성이 적을수록 0에 가깝다. 표 1의 값을 통해 키 스케줄 1~3 의 독립성 정도를 알 수 있다.

키 스케줄	각 키쌍의 상관계수 절대값 평균
KS1	0.4
KS2	0.1313
KS3	0.0211

표 1. 키 스케줄 별 상관계수

Ⅱ.2 실험방법

각각의 키 스케줄 마다 가능한 모든 마스터 키와 입력차분, 출력차분을 모두 조사하여 실제 입출력 차분쌍이 나올 확률을 구한다. 예를 들어, 첫 번째 키 스케줄에 대한 실험을 진행할 때는 키 스케줄을 1번으로 고정한 후, 모든 가능한 키 값

0~1023에 대해 입력 차분을 고정한 후, 모든 평문 값 0~1023에 대해 P, P'을 암호화 하여, 암호문 C, C'의 차분을 구해서 기록한다. 즉, 각각의 키 마다 입력차분 0을 제외한 $2^{10} \times (2^{10}-1)$번의 탐색을 진행한다. 따라서 각각의 키 스케줄마다 총 $2^{10} \times 2^{10} \times (2^{10}-1)$번의 탐색이 진행되며, 세 번째 키 스케줄은 마스터 키 로부터 확장되는 스케줄이 아니기 때문에, 키가 변하는 시점마다 새로운 임의의 키를 생성하여 실험을 진행한다.

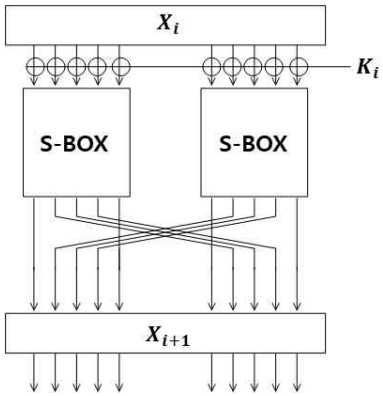


그림 1. 실험에 사용한 라운드 함수 구조

III. 결과 및 분석

Ⅲ.1 결과

진행 된 10비트 SPN구조 암호에 대한 실험결과는 다음 표 2와 같다.

R	KS	Best Difference	Prob.	Std. dev. best	Std. dev.
2	1	19 -> d1	64.00	0.0000	0.8842
2	2	19 -> d1	64.00	0.0000	0.8627
2	3	19 -> d1	64.00	-	0.8469
3	1	d -> 180	18.37	0.4170	0.5366
3	2	2a0 -> 4	13.78	0.2977	0.5262
3	3	180 -> 8	13.41	-	0.5246
4	1	d -> 8	5.65	0.1307	0.5013
4	2	d -> 8	4.92	0.1279	0.5011
4	3	d -> 8	5.22	-	0.5003
5	1	2a0 -> 4	2.28	0.0727	0.5000
5	2	220 -> 8	2.33	0.0612	0.4998
5	3	2a0 -> 8	2.25	-	0.4997

표 2. 10비트 SPN구조 암호 실험결과

표의 각 열에 대해서 설명하면, 'R'은 라운드 수를 의미하고 'KS'는 키 스케줄을 'Best Difference'는 최고 확률을 갖는 입력 및 출력차분을 'Prob.'는 최고 확률을 의미한다. 기재된 모든 값은 편의상 계산 과정에서 1024가 곱해진 값으로 표기했다. 'Std. dev. best'와 'Std dev.'는 각각 최고 확률을 갖는 differential에 대한 표준편차와 모든 differential에 대한 표준편차 이다. 최고 확률을 갖는 입력 및 출력차분은 모든 키에 대해서 나올 수 있는 차분 값을 모두 누적시킨 후, 제일 높은 값을 갖는 차분 쌍이다.

Ⅲ.2 분석

Feistel 구조를 분석한 [1]의 결과와 동일하게, 키 스케줄이 복잡할수록 표준편차가 감소하는 것을 확인할 수 있다. 또한, 라운드수가 증가함에 따라 표준 편차는 점점 0으로 가까워지며, 많은 라운드 수를 거치게 되면 균일분포로 수렴할 것이다. 이는 키 스케줄의 강도가 높아짐에 따라 암호가 차분공격에 대한 더 높은 저항성을 갖는다는 것으로 해석할 수 있다. 키 스케줄의 강도가 높아짐에도 확률이 증가하여 차분공격에 대한 키 스케줄의 영향이 다소 모호하게 보이는 부분이 존재하지만 이는 단순한 암호구조와 단순한 키 스케줄 알고리즘에 기인한 것이며, 이러한 경우에도 표준편차 값은 점점 줄어들고 있는 것을 통해 키 스케줄의 강도가 차분공격에 대한 저항성을 높여주고 있음을 주장할 수 있다.

IV. 향후연구

본 논문은 반복 암호에서 키 스케줄의 역할을 SPN구조 암호에 대해 확인해봄으로써 [1]에서 제시한 결론이 Feistel 뿐만 아니라 SPN구조를 가지는 블록암호에서도 적용된다는 사실을 검증하였다. 작은 크기의 암호의 전수조사를 통한 실험은 분명한 한계점이 존재한다. 그것은 실제 사용되는 암호들은 블록 사이즈가 크기 때문에 본 논문과 같은 전수조사를 하는 것이 불가능 하다는 것이다. 즉, 본 논문에서 진행한 실험에서 쓰이는 작은 암호에서 도출되는 결과

를 실제 사용되고 있는 암호로 가지고 올 수 있을지에 대한 검증이 필요하다. [1]에서는 이러한 결과가 '크기가 큰 암호에서 적용되지 않아야 하는 이유를 발견하지 못했다'고 말하며 큰 크기의 암호에서도 동일하게 적용될 것이라고 예측하고 있다.

Ⅳ.1 통계적 추정

실제 사용되고 있는 암호들, 즉 큰 크기를 가지는 암호들에 대해서 키 스케줄이 암호의 차분 특성에 미치는 영향을 알아보기 위해서는 전수조사가 불가피하다. 그러나 통계적인 방법을 이용한다면 그 결과를 근사적으로 나마 추정할 수 있다. 실제로 전수조사를 통해 얻어진 결과 값이 아니기 때문에 추정에 대한 신뢰도를 증명해야 하는 문제가 있으나, 통계적 추정을 통하여 해당 결과를 얻는 것은 가능하다. HIGHT[5]의 경우 64비트 평문과 128비트 키를 사용하는데, 이러한 경우에 전수조사는 2^{256}개의 입력을 대상으로 진행되어야 하므로 현실적으로 불가능하다. 그러나 탐색공간에서 독립적으로 표본을 추출하고 추출한 표본을 통해 얻을 결과로 전체의 특징을 추정할 수 있다. 이러한 추정을 통해 실제 사용되는 암호에 대해서도 위와 같은 표를 작성하는 것이 가능해지며, 키 스케줄과 해당 암호에 대한 차분 특성을 분석할 수 있다. 해당 실험은 표본을 추출하는 기준, 표본을 통한 정확한 추정, 사용된 통계적 추정 방식의 신뢰성 등, 여러 가지 고려사항이 있기 때문에 현재 진행 중에 있다.

Ⅴ. 결론

본 논문에서는 SPN구조 암호가 키 스케줄의 강도에 따라 차분 공격에 대한 저항성이 높아지는 것을 실험적으로 보였다. 전수조사가 가능한 비교적 간단한 장난감 암호(toy cipher)에서 실험이 진행되었으나, 실제 환경에서 사용되는 큰 암호에 대한 결과도 통계적으로 추정할 계획이다.

[참고문헌]

[1] Knudsen, Lars R., and John E. Mathiassen. "On the role of key schedules in attacks on iterated ciphers." European Symposium on Research in Computer Security. Springer, Berlin, Heidelberg, 2004.

[2] Biham, Eli, and Adi Shamir. "Differential cryptanalysis of DES-like cryptosystems." Advances in Cryptology-CRYPTO. Vol. 90. 1991.

[3] Matsui, Mitsuru. "Linear cryptanalysis method for DES cipher." Workshop on the Theory and Application of of Cryptographic Techniques. Springer, Berlin, Heidelberg, 1993.

[4] Lai, Xuejia, James Massey, and Sean Murphy. "Markov ciphers and differential cryptanalysis." Advances in Cryptology—EUROCRYPT'91. Springer Berlin/Heidelberg, 1991.

[5] Hong, Deukjo, et al. "HIGHT: A new block cipher suitable for low-resource device." CHES. Vol. 4249. 2006.

2017 한국정보보호학회 동계학술대회

E5 암호이론(II)

안 효 범 (공주대)

- **내적연산을 지원하는 함수암호 연구동향**
  김훈기, 이영경, 이동훈 (고려대학교)
- **RTOS가 동작하는 Cortex-M4보드의 암호알고리즘 소비 전력 측정**
  송행권, 이옥연 (국민대학교)
- **표준 모델에서 증명 가능한 LWE 기반 BGN-타입 ID기반 동형 암호**
  우주, 김종현, 이동훈 (고려대학교)
- **동적 암호 기반 Card OTP 모듈을 사용한 복제방지 RFID Card 제안**
  권성민, 박기현, 김재성, 원요한, 이동현, 이예지, 지한별,
  박의성 (KITRI BoB), 이종호 (고려대학교)

내적연산을 지원하는 함수암호 연구동향*

김훈기[1†], 이영경[1], 이동훈[1‡]

[1]고려대학교 정보보호대학원

A Study on Functional Encryption for Inner Products

Hunki Kim[1†], Youngkyung Lee[1], Dong Hoon Lee[1‡]

[1]Graduate School of Information Security, Korea University.

요 약

함수암호는 암호문을 복호화한 결과가 키가 갖고 있는 특정 값과 암호문에 해당하는 평문의 연산으로 출력되는 암호 기법이다. 함수암호 설계에 대한 연구가 이루지고 있지만 현재까지 설계된 모든 연산을 지원하는 함수암호는 실용적으로 사용되기에는 무리가 있었다. 2015년 Abdalla등이 하나의 암호문에 대한 내적연산만을 지원하는 실용적인 함수암호 기법을 설계하였다. 이후, 여러 개의 암호문에 대한 내적연산을 지원하는 함수암호 기법들이 제시되며 비교적 실용적인 내적연산을 지원하는 함수암호 연구가 지속적으로 이루어지고 있다. 본 논문에서는 내적연산을 지원하는 함수암호 기법을 비교·분석하고, 앞으로의 연구방향을 제시한다.

I. 서론

일반적인 현대 암호 시스템은 암호화하기 전 평문과 복호화 한 후의 평문이 같은 형태를 갖는 시스템이다. 이와 달리 함수 암호(Function Encryption, FE) 시스템은 암호문을 복호화한 결과가 키가 갖고 있는 특정값과 암호문에 해당하는 평문의 연산으로 출력되는 시스템이다. 현재 모든 연산을 지원하는 FE에 대한 이론적 설계는 이루어 졌지만 실용적으로 사용하기에는 무리가 있다.

2011년 Boneh등은 암호화 된 메시지에 대한 정보가 복호화된 결과에 노출이 되면 안 되는 경우를 예를 들면서 처음으로 functionality에 대하여 정의[3]하였다. 이후 내적연산의 값이 조건을 만족할 경우 복호화를 해주는 Inner Product Predicate[4]와는 달리 2015년 Abdalla등에 의해 내적연산의 결과를 복호화 해주는 연구 결과가 제시[1]되었다. 겹선형 함수를 사용하여 연산이 효율적이라는 장점이 있지만 하나의 암호문에만 연산을 지원하기 때문에 응용이 제한되는 단점이 존재하였다. 이후 여러 개의 암호문에 대한 연산 결과가 암호문 형태로 복호화 되는 완전동형암호(Fully Homomorphism Encryption, FHE)[5]처럼 여러 개의 암호문에 대한 연산 결과가 복호화 되는 FE에 대한 연구가 이루어진다. 2016년 Lee등에 의해 두 개의 암호문으로 내적연산을 지원되는 함수암호(Two Input Functional Encryption for Inner Product, TI-FE for IP)가 제안[6]되었다. 암호문이 두 개로 늘었지만 응용할 수 있는 범위가 크게 증가하지 않았다는 단점이 존재하였다. 2017년 Raykova등은 n개의 암호문으로 내적연산을 지원하는 함수암호(Multi Input Fuctional

* 이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2017-0-01863, 함수암호 기법 설계·분석 및 구현기술 연구)

† 주저자, hunki3882@korea.ac.kr

‡ 교신저자, donghlee@korea.ac.kr

Encryption for Inner Product, MIFE for IP)를 제안[2]하였다. 암호문의 수가 늘어 여러 방면으로의 응용이 가능하지만 설계가 복잡하며 많은 연산을 필요로 한다는 단점이 존재한다.

이 논문에서는 겹선형 함수를 사용하여 내적연산을 지원하는 함수암호들의 연구 동향을 살펴볼 것이며 본 논문의 구성은 다음과 같다. Ⅱ장에서는 기법의 이해에 필요한 배경지식을 설명한다. Ⅲ장에서는 현재 제안된 내적연산을 지원하는 함수암호의 연구동향에 대해 살펴본다. 마지막으로 Ⅳ에서 결론을 맺는다.

II. 배경지식

본 장에서는 내적연산을 지원하는 함수암호의 동향을 살펴보기에 앞서 필요한 배경지식을 서술한다.

2.1 Functionality definition

K를 키공간, X를 메시지 공간, Σ를 결과값 공간, ⊥를 Σ에 포함되지 않은 특수한 문자들의 공간이라 할 때 (K, X)상에서 정의된 functionality F는 $K \times X \rightarrow \Sigma \cup \{\perp\}$인 함수로 정의된다. 즉, 키가 K에 없거나, 메시지가 X에 없다면 functionality는 정의될 수 없다는 것을 의미한다.

2.2 Functional Encryption definition

functionality F에 대한 함수암호의 설계는 4가지의 알고리즘으로 정의 될 수 있다.

- Setup $(1^\lambda) \rightarrow (mpk, msk)$
  : 입력받은 보안상수 λ에 대한 공개키와 마스터 비밀키를 출력한다.
- KeyDer $(msk, k) \rightarrow sk_k$
  : msk와 키 $k \in K$를 입력하여 비밀키 sk_k을 출력한다.
- Encrypt $(mpk, x) \rightarrow$ Ct
  : mpk와 암호화 할 메시지 $x \in X$를 입력받고 암호문 Ct를 출력한다.
- Decrypt (mpk, Ct, sk_k)
  : mpk와 암호문 Ct , 비밀키 sk_k를 입력받고 복호화하여 k와 m을 연산한 결과 $F(k,m)$을 출력한다.

III.내적연산을 지원하는 함수암호 연구 동향

본 장에서는 겹선형 함수를 사용하여 내적연산을 지원하는 함수암호의 기법에 대해 살펴본다.

3.1 Abdalla등의 연구[1]

2015년 Abdalla등은 하나의 암호문에 대한 내적연산을 지원하는 간단한 함수암호 기법을 4개의 알고리즘으로 제안하였다.

- Setup $(1^\lambda, 1^\ell) \rightarrow (mpk, msk)$
  : 1^λ와 1^ℓ을 입력받아 **GroupGen** 알고리즘을 통해 $(\mathbb{G}, p, g)$를 출력 값으로 갖고, $\mathbb{Z}_p^\ell$에서 $\mathbf{s} = (s_1, \cdots, s_\ell)$을 뽑아 $mpk = (h_i = g^{s_i})_{i \in [\ell]}$, $msk = \mathbf{s}$ 로 설정한다.
- Encrypt $(mpk, \mathbf{x}) \rightarrow \text{Ct} = (ct_0, (ct_i)_{i \in [\ell]})$
  : mpk와 암호화 할 메시지 $\mathbf{x} = (x_1, \cdots, x_\ell)$을 입력받고 $\mathbb{Z}_p$에서 임의의 r을 뽑아 $ct_0 = g^r$과 $i \in [\ell]$에 대한 $ct_i = h_i^r \cdot g^{x_i}$을 계산하여 암호문 $\text{Ct} = (ct_0, (ct_i)_{i \in [\ell]})$을 출력한다.
- KeyDer $(msk, \mathbf{y}) \rightarrow sk_y = \langle \mathbf{y}, \mathbf{s} \rangle$
  : msk와 벡터 $\mathbf{y} = (y_1, \cdots, y_\ell) \in \mathbb{Z}_p^\ell$을 입력받아 비밀키 $sk_y = \langle \mathbf{y}, \mathbf{s} \rangle$를 출력한다.
- Decrypt $(mpk, \text{Ct}, sk_y) \rightarrow \prod_{i \in [l]} ct_i^{y_i} / ct_0^{sk_y}$
  : mpk와 암호문 $\text{Ct} = (ct_0, (ct_i)_{i \in [\ell]})$, 벡터 $\mathbf{y}$에 대한 비밀키 sk_y을 입력받고 $\prod_{i \in [l]} ct_i^{y_i} / ct_0^{sk_y}$을 출력 후 g에서 이산로그 계산을 하여 $\langle \mathbf{x}, \mathbf{y} \rangle$를 결과로 갖는다.

복호화시 ct_i에 l번의 지수연산과 ct_0에 한 번의 지수연산이 되어 총 $l+1$번의 지수연산이 이루어진다. 하나의 암호문에 대한 복호화의 결과가 내적연산 값이기 때문에 가중평균을 구할 때 여러 응용이 가능하다. 예를 들어 학생들의 점수를 입력으로 받지만 복호화 된 결과로 학생들의 성적만 알 수 있게 된다. 하지만 cloud환경과 같이 여러 사용자의 입력이 필요한 경우에서 응용이 제한된다.

3.2 Lee등의 연구[6]

2016년 Lee등은 두 개의 암호문에 대한 내적 연산을 지원하는 함수암호 기법을 4개의 알고리즘으로 제안하였다.

- Setup $(1^\lambda, 1^\ell) \rightarrow EK_1, EK_2, PP, MK$
  : 1^λ을 입력받아 그룹의 위수들의 조합 $N = p_1 p_2 p_3$에서 GroupGen 알고리즘을 통해 (p_1, p_2, p_3, G, G_T)를 출력 값으로 갖는다. g_1을 부분군 G_{p1}의 임의의 생성자라 하고, 임의의 값 $\{\omega_{1,i}, \omega_{2,i}\}_{i\in[\ell]} \in \mathbb{Z}_N$, $Q, \{Q_i\}_{i\in[\ell]} \in G_{p_2}$와 $R, \{R_i\}_{i\in[\ell]} \in G_{p_3}$을 뽑는다. 마스터키를 $MK = \{\omega_{1,i}, \omega_{2,i}\}_{i\in[\ell]}$로, 두 개의 암호화키를 $EK_1 = (\{W_{1,i} = g_1^{\omega_{1,i}} Q_i\}_{i\in[\ell]})$, $EK_2 = (\{W_{2,i} = g_1^{\omega_{2,i}} R_i\}_{i\in[\ell]})$로, 공개 상수를 $PP = ((N, G, G_T, e), g = g_1, Q, R)$로 설정한다.

- GenKey $(\vec{y}, MK, PP) \rightarrow SK_{\vec{y}}$
  : $\overrightarrow{y_j} = (y_{j,1}, \cdots, y_{j,\ell})$인 벡터 $\vec{y} = (\overrightarrow{y_1}, \overrightarrow{y_2})$와 MK, PP를 입력하여 개인키 $SK_{\vec{y}} = (K = \sum_{j=1}^{2}\sum_{i=1}^{\ell} \omega_{j,i} \cdot y_{j,i}) \in \mathbb{Z}_N$ 을 출력한다.

- Encrypt $(j, \overrightarrow{x_j}, EK_j, PP) \rightarrow CT_1 / CT_2$
  : $j \in \{1,2\}$와 벡터 $\overrightarrow{x_j} = (x_{j,1}, \cdots, x_{j,\ell})$와 암호화키 $EK_j = (\{W_{j,i}\}_{i\in[\ell]})$ 그리고 PP를 입력한다. 임의로 $t_j \in \mathbb{Z}_N$, $Q_1, \{Q_{2,i}\}_{i\in[\ell]} \in G_{p_2}$, $R_1, \{R_{2,i}\}_{i\in[\ell]} \in G_{p_3}$을 뽑아 만약 $j = 1$일 경우 $CT_1 = (C_1 = g^{t_1} Q_1, \{D_{1,i} = (g^{x_{1,i}} W_{1,i})^{t_1} Q_{2,i}\}_{i\in[\ell]} \in G_{p_1 p_2}^{n+1}$의 암호문을 출력하고 $j = 2$일 경우는 $CT_2 = (E_2 = g^{t_2} R_1, \{F_{1,i} = (g^{x_{2,i}} W_{2,i})^{t_2} R_{2,i}\}_{i\in[\ell]} \in G_{p_1 p_3}^{n+1}$의 암호문을 출력한다.

- Decrypt $(CT_1, CT_2, SK_{\vec{y}}, PP) \rightarrow g_t^{\sum_{i=1}^{\ell} < X_i, Y_i >}$
  : 벡터 $\overrightarrow{x_1} = (x_{1,1}, \cdots, x_{1,\ell})$에 대한 암호문 $CT_1 = (C_1, \{D_{1,i}\}_{i\in[\ell]})$, 벡터 $\overrightarrow{x_2} = (x_{2,1}, \cdots, x_{2,\ell})$에 대한 암호문 $CT_2 = (E_1, \{F_{1,i}\}_{i\in[\ell]})$, 벡터 $\vec{y} = (\overrightarrow{y_1}, \overrightarrow{y_2})$에 대한 비밀키 $SK_{\vec{y}} = K_1$ 그리고 공개사수 PP를 입력한다. 입력값에 대해 $A = e(C_1, E_1)$, $B = \prod_{i=1}^{\ell} e(E_1, D_{1,i})^{y_{1,i}} \cdot \prod_{i=1}^{\ell} e(C_1, F_{1,i})^{y_{2,i}} \cdot A^{-K}$의 페어링연산 후 이산로그 계산을 하여 내적연산 결과 $< \vec{x}, \vec{y} > = \sum_{j=1}^{2}\sum_{i=1}^{\ell} x_{j,i} \cdot y_{j,i}$를 출력한다.

복호화시 B에서 페어링연산에 대한 지수연산이 2ℓ번, A에 대한 지수연산이 한 번 이루어져 총 $2\ell+1$번의 지수연산이 이루어지며, A의 페어링 연산을 포함해 총 $2\ell+1$번의 페어링 연산이 이루어진다. 기존 기법과 달리 복호화 결과가 내적연산의 합으로 나온다. 하나의 내적연산 값이 노출되는 것을 방지하기 위해 개인키 설정으로 암호화를 해주었으며 복호화시 페어링 연산을 사용해 연산횟수가 늘어났음을 알 수 있다.

3.3 Raykova등의 연구[2]

2017년 Raykova등은 n개의 암호문에 대한 내적연산을 지원하는 함수기법을 4개의 알고리즘으로 제안하였다.

- Setup $(1^\lambda, 1^\ell) \rightarrow (msk)$
  : 1^λ와 1^ℓ을 입력받아 GroupGen 알고리즘을 통해 $(\mathbb{G}_1, p_1, g_1)$, $(\mathbb{G}_2, p_2, g_2)$, $(\mathbb{G}_t, p_t, g_t)$를 출력 값으로 갖고, $\mathbb{Z}_p^\ell$에서 $S_i = (s_{i,j})_{i\in[n], j\in[\ell]}$을 뽑고 $\mathbb{Z}_p$에서 임의의 $(v_i)_{i\in[n]}$과 $(z_i)_{i\in[n]}$ 을 뽑아 $msk = (g_1^{S_i}, g_1^{v_i}, g_1^{z_i})_{i\in[n]}$로 설정한다.

- Encrypt $(msk, (X_i)_{i\in[n]}) \rightarrow (ct_{x_i})_{i\in[n]}$
  : msk와 암호화 할 메시지 $X_i = (x_{i,j})_{i\in[n], j\in[\ell]}$을 입력받고 $\mathbb{Z}_p$에서 임의의 $(r_i)_{i\in[n]}$을 뽑아 $i \in [n]$에 대한 암호문 $ct_{x_i} = (g_1^{r_i}, g_1^{X_i + S_i r_i}, g_1^{z_i + v_i r_i})$을 출력한다.

- KeyDer $(msk, (Y_i)_{i\in[n]}) \rightarrow (sk_{y_i})_{i\in[n]}$
  : msk와 벡터 $Y_i = (y_{i,j})_{i\in[n], j\in[\ell]} \in \mathbb{Z}_p^\ell$을 입력받고 $\mathbb{Z}_p$에서 임의의 w값을 뽑아 비밀키 $sk_{y_i} = ((g_2^{< S_i, Y_i > + v_i w})_{i\in[n]}, g_2^w, g_t^{(z_1 + \cdots + z_n)w})$을 출력한다.

- Decrypt $(msk, (ct_{x_i})_{i\in[n]}, (sk_{y_i})_{i\in[n]}) \rightarrow g_t^{\sum_{i=1}^{n} < X_i, Y_i >}$

: msk와 $i \in [n]$에 대한 암호문 ct_{x_i}, 벡터 Y_i에 대한 비밀키 sk_{y_i}을 입력받고

$$g^{< X_i, Y_i > + z_i w} = e(g_1^{(X_i + S_i r_i)^T}, g_2^{Y_i}) \cdot e(g_1^{(z_i + v_i r_i)^T}, g_2^{w}) \cdot e(g_1^{r_i}, g_2^{< S_i, Y_i > + v_i w})^{-1}$$

입력 값에 대한 페어링 연산 한 결과 값을 $g_t^{((z_1 + \cdots + z_n)w)^{-1}} \cdot \prod_{i=1}^{n} g^{< X_i, Y_i > + z_i w}$로 계산하여 암호문을 복호화 한 $g_t^{\sum_{i=1}^{n} < X_i, Y_i >}$을 후 g_t에서 이산로그 계산을 하여 $\sum_{i=1}^{n} < X_i, Y_i >$를 결과로 갖는다.

복호화시 g_1, g_2에 6ℓ번의 지수연산이 n번 이루어 진 후 g_t에 한 번의 지수연산이 되어 총 $6\ell n + 1$번의 지수연산이 이루어진다. 페어링 연산은 3ℓ번씩 n번 이루어져 $3\ell n$번 이루어진다. $< X_i, Y_i >$를 노출하지 않고 $\sum_{i=1}^{n} < X_i, Y_i >$만 결과로 갖기 위해 비밀키 설정을 사용하였다. n개의 입력을 받을 수 있어 기존의 기법들보다 cloud환경과 같은 다양한 분야에서 응용이 이루어 질 수 있다.

IV. 결론

본 논문에서는 내적연산을 지원하는 함수암호의 동향을 살펴보고 효율성과 제공하는 기능에 대해 비교 분석하였다.

참고문헌	입력 개수	설정	페어링	연산횟수	
				지수연산	페어링
[1]	1	공개키	X	$\ell+1$	-
[6]	2	개인키	O	$2\ell+1$	$2\ell+1$
[2]	n	개인키	O	$6\ell n+1$	$3\ell n$

n : 입력 벡터의 수, ℓ : 단위벡터의 길이

Abdalla등의 연구[1]는 입력 개수가 하나인 함수암호를 제시하였는데 여러 개의 입력 개수가 필요한 경우에서의 응용이 제한된다. 공개키 설정에서 페어링 연산 없이 암호화와 복호화가 이루어지기 때문에 간단하게 설계되었으며 복호화시 $\ell+1$번의 지수연산이 이루어진다.

Lee등의 연구는 입력 개수가 하나 늘어 두 개의 입력이 가능한 함수암호를 제시하였다. 기존 연구보다 응용범위가 증가하였지만 개인키 설정에서 페어링 연산을 사용하여 복호화시 $\ell+1$번의 지수연산과 $2\ell+1$번의 페어링 연산이 이루어 진다.

Raykova등의 연구는 여러 개의 입력을 받아 내적연산을 지원하는 함수암호를 제시하였다. 기존 연구와 비교해 n개의 입력이 가능하기 때문에 많은 분야에서 응용이 될 수 있지만 개인키 설정에서 페어링 연산을 사용함으로서 기법 설계가 복잡해지고 페어링 연산을 하며 연산속도가 느려지게 되었다. 복호화시 $6\ell n+1$번의 지수연산과 $3\ell n$번의 페어링 연산이 이루어 진다.

현재까지 나온 연구의 단점을 보완하기 위해 MIFE for IP를 페어링 없이 설계하여 연산속도를 줄이거나, Lattices로 설계하여 양자컴퓨터 환경에서 안전한 기법 또한 의미 있는 연구 주제이다.

[참고문헌]

[1] M. Abdalla, F. Bourse, A.D. Caro, D. Poincheval, "Simple Functional Encryption Schemes for Inner Products," in PKC'15, LNCS 9020, pp. 733-751, 2015.

[2] M. Abdalla, R. Gay, M. Raykova, H. Wee "Multi-Input Inner-Product Functional Encryption from Pairings," in Annual International Conference on the Theory and Applications of Cryptographic Techniques'17, pp. 601-626, 2017.

[3] D. Boneh, A. Sahai, B. Waters, "Functional Encryption : Definitions and Challenges," in TCC'11, LNCS 6597, pp. 253-273, 2011.

[4] J. Katz, A. Sahai, B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in EUROCRYPT'08, LNCS 4965, pp. 146-162, 2008.

[5] C. Gentry, "Fully homomorphic encryption using ideal lattices," in STOC'09, pp. 169-178, 2009.

[6] K. Lee, D.H. Lee, "Two-input functional encryption for inner products from bilinear maps," in IACR'16, pp. 432, 2016.

RTOS가 동작하는 Cortex-M4보드의 암호알고리즘 소비 전력 측정

송행권*, 이옥연*

*국민대학교 금융정보보안학과

Cryptographic algorithm power consumption measurement of Cortex-M4 board with RTOS

Haeng-Gwon Song*, Ok-Yeon Lee*

*Division of information security, Korea University.

요 약

RTOS가 동작하는 Cortex-M4보드에서 블록암호, 해시함수의 전력 측적 결과 값을 제시한다. IoT 환경의 하위 단말에서 주로 사용되는 Cortext-M4보드에서 암호 알고리즘의 소비전력을 측정함으로써 보안을 추가하였을 때 생기는 비용을 계산 할 수 있다. 구현한 암호알고리즘의 소비전력 측정 값을 비교하기 위하여 KISA에서 제공하는 알고리즘과, OpenSSL에서 제공하는 암호알고리즘과의 비교를 수행하였다. 이를 통해 최대 LEA알고리즘에서 약 27%의 전력 사용량의 효율을 확인 할 수 있었다.

I. 서론

다양한 사물들이 인터넷과 연결되어 사람의 개입 없이 서로 정보를 교환하고 공유하는 사물인터넷(Internet of Things, IoT)기술이 빠른 속도로 확산되고 있다. 현재 IoT 기술은 실생활에서 사용되는 스마트 가전, 스마트 디바이스, 스마트그리드 등 다양한 분야에서 폭 넓게 적용되고 있다. 다양한 형태의 디바이스들이 인터넷과 결합하면서 개인의 프라이버시를 침해할 수 있는 개인정보 혹은 비밀정보들이 인터넷 상에서 전송되고 있다. 이러한 데이터들은 해커나 악의적인 소프트웨어, 바이러스 등에 의한 정보 유출, 조작, 변조 되는 보안 위협에 노출될 수 있다.

IoT환경에서 사용되는 데이터의 보안 위협을 막기 위하여 사용하는 것이 암호이다. 데이터의 보안을 제공하기 위하여 지켜져야 할 것은 기밀성, 무결성, 가용성(CIA, Confidentiality · Integrity·Availability)이다. 각각의 요소들은 대칭키, 해시함수, 메시지인증코드, 공개키 암호알고리즘을 이용하여 제공 할 수 있다.

본 논문에서는 저 전력이 요구되는 센서 디바이스 혹은 IoT 하위 디바이스에 사용되는 Cortex-M4 보드에서 동작하는 암호 알고리즘의 전력 값 측정 방법 및 측정 결과 값을 제시한다. 측정한 알고리즘은 기밀성을 제공할 수 있는 ARIA[1], HIGHT[2], LEA[3], SEED[4]등 국내 표준으로 지정된 암호 알고리즘과, 블록암호와 함께 사용하여 기밀성 및 무결성을 제공할 수 있는 CBC, CFB, OFB운영모드[5]와 마지막으로 무결성을 제공 할 수 있는 SHA2[6] 알고리즘을 측정한 결과를 제시한다.

본 논문은 2장에서는 전력 측정 방법 및 전력측정을 위한 간단한 개념정리를 소개하고, 3장에서는 소비전력을 측정하기 위한 실험환경 및 측정 결과 값을 제시하고 4장에서는 결론을 맺는다.

II. 전력 측정 방법

전력이란 전기에너지가 한 일을 의미하며, 즉 부하에서 소모된 에너지를 뜻한다. 디바이스에서 소비되는 에너지(소비 전력량)를 계산하기 위하여 전압과 전류 값을 측정해야 한다.

$$\text{전력}(Watt) = \text{전압}(V) \times \text{전류}(A)$$

먼저 전류를 측정하고자 할 때, 측정하고자 하는 부분에 직렬로 연결하여 전류를 측정 할 수 있다. 전류를 측정 할 때, 전류계의 저항은 아주 작으며 0에 가까울수록 좋다. 만약 전류 측정 시 병렬로 연결할 경우 과전류로 인해 전류계가 고장 날 수 있으므로 반드시 직렬로 연결하해야 한다.

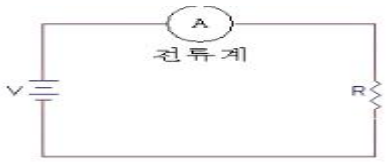


그림 1 전류계 결선

전압을 측정하고자 할 때, 측정하고자 하는 부분에 병렬로 연결한다. 전압계의 저항은 매우 크며, 전압을 측정할 때 저항이 클수록 좋다. 전류와 반대로 직렬로 연결 할 경우 측정이 제대로 이루어지지 않는다.

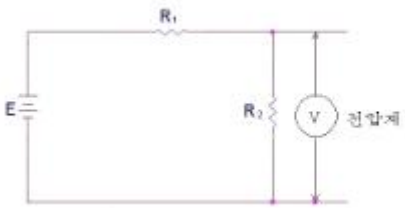


그림 2 전압계 결선

전력계의 전압단자는 부하와 병렬로 연결하고, 전류 단자는 부하와 직렬로 연결함으로써 디바이스에서 전기에너지를 사용한 양을 계산할 수 있다.

III. 실험환경 및 전력측정 결과

본 논문에서 사용한 디바이스는 5V의 교류(DC, Direct, Current)전압을 갖는 디바이스이다. 디바이는 RTOS가 동작하고 있으며, 해당 디바이스에서 알고리즘을 구동하기 위하여 Windows 7 환경에서 알고리즘 소스크를 크로스 컴파일하여 STM32F415 보드에 탑재하였다. 디바이스의 소비전력을 측정하기 위하여 그림3과 같은 방법으로 전력 측정 실험환경을 구성하였다.

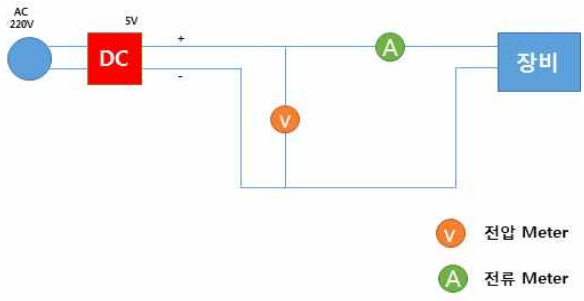


그림 3 소비전력 측정 개념도

알고리즘 호출 시 소비되는 전력을 계산하기 위하여 디바이스가 기본적인 동작을 수행하는 전력 값과, 암호 연산을 수행 할 때 측정되는 전력 값을 측정한다. 그 후 소비전력을 계산하기 위해 측정한 두 결과 값의 차를 이용하여 암호 알고리즘의 소비전력 값을 계산하였다.

Specification	
MCU	STM32F415, Flash 1MB, RAM 192 KB
Flash	1 MB
RAM	192 KB
OS	NeosCM

표 1 디바이스 실험환경

또한 직접 구현 한 알고리즘과 비교를 위하여 OpenSSL[7]의 일부 알고리즘과 KISA에서 제공하는 알고리즘을 사용하였다. 각각의 소스코드는 최적화 옵션 -O1, 디버그 옵션 -g3를 이용하여 동일한 조건에서 컴파일을 수행하였다.



그림 4 실제 소비전력 측정 환경

RTOS의 기본적인 동작만을 수행 하였을 때 측정되는 전류 값은 127.8mA였고, 전압 값은 5.17(w)였다. 그 결과 RTOS가 기본적으로 소비하는 소비전력은 660.726mW였다.

$$\text{기본소비전력} = 5.17(w) \times 127.8(mA) = 660.726mW$$

위와 같은 방법으로 각각의 암호알고리즘의 소비 전력을 측정한 후 기본 소비 전력을 뺀 결과 값이 표2와 같다.

분류	비교 소스(mW)	최적화 소스(mW)	전력 소모 효율
ARIA	78.067	67.21	14%
HIGHT	65.659	69.795	-6%
LEA	88.924	64.625	27%
SEED	71.346	63.591	11%
CBC_ MODE	73.414	71.346	3%
CFB_ MODE	73.931	67.21	7%
OFB_ MODE	74.965	70.312	6%
SHA224 /256	77.033	69.278	10%
SHA384 /512	81.169	72.897	10%

표 2. 저전력 측정 결과 표

전력 소모 효율 측정 방법 =
(비교소스 - 최적화 소스) / 비교소스

IV. 결론

기존에 폐쇄망 혹은 인터넷과 결합되어 있지 않은 하위의 단말들이 최근 인터넷과 결합되어 정보를 전달하는 주최적인 역할을 수행하고 있다. 이에 따라서 저사양, 저전력의 장비에서도 보안의 필요성이 증대되고 있다. 하지만 저 전력의 장비의 특성상 보안을 적용하였을 때 에너지 소비량을 증가시키거나 데이터의 전송이 제대로 이루어지지 않는다면 디바이스의 효용성을 침해하는 일이 발생한다.

본 논문에서는 국내에서 사용되고 있는 ARIA, HIGHT, LEA, SEED알고리즘과 운영모드, 해시함수에 대한 전력 측정을 수행하여 저사양의 장비에서 암호 알고리즘의 전력 소모량을 측정하였다. 또한 그 결과 값을 필드에 많이 사용되고 있는 Openssl과 KISA의 소스코드와 비교를 수행함으로써 암호 알고리즘 구현이 전력 소모량에 미치는 영향에 대한 결과도 확인하였다. 실험의 결과로 LEA알고리즘의 경우 최대 27%의 에너지 효율을 확인 할 수 있었으며 평균적으로 약 9%의 에너지 효율이 있는 것을 확인 할 수 있었다.

본 실험 결과를 바탕으로 저전력 장비에서 보안을 적용하였을 때 소모되는 전력량을 계산하여 디바이스의 수명을 계산 할 수 있을 것이다. 추후 암호 알고리즘 구현 시, 연산량 등의 값들이 전력 소모에 어떤 영향을 미치는지에 대한 연구를 통하여 저전력 장비에서도 안전한 보안을 제공할 수 있다.

[참고문헌]

[1] KS X1213:2004, 128비트 블록 암호알고리즘 ARIA – 제1부: 일반

[2] HIGHT Algorithm Specification, 한국인터넷진흥원, 07, 2009

[3] Deukjo Hong, Jung-Keun Lee, Dong-Chan Kim, Daesung Kwon, Kwon Ho Ryu, and

Dong-Geon Lee, LEA: A 128-Bit Block Cipher for Fast Encryption on Common Processors, 8, March, 2014

[4] SEED 128 Algorithm Specification, 한국인터넷진흥원, 07, 2009

[5] KS X1213:2004, 128비트 블록 암호알고리즘 ARIA – 제2부: 운영모드

[6] FIPS 180-2 - Secure Hash Standard, U. S. Department of Commerce, Technology Administration, National Institute of Standards and Technology, 1 August 2002.

[7] openssl source code, https://www.openssl.org/source/

표준 모델에서 증명 가능한 LWE 기반 BGN-타입 ID기반 동형 암호*

우 주[1†], 김종현[1], 이동훈[1‡]

[1]고려대학교 정보보호학원

A BGN-type ID-based Homomorphic Encryption from LWE in Standard Model*

Joo Woo[1†], Jong hyun Kim[1], Dong Hoon Lee[1‡]

[1]Graduate School of Information Security, Korea University

요 약

본 논문은 worst-case 레티스 문제와 환원 관계가 있다고 알려진 Learning with Errors 문제를 이용하여 랜덤 오라클을 사용하지 않고 표준 모델에서 INDr-sID-CPA에 대한 안전성을 증명할 수 있는 ID 기반 동형 암호 기법을 제시한다. 본 논문의 기법은 Gentry, Halevi, 그리고 Vaikuntanathan에 의해 제시된 동형 암호 기법과 같이 다항식 개수만큼의 행렬 합 연산과 한 번의 행렬 곱 연산을 지원한다. 또한, 본 논문의 기법은 Gentry, Halevi, 그리고 Vaikuntanathan의 ID 기반 동형 암호 기법이 한 비트의 평문만을 지원하는 것과 달리, $Z_2^{2m\times 2m}$라는 큰 메시지 공간에서의 행렬 합 및 행렬 곱 연산을 제공한다.

I. 서론

동형 암호(Homomorphic Encryption)는 암호화된 데이터들에 대하여 해당 평문을 구하지 않고 암호문들의 연산 값만을 이용하여 연산된 평문에 해당하는 암호문을 구할 수 있는 기법이다. Paillier 암호, RSA 암호와 같이 합 혹은 곱 연산을 단독으로 지원하는 공개키 암호 시스템이 설계되었지만, 두 연산을 동시에 지원하는 암호 기법의 설계는 쉽지 않았다[1][2].

2005년 Boneh, Goh, 그리고 Nissim은 페어링 연산을 이용하여 임의의 개수의 합 연산과 한 번의 곱 연산을 지원하는 암호시스템을 제안하였다[3]. 하지만 해당 기법은 양자컴퓨팅 공격에 안전하지 않은 단점이 있다. 2010년 Gentry, Halevi, 그리고 Vaikuntanathan[4]는 [3] 기법의 단점을 해결하기 위해 양자컴퓨팅 공격에 안전한 Learning with Errors(LWE) 가정을 이용하여 다항식 개수의 $Z_2^{m\times m}$ 크기 행렬 합 연산과 한 번의 행렬 곱 연산을 지원하는 동형 암호 기법을 제안하였다. 또한 [4]에서는 기존 [GPV08]의 연구결과를 재해석하여 랜덤 오라클 모델에서 안전한 ID 기반 동형 암호 시스템을 제안하였다. 해당 기법은 기존 [3] 동형 암호 기법과 달리 한 비트 연산만을 지원한다.

본 논문은 Agawal, Boneh, 그리고 Boyen[5]가 제안한 LWE 기반 IBE 기법을 이용하여 논문 [4]에서 제안한 BGN 타입 ID 기반 동형 암호 기법을 개선하여 ID에 대한 선택 평문 공격 구분 불가능(INDr-sID-CPA) 모델에 안전하면서 표준 모델에서 증명 가능한 기법을 제안한

* 이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2017-0-01860, 함수서명 설계기법 및 응용기술 연구)

† 주저자, woojoo0121@korea.ac.kr

‡ 교신저자, donghlee@korea.ac.kr

다. 제안 기법은 기존 연구[4]에서 한 비트였던 메시지 공간을 $Z_q^{2m\times 2m}$으로 확장하였다.

II. 배경지식

본 장은 LWE 문제, 레티스 관련 알고리즘, ID기반 동형 암호 및 안전성모델 등을 정의한다.

2.1 LWE 문제[6]

결정적 LWE 문제는 임의로 선택된 $A\in Z_q^{n\times m}$, $S\in Z_q^{n\times m}$와 [6]에서 정의된 분포 $\overline{\Psi_\beta}^{m\times m}$에서 추출한 $X\in Z_q^{m\times m}$, 임의로 선택한 $R\in Z_q^{n\times m}$에 대하여 순서쌍 (A, A^tS+X)와 순서쌍 (A,R)을 구분하는 문제이다.

2.2 레티스 관련 알고리즘[5]

본 논문은 [5]에서 정의된 레티스 $\Lambda_q^\perp(A) := \{e\in Z^m s.t. Ae=0 (\mathrm{mod} q)\}$의 샘플링 알고리즘을 이용한다. 샘플링 알고리즘 구성은 다음과 같다.

- $TrapGen(q,n)$: q,n을 입력받아 $A\in Z_q^{n\times m}$와 $\Lambda_q^\perp(A)$의 짧은 기저 $T\in Z^{m\times m}$를 출력한다.
- $SampleBasisLeft(A,M,T_A)$: 행렬 $A\in Z_q^{n\times m}$, $M\in Z_q^{n\times m_1}$과 $\Lambda_q^\perp(A)$의 기저 T_A를 입력받아 $\Lambda_q^\perp(A\|M)$의 기저 T를 출력한다.
- $SampleBasisRight(A,B,R,T_B)$: 행렬 $A\in Z_q^{n\times m}$, $B\in Z_q^{n\times m}$, $R\in Z_q^{k\times m}$과 $\Lambda_q^\perp(B)$의 기저 T_B로 $\Lambda_q^\perp(A\|AR+B)$의 기저를 출력한다.

2.3 ID 기반 동형 암호

IBHE는 아래의 5개 알고리즘으로 구성된다.

- $Setup(\lambda)\rightarrow PP,MK$: 보안 상수λ를 입력 받고 마스터키 MK와 공개 파라미터 PP를 출력한다.
- $Extract(PP,MK,id)\rightarrow SK_{id}$: 공개 파라미터 PP와 마스터키 MK, id로 비밀키 SK_{id}를 출력한다.
- $Encrypt(PP,id,M)\rightarrow C$: 공개 파라미터 PP와 id, 메시지 M로 암호문 C를 출력한다.
- $Decrypt(PP,SK_{id},C)\rightarrow M$: 공개 파라미터 PP와 비밀키 SK_{id}, 암호문 C로 메시지 M을 출력한다.
- $Evaluate(PP,C_1,...,C_t,\Psi)\rightarrow C'$: 공개 파라미터 PP와 주어진 여러 개의 암호문 $C_1,...,C_t$을 입력받아 연산 함수 Ψ를 통해 새로운 암호문 C'를 출력한다. 본 논문에서는 다항식 개수의 행렬 합 과 한 번의 행렬 곱 연산으로 구성된다.

2.4 인코딩 함수[5]

본 논문에서 $id\in Z_q^n$는 인코딩 함수 H에 의해 $n\times n$행렬 값으로 인코딩된다. H는 전사 함수로 정의되며 다음 두 성질을 만족한다.

1. 서로 다른 $u,v\in Z_q^n$에 대하여 $Z_q^{n\times n}$행렬 $H(u)-H(v)$의 계수는 n이다.
2. H는 다항식 시간 안에 계산할 수 있다.

2.5 안전성 모델[5]

본 논문에서 사용하는 INDr-sID-CPA 모델의 정의는 다음과 같다.

- $Init$: 공격자는 챌린지 id^*를 전달한다.
- $Setup$: 챌린저는 $Setup$알고리즘을 실행하여 공개 파라미터 PP와 마스터키 MK를 생성하고 공개 파라미터 PP는 공격자에게 전달한다.
- $Queries$: id^*가 아닌 id에 대해서 공격자는 챌린저에게 비밀키 질의를 한다. 질의를 받은 챌린저는 $Extract$알고리즘을 통해 해당 id의 비밀키 SK_{id}를 공격자에게 전달한다.
- $Challenge$: $Queries$과정이 끝나면 공격자는 챌린지 메시지 M을 챌린저에게 보낸다. 챌린저는 암호문 공간에서 임의의 암호문을 선택하고, $b\in\{0,1\}$에 대해 동전을 던져 0이 나온 경우는 주어진 id^*와 평문을 입력하여 생성한 암호문을, 1이 나온 경우는 임의로 선택한 암호문을 공격자에게 전달한다.
- $Guess$: 공격자는 챌린지에게 받은 암호문이 자신이 보낸 평문으로부터 암호화 알고리즘을 통해 만들어진 암호문인지 임의로 선택한 암호문인지를 추측하여 b'값을 챌린저에게 전달한다. b'값이 b값과 같은 경우 공격자가 게임에서 이긴다.

III. 암호 기법

본 논문의 제안 기법은 다음과 같다.

- $Setup(\lambda)$: 보안 상수λ를 입력 받아 q,n,m,σ,α를 설정하고 다음 과정을 거친다.

  1. $TrapGen(q,n)$ 알고리즘으로 임의의 행렬 $A_0\in Z_q^{n\times m}$와 $\Lambda_q^\perp(A_0)$의 짧은 기저 $T_{A_0}\in Z^{m\times m}$을 생성한다.

2. 임의의 행렬 $A_1, B \in Z_q^{n\times m}$를 선택한다.

3. 마스터키를 $MK = T_{A_0}$로, 공개 파라미터를 $PP = (A_0, A_1, B)$로 설정하여 출력한다.

- $Extract(PP, MK, id)$: PP, MK, 그리고 $id \in Z_q^n$를 입력받아 다음 과정을 거친다.

1. $SampleBasisLeft(A_0, A_1 + H(id)B, T_{A_0})$ 으로 $\Lambda_q^{\perp}(F_{id})$의 기저 $T_{id} \in Z^{2m\times 2m}$을 생성한다. (단, $F_{id} = (A_0 \| A_1 + H(id)B)$)

2. 비밀키 $SK_{id} := T_{id}$를 출력한다.

- $Encrypt(PP, id, M)$: $PP, id, M \in Z_2^{2m\times 2m}$을 입력받아 다음 과정을 거친다.

1. 임의의 $S \in Z_q^{n\times 2m}$와 $R \in \{-1,1\}^{m\times m}$, 가우시안 에러 분포 $\overline{\Psi_\beta}$에서 $X_0 \in Z^{m\times 2m}$를 선택한다.

2. $X = \begin{bmatrix} X_0 \\ X_1 = R^t X_0 \end{bmatrix} \in Z^{2m\times 2m}$를 설정한다.

3. 암호문 $C = F_{id}^t S + 2X + M (\mathrm{mod} q)$를 출력한다.

- $Decrypt(PP, SK_{id}, C)$: PP, SK_{id}, C를 입력받아 다음 과정을 거친다.

1. $E = T_{id}^t C T_{id} (\mathrm{mod} q)$로 설정한다.

2. $M' = ((T_{id}^{-1})^t E T_{id}^{-1} (\mathrm{mod} q))(\mathrm{mod} 2)$를 출력한다.

*** 복호화 정확성**

1. $E \equiv T_{id}^t F_{id}^t S T_{id} + 2T_{id}^t X T_{id} + T_{id}^t M T_{id}$
$\equiv (F_{id} T_{id})^t S T_{id} + 2T_{id}^t X T_{id} + T_{id}^t M T_{id}$
$\equiv 2T_{id}^t X T_{id} + T_{id}^t M T_{id} (\mathrm{mod} q)$

2. $(T_{id}^{-1})^t E T_{id}^{-1} (\mathrm{mod} q)$
$= (T_{id}^{-1})^t (2T_{id}^t X T_{id} + T_{id}^t M T_{id}) T_{id}^{-1} (\mathrm{mod} q)$
$= 2X + M$

3. $M' = ((T_{id}^{-1})^t E T_{id}^{-1} (\mathrm{mod} q))(\mathrm{mod} 2) = M$

3.1 동형사상

본 논문의 기법은 M_1, M_2에 대한 암호문 $C_1 = F_{id}^t S_1 + 2X_1 + M_1, C_2 = F_{id}^t S_2 + 2X_2 + M_2$에 대해 다항식 개수의 행렬 합 연산과 한 번의 행렬 곱 연산을 지원한다.

- 행렬 합 : $C_1 + C_2 = F_{id}^t (S_1 + S_2) + 2(X_1 + X_2) + (M_1 + M_2)$을 만족하여 합 연산을 지원한다.
- 행렬 곱 :
$C_1 \cdot C_2^t = (F_{id}^t S_1 + 2X_1 + M_1)(F_{id}^t S_2 + 2X_2 + M_2)^t$
$= F_{id}^t S + 2X + M_1 M_2^t + S' F_{id}$
$(S = S_1 C_2^t,\ X = X_1 (2X_2 + M_2)^t + M_1 X_2^t,$
$S' = (2X_1 + M_1) S_2^t)$

*** 복호화 정확성**

$C = C_1 C_2^t (\mathrm{mod} q)$에 대한 암호문은 $\Lambda_q^{\perp}(F_{id})$의 trapdoor T_{id}를 이용해 다음과 같이 연산한다.

1. $E \equiv T_{id}^t C T_{id} \equiv T_{id}^t F_{id}^t S^t T_{id} + 2T_{id}^t X T_{id}$
$+ T_{id}^t M T_{id} + T_{id}^t S' F_{id} T_{id}$
$\equiv 2T_{id}^t X T_{id} + T_{id}^t M T_{id} \mod q$

2. $(T_{id}^{-1})^t E T_{id}^{-1} (\mathrm{mod} q)$
$= (T_{id}^{-1})^t (2T_{id}^t X T_{id} + T_{id}^t M T_{id}) T_{id}^{-1} (\mathrm{mod} q)$
$= 2X + M$

3. $M' = ((T_{id}^{-1})^t E T_{id}^{-1} (\mathrm{mod} q))(\mathrm{mod} 2) = M$

3.2 안전성 증명

본 논문의 제안 기법은 INDr-sID-CPA 모델에서 안전하다. 증명을 위해 아래 게임을 정의한다.

▶ Game 0.

INDr-sID-CPA game과 동일하다.

▶ Game 1.

Game 0과 달리 A_1는 임의의 $R^* \in \{-1,1\}^{m\times m}$을 선택하여 $A_1 = A_0 R^* - H(id^*)B$로 설정한다.

▶ Game 2.

Game 1과 달리 $A_0 \in Z_q^{n\times m}$를 임의로 선택하고, B는 $TrapGen$으로 $T_B \in Z^{m\times m}$와 함께 생성한다. 비밀키 질의는 $F_{id} = (A_0 \| A_1 + H(id)B)$ $= (A_0 \| A_1 R + (H(id) - H(id^*))B)$ 관계를 이용하여 $SK_{id} = T_{F_{id}} \leftarrow SampleBasisRight(A_0, B', R^*, T_B)$를 생성하여 처리한다. (이때, $B' = (H(id) - H(id^*))$)

▶ Game 3.

Game 2와 달리 챌린지 암호문 $C_{id^*}^*$을 $Z_q^{2m\times 2m}$에서 임의로 선택하여 생성한다.

정리 1. 본 논문의 제안 기법은 LWE 가정이 성립하면, INDr-sID-CPA 모델에서 안전하다.

증명. Game 0과 Game 1, Game 1과 Game 2는 각각 구분하기 어렵다. 자세한 증명은 생략한다. 보조 정리 2는 Game 2와 Game 3이 구분하기 어려움을 증명한다. 이때, Game 3는 챌린저가 선택한 b 값과 무관하게 챌린지 암호문 $C^*_{id^*}$을 임의의 메시지로 선택하기 때문에 이점이 0이다. 따라서 Game 0의 이점은 무시할 수 있다.

보조 정리 1. LWE 가정이 성립할 때, Game 2과 Game 3는 구분불가능하다.

증명. Game 2와 Game 3을 구분할 수 있는 공격자 $\mathcal{A}$를 이용해 LWE 문제를 해결하는 알고리즘 $\mathcal{B}$를 설계함으로써 보조 정리를 증명한다. LWE 문제 $(U,V)\in Z_q^{n\times m}\times Z_q^{n\times 2m}$를 입력받은 $\mathcal{B}$는 공격자 $\mathcal{A}$에게 아래 환경을 제공한다.

- *Init* : $\mathcal{A}$는 챌린지 id^*를 $\mathcal{B}$에게 전달한다.
- *Setup* : $A_0 = U$로 설정하고, id^*와 R^*를 이용해 **Game 2**와 같이 A_1과 B를 생성해 공개 파라미터 (A_0, A_1, B)를 $\mathcal{A}$에게 전달한다.
- *Queries* : **Game 2**와 같이 질의에 답한다.
- *Challenge* : id^*에 대한 챌린지 암호문 C^*을 다음과 같이 생성하여 공격자에게 전달한다.
  1. $C=\begin{bmatrix} V \\ (R^*)^t V \end{bmatrix} \in Z_q^{2m\times 2m}$
  2. $C^* = 2C + M \pmod q$
- *Guess* : 공격자 $\mathcal{A}$는 자신이 수행하고 있는 Game이 Game 2 혹은 Game 3인지 응답한다.

만약 U,V가 LWE 분포를 따른다면,

$$C=\begin{bmatrix} V \\ (R^*)^t V \end{bmatrix}=\begin{bmatrix} A_o^t S+X \\ (R^*)^t A_0^t S+(R^*)X \end{bmatrix}$$
$$=(F_{id^*})^t S+\begin{bmatrix} X \\ (R^*)X \end{bmatrix}$$

를 만족하여 Game 2를 수행하는 것이고, U,V가 랜덤 분포를 따를 경우, C^*가 랜덤이 되어 Game 3를 수행하게 된다. 따라서 알고리즘 $\mathcal{B}$는 공격자 $\mathcal{A}$를 이용하여 Game 2와 Game 3를 구분하는데 성공한다.

IV. 결론

본 논문에서는 Agawal, Boneh, 그리고 Boyen이 제안한 LWE 기반 IBE 기법을 이용하여 논문 [4]에서 제안한 BGN 타입 ID 기반 동형 암호 기법을 개선하여 ID에 대한 선택 평문 공격 구분 불가능(INDr-sID-CPA) 모델에 안전하면서 표준 모델에서 증명 가능한 기법을 제안하였다. 또한 기존의 연구에서 한 비트로 제한된 메시지 공간을 $Z_2^{2m\times 2m}$으로 확장하였다.

하지만 메시지 공간을 확장하면서 암호문 공간 또한 $Z_2^{2m\times 2m}$로 길어졌기 때문에 암호문 길이를 줄여야 효과적인 암호화 기법이 될 것이다. 또한 본 논문의 기법은 다항식 개수의 행렬합을 지원하지만 행렬 곱에 대해서는 한 번의 연산만을 지원하므로 완전 동형성을 갖는 암호 기법을 구성하기 위한 기반이 마련되었다고 볼 수 있다.

[참고문헌]

[1] P. Pallier, "Public-key Cryptosystems Based on Composite Degree Residuosity Classes," In Eurocrypt, pp. 223-238, 1999.

[2] R. Rivest, A. Shamir, L. Adleman, "A Method for Obtaining Digital Signatures and Public-key Cryptosystems," In Communications of the ACM, pp.120-126, 1978

[3] D. Boneh, E.J. Goh and K. Nissim, "Evaluating 2-DNF Formulas on Ciphertexts," In Theory of Cryptography, pp. 336-352, 2005.

[4] C. Gentry, S. Halevi and V. Vaikuntanathan, "A Simple BGN-type Cryptosystem from LWE," In Eurocrypt, pp. 518-534, 2010.

[5] S. Arawal, D. Boneh and X. Boyen, "Efficient Lattice (H)IBE in the Standard Model*," In Eurocrypt, pp. 565-584, 2010.

[6] O. Regev, "On Lattices, Learning with Errors, Random linear codes, and Cryptography," In Symposium on Theory of Computing, pp. 84-93, 2005.

동적 암호 기반 Card OTP 모듈을 사용한 복제방지 RFID Card 제안

권성민*, 박기현*, 김재성*, 원요한*, 이동현*, 이예지*, 지한별*, 박의성*, 이종호**

*KITRI 차세대 보안리더 양성 프로그램(Best of the Best)

**고려대학교 정보보호대학원

Copy Protection RFID Card Proposal Using a dynamic password-based Card OTP module

Seongmin Kwon*, Gihyun Park*, Jaesung Kim*, Yohan Won*, Donghuyn Lee*, Yeji Lee*, Hanbyeol Ji*, Uiseong Park*, Jongho Lee**

*BoB Korea Information Technology Research Institute

**Graduate School of Information Security, Korea University

요 약

13.56MHz HF 대역을 사용하는 비접촉식 IC 카드는 공통적으로 ISO/IEC 14443-1:2016 개정안을 표준으로 사용한다. 스마트 RFID Card들은 기업, 교육, 관광, 레저, 주거 등 다양한 영역에서 널리 쓰이고 있으며 점점 더 확장되고 있다. 지속적으로 보안에 대한 이슈가 발생하여 보완을 해왔으나 대다수의 RFID Card들은 취약하게 쓰일 수 있는 구조로 이루어져 있다. 본 논문에서는 취약하게 쓰일 수 있는 구조상의 문제점을 기술하고 개인정보가 안전하게 보호될 수 있는 동적 암호 기반 Card OTP(One-Time Pad) 모듈을 활용한 복제방지 RFID Card를 제안한다.

핵심어: RFID, ID Card, 카드 OTP 모듈, 해시함수.

I. 서론

RFID(Radio Frequency IDentification)는 IoT(Internet Of Things) 다양한 분야와 방면으로 사용될 수 있는 차세대 핵심 기술로 전파를 통해 상품이나 사물의 정보를 IC에 저장하고 식별하는 핵심 기술[1]이다. RFID는 기업, 교육, 관광, 레저, 주거 등의 다양한 산업 분야에서 사용되고 있으며 그 중에서 빌딩, 오피스텔 같은 건물 시스템에서 사용되는 RFID Card는 스피드게이트, 통제구역, 신원을 확인하기 위한 식별(identification)용으로 사용되고 있다.

현재 사용되는 RFID Card들은 대부분 정적 암호 기반의 패시브 태그이다. 다양한 방법을 통해서 보안성을 높이고 있지만 실무에서 취약하게 사용되고 있음을 확인했다. 즉, 정적 암호 기반은 암호화 기법 등의 보안 기술이 제공되더라도 운영상에서 취약점이 드러날 수 있다는 한계점이 있다.

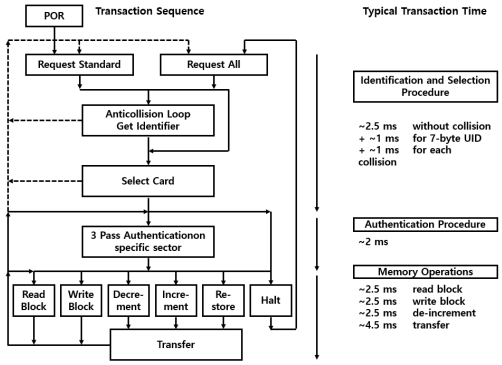
본 논문에서는 이러한 문제점에 대해서 운영과 상관없이 안전하게 사용될 수 있는 동적 암호 기반 OTP RFID Card를 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 정적 암호 기반 RFID Card의 설계와 실제 용례들을 통해서 한계점을 기술하고 3장에서는 동적 암호 기반의 OTP RFID Card에 대해서 제안한다. 마지막으로 4장에

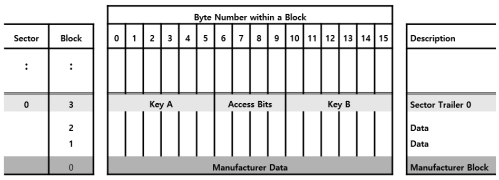
서 결론을 맺는다.

II. 정적 암호 기반 RFID Card

통계적으로 가장 많이 사용되는 NXP Mifare를 통해서 정적 암호 기반이 취약할 수 있음을 증명했다. RF 인터페이스는 ISO/IEC 14443A 표준을 따르고 있으며 공식 문서[2]에서 보안 및 개인 정보 보호를 위해 Three pass authentication 절차(ISO/IEC DIS 9789-2)와 2개의 키를 사용하고 있다. 아래에서 그 보안 설계에 관해 기술한다.

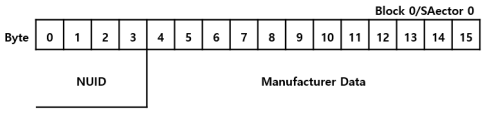


[그림 1] command flow diagram



[그림 2] Memory organization

메모리 조작 이전에 인증 절차를 수행하면 각 블록에 대해 지정된 두 개의 키(KEY)를 통해서 블록에 액세스할 수 있도록 설계되어있으며 RFID Card를 읽기 위해서 리더기는 다음 메모리 액세스의 메모리 위치를 지정하고 해당 키를 사용해서 3단계 인증절차를 진행한다. 인증에 성공하면 모든 명령과 응답이 암호화되어 정보를 주고받게 되도록 설계되었다.

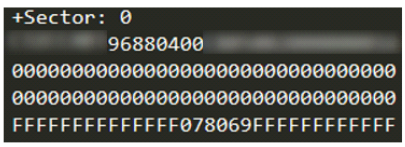


[그림 3] UID position

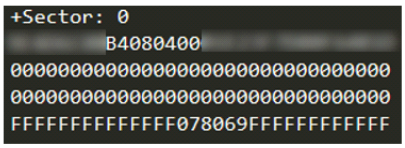
Mifare Classic RFID Card에서 0번 섹터에는 UID와 제조자 정보가 Read-Only로 저장되어있으며 해당 섹터는 Write가 불가능하다. 이것은 Radio 표준으로 규정된(ISO 18000) 사항이므로 공통적인 하드웨어적 특징이다. 이러한 표준에 따라 설계된 RFID Card는 내부 정보가 공개되어있지 않고 Read-Only의 하드웨어적 제약으로 인해서 안전해 보이지만, 2008년 Henryk Plötz and Karsten Nohl에 의해 USENIX 보안컨퍼런스에서 부분적인 리버스 엔지니어링 알고리즘이 게시[3]되었으며 같은 해 Radbound 대학 Nijmegen은 완전한 리버스 엔지니어링과 데이터를 복제하고 조작될 수 있음을 입증[4]하기도 했었다. 또한, 2013년 BlackHat에서도 RFID Hacking과 관련된 발표[5]가 있었다.

이처럼 RFID Card에 대한 복제 위협은 지속적으로 존재해왔다. 또한, 새로운 기술이 개발되더라도 운영상의 문제로 인한 사례들이 많다. 그 중에서도 초기값 혹은 널리 알려진 범위에서 사용되는 KEY는 지속적으로 문제가 되고 있다.

본 논문에서 실제로 주변에서 사용되는 정적 암호 기반 RFID Card들의 취약점을 증명하고자 RC522 모듈과 UID Writable RFID Card를 사용했다. 아래의 그림들은 초기값 혹은 취약한 키로 인해서 정보가 노출되는 대상들의 데이터이다.



[그림 3] 모 공공기관



[그림 4] 모 건물 브랜드

[그림 3, 4]는 실제로 읽어 들인 RFID Card의 0번 섹터(UID와 제조사 정보)이다. 이름만 들어도 익숙한 대표적인 모 공공기관과 건물에서 사용되는 RFID Card들이 초기값 키(FFFFFFFFFFFF, 6 byte)를 사용하는 것을 볼 수 있다. 이렇게 노출된 정보들은 0번 섹터 쓰기가 가능한 특수 Card를 통해서 복제될 수 있으며 ID로써 고유성을 잃고 악용될 수 있는 위험성이 높아진다. 모 브랜드 건물의 경우, 오피스텔에서 복제된 Card에 의해서 건물 출입뿐만 아니라 Card에 해당하는 호실 또한 그대로 열린다는 점은 복제 가능한 RFID Card가 얼마나 위험한지 알 수 있다.

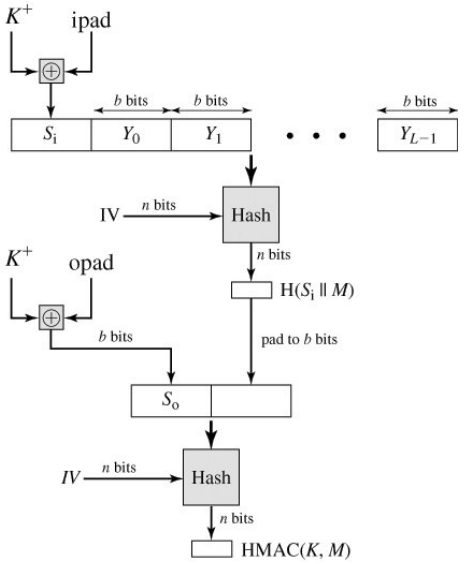
이러한 문제들은 기술적으로 예방할 수 있는 방법이 명확히 제공되고 있지만 운영상의 원인으로 발생한다고 말할 수 있다. 동적 암호 key의 OTP RFID Card는 운영과 상관없이 안전한 ID로써 존속될 수 있다는 점에서 정적 암호 기반과 대비해 더 안전하게 사용될 수 있다.

Ⅲ. 동적 암호 기반 OTP RFID Card

OTP(One Time Password)[6]는 특정 인증번호 대신 무작위로 생성되는 일회용 인증번호를 이용하는 인증 방식이다. 매번 인증번호가 변경되며 다음 인증번호의 유추가 수학적으로 불가능하다는 점에서 안전하다고 평가받으며 그로 인해 금융, 전자상거래 등 여러 분야에서 사용되고 있다.

OTP를 사용하게 된다면 2장에서 증명되었던 취약점 실태를 완화할 수 있다. 제안의 핵심은 동적으로 변화하는 값을 암호로 사용하여 운영과 무관하게 일관된 안정성을 제공하고 복제로부터 강한 내구성을 가질 수 있다는 점이다. 다만, 시간 동기화 OTP[7]로 만들어진 값만을 사용할 경우 값이 예측될 수 있다는 한계점이 존재하기 때문에 이러한 부분을 보완하기 위해 RFC-2104 HMAC(Hash-based Message Authentication Code)을 이용한 RFC-6238 TOTP(Time-Based One-Time Password) 알고리즘을 사용한다.

$$\mathrm{HMAC}(K,m) = H\Big((K' \oplus opad) \,||\, H\big((K' \oplus ipad) \,||\, m\big)\Big)$$

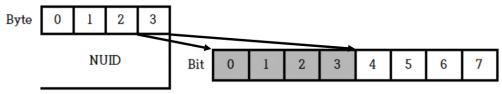


[그림 5] HMAC diagram

HMAC을 사용하면 키값(OTP값)과 메시지(UID)를 활용하여 무결성과 신뢰성을 보장하는 위조방지를 가능하게 한다. Time-based OTP에서 생성된 값과 UID를 조합, HMAC을 적용하여 동적 암호 KEY를 구현하는 것이 OTP RFID Card의 핵심이다.

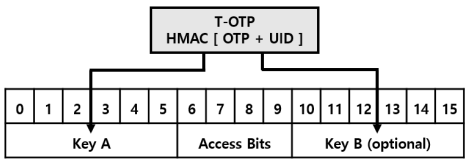
3장에서 제안하고 있는 동적 암호 Key OTP RFID Card에 대해 몇 가지 짚고 넘어가야할 부분이 있다. Mifare Classic의 RW속도는 대략 less than 100ms이고 모 브랜드 건물의 경우 한 건물에 존재하는 가구의 수는 약 300여 가구이다. 결과적으로 각 가구의 수와 동일한 개수의 KEY가 DB에 저장된다. 문제는 이러한 경우, 출입

을 위한 인증이 시간이 매우 길어질 수 있다는 점이다. IoT와 현대사회에서는 보안이 좋아야 하는 인식만큼 사용 편의성에 대한 욕구가 높다. 그러한 측면에서 RFID Card 인식을 빠르고 정확하게 이루어져야 한다.



[그림 6] 제안 Seed Uid (4bit)

그렇기 때문에 전체 UID(4byte)를 사용하는 것이 아니라 일부분(4bit)을 시드로 사용함으로써 300가구 각각의 키는 16개의 Key Pool로 나뉘게 되고, 이는 충분히 신속한 속도로 인증을 진행할 수 있게 해준다. 16개의 Key pool은 현실적으로 사용될 수 있는 모델의 제약 요소를 극복할 수 있다.



[그림 7] Dynamic KEY

이러한 동적 암호 기반 RFID Card는 정적 암호 기반에 비해서 구조적으로 안전하고, 복제로부터 자유로울 뿐만 아니라 사용자 측면에서 어떻게 사용하는 지에 관계없이 Card의 유일성이 훼손될 가능성은 낮아진다.

Ⅳ. 결론

정적 암호 기반 RFID Card는 패시브 태그이기에 별도의 전원이 필요하지 않다. 하지만 제안 OTP RFID Card는 자체 전원이 필요하므로 일종의 액티브 태그에 속할 수 있다. 이러한 연유로 약간의 비용적인 증가는 있을 수 있다. 하지만 복제로부터 더욱 안전해지고 ID로서의 유일성을 지킴으로써 발생할 수 있는 사고들에 대비하여 합리적이라고 할 수 있다.

최근 IoT에 대한 관심이 많아지는 만큼 RFID가 사용될 수 있는 영역 또한 넓어지고 있다. 지금까지 사용되고 있었던 정적 암호 기반의 RFID Card들은 운영상에서 발생할 수 있는 취약점이 있을 뿐 아니라 지속적으로 복제에 대한 공격이 가능했다. 하지만, 동적 암호 기반 OTP RFID Card는 운영상에서 문제가 발생할 수 있는 여지와 복제에 대한 위험도를 줄일 수 있다.

본 논문에서는 정적 암호 기반 RFID Card 취약점과 한계에 대해서 증명하고, 동적 암호 기반 OTP RFID Card를 통해서 얻을 수 있는 보안 이점을 제시했다. RFID 사용 영역이 증가함에 따라 그에 대한 보안적인 요소도 고려되어야 할 것이다.

[참고문헌]

[1] Jennifer Zaino,RFID Trends: What's Ahead 2017 in IOT journel

[2] NXP, Mifare MF1S50YYX_V1

[3] Karsten Nohl and David Evans, Starbug and Henryk Plotz. Reverse-Engineering a Cryptographic RFID Tag

[4] Flavio D. Garcia. Dismantling MIFARE Classic. in ESORICS 2008

[5] Francis Brown. Live Free or RFID Hard. in Blackhat USA 2013

[6] RFC 2289. A One-Time Password System

[7] Chao-Hsi Huang,Shih-Chih Huang. RFID systems integrated OTP security authentication design

[8] R.K. Pateriya, Sangeeta Sharma. The Evolution of RFID Security and Privacy: A Research Survey

[9] David M'Raihi, US 9258124 B2, Time and event based one time password

[10] John K. Bona, Mark A. Cox. US 9004365 B2, One-time password card for secure transactions

2017 한국정보보호학회 동계학술대회

포스터발표 트랙 1

고출력 전자기파 위협에 대한 원전의 통신시스템 보호 방안 연구

송동훈*

*한국원자력통제기술원

A Study on the Protection Method for Communication Systems in Nuclear Power Plants to Protect Electromagnetic Pulse

Dong-Hoon Song*

*Korea Institute of Nuclear Nonproliferation and Control

요 약

북한 핵실험, 고출력 전자기파(EMP) 발생장치의 지속적인 개발 등으로 신규 위협으로 대두되는 EMP는 국가 중요시설로 대한민국의 기저부하를 담당하는 원자력발전소의 통신시스템에 영향을 줄 수 있다. EMP로 인해 통신시스템이 영향을 받을 경우, 원전의 안전기능을 수행하는 시스템 간의 제어 명령 전달이 불가능해 심각한 피해를 야기할 수 있다. 이에 따라, 본 논문에서는 EMP 영향 하에서도 원전의 통신시스템의 건전성을 유지하는 보호 방안을 연구하였다.

I. 서론

원전의 통신시스템은 안전기능을 수행하는 안전시스템과 전력생산을 담당하는 비안전시스템의 제어 및 정보 교환을 위해 사용되는 중요한 시스템으로, 특히 원전을 안전하게 정지하고 유지하기 위해서는 안전시스템 사이의 통신시스템의 건전성 유지가 필요하다. 따라서 원전의 안전기능을 수행하는 중요 통신시스템은 여러 안전 규제 요건들을 적용하여 설계 및 설치되었다.

하지만 북한 핵실험, EMP 발생장치의 지속적인 개발 등으로 신규 위협으로 대두된 EMP 위협에 대한 방호 설계는 통신시스템에 고려되지 않았다. 이에 따라, EMP 영향 하에서도 원전의 안전기능에 필요한 통신시스템의 건전성 확보를 위해 적절한 보호 방안에 대한 연구가 필요하다.

II. 본론

원전의 안전기능 수행에 필요한 통신시스템을 EMP 위협으로부터 보호하기 위해서는 EMP 위협의 특징을 분석하고 특징별로 EMP 방호 대책을 개발하여야 한다.

2.1 EMP 위협

EMP 위협은 발생원에 따라 핵EMP와 비핵EMP로 구분할 수 있다.

2.1.1 핵EMP

핵EMP는 핵폭발 과정에서 발생되는 감마선이 공기 입자와 부딪혀 콤프턴 효과에 의해 전자가 발생되어 나타는 EMP로, 핵EMP 중 고 고도 EMP(High-altitude EMP)는 상공 30㎞ 이상에서 핵폭발이 발생할 경우 광범위한 지역에 높은 전계강도를 가진 EMP가 생성된다.

2.1.2 비핵EMP

비핵EMP는 전자기파 발생장치를 이용해 발생되는 EMP로 비교적 한정된 범위에 높은 전계강도를 가진 EMP가 발생된다. EMP 발생장치는 휴대성이 좋은 EMP 총부터 군사용 EMP 발생장치까지 크기와 모양에 따라 EMP 강도 및 범위가 다르다.

2.2 EMP 피해 경로

핵EMP와 비핵EMP가 발생될 경우, EMP는 두 가지 경로를 통해 전자장비에 영향을 줄 수 있다. 공기 중을 통한 복사성 EMP는 노출된 전자장비에 영향을 줄 수 있으며, 전력선 등에 유기되어 전달되는 전도성 EMP는 외부와 전력·통신선 등을 통해 연결된 전자장비에 피해를 줄 수 있다.

2.3 통신시스템에 대한 EMP 방호 대책

원전의 안전기능 수행에 필요한 통신시스템을 EMP 위협으로부터 보호하기 위해서는 EMP 피해경로에 따라 EMP를 완전 차단하여 통신시스템의 건전성을 유지하여야 한다.

2.3.1 복사성 EMP 보호를 위한 차폐

공기 중으로 전파되는 복사성 EMP를 막기 위해서는 EMP를 방호할 수 있는 차폐체로 통신시스템을 보호하여야 한다. 대표적인 차폐체로는 EMP가 투과할 수 없는 철판 또는 두꺼운 콘크리트 구조물이 있으며, 이러한 차폐체를 통해 보호대상인 통신시스템을 감싼다면 효과적으로 복사성 EMP 방호가 가능하다.

2.3.2 전도성 EMP 보호를 위한 EMP 필터

외부의 전력·통신선을 통한 전도성 EMP는 통신시스템과 연결되는 외부와의 전선 중간에 EMP 필터를 설치하여 강력한 EMP를 통신시스템으로 전달되는 것을 막을 수 있다.

2.3.3 기타 EMP 방호 대책

EMP를 효과적으로 방호하기 위해서는 통신시스템을 대지와 접지시켜 EMP로 인한 과도 전류를 대지로 우회시킬 수 있다. 또한, 전기적 영향을 덜 받는 광통신 방식으로 통신시스템을 구성하면 EMP에 대한 영향을 줄일 수 있다.

III. 결론

원전은 국가의 기저부하를 담당하는 국가 중요시설로 안전설계 뿐만 아니라 여러 위협들로부터 안전하게 보호하여야 한다. 특히, 원전을 사고 시에도 안전하게 정지할 수 있는 안전기능에 사용되는 통신시스템에 EMP 방호 대책을 적용하여 EMP 위협 하에서도 정상 동작을 보장하여야 한다.

본 논문에서는 원전의 통신시스템을 EMP 위협으로부터 보호하기 위한 EMP 방호 대책을 제시하였다. 원자력시설 등의 방호 및 방사능 방재 대책법에 따라 원자력시설에 대한 EMP 방호 체제를 구축하기 위해서는 원전의 EMP 영향 하에서도 안전기능 수행 가능 여부에 대한 연구가 추가적으로 필요하며, 연구 결과에 따라 EMP 취약 부분에 대해 EMP 방호 대책을 적용하여 원전뿐 아니라 원자력시설에 대한 EMP 방호 체계를 구축해 나갈 예정이다.

[참고문헌]

[1] 한국원자력통제기술원, “원자력시설 등의 방호 및 방사능 방재 대책법령집”, 2017.

[2] US NRC, Sandia National Laboratories, "Interaction of Electromagnetic Pulse with Commercial Nuclear Power Plant Systems," 1983

소프트웨어 취약점 분석 자동화 도구에 관한 연구

나건배, 전지수, 김태은, 김환국
한국인터넷진흥원

A Study on Software Vulnerability Analysis Automation Tool

Geonbae Na, Jee-Soo Jurn, Taeeun Kim and Hwankuk Kim
Korea Internet & Security Agency

요 약

자동화된 공격의 양·질적 증가가 예상되는 가운데, 전문가 수준의 자동화된 대책 및 예방 체계의 필요성이 커지고 있다. 국외에서는 美 국방성을 중심으로 이러한 필요성을 인지하고 사이버 자동화 공격과 방어 체계에 대한 연구가 활발히 진행되고 있다. 이에 국내에서도 여러 연구기관들이 협력해 취약점 탐색 및 분석 자동화와 패치 자동화를 목표로 연구가 진행 중이다. 본 논문에서는 취약점 탐색 및 분석 자동화에 초점을 맞춰, 주요 기술인 기호실행에 대한 기술 동향과 도구 비교 분석을 통한 결과를 바탕으로 연구 방안에 대해 논한다.

I. 서론

포티가드랩(FortiGuard Labs)과 한국인터넷진흥원의 2017년 보안 위협 전망[13][14]에 따르면 사람같은(Human-like) 자동화된 공격이 증가할 것으로 전망했다. 한편, 국외에선 14년부터 자동화된 취약점 탐색과 패치 시스템 개발을 위해 DARPA CGC[15] 대회가 매년 열려왔다. CGC는 인간의 개입을 최소화한 AI 사이버전을 다룬 대회로써 자동화된 시스템이 바이너리 수준의 취약점 분석을 통한 공격과 방어가 가능함을 증명했다. DARPA는 해당 분야를 크게 취약점 탐색 자동화와 패치 자동화, 그리고 인공지능 적용으로 나눠 각각 3년, 10년, 20년을 목표로 하고 있다('15 기준). 이처럼 자동화된 대책 및 예방 체계에 대한 인식과 필요성이 커지고 있는 추세 속에서 국내 역시 취약점 탐색과 패치 자동화를 목표로 연구가 진행 중에 있다.

본 논문에서는 취약점 탐색 및 분석 자동화에 초점을 맞춰, 그 근간을 이루는 기호실행(Symbolic Execution)의 동향과 각 기술간 비교 분석을 바탕으로한 앞으로의 취약점 분석 기술 자동화 연구 방안에 대해 논하고자 한다.

II. 관련 연구

2.1 기호실행

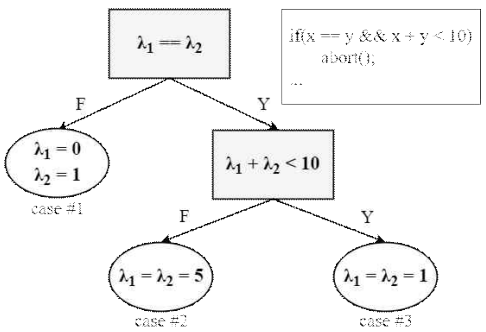


그림 1. 기호실행 경로탐색 예시

기호실행은 프로그램 분석 기술로써 최초 개념은 1970년대 중반에 제안됐다[1][2][3]. 기호실

행의 핵심 아이디어는 프로그램의 입력 값을 구체적인(Concrete)값이 아닌, [그림 1]과 같이 기호값을 이용해 프로그램의 경로를 탐색하는 것이다. 탐색과정에서 기호식(Symbolic Expression)이 생성되고, 해당 식은 Z3[4]나 STP[5]와 같은 제약조건 해결자(Constraint Solver)를 통해 풀이된다. 기호실행의 주된 목적은 프로그램의 모든 유효한 실행 경로를 판별하고, 각 경로에 도달할 수 있는 입력 셋을 구성하는 것이다. 경로는 프로그램의 시작 지점부터 종료 지점(Exit Statement), 혹은 에러 지점(프로그램 충돌 등)까지를 나타낸다.

현재 기호실행은 콘콜릭(Concolic) 실행 형태로 발전했다. 콘콜릭은 콘크리트(Concrete)와 기호(Symbol)의 합성어로써 명칭대로 구체적 실행과 기호실행을 혼합한 기술이다. 기호실행이 경로 탐색을 위한 모든 과정에서 기호를 이용했던 것과 달리 콘콜릭 실행은 초기 입력값을 구체적인 값으로 설정한다. 입력값은 사용자에 의해 주어진 값일 수 있고, 무작위 값일 수도 있다. 이렇게 설정된 구체적인 값으로 최초 탐색을 시작하며 프로그램의 제약조건에 대한 정보를 모아 기호식을 생성한다. 프로그램의 종료 지점에 도달하면, 기호식의 일부 조건을 부정하는 작업을 반복적으로 수행해 미탐색 경로로 유도할 수 있는 새로운 입력값을 찾아내는 방식이다.

2.2 기호실행 도구

기호실행과 관련된 도구나 기술들은 약 100여개[16]가 존재하며, 기본적인 알고리즘부터 시스템 구성까지 큰 특징들이 존재한다. 주요 도구들의 비교 내용은 [표 1]과 같다.

동시에 여러 경로를 탐색하는 온라인 실행(Online Execution) 방식의 KLEE('08)[6]는 무작위 경로 선택 전략을 선택했다. 우선, 온라인 실행 방식의 이점은 탐색의 중복성을 제거할 수 있으나, 여러 경로에 대한 정보를 유지 및 관리하기 위한 메모리 오버헤드가 크게 발생할 수 있는 방식이다. KLEE는 이러한 문제를 Copy-on-write 방식으로 해결하고자 하였으며, Cloud9('11)[7]은 클라우드 환경을 기반으로 하여 병렬 처리 접근방식을 취했다.

다음으로 무작위 경로 선택 전략이라 함은 동시에 경로를 탐색하는 구조상 다음 경로를 선별하는 단계가 필요한데, 깊이 우선 탐색이나 너비 우선 탐색과 같은 정해진 원칙이나 규칙에 의존하지 않고 무작위로 선택하여 진행하는 기호실행 전략이다. KLEE는 완전한 무작위 선택이 아닌, 길이와 분기점을 기반으로 각 경로에 확률을 할당한 후, 이를 기반으로 매번 새로운 경로를 선택한다.

이러한 KLEE는 후에 QEMU 가상화 환경과 x86-to-LLVM 리프터가 추가된 형태로 발전했다. 해당 도구는 S^2E('12)[8]로써 바이너리 분석을 위해 머신 코드를 중간 표현(IR) 수준으로 끌어올린 후, 기존 KLEE의 기호실행을 적용하는 방식이다.

기호실행에서 중간 표현은 중요한 요소로써 아키텍쳐에 의한 제약을 없애고, 다양한 플랫폼 지원을 가능하게 한다. 여러 언어로 작성된 소스 코드를 하나의 중간 표현으로 변환하거나, 전술한 S^2E와 같이 바이너리 수준의 코드를 중간 표현으로 변환함으로써 기호실행 도구의 적용 대상을 대폭 넓힐 수 있는 것이다. 비교적 최근에 개발된 ANGR('16)[9]의 경우 VEX 중간 표현을 사용하는데, VEX는 구조가 단순한 RISC 형태로써 x86과 ARM 등 다양한 아키텍

표 1. 기호실행 도구 비교

도구명	해결자	스케줄링	중간표현	운영체제	지원언어	바이너리지원	아키텍쳐
DART	lp_solve	Offline	CIL	Linux	C	○	x86
KLEE	STP	Online	LLVM	Linux	C/C++	X	-
Bitblaze	STP	Offline	VEX	Linux	C/C++	○	x86/ARMv4
Cloud9	자체개발	Online	LLVM	Linux	C++	X	-
S^2E	Z3	Online	LLVM	Lin./Win./Mac.	C++	○	x86/x64/ARM
Angr	Z3	Offline	VEX	Lin./Win./Mac.	Python	○	x86, MIPS, ARM, etc.
Driller	Z3	Offline	VEX	Lin./Win./Mac.	Python	○	

쳐를 지원하는 범용성을 갖는다.

III. 취약점 분석 자동화 연구방안

3.1 한계점

기호실행을 이용한 취약점 분석 자동화 기술을 모델링을 하는데 있어 여러 고려사항이 존재한다. 온라인/오프라인 실행 방식부터 메모리 모델링, 파일 시스템 구성 등 목적과 분석 대상에 대한 심도 있는 고려가 바탕이 되어야 한다. 이러한 다양한 고려 사항 중 특히 중요시 언급되는 부분은 경로의 폭발(Path Explosion)이다. 경로폭발이란 복잡한 프로그램, 혹은 큰 프로그램을 분석하면서 생기는 수 많은 경로의 수 때문에 논리적·물리적 한계가 발생하는 것을 뜻한다. 이를 해결하기 위한 다양한 방법이 제안됐는데, 앞서 서술한 KLEE의 무작위 경로 선택이나 가지 치기, CFG(Control Flow Graph)를 활용한 휴리스틱 탐색 등이 있다.

가지 치기는 불필요하게 반복되는 탐색 범위를 제거하는 것으로써, 새로운 경로 탐색 시 이전의 탐색 범위가 포함될 경우 해당 범위에 대한 기존 정보를 재사용하여 반복 연산을 방지하는 것이다. CFG 기반 휴리스틱 탐색의 경우, 프로그램의 정적 CFG를 활용하여 탐색하지 못한 명령어로부터 가장 가까운 경로로 유도해 나가는 방법이다. 이 방법은 효과적인 것[10]으로 나타났지만, CFG 복구와 관련된 퍼포먼스와 간접 점프 이슈가 남아있다. 즉, CFG 복구에 따른 추가 시간 및 자원의 발생과 런타임 시에 결정되는 간접 점프를 올바르게 표현할 수 없는데에서 오는 한계점이 발생한 것이다.

이처럼 기호실행의 근본적 이슈인 경로폭발을 해소하기 위해 기호실행의 내부적인 측면에서 다양한 접근과 시도가 있었다. 그리고 최근에는 이러한 내부적인 측면에서 벗어나 타 기술과의 융합이 활발히 이루어지고 있는 추세다.

3.2 연구방안

퍼저의 불완전성을 보완하고, 기호실행의 경로폭발 문제를 해소하기 위해 Driller('16)[11]는 ANGR과 AFL 퍼저[17]를 융합했다. Driller의 주 아이디어는 퍼저로 더 이상 진행할 수 없는 상태가 되면 기호실행을 통해 문제를 해결하고, 다시 퍼저를 재개하는 것이다. 퍼저가 더 이상 진행할 수 없는 상태라 함은 퍼저의 특성상 파일의 매직값이나 해쉬값 등 특정값(Specific Value)에 따른 조건문에 의해 지나치게 많은 시간이 소요되거나 탐색 실패에 이르게 되는 경우를 말한다. Driller는 이 때, 기호실행 엔진을 호출하여 특정값 추론 후 퍼저를 재호출하여 탐색을 재개한다. 필요시마다 퍼저와 기호실행을 번갈아가며 호출하는 Driller의 방식은 기호실행의 경로폭발 문제와 퍼저의 멈춤(Stuck) 문제 및 코드 커버리지 저하 문제를 해소한 예이다.

Triton('15)[12]의 경우 기호실행과 테인트 분석을 융합한 예이다. 테인트 분석은 패킷, 입력 파일, 사용자 입력 정보 등의 신뢰할 수 없는 입력값을 오염된 것으로 간주하고, 오염된 값이 프로그램에 영향을 끼치는 범위와 그 정도를 분석하기 위한 기술이다. Triton은 기호실행 시 테인트 분석 엔진을 활용하여 메모리와 레지스터의 오염 여부를 체크하고, 외부 개입이 가능한 경우에 한해서만 해결자에게 질의를 한다. 즉, Triton은 복잡한 기호식을 푸는데 사용되는 자원을 최소화하고, 실제로 위협 가능성이 존재하는 경로만을 선별하는 성능 최적화 예시로 볼 수 있다.

이러한 사례를 바탕으로 취약점 분석 자동화 연구를 위해 우리는 해당 분야에 Driller와는 또 다른 퍼징과 기호실행의 융합을 연구하고 있다. 분석 대상 바이너리의 CFG로 각 기본블록의 복잡도를 계산한 후, 기호실행과 퍼징을 선택적으로 운용할 수 있는 방안을 연구중이다. 더불어 테인트 분석을 활용한 취약점 자동 원인 분석 방안을 모색 중이다. 자동으로 탐색된 취약점의 원인을 규명하고, 패치 역시 자동화함으로써 고도화되는 사이버 공격에 보다 능동적으로 대응할 수 있는 효과적인 자동화 방어 체계를 이룰 수 있을것으로 기대한다.

IV. 결론

자동화된 공격의 양·질적 증가가 예상되는 가운데, 전문가 수준의 자동화된 대책 및 예방

체계의 필요성이 커지고 있다. 본 논문에서는 취약점 탐색 및 분석 자동화에 초점을 맞춰 그 근간을 이루는 기호실행의 현주소를 살펴보고, 이를 활용한 연구방안에 대해 논하였다. 향후에 서술한 연구방안의 기술 검증 및 테스트를 진행할 예정이다.

ACKNOWLEDGEMENT

이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2017-0-00184, 자기학습형 사이버 면역 기술 개발)

[참고문헌]

[1] Boyer, Robert S., Bernard Elspas, and Karl N. Levitt. "SELECT-a formal system for testing and debugging programs by symbolic execution." ACM SigPlan Notices 10.6. pp. 234-245. 1975.

[2] King, James C. "Symbolic execution and program testing." Communications of the ACM 19.7. pp. 385-394. 1976.

[3] King, James C. "A new approach to program testing." ACM SIGPLAN Notices. Vol. 10. No. 6. ACM, 1975.

[4] De Moura, Leonardo, and Nikolaj Bjørner. "Z3: An efficient SMT solver." Tools and Algorithms for the Construction and Analysis of Systems. pp. 337-340. 2008.

[5] Ganesh, Vijay, and David L. Dill. "A decision procedure for bit-vectors and arrays." CAV. Vol. 4590. 2007.

[6] Cadar, Cristian, Daniel Dunbar, and Dawson R. Engler. "KLEE: Unassisted and Automatic Generation of High-Coverage Tests for Complex Systems Programs." OSDI. Vol. 8. 2008.

[7] Bucur, Stefan, et al. "Parallel symbolic execution for automated real-world software testing." Proceedings of the sixth conference on Computer systems. ACM, 2011.

[8] Chipounov, Vitaly, Volodymyr Kuznetsov, and George Candea. "The S2E platform: Design, implementation, and applications." ACM Transactions on Computer Systems (TOCS) 30.1. pp. 2. 2012.

[9] Shoshitaishvili, Yan, et al. "Sok:(state of) the art of war: Offensive techniques in binary analysis." Security and Privacy (SP), 2016 IEEE Symposium on. IEEE, 2016.

[10] Burnim, Jacob, and Koushik Sen. "Heuristics for scalable dynamic test generation." Automated Software Engineering, 2008. ASE 2008. 23rd IEEE/ACM International Conference on. IEEE, 2008.

[11] Stephens, Nick, et al. "Driller: Augmenting Fuzzing Through Selective Symbolic Execution." NDSS. Vol. 16. 2016.

[12] Saudel, F., and J. Salwan. "Triton: A dynamic symbolic execution framework." Symposium sur la securite des technologies de l'information et des communications, SSTIC, France, Rennes. 2015.

[13] Fortinet 2017 Cybersecurity Predictions, https://blog.fortinet.com/2016/11/21/fortinet-2017-cybersecurity-predictions-accountability-takes-the-stage

[14] KISA 2017년 7대 사이버 공격 전망, https://www.kisa.or.kr/notice/press_View.jsp?mode=view&p_No=8&b_No=8&d_No=1516

[15] DARPA Cyber Grand Challenge, http://archive.darpa.mil/cybergrandchallenge/

[16] Symbolic execution timeline, https://github.com/enzet/symbolic-execution

[17] American Fuzzy Lop. http://lcamtuf.coredump.cx/afl/

Monitor Mode를 이용한 모바일 디바이스 사용자 추적 방안에 대한 연구

김건용* 김형철** 이원용** 유명성** 박강민**
정재완** 오동현** 심기욱** 이경문**

*KITRI BoB - Team G0Host **KITRI BoB - Team G0Host

A Study on User Device Tracking Method Using Monitor Mode

Geon Yong Kim* Hyeong Cheol Kim** Won Yong Lee**
Myoung Seoung You** Kang Min Park** Jae Wan Jeong**
Dong Hyeon Oh** Ki Wook Shim** Kyoung Moon Lee**

*KITRI BoB - Team G0Host **KITRI BoB - Team G0Host

요 약

Wi-Fi가 지원되는 Station들은 이전에 접속하였던 Access Point에 빠르게 접근하거나, Hidden Wifi를 찾기 위해서 주기적으로 자신의 하드웨어 주소 및 신호 세기를 802.11 Probe Request type으로 Broadcast 하여 AP에 전달한다. 이러한 기술로 인해 해당 Station 주위에서 Monitor Mode Packet 수집 장치를 이용하여 Sniffing 할 수 있고, 이를 이용해 모바일 디바이스 사용자가 어느 시점에 특정 공간에 위치해 있었는지를 추정할 수 있다. 본 논문에서는 이러한 부분에 초점을 맞추어 비식별 MAC Address를 이용하여 타깃을 추적할 방안을 제시한다. 또한, 이러한 기능을 응용하여 발생할 수 있는 시나리오들을 제시하고자 한다.

I. 서론

1.1 연구 배경

Wi-Fi는 최근 단거리 통신의 핵심 솔루션이며, 이용 빈도가 증가하는 추세이다. 특히, 최근의 모바일 디바이스는 대부분 Wi-Fi 솔루션을 탑재하여 출시하고 있다. Wi-Fi 통신은 802.11i 보안 표준인 AES 암호화 방식을 이용한 WPA2를 주로 사용하여 암호화된 통신을 하고 있다. 하지만, AP와 Station의 연결을 위해서 필요한 Packet을 주고받는 경우, 이 과정에서 각각의 AP, Station 구분을 위한 고유 구분자들은 암호화가 이루어지지 않고 Plain Text로 전송되기 때문에 이를 수집하여 재식별한다면 각 장치를 소유한 사람을 식별할 수 있다.

이러한 개인정보 수집은 Wi-Fi를 사용하는 모든 모바일 디바이스 사용자에게 이루어질 수 있다. Wi-Fi 모드를 활성화하면 AP 검색을 위해서 Station Device의 인터페이스는 AP를 향해서 Broadcast Probe Request Packet을 전송하고, 해당 과정에서 고유 식별자인 MAC Address가 유출된다. 이를 이용한다면, Signal Strength를 이용하여 사용자를 추적할 수 있고, 모니터 모드 Packet Sniffer가 미리 여러 곳에 설치되어 있다면 특정 MAC Address 소유자의 이동 경로를 추적할 수 있다.

1.2. 기술적 배경

Wi-Fi 는 Access Point(AP)와 이를 이용하는 Station으로 구성된다. AP와 Station은 상호간 Key를 생성하고, 암호화된 802.11 Frame Packet을 전송해 이를 Ethernet 형태로 변환하여 통신한다.

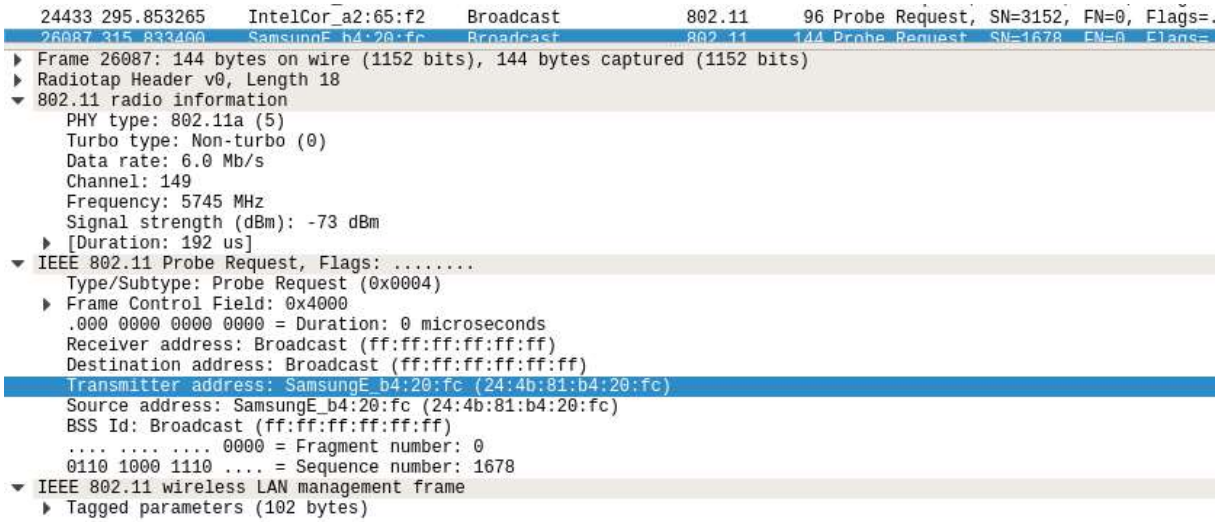


Fig 1. Probe Request Packet Sniffing with Wireshark

802.11 Frame은 그 이용 방식에 따라 관리 프레임, 제어 프레임, 데이터 프레임으로 나뉜다. 여기서 이용하는 것은 관리 프레임의 Subtype인 802.11 프로브 요청 프레임이다. 이는 찾기를 원하는 가까운 AP를 능동적으로 탐색하기 위해(802.11 Scanning), 주어진 주파수 채널 각각에 송출하도록 하는 프레임이다. 해당 프레임은 Frame Control, Duration, Destination Address, Source Address, BSSID, Sequence Control로 이루어진 프레임 헤더를 가진다[2]. 여기서 DA, SA는 각각 Station이 찾기를 원하는 AP의 하드웨어 주소, SA는 해당 Station의 하드웨어 주소를 의미한다. 따라서 SA를 파싱한다면 우리는 Probe Request를 보내는 AP 사용자의 MAC Address를 가져올 수 있다.

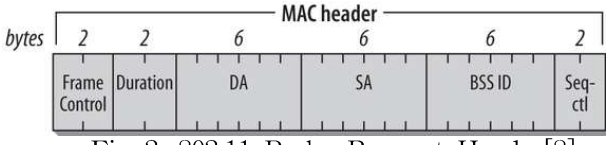


Fig 2. 802.11 Probe Request Header[8]

Monitor Mode 802.11 패킷을 잡기 위해서는 모니터 모드를 이용해야 한다. 우리가 기본적으로 이용하는 모드는 Managed Mode로, 많은 Access Point로 구성된 네트워크에 연결하여 AP와 직접 통신할 수 있도록 하는 모드이다[3]. 반면, Monitor Mode는 주위의 모든 패킷을 수신하여 보는 것으로, 이는 AP와 Station의 연결과는 상관없이 패킷을 수집할 수 있다[4].

이러한 프로그램은 대표적으로. Aircrack-ng의 Airodump-ng가 있다. 해당 프로그램은 Monitor Mode Adapter 주위에서 흘러 다니는 Monitor Mode Packet을 읽어 BSSID, ESSID, 암호화 방식 등을 보여주는 프로그램이다[7].

II. 연구 기술 활용 방안

위의 MAC 주소를 Parsing하는 것을 이용하여 Wireless Tracking을 할 수 있다. 여기서는 두 가지 추적 방안을 제시한다. 첫 번째 방식은 많은 장소에 Monitor Mode Adapter를 설치하고 수집된 MAC Address를 분석하여 인구 분석 통계를 내거나 특정 MAC Address를 가진 대상의 이동 경로를 추적할 수 있는 방법으로, 이를 Wireless Footprint라고 명명한다. 다른 방법은 하나의 Monitor Mode Adapter를 이용하여 단일 대상 혹은 단일 집단을 추적하는 방법으로, 이를 Wireless Tracking이라고 명명한다.

2.1 특정 지역의 유동 인구 추정

광범위한 지역에 Monitor Mode Adapter를 설치하여 여기서 가져온 MAC 주소를 이용해 특정 인물의 이동 경로를 추정하거나, 특정 지역의 유동 인구를 분석할 수 있다.

각 지점에 설치되는 Adapter에서는 802.11 패킷을 Sniffing 하여 Probe Request Frame을 필터링하고, Probe Request Frame에서 Source

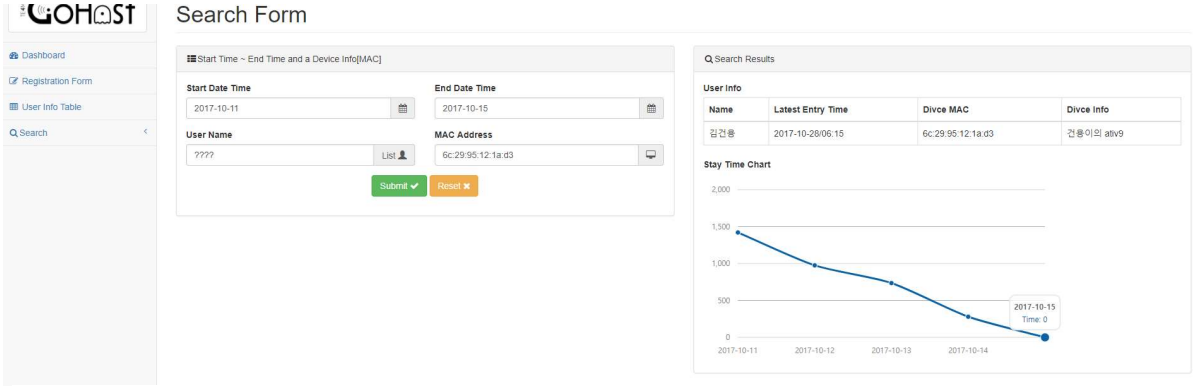


Fig 3. Wireless Footprint with Web GUI (Developed by Team G0Host)

Address를 Parsing하여 중앙 서버로 전송하도록 한다. 802.11 Frame Packet에서 가장 먼저 등장하는 Frame은 Radiotap Header이다. Radiotap Header에서 적절한 Signal Strength를 지정하여 해당 신호 세기보다 세기가 약하다면 valid packet이 아니라고 판단하고 이를 무시하도록 신호 세기에 대한 필터링을 걸어 준다. 또한, TimeStamp를 Parsing 하여 이를 날짜와 시간으로 변환하여 준다. 그 다음으로 오는 Frame은 IEEE 802.11 Frame으로, 여기서는 Probe Request Frame의 Source Address를 Parsing 하여 TimeStamp 정보와 함께 중앙 서버로 전송한다.

중앙 서버에서는 수집되는 Data를 Database에 저장한다. 어느 시점에 어떠한 MAC Address가 해당 위치에 있었는지 알리기 위해 Fig 3과 같이 CLI 혹은 GUI를 이용하여 이를 표현해 줄 수 있어야 한다.

위에서 제시한 기술을 통계적으로 활용할 수 있다. 대규모 집회나 콘서트 등 많은 사람이 드나드는 곳에서 유동 인구수를 추정해야 하는 경우 AP를 제공하여 사용할 수 있도록 하고 Unique MAC Address를 식별하여 얼마나 많은 인원이 그곳에 있었는지를 추정할 수 있다. 이를 상업적으로 활용할 수도 있다. Monitor Mode Adapter를 AP 근처에 설치하여 얼마나 많은 고객이 매장을 방문했고, 인기가 많은 지점과 인기가 떨어지는 지점을 점검할 수 있다. 이를 통해서 점포 위치를 변경하거나 점포를 교체하여 이익을 창출하도록 할 수 있다[9].

2.2 단일 대상에 대한 추적

단일 대상을 추적하고 싶은 경우, Wireless Footprint를 이용하지 않고, 직접 추적할 수 있는 방법이 있다. 기본적으로 Monitor Mode Packet Sniffier는 실시간으로 Packet을 Monitoring할 수 있다. 또한, RadioTap의 Signal Strength를 이용하여 해당 MAC Address 소유자와 자신이 대략 어느 정도 거리에 떨어져 있는지를 파악할 수 있다. 방법은 간단하다. 인파가 많은 곳에서는 끊임없이 수신되는 Source Address가 바뀐다. 그러나 추적하는 대상이 송신하는 Probe Request Packet은 그대로일 것이다. 이를 통해서 후보 그룹 군을 만들어 놓고, 신호 세기를 측정하여 어떤 MAC Address가 우리가 추적하고자 하는 인물의 것인지 찾아낸다. 이렇게 한다면 상대에게 들키지 않는 거리에서 계속해서 추적을 진행할 수 있다[6].

2.1의 방식과 혼합하여 이용하는 경우, 만약 Monitor Mode Adapter가 여러 군데 설치되었다는 가정을 한다면 이를 수사 목적으로 이용할 수 있다. CCTV 등으로 수배 대상자가 어떠한 곳에 있었는지 확인할 수 있으면, 해당 시간에 해당 위치에서 발생한 Probe Request Packet에서 Source Address를 추출한다. 그 후, 예상 도주 경로에서 발생한 Packet들을 확인하고 여기서 중복되는 Source Address를 식별하여 범인의 것으로 추정한다. 범인의 것으로 추정되는 MAC을 식별한다면 2.1의 방식과 2.2의 방식을 혼합하여 대상을 추적할 수 있다.

III. 결론

802.11 Packet을 이용하여 모바일 디바이스의 통신상에 담긴 비식별 개인정보들을 가져올 수 있다. 하지만, 해당 정보만으로는 비식별 정보를 재 식별하기는 어렵다. 본 논문에서는 비식별 개인정보를 가져오는 과정과, 가져온 비식별 개인정보를 재 식별하는 방법. 그리고 재 식별한 개인정보를 활용할 수 있는 시나리오들을 제시하였다. 이를 이용한다면 수사 과정에서 범인 추적이 보다 수월해질 수 있을 것이다. 또한, 여러 통계적 추산을 위한 데이터 축적에도 많은 도움을 줄 수 있을 것이다.

본 논문에 기술한 내용을 이용하면 누구나 쉽게 모바일 디바이스 사용자를 추적할 수 있다. WPA2 방식으로 암호화가 되어 안전한 통신이라고 생각하였던 802.11 Frame이었지만, 디바이스 식별 정보를 쉽게 Parsing 할 수 있었다. 해당 추적을 방지하기 위해서는 공공 장소에서 제공하는 Public Wi-Fi 이용을 자제하고, LTE 및 3G 데이터를 이용하여 통신하는 것을 추천하는 바이다. 혹은, 각 모바일 디바이스의 벤더 사에서 802.11 Probe Request의 발생량을 최소화하도록 패치하는 것 역시 방법이다.

향후 연구에서는 802.11 Probe Request Frame 뿐만이 아닌, 802.11 Frame 전반에 대한 연구를 진행한다. 암호화된 802.11 Frame을 해독하기 위한 키 재생성 및 이를 통한 패킷 Analysis 방법을 제시하고자 한다.

최근 Wi-Fi 이용률이 증가하는 만큼, 무선 네트워크상의 공격이 발생한다면 개인정보 유출 등의 문제가 발생할 수 있다. 이를 예방하기 위해서는 WIPS와 같은 무선 보안 장비들이 필요하지만, 비싼 가격과 In path 운영 방식으로 인한 네트워크 성능 저하 등이 문제가 되어왔다. 해당 문제를 해결하기 위해서 Monitor Mode를 이용해 Out of path에서 동작하도록 하는 패킷 분석기와 이를 응용한 보안 솔루션에 대한 연구를 진행하고자 한다. 이는 향후 무선 네트워크 보안 솔루션 시장의 스펙트럼을 넓힐 수 있고, 기존의 WIPS 가격이 부담되는 곳에서 해당 솔루션을 활용할 수 있기 때문에 공공 Wi-Fi에 대한 보안성이 향상될 수 있을 것이다.

[참고문헌]

[1] MINISTRY OF THE INTERIOR AND SAFETY. “Personal information non - discrimination measures and industrial applications..” https://www.privacy.go.kr/edu/inf/9.do (2017.10.28.)

[2] Korean ICT Glossary. “802.11 Probe Request Frame”.http://www.ktword.co.kr/abbr_view.php?m_temp1=2344. (2017.10.,28.)

[3] Linux Iwconfig Manual https://linux.die.net/man/8/iwconfig

[4] Wireshark Packet Capture, Auditing Tool “Monitor Mode”, https://wiki.wireshark.org/FrontPage (2017.10.28.)

[5] FRENCH, Clark; WHITE, Peter W. Method and apparatus for indexing database columns with bit vectors. U.S. Patent No 5,649,181, 1997.

[6] ABE, Ryuta, et al. Network-Based Pedestrian Tracking System with Densely Placed Wireless Access Points. In: Information Search, Integration, and Personlization. Springer, Cham, 2017. p. 82-96.

[7] Aircrack-ng wireless network analyzer. “airodump”, https://www.aircrack-ng.org/ (2017.10.28.)

[8] sarwiki.“802.11 Network Structures” https://sarwiki.informatik.hu-berlin.de/ (2017.10.28.)

[9] Zoyi Corp. “WALK INSIGHTS” http://walkinsights.com/ (2017.10.28.)

원자력시설 사이버보안 강화를 위한 인적 보안과 시스템 및 정보의 무결성에 대한 평가 방법론 개발

송동훈*

*한국원자력통제기술원

Development of the Evaluation Methodology for Personnel Security and System and Information Integrity to enhance Cyber Security in Nuclear Facilities

Dong-Hoon Song*

*Korea Institute of Nuclear Nonproliferation and Control

요 약

원자력시설 등의 방호 및 방사능 방재 대책법(방사능방재법)에 의거하여 한국원자력통제기술원은 원전 등 원자력시설에 대한 사이버보안 체계를 단계별로 구축해 나가고 있다. 현재 가동 중인 원자력시설은 사이버보안이 설계단계부터 고려가 되지 않아 한국원자력통제기술원의 기술기준 KINAC/RS-015 중 기술적 보안대책을 모두 적용하기 어려운 실정이기에 운영적 및 관리적 보안대책으로 사이버보안 규제 체계를 만들어야 한다. 따라서 본 연구에서는 KINAC/RS-015의 운영적 보안대책 중 인적 보안과 시스템 및 정보의 무결성 기준에 대한 평가 방법론을 개발하였다.

I. 서론

2014년 원전반대그룹의 한국수력원자력 해킹 사건과 신규 사이버 위협 증가로 인해 국가 중요시설인 원전 등 원자력시설에 대한 사이버보안 규제 체계 구축 필요성이 대두되어 방사능 방재법에 따라 원자력시설에 대한 사이버보안 규제를 강화해 나가고 있다. 한국원자력통제기술원은 사이버보안 강화를 위해 기술기준 KINAC/RS-015를 개발하였고, 이에 따라 원자력시설에 단계적으로 사이버보안 규제를 추진해 나가고 있지만, 현재 사이버보안 규제 대상인 원자력시설은 사이버보안을 고려하여 설계되지 않아 기술적 보안대책을 모두 적용하기 어려운 실정이다.

따라서 본 논문에서는 운영 중 원자력시설에 대한 사이버보안 규제 적용을 위해 KINAC/RS-015의 운영적 보안대책 중 인적보안 및 시스템 및 정보의 무결성 총 10가지 항목에 대한 평가 방법론을 개발하였다.

II. 본론

KINAC/RS-015는 총 101가지 보안대책을 제시하고 있으며, 기술적 보안대책(62), 운영적 보안대책(31) 및 관리적 보안대책(8)으로 구성된다.

본 논문에서는 표 1과 같이 운영적 보안대책 중 10가지 항목에 대한 상세 평가 방법론을 제시하고자 한다.

순번	대제목	소제목
1	2.1 인적 보안	2.1.1 사용자 승인
2		2.1.2 업무 종료 혹은 이전
3	2.2 시스템 및 정보의 무결성	2.2.1 결함 제거
4		2.2.2 악성코드 유입 방지
5		2.2.3 모니터링 도구 및 기술
6		2.2.4 보안 경고
7		2.2.5 보안기능 확인
8		2.2.6 소프트웨어 및 정보의 무결성
9		2.2.7 정보의 입력 제한
10		2.2.8 에러에 대한 조치

2.1 인적 보안

인적 보안 항목은 총 2가지 항목으로 구성되며, 사이버보안 규제 대상인 필수디지털자산(CDA, Critical Digital Asset)에 접근 가능한 인력에 대한 사이버보안 규제 요건이다.

2.1.1 사용자 승인

사용자 승인 항목을 평가하기 위해서는 CDA에 접근 가능한 사용자에 대한 승인 절차를 검토하여야 한다. 사용자 승인 절차 내에는 CDA에 접근 가능한 사용자 및 단독 접근 가능자에 대한 식별, 분류, 검토 및 승인 절차, 사용자별 권한 설정 절차, 사용자별 직무 분류 절차의 포함 여부를 검토하고 내용의 적절성을 평가하여야 한다.

2.1.2 업무 종료 혹은 이전

업무 종료 혹은 이전 항목을 평가하기 위해서는 CDA에 접근 가능한 사용자에 대한 변경 절차를 검토하여야 한다. CDA 접근 가능 사용자 변경 절차 내에는 사용자 변경에 따른 통보 절차, 접근 권한 취소 절차, 사용자 변경 및 종료 시에 대한 보안 인터뷰 절차, 권한 취소된 사용자의 CDA 반납 절차 및 권한 취소된 사용자의 CDA 접근 기록 유지 절차의 포함 여부를 검토하고 내용의 적절성을 평가하여야 한다.

2.2 시스템 및 정보의 무결성

시스템 및 정보의 무결성 항목은 총 8가지 항목으로 구성되며, CDA와 포함된 정보의 무결성에 대한 사이버보안 규제 요건이다.

2.2.1 결함 제거

결함 제거 항목을 평가하기 위해서는 CDA에 대한 결함 제거 절차를 검토하여야 한다. 결함 제거 절차 내에는 CDA 취약점 정보 획득 절차, 획득된 취약점에 대한 평가 절차, 취약점 제거 절차, 취약점 제거 후 정상 동작 여부 점검 절차 및 취약점 제거 여부에 대한 검증 절차의 포함 여부를 검토하고 내용의 적절성을 평가하여야 한다.

2.2.2 악성코드 유입 방지

악성코드 유입 방지 항목을 평가하기 위해서는 CDA에 대한 악성코드 탐지 및 제거 절차를 검토하여야 한다. 악성코드 탐지 및 제거 절차 내에는 신규 악성코드 탐지 및 제거 방안에 대한 주기적 갱신 절차, 악성코드 감염 여부 점검 절차, 악성코드 감염 사건에 대한 대응 및 제거 절차, 악성코드 탐지를 위한 자동화 소프트웨어 사용 절차, 악성코드 탐지 및 제거를 위한 사용자 식별, 분류, 검토 및 승인 절차, CDA 접근 가능한 휴대용매체에 대한 주기적 점검 절차, 휴대용매체 인터페이스 통제 절차의 포함 여부를 검토하고 내용의 적절성을 평가하여야 한다.

2.2.3 모니터링 도구 및 기술

모니터링 도구 및 기술 항목을 평가하기 위해서는 CDA에 대한 감시 및 사이버 공격 탐지 절차를 검토하여야 한다. 감시 및 사이버 공격 탐지 절차 내에는 CDA 정상 운전 중의 감시 절차, 사이버 공격, 비정상 및 불법적 연결에 대한 탐지, 분석, 기록 및 관리 절차, 사이버 위기 상황별 감시 절차, 사이버 공격, 비정상 및 불법적 연결 탐지를 위한 자동화 소프트웨어 사용 절차, 보안장비의 사용 및 관리 절차 및 보안장비에 대한 주기적 기능 점검 시 CDA로의 악영향 점검 절차의 포함 여부를 검토하고 내용의 적절성을 평가하여야 한다.

2.2.4 보안 경고

보안 경고 항목을 평가하기 위해서는 CDA에 대한 보안 정보 활용 절차를 검토하여야 한다. 보안 정보 활용 절차 내에는 보안 정보 공유를 위한 신뢰 업체 선정 절차, 보안업체와의 사이버보안 정

보 공유 절차, 사이버보안 정보에 대한 분석 및 평가 절차, 분석된 사이버보안 정보에 대한 내부 처리 절차의 포함 여부를 검토하고 내용의 적절성을 평가하여야 한다.

2.2.5 보안기능 확인

보안기능 확인 항목을 평가하기 위해서는 CDA에 적용된 보안기능 확인 절차를 검토하여야 한다. 보안기능 확인 절차 내에는 CDA 정상 또는 이상 시의 보안기능 점검 절차, 보안기능 점검을 위한 자동화 소프트웨어 사용 절차의 포함 여부를 검토하고 내용의 적절성을 평가하여야 한다.

2.2.6 소프트웨어 및 정보의 무결성

소프트웨어 및 정보의 무결성 항목을 평가하기 위해서는 CDA에 대한 소프트웨어 및 정보의 무결성 검증 절차를 검토하여야 한다. 소프트웨어 및 정보의 무결성 검증 절차 내에는 소프트웨어 및 정보의 불법 변경 탐지 절차, 소프트웨어 및 정보에 대한 물리적 접근 통제 절차, 소프트웨어 및 정보에 대한 주기적 점검 절차, 소프트웨어 및 정보 무결성 침해에 따른 통보 절차의 포함 여부를 검토하고 내용의 적절성을 평가하여야 한다.

2.2.7 정보의 입력 제한

정보의 입력 제한 항목을 평가하기 위해서는 CDA로의 입력 정보 제한 절차를 검토하여야 한다. 입력 정보 제한 절차 내에는 CDA에 정보 입력 가능 사용자에 대한 식별, 분류, 검토 및 승인 절차, 정보 입력 시 입력 값에 대한 사전 정의 절차, 정보 입력 값에 대한 검증 절차의 포함 여부를 검토하고 내용의 적절성을 평가하여야 한다.

2.2.8 에러에 대한 조치

에러에 대한 조치 항목을 평가하기 위해서는 CDA 고장에 따른 조치 절차를 검토하여야 한다. 고장에 따른 조치 절차 내에는 CDA 발생 가능 고장 유형에 대한 정의 절차, CDA 고장 발생에 따른 고장메시지 정의 절차, CDA 고장 발생에 따른 고장메시지 내 불필요 정보 포함 여부 점검 절차의 포함 여부를 검토하고 내용의 적절성을 평가하여야 한다.

III. 결론

원자력시설에 대한 사이버보안 강화를 위해 한국원자력통제기술원은 KINAC/RS-015를 규제 기준으로 제시하고 있으며, 본 논문에서 연구된 KINAC/RS-015의 운영적 보안대책은 운영 중인 원자력시설에 사이버보안을 적용하기 위한 규제 기준이다.

본 논문에서는 운영적 보안대책 중 인적 보안과 시스템 및 정보의 무결성 총 10개 항목을 평가하기 위한 최소 검증 방법을 제시하였다. 한국원자력통제기술원은 추가적인 연구를 통해 운영적 보안대책 뿐만 아니라 기술적 및 관리적 보안대책에 대한 평가 방법론을 개발하여 원자력사업자의 사이버보안 이행 결과를 검증 및 평가해 나갈 예정이다.

[참고문헌]

[1] 한국원자력통제기술원, “원자력시설 등의 방호 및 방사능 방재 대책법령집”, 2017.

[2] 한국원자력통제기술원, “원자력시설등의 컴퓨터 및 정보시스템 보안 기술기준”, KINAC/RS-015, 2016.

[3] U.S. NRC, “Cyber Security Programs for Nuclear Facilities”, Regulatory Guide 5.71, 2009.

제어시스템 테스트베드를 활용한 기술적 보안조치 적용 시 고려사항 및 대처방안 연구

임현종*, 송동훈*, 신익현*
*한국원자력통제기술원

A Study on the Considerations and Alternative methods for applying the technical security controls by utilizing control system testbed

Hyun-Jong Lim*, Dong-Hoon Song*, Ick-Hyun Shin*
*KOREA INSTITUTE OF NUCLEAR NONPROLIFERATION AND CONTROL

요 약

디지털 기술의 발전으로 우리나라 기반시설 대부분이 점차 디지털화 되어가면서 이에 따른 사이버보안 적용이 필수적으로 요구되고 있다. 특히, 원자력시설은 방사능방재법 개정에 의해 사이버보안 이행에 관한 기준(KINAC/RS-015)에 따라 사이버 보안조치가 적용 및 이행되어야 한다. 하지만 실제 원자력시설에 사이버보안 기술을 적용하는데 많은 어려움이 있다. 이에 본 논문에서는 사이버보안 연구를 위해 구축한 제어시스템 테스트베드를 통해 기술적 사이버 보안조치를 이행하는데 발생할 수 있는 한계점 및 고려사항에 대해 서술하고 그에 대한 대안적 조치 방안에 대해 알아보고자 한다.

I. 서론

오늘날 우리나라 기반시설은 디지털 기술의 발전으로 아날로그 형태의 시스템 구조에서 디지털 기술들이 점차 적용되고 있는 실정이다. 이에 따라 기반시설의 대부분의 시스템들은 운영적 측면에서 보다 높은 효율성과 편의성을 확보할 수 있었다.

하지만 이러한 디지털 시스템의 도입은 IT 분야에서 발생하는 각종 사이버 위협이 기반시설의 주요 시스템에서도 발생되는 단점을 가지게 되었다. 특히, 원자력시설의 경우 실제 2010년 이란 원자력시설내 우라늄 원심분리기 가동을 중단시킨 스턱스넷(Stuxnet)을 기점으로 점차 사이버 위협이 증대되고 있어 원자력시설을 보호하기 위한 사이버보안 기술 적용이 필수적으로 요구되고 있다[1].

이에 국내에서는 [원자력시설 등의 방호 및 방사능 방재 대책법령] 개정에 따라 원자력통제기술원(KINAC)에서 KINAC/RS-015 [원자력시설등의 컴퓨터 및 정보시스템 보안 기술기준]을 마련하여, 원자력시설의 사이버보안대책 수립 및 이행 상태에 대한 적합성 등을 심·검사하고 있다.

이에 본 논문에서는 연구를 위해 구축한 제어시스템 테스트베드의 전반적인 통신망 네트워크 구성 형태와 이러한 제어시스템에 적용되는 기술적 사이버 보안조치들의 이행방안 마련 시 한계점 및 고려사항 등에 대해 서술하고 앞으로의 대안 방안에 관해 알아보고자 한다.

II. 본론

본 장에서는 주요기반시설에 사용되고 있는 제어시스템들 중 연구를 위해 구축한 테스트베드의 통신망 네트워크 관련 설명과 KINAC/RS-015 기술기준 중 기술적 사이버 보안 조치에 관한 이행 방안 마련 시 고려사항에 대해 알아본다.

2.1 원전 제어시스템 테스트베드

제어시스템에 대해서는 다양한 제어시스템 중에서 KINAC에 구축되어 있는 원자력시설 제어시스템 테스트베드를 기준으로 사이버위협에 가장 영향을 받을 것으로 예측되는 통신망 네트워크에 대해 설명하겠다.

일반적으로 원자력시설은 안전 계통과 비안전 계통으로 나눠지며, 제어시스템은 단일 고장 시 빠른 정상 운행 복구를 위해 이중화 구조로 되어있다. 현재 구축된 테스트베드의 경우 PLC(Programmable logic controller)와 DCS(Distributed control system)시스템으로 시스템 내 각각의 모듈들은 각 통신망 네트워크로 연결된다.

2.1.1 Programmable logic controller

PLC의 경우, 각 모듈들은 Profibus FDL 프로토콜을 사용하는 1대1 통신 및 1대N 통신, N대N 통신으로 통신 라인이 구성되며, 각 통신 네트워크를 통해 프로세서, 통신, 입출력 모듈 및 EWS 소프트웨어와 연결된다.

2.1.2 Distributed control system

DCS의 경우, 시스템 감시 제어 시스템인 OIS(Operator Interface System), 시스템 구성 및 제어로직 설정 시스템인 EWS(Engineering Workstation System), 각종 Process 데이터를 관리 수행하는 FCS(Field Control Station) 및 네트워크 장치로 구성된다. 전체 네트워크를 구성하는 통신망은 Modbus TCP/IP 프로토콜 기반으로 제어망과 정보망으로 분리되어 제어 신호의 안정성과 신뢰성을 보장하도록 구성되며, 100 Mbps Ethernet 방식을 사용한다.

2.2 기술적 보안조치

전자적 침해행위에 대한 원자력시설의 컴퓨터 및 정보시스템 보안에 대한 국내 기술기준인 KINAC/RS-015에 의하면 사이버 보안조치 이행 사항에 대해 기술적, 운영적, 관리적 측면으로 나누어 적용 사항을 기술하고 있다.

그 중 본 논문에서는 테스트베드에 적용하고자 하는 기술적 보안조치 사항에 대해 알아보고자 한다. 기술적 보안조치 사항은 접근 통제, 감사 및 책임, 시스템 및 통신 보호, 식별 및 인증, 시스템 보안 강화의 통제 항목으로 총 62개의 점검 기준이 분류되어 있다[2][표1].

[표1] 기술적 보안조치 사항 목록

통제항목	점검 기준(개)	설명
접근통제	19	인가된 사용자가 필수디지털자산에 접근하여 인가된 행위만이 수행되도록 보장
감사 및 책임	11	사이버보안계획의 사항들을 감사하기 위해 책임 및 관리되도록 보장
시스템 및 통신 보호	19	시스템 및 통신설비에 악영향을 유발 할 수 있는 비인가 접근 위험으로부터 보장
식별 및 인증	8	주요 네트워크 사용자, 호스트, 응용프로그램, 서비스 등에 대한 식별 및 인증 보장
시스템 보안 강화	5	필수디지털자산을 불법 접근 및 불법 사용으로부터 보호하기 위한 기술적 보장

2.3 기술적 보안조치 적용 시 고려사항 및 대처 방안

제어시스템의 주요 특징은 원자력의 제어를 위해 주어진 특정 시간 내에 인터럽트 처리할 수 있는 실시간 운영체제(RTOS)를 기반으로 구성되어 있으며, 이를 통해 다수의 task가 일정 시간 내에 정해진 이벤트를 순차적으로 수행되어야 한다.

또한 원자력시설의 통신망은 일반 네트워크 망과 달리 폐쇄망으로 이루어져 있으며, 외부망과는 별도의 접점을 통해 정보 교환이 가능하다. 때문에 다음의 통제 항목들의 보안조치에 관한 이행 방안을 마련 시, IT분야의 일반 시스템과는 다르게 구성된 제어시스템의 특징들을

고려하여 더 특화된 시각에서 그 보안조치 사항을 마련하여야 한다.

2.3.1 제어시스템 내 운영체제에서의 고려사항

일반적으로 원자력 시설에서의 제어시스템의 통신망 구성은 폐쇄망 형태로 구성되며 별도의 보안절차 없이는 쉽게 접근할 수 없도록 되어있다. 실제로 구축된 제어시스템 테스트베드 내의 DCS의 RTOS인 “VXWORKS”에 접근하여 점검을 수행하였지만, 계정 확인 등의 일반적인 기능 확인 외에는 어떠한 설정 변경 및 삭제가 불가능하였다. 이는 가동되는 시스템에 불필요한 부하를 주지 않기 위해 제작 단계에서의 설정 외에는 대부분의 기능 변경을 하지 못하도록 하였기 때문이다. 이처럼 제어시스템의 사이버보안을 적용하기 위해서는 제어기기의 설계 단계에서부터 사이버보안의 기술적 보안조치 적용이 고려되어야 한다. 또한, 현 운영되는 제어기기의 기술적 보안조치 적용의 어려움을 보완할 수 있도록 제어기기 보안 관리 지침, 접근통제 및 관리대장 등의 절차적 점검 방법의 수립이 필요하다[그림1].

[그림1] 접근통제 이행 항목 점검 방법(예시)

점검사항	수립된 접근통제 정책 및 절차에 따라 필수디지털자산에 대한 접근통제 이행 여부 점검	분류	절차적
구분	내용		
절차적 점검	① 접근통제 이행 및 절차 수립 여부 확인 - 필수디지털자산에 대한 접근 및 사용가능한 사용자를 사전 정의하고, 차등 권한 및 특권의 부여 적절성 확인 - 필수디지털자산의 특정 기능을 정의하고, 해당 기능에 대한 사용자별 접근 및 사용 권한을 차등 부여 적절성 확인 - 시스템에 악영향 주는 기능 문서화 여부 확인 - 악영향을 미치는 기능 수행 시 두 명의 관리자 승인 여부 확인 - 접근통제 지침 적용 시, 필수디지털자산에 미치는 영향평가 결과를 확인하고 대안 조치 수행 여부 확인		

2.3.2 통신망 네트워크에서의 고려사항

네트워크 통신망의 보안 대책에 대해서는 현재 구성된 시스템의 통신망 분석을 통해 예상되는 취약점을 파악하거나 구축된 통신망의 배포된 취약점을 파악하여 대처방안을 수립하는 것이 중요하다.

일반적으로 profibus 프로토콜은 각 시스템 모듈 간 Broadcasting 방식으로 정보를 전달하며 특정 수신 데이터에 부합하는 장비에서만 수신 가능하도록 Master 장비에 의해 제어된다[3]. PLC와 EWS 사이의 정보 교환은 데이터가 암호화되어 전송되며, serial cable을 통해 자료의 업로딩을 수행한다. 각 모듈마다 자가 진단 기능을 보유하고 있지만, 데이터 정보의 변형 가능성에 대한 판단을 할 수는 없다. 이처럼 대부분의 PLC 간 연결되는 통신 프로토콜은 기밀성 유지를 위한 암호화 등의 보안 조치가 적절히 이루어져있지 않기 때문에, Man-in-the-Middle 공격 등의 사이버 위협에 취약한 부분이 있다[4].

DCS에 적용된 MODBUS프로토콜은 TCP/IP 기반의 프로토콜로 적용되며, 제어망과 정보망을 분리하여 네트워크를 구성한다. 일반적으로 MODBUS는 오래된 프로토콜로 통신 프로토콜 Specification이 모두 공개되어 있으며, 산업 통신 프로토콜로서 많은 시스템에 적용되어 있는 대표적인 통신 프로토콜이다. 하지만 최근 대두되고 있는 사이버보안을 고려하여 설계된 프로토콜이 아니다보니 사이버 위협에 대해 취약한 부분을 가지고 있다. 특히, Master와 Slave 간 정보 전송 시 그 데이터의 변형 여부를 증명하는 것이 불가능하며, 각 장치 간의 인증 과정이 반영되어 있지 않기 때문에 인가되지 않은 데이터의 유입이나 Man-in-the-Middle 공격 등에 대한 취약성을 가지고 있다[5].

이러한 산업 제어시스템은 예전에 구성되어 많은 정보들이 공개된 반면 보안적 측면이 적절히 적용되지 않았기 때문에 Master 장비의 물리적 접근만 가능하다면 쉽게 제어시스템에 위협을 가할 수 있을 것이라 생각된다. 산업 제어시스템의 피해 사례도 대부분 이동식 미디어 및 외부 하드웨어를 통한 악성코드 침투 공격이 가장 많이 시도되고 있다. 하지만 산업 현장의 특성상 가격, 가용성, 편의성 등이 더욱 우선시 되고 있으며, 가동 중인 시스템을 변경하기도 쉽지 않다. 때문에 우선적으로 보완해야 할 사항이 물리적 접근 통제를 통한 일차적인 보안 대책을 세워야 한다. 점진적으로는 제어망 내의 데이터의 제한적 이동 통제를 위해 제어시스템의 운영에 미치는 영향을 최소하는 방향

으로 방화벽 및 단방향 전송 장치를 사용할 수도 있다. 또한, 이러한 대안적 보안적 조치 적용에 앞서 실제 설계 단계 중인 제어시스템 체계 내에 사이버 보안을 적용하여 구축되어야 하는 것이 가장 효과적인 제어시스템 보안 조치일 것이다.

III. 결론

구축된 제어시스템 테스트베드를 기준으로 기술적 보안조치에 대한 적용을 통해 그 적용 가능성과 문제점에 대해 알아보았다. 제어시스템은 일반 시스템과는 다르게 많은 제한 사항들이 있었으며, 그에 대한 많은 대안적 조치들이 요구된다. 실제 운영되는 원자력시설의 제어시스템은 이러한 조치들을 이행하는데 필요한 추가 장치 도입에 대해 더욱 민감하게 작용될 요지가 많다. 때문에 가장 이상적인 사이버보안 대책은 설계 단계에서부터 제어시스템 내 사이버보안조치가 적용되는 것이다. 또한, 현 원자력 시설에 기술적 보안조치를 적용하기 위해서는 원자력 시설의 시스템에 대한 가용성과 사이버보안 모두가 절충될 수 있는 대안을 알아보고, 발생될 수 있는 취약성에 대해 적절한 조치를 마련하여야 한다.

ACKNOWLEDGMENT

This work was supported by the Nuclear Safety Research Program through the Korea Foundation Of Nuclear Safety(KoFONS), granted financial resource from the Nuclear Safety and Security Commission (NSSC), Republic of Korea. (No. 1305034)

[참고문헌]

[1] D.Y.Kim, “Vulnerability Analysis for industrial Control System Cyber Security”, J. of the Korea Institute of Electronic Communication Sciences, vol 9, 2014

[2] 한국원자력통제기술원, “원자력시설등의 컴퓨터 및 정보시스템 보안 기술기준”, KINAC/RS-015, 2016.

[3] HU Li-kun, “Analysis of Profibus FDL”, Control and Instruments in Chemical Industry, 2005-02

[4] Young-Jin Kim, “A Study on the Secure Plan of Security in SCADA Systems”, Jonornal of the Korea Institue of information Security & Cryptology, 19(6), Dec. 2009

[5] MODBUS, 'MODBUS APPLICATION PROTOCOL SPECIFICATIOIN v1.1b3, April 26, 2012

PPTP 인증 프로토콜 취약점 동향 분석

손민우[1], 이상익[1], 배윤겸[1], 김한기[2], 김지훈[2], 김종성[1,2]

[1]국민대학교 정보보안암호수학과,

[2]국민대학교 금융정보보안학과

PPTP Authentication Protocol Trend Analysis

Min-woo Son[1], Sang-ick Lee[1], Yun-gyeom Bea[1], Han-gi Kim[2], Jihun Kim[2],

Jongsung Kim[1,2]

[1]Dept. of Information Security, Cryptology and Mathematics, Kookmin University.

[2]Dept. of Financial Information Security, Kookmin University.

요 약

PPTP는 Point-to-point protocol (PPP, 지점 간 프로토콜)를 캡슐화하기 위해 Microsoft의 주도 하에 만들어져 1997년 7월에 RFC2637로 등록되었다. 현재 많은 VPN 서비스 제공 업체와 기업 등이 PPTP를 사용하고 있다. 본 논문은 PPTP에서 사용하는 인증 프로토콜 PAP, MS-CHAP v1, MS-CHAP v2, PEAP-MS-CHAP v2의 정의와 취약점을 설명한다.

I. 서론

Virtual Private Network (VPN, 가상사설망)은 공중전화망 상에 사설망을 구축하여 공중망을 사설망처럼 사용할 수 있도록 하는 서비스로, 저렴한 비용으로 전용회선을 구축할 수 있다. VPN은 공중망을 사설망처럼 사용하므로 이에 대한 기밀성 보장이 필요하기 때문에 프로토콜의 취약성 분석이 필요하다.

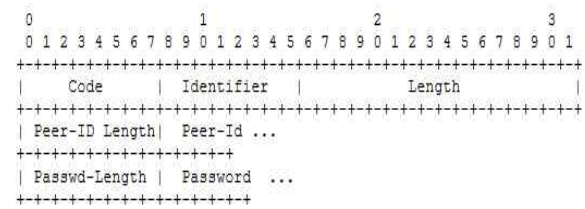
본 논문은 VPN 지원에 사용되는 Point-to-Point Tunneling Protocol (PPTP) 프로토콜의 세부 프로토콜인 PAP, MS-CHAP v1, MS-CHAP v2, PEAP-MS-CHAP v2의 개념과 취약성을 다룬다. 2, 3, 4장에서는 각각 PAP, MS-CHAP v1, MS-CHAP v2의 정의와 발견된 취약점을 살펴보며, 5장에서는 PEAP-MS-CHAP v2의 정의, 인증 절차, 취약점을 다룬다. 6장에서는 PPTP 인증 프로토콜의 취약점을 강조한다.

II. PAP

2.1 프로토콜의 개념

Password Authentication Protocol (PAP, 비밀번호 인증 규약)는 2-way Handshaking 프로토콜을 이용하며 Link Establishment[1] 단계가 완료되면 인증이 확인되거나 연결이 종료될 때까지 ID/Password 쌍이 인증자에게 반복적으로 전송된다.

2.2 취약점



[그림 1] PAP Authenticate-Request

[그림 1]은 PAP가 Authenticate-Request 하는 과정으로, ID와 Password를 평문으로 전송하여 쉽게 도청당할 수 있으며, 이를 이용하여 무단으로 접근이 가능하다. 또한 Link Establishment가 완료되면 반복적으로 클라이언트에게 인증의뢰가 가능하기 때문에 전사공격에 취약하다[1].

1) 2개의 인접한 스테이션 사이에서 일어날 수 있는 매체를 신뢰할 수 있게 만들고 정보를 교환할 수 있는 링크를 확립하는 것

III. MS-CHAPv1

3.1 프로토콜의 개념

Microsoft-Challenge Handshake Authentication Protocol v1 (MS-CHAP v1, 챌린지-핸드셰이크 인증 규약v1)는 CHAP과 유사한 인증 프로토콜이다. 이 프로토콜은 먼저 원격 액세스 서버가 세션 ID와 임의의 챌린지 문자열로 구성된 챌린지를 원격 클라이언트에 보낸다. 그 후 클라이언트는 챌린지 문자열, 세션 ID 및 MD4 해시 암호의 사용자 이름과 해시를 반환하여 인증한다.

3.2 취약점

MS-CHAP v1에서는 다음과 같은 취약점이 발견되어 현재는 사용되고 있지 않다.

- 관리자의 패스워드 변경에 대한 인코딩 과정에 사용되는 알고리즘이 암호학적으로 취약하다.
- 오직 일방향 인증만이 가능하기 때문에 원격으로 접근하는 클라이언트가 기관의 서버를 호출하는지 혹은 기관으로 위장한 서버를 호출하는지를 확인할 수 없다. 따라서 클라이언트는 위장한 서버에 인증할 수 있기 때문에 보안에 취약하다.
- 사용자 패스워드를 바탕으로 하는 40 비트의 암호화키만을 사용하여 암호화를 한다. 이러한 비트 수가 적은 사용자 패스워드 기반 암호화키만을 사용하기 때문에 안전성이 떨어진다.
- 데이터 송수신에 하나의 암호화키만 사용되기 때문에 키 유출에 대한 위험성이 있다[2].

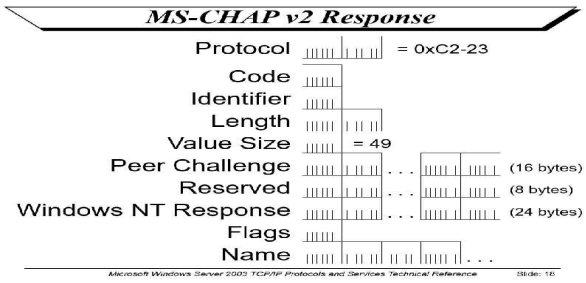
IV. MS-CHAPv2

4.1 프로토콜의 개념

Microsoft-Challenge Handshake authentication protocol v2 (MS-CHAPv2, 챌린지-핸드셰이크 인증 규약v2)의 약칭이다. 주요 사용 알고리즘으로는 DES, SHA-1, MD4 등이 있다. MS-CHAP v2는 위에 기술한 MS-CHAP v1의 취약성을 보완하여 나온 프로토콜이다. MS-CHAP v2는 다음과 같은 방법으로 MS-CHAP v1의 취약성을 보완했다.

- 일방향 인증만이 가능하여 클라이언트가 서버를 확인할 수 없는 MS-CHAP v1과 달리 MS-CHAP v2는 상호 인증이 가능하다. 클라이언트는 사용자 패스워드를 통해 인증된 서버의 확인을 받는다.
- 사용자 패스워드를 기반으로 하는 암호화키만으로 암호화를 하는 MS-CHAP v1과 달리 MS-CHAP v2의 암호화 키는 사용자 패스워드와 임의의 챌린지 문자열을 바탕으로 만들어진다. 따라서 사용자가 같은 패스워드를 사용하더라도 다른 암호화키로 암호화를 진행한다.
- 데이터 송수신에서 하나의 암호화키를 사용하는 것이 아닌 각각 다른 암호화키를 생성하여 사용된다[3].

4.2 취약점



[그림 2] MS-CHAP v2 Response

[그림 2]는 MS-CHAP v2의 응답 과정의 구조이다. 클라이언트의 응답은 다음과 같은 과정으로 진행된다.

① 인증자는 16 바이트의 챌린지(CS)를 보낸다.

② 클라이언트는 16 바이트의 챌린지(CC)를 만든다.

③ 클라이언트는 CS, CC 그리고 사용자 이름($UNAME$)을 해시하여 8 바이트의 블록(C)으로 만든다.

$$C = SHA1(CS, CC, UNAME)$$

④ 클라이언트가 MD4 알고리즘을 사용해 패스워드를 해시한다. (해쉬값 : H)

⑤ 클라이언트는 H에 5 바이트를 0으로 패딩하여 21 바이트의 블록을 만들고 이를 다시 3개의 DES 키 (k_1, k_2, k_3)로 나눈다.

$$P = MD4(\text{패스워드})$$

$$P \mid 00000 = k_1 \mid k_2 \mid k_3$$

⑥ 클라이언트는 k_1, k_2, k_3로 각각 블록 C를 암호화하여 응답(R)을 만든다.

$$R = DES(k_1, C) \mid DES(k_2, C) \mid DES(k_3, C)$$

⑦ 클라이언트는 R, C 그리고 $UNAME$을 보낸다.

응답 과정의 알고리즘에는 다음과 같은 취약점이 있다. 첫 번째로, 비밀번호의 MD4 옆의 모든 챌린지 및 응답 요소가 전송 과정에서 쉽게 계산될 수 있다. 따라서 해시값이 쉽게 계산되기 때문에 중간자 공격에 취약하며 제 3자가 인증할 수 있다. 두 번째로, H에 5바이트를 0으로 패딩하기 때문에 실제 마지막 DES키(k_3)의 비트는 16비트가 된다. 따라서 k_3의 키공간이 2^{16}이므로 키의 암호학적 안전성을 보장하지 않는다[4].

그 외에도 다음과 MS-CHAP v2에서는 다음과 같은 취약점이 발견되었다.

- 무선 액세스 지점의 스푸핑 방지가 되지 않는다.

- 권한이 없는 서버가 보안 수준이 가장 낮은 인증 방법으로 협상하게 한다.

- 공개키 암호를 통해 생성된 TLS 키를 사용하지 않는다.

- 종단 간에서 암호화를 제공하지 않는다.

- 사전 공격, 전사 공격, 재생 공격(replay attack)에 취약하다.

- 서버가 제공한 인증서에 대한 클라이언트의 신뢰를 요구하지 않는다[5].

V. PEAP-MS-CHAPv2

5.1 프로토콜의 개념

Protected Extensible Authentication Protocol (PEAP, 보호 확장성 인증 프로토콜)은 서버 인증만을 통해 생성한 클라이언트와 서버 간 보안 터널을 통해 ID, 패스워드, 인증서 같은 기존 암호 기반 프로토콜의 인증 데이터를 안전하게 전송할 수 있도록 하는 인증 프로토콜이다[6].

PEAP 인증절차는 클라이언트와 인증 서버 간 암호화 된 TLS 터널을 만드는 단계와 그 후 서버측의 PKI 인증서와 공개키 인증서를 사용해 서버를 인증하는 단계로 이루어진다.

5.2 TLS 암호화 채널

무선 PEAP 클라이언트에 대해 TLS 암호화 채널을 만드는 방법은 다음과 같다.

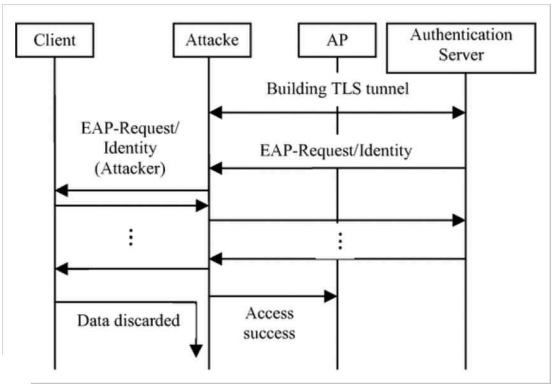
① PEAP 클라이언트가 NPS(Network Policy Server)를 실행하는 서버와 무선 엑세스 지점과 연결한다. PEAP 클라이언트와 엑세스 지점 간에 보안 연결이 만들어지기 전에 개방 시스템 또는 공유키 인증을 제공한다.

② 클라이언트와 액세스 지점 간에 연결이 성공적으로 설정된 후 액세스 지점과 TLS 세션이 협상된다.

③ 무선 PEAP 클라이언트와 NPS 서버 간에 컴퓨터 수준 인증이 성공적으로 완료된 후 클라이언트와 서버 간에 TLS 세션이 협상된다. 이 협상 중에 파생된 키는 사용자가 조직 네트워크에 연결하도록 허용하는 네트워크 액세스 인증을 비롯하여 이후 모든 통신을 암호화하는 데 사용된다[5].

5.3 취약점

PEAP는 필수 인증 클라이언트를 시행하지 않으므로 중간자 공격에 취약하다.



[그림 3] 중간자 공격 과정

중간자 공격 과정은 다음과 같다[그림 3].

① 공격자와 인증 서버는 키 합의를 만들어 터널 키를 만든다.

② 공격자는 AP로 위장하여 PEAP-MS-CHAPv2 프로토콜 패킷을 통해 합법적인 클라이언트가 AP에 액세스하도록 유도하고 클라이언트 패킷을 캡처하여 인증 서버로 전달한다. 동시에 공격자는 인증 서버에서 설정한 개인키 복호화 인증 서버에서 얻은 데이터 패킷을 합법적인 클라이언트에 보낸다.

③ 합법적인 클라이언트가 인증을 완료하면 공격자는 인증 서버에서 자체 인증을 받고 클라이언트는 AP로부터 응답을 받지 않기 때문에 결국 연결이 끊어진다[7].

VI. 결론

본 논문은 PPTP 인증 프로토콜에 대한 정의와 취약점에 대해 기술하였다. PPTP 인증 프로토콜은 기존의 취약점을 보완하면서 점점 진화해 왔지만 새로운 취약점 또한 계속 발견되고 있다. IOS 운영체제는 10버전부터 PPTP를 지원하지 않지만, iptime 공유기나 안드로이드, Windows 등 여러 운영체제에서는 여전히 PPTP 방식을 지원하고 있다. PPTP의 취약점은 지속적으로 발견되고 있기 때문에 L2TP, IPSec와 같이 PPTP보다 안전한 프로토콜을 사용해야 한다.

[참고문헌]

1] 패킷 통신 과정, RFC 1334, Brian William Allen and Lloyd Simpson, October 1992

[2] MS-CHAP v1 인증과정, https://technet.microsoft.com/en-us/library/cc957985.aspx

[3] MS-CHAP v2 인증과정, https://technet.microsoft.com/ko-kr/library/dd197526(v=ws.10).aspx

[4] MS-CHAPv2 응답과정 취약점, https://blogs.technet.microsoft.com/srd/2012/08/20/weaknesses-i

n-ms-chapv2-authentication/

[5] MSCHAPv2 취약점 정리, PEAP 인증과정, https://technet.microsoft.com/ko-kr/library/cc754179(v=ws.11).aspx

[6] PEAP 정의, http://terms.naver.com/entry.nhn?docId=857176&cid=42346&categoryId=42346

[7] PEAP 공격과정, http://ieeexplore.ieee.org.proxy.kookmin.ac.kr:8010/document/6291453/

개인 건강정보 보안을 위한 소프트웨어 설계방안

- A사의 개인 건강정보 활용 시스템을 중심으로

강기원

고려대학교 컴퓨터정보통신대학원 소프트웨어보안학과

Software Design Method for Personal Health Information Security

- Focusing on the Personal Health Information Utilization System of A Company

Kee-Won Kang

Dept of Software Security Engineering, Korea University Graduate School of Computer & Information Technology.

요 약

건강한 생활을 오래도록 유지하려는 인간의 노력은 의료서비스를 발전시키며 헬스 케어(Health Care) 분야를 새로운 블루오션으로 떠오르게 하고 있다. 수많은 기업들이 참여하고 있으며 국내에서도 다양한 분야에서 활발히 연구되고 있다. 그러나 의료서비스의 특성상 의료전문가의 참여의 중요성이 강조되는 경향이 있고 그 영향으로 의료서비스 발전 속도에 비해 건강정보는 극히 개인적인 정보임에도 보안 취약점(Vulnerability)이 금융정보 등 다른 분야 보다 소홀히 다루어지는 경향이 있다. 이에 따라 건강정보와 관련한 소프트웨어 구현 시 사이버 보안 위협(Threat) 및 보안취약점(Vulnerability) 해소를 위해 좀 더 안전한 설계를 할 수 있는 방안을 제시하고자 한다.

I. 서론

현대인의 과도한 스트레스 및 피폐한 정신건강으로 인하여 개인적, 사회적 문제가 발현되고 있다. A사는 웨어러블 디바이스를 이용하여 개인 건강정보인 심박수, 혈압, 스트레스 지수 등을 수집하고 수집된 Data를 분석하여 스트레스 수치를 낮추기 위해 챗봇을 활용한 채팅, 음악, 영화, 운동 등 건강정보에 따른 개인 맞춤형 콘텐츠 추천 등의 서비스를 제공하고 향후 건강데이터 이력 데이터를 활용하여 의사 상담 추천, 약품 재구매 시기 알림 등의 서비스를 제공하고자 시스템을 구축하려 한다. Software System 설계는 기능 요구사항이 중요시되던 과거에서 보안 요구사항을 포함하는 시대적 변화에 따라 설계 시점부터 안전한 Software를 구상하여야 하며, S/W 개발단계별 결함 수정비용을 보면 단계별로 오류 발견 시 수정 비용은 기하급수적으로 증가하는 것을 알 수 있다.

<표1> SW 개발단계별 결함 수정비용 분석

구분	설계 단계	코딩 단계	통합 단계	베타 제품	제품 출시
설계 과정	1배	5배	10배	15배	30배
코딩 과정		1배	10배	20배	30배
통합 과정			1배	10배	20배

출처 : NST. The Economic Impacts of Inadequate Infrasturcture for Software Testing, (2002년 5월)

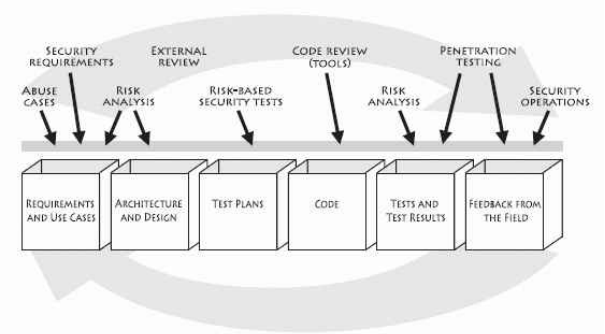
S/W 설계시점부터 기능 및 보안 요구사항을 만족하는 S/W를 설계하여야 향 후 수정비용을 감소시켜 결과적으로 적은 비용으로 운영이 가능한 시스템을 보유하게 된다.

II. 관련연구

2.1 7-TouchPoints

미국 국토안보부(United States Department of Homeland Security)에서 채택한 7-TouchPoints는 현재의 개발방법론을 유지하면서 보안을 강화 할 수 있는 대안으로 적절하게 활용이 가능하다.

<그림1> 7-Touchpoints



출처 : http://www.swsec.com/resources/touchpoints/

7-TouchPoints[7]는 기존 방법론의 주요 요소에

- Code Review(코드 리뷰)
- Architectural risk analysis(아키텍쳐 위험 분석)
- Penetration testing(침투 테스트)
- Risk-based security tests(위험 기반 보안 테스트)
- Abuse cases(악용 사례)
- Security requirements(보안 요구사항)
- Security operations(보안 작업-모니터링)
- External Review(외부 리뷰)

를 활용하여 보안성을 강화하는 방법이다.

7-TouchPoints의 보안 요구사항은 <그림2> 소프트웨어 보안 요구사항의 유형에 요소를 포함하여야 한다.

<그림2> 소프트웨어 보안 요구사항



출처 : Official ISC2 Guide to the CSSLP CBK Second Edition

2.2 개인 정보보호 요구사항 과 개인정보 테스트

개인 정보보호 요구사항을 보안 요구사항에 포함시키고 개인정보 테스트(Privacy Testing)는 침투 테스트와 함께 활용하면 더 나은 개인정보보호를 보장할 수 있다.

개인 데이터를 처리하는 소프트웨어의 경우 개인정보 테스팅은 테스트 계획의 일부여야 한다[2].

- 개인 정보 보호에 영향을 미치는 조직 정책 제어의 검증 포함.
- 또한 네트워크 트래픽 및 통신 모니터링을 포괄.
- 개인 정보가 공개되지 않도록 보증.
- 개인 정보가 수집 될 때 통지 및 부인의 적합성에 대한 테스트 수행.
- 자녀 및 보호자 통제 연령과 같은 보호 자료의 개인 정보 보호 테스트.

최소한 위의 5가지 내용은 포함하여야 한다.

2.3 암호화 및 시큐어 코딩

데이터에 대한 RSA, AES, SHA등 강한 암호화가 필요하고 개발단계부터 시큐어 코딩을 적용한 보다 세심한 방어가 이루어져야 한다.

국제적으로 기관별, 분야별로 다양한 시큐어 코딩 규칙이 존재한다. 영국 자동차산업 신뢰성 협회에서 개발한 MISRA-C, 미국 카네기 멜론 대학 내의 Software Engineering Institute 에서 개발한 CERT C, CERT Java[3] 등이 있다.

국내의 경우 행정안전부의 'SW 개발보안 가이드'를 발표하여[4] '소프트웨어 개발보안(시큐어코딩)'을 의무화하는 정보시스템 구축•운영 지침 개정안을 마련하였으며 최근에는 의무화 되어가고 있다.

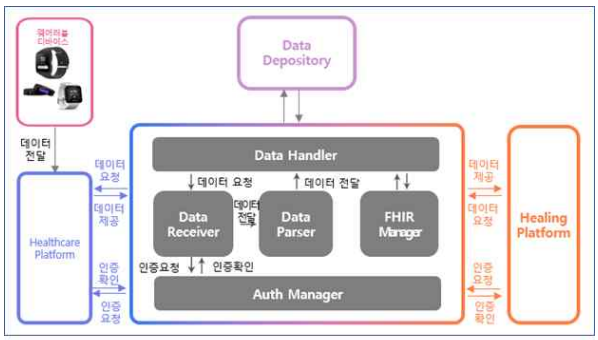
III. 건강정보를 활용한 시스템 설계

3.1 개인 건강정보 시스템 구조

개인 건강정보 시스템의 정보수집 및 콘텐츠 추천 시스템의 구조를 보면 웨어러블 디바이스에서 수집되는 생체정보 데이터를 전달받는 HealthCare Platform과 Data를 저장하고 인증

정보를 주고 받는 Middleware 그리고 콘텐츠를 추천하는 Healing Platform으로 구성되어 있는 것을 볼 수 있다.

<그림3> 개인 건강정보 수집 및 콘텐츠 추천 시스템 계략도



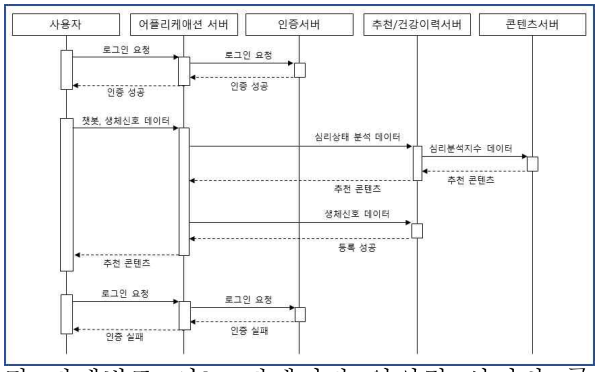
이 시스템에서 Data 및 콘텐츠를 주고받는 과정에서 통신과정에서 취약점들이 발생할 수 있는 확률이 높아진다.

3.2 개인 건강정보 Data 수집 및 활용

개인 건강정보의 Data 수집과정은 웨어러블 디바이스의 사용하여 수집하며 수집된 Data를 활용하는 과정을 보면 다음과 같다.

1. 사용자의 로그인 요청 및 인증.
2. 생체신호 Data 전송 및 이력 Data 저장
3. 생체신호 분석에 의한 심리상태 Data 추출
4. 심리분석 알고리즘에 의한 심리지수 분석
5. 심리지수에 따른 맞춤 콘텐츠 선정
6. 콘텐츠 서버의 맞춤 콘텐츠 전송

<그림4> Data 수집 및 활용 시나리오



각 단계별로 어느 단계이건 악의적 사이버 공격을 받을 수 있다.

소프트웨어 취약점은 시스템의 정지, Data 유출 및 오작동 등으로 연결될 수 있으므로 세심한 주의와 지속적인 관찰이 필요하다.

3.3 A사의 소프트웨어 설계 시 고려사항

외부공격으로부터 보다 안전한 소프트웨어를 개발하기 위한 방법은 다음 3가지로 분류할 수 있다.

1. 기존 개발방법론을 그대로 사용한다.
2. 보안방안을 적용한 새로운 개발방법론을 만든다.
3. 기존 개발방법론을 활용하면서 7-Touch Points를 활용한다.

3가지 방법 중에 1번은 기존 개발방법론을 활용하며 보안취약점(Vulnerability)을 그대로 가져가므로 위험성을 그대로 내포하기 때문에 사용할 수 없으며 2번은 새로운 개발방법론의 개발 및 검증과 보완에 막대한 물리적, 시간적 비용이 발생하므로 현실적으로 도입할 수 있는 기업이 극소수의 이며 3번 7-Touchpoints[7]를 활용하는 방안이 현실적인 방법이라 할 수 있다.

각 시스템의 보안요건에 따른 다음과 같은 서버, 네트워크, UI, 소스, 암호화, 접근통계 등에 대한 기술요소의 영향도 매트릭스를 작성하고 활용할 수 있다.

<표2>와 같이 메트릭스를 작성하여 보안 요구사항 및 침투 테스트 등에 활용하면 보안성 강화에 도움이 된다고 할 수 있다.

<표2> 보안 아키텍처 매트릭스

영역	전용선	방화벽	서버 접근 제어	DB 접근 제어	위변조 방지	업무별 접근 권한	탬보안 모듈	SSL	코딩 보안	마스킹 처리	이력 관리	단방향 암호화	양방향 암호화
외부사용자		●					●	●					
화면 내 개인정보						●		●		●	●		
사용자 비밀번호										●		●	
어플리케이션									●	●	●	●	●
데이터				●							●	●	●
인터페이스 (외부)	●	●	●					●					●
로그			●									●	●
첨부파일			●			●					●		
인쇄물					●	●				●	●		

출처 : 정윤석, 2014 한국소프트웨어 아키텍트 대회

보안 강화를 위하여 기본적인 암호모듈, 인증모듈, 접근제어 뿐 아니라 감사기록의 기능을 구현해야한다.[5]

소프트웨어 설계시점부터 보안을 강화할 수 있는 방법으로 7-TouchPoints를 통해 악용사례, 보안 요구사항, 아키텍쳐 위험 분석, 위험기반 보안테스트, 정적분석도구를 통한 코드 리뷰, 침투 테스트, 보안 모니터링, 구현과 관계없는 사람에 의한 외부검토 등을 추가하여 반복하여 보안성을 높일 수 있는 방법을 활용하며 개인 정보보호 요구사항을 포함시켜야 한다.

개발단계에서는 시큐어 코딩으로 보안약점을 최소화하며 SSL/TLS 구간암호화와 적극적인 암호화를 활용하고 테스트 단계에서 침투 테스트 및 개인정보 테스트를 함께 활용하여 보안성을 향상하는 것이 시스템 설계 시 고려해야 할 사항이라 하겠다.

IV. 결론

본 논문에서는 언제 어디서나 접근 가능한 개인 건강정보를 다루는 웨어러블 디바이스 등의 다양한 장비를 포함하는 시스템에서 기능 요구사항과 보안 요구사항을 포함하는 S/W를 구상하기 위해 보안강화 방안을 검토하고 S/W 설계를 보강하는 방법을 제시하였다.

개인정보 보호의 관점에서 접근에 대한 익명성, 편리함과 안정성을 보장하는 접근제어 방법 등의 활발한 연구가 이루어지고 있고 균형감 있게 발전해 나아가야 할 것이다.

개인 건강정보의 유출 및 이에 따른 의료사고 및 보안사고에 대한 우려를 해소하고 보다 건강하고 편리한 헬스케어 환경을 만들기 위해 법제도 개선, 의료적, 기술적 노력과 함께 악의적 사이버 공격에 강한 소프트웨어 설계에 지속적인 연구와 보완이 필요할 것으로 사료된다.

[참고문헌]

[1] NST. The Economic Impacts of Inadequate Infrasturcture for Software Testing, (2002년 5월)

[2] ISC2, “Official ISC2 Guide to the CSSLP CBK Second Edition”, 2013.07

[3] Joe McManus MGR, The CERT Oracle Secure Coding Standard for Java, Feb 2014.

[4] 행정안저부, ‘SW 개발단계부터 보안약점 제거(시큐어 코딩) 의무화, 2012.12

[5] Myung-Kyu Yi*, Hee-Joung Hwang, "A Study on Security Reference Model in Personal Health Lifelog Services" The Journal of The Institute of Internet, Broadcasting and Communication (IIBC) Vol. 16, No. 4, pp.109-115, Aug. 31, 2016

[6] 정윤석, “금융 및 Commerce 영역 보안아키텍처정의”, 2014 한국소프트웨어 아키텍트 대회

[7] http://www.swsec.com/resources/touchpoints/

안티바이러스 취약점 분석 방법 및 주요 보안 문제

전형린*, 김도연*, 김도혁*, 안정언*, 오우진*, 이상진**, 이승진***, 한승훈***

A Method for finding security vulnerabilities in antivirus and major security issues

Hyoung-Lin Jun*, Do-Yeon Kim, Do-Hyeok Kim, Jung-Un Ahn, Woo-Jin Oh

*Korea Information Technology Research Institute(KITRI), Best of Best(BoB).

요 약

안티바이러스(Antivirus)는 공격자의 공격으로부터 사용자 컴퓨터를 안전하게 보호한다. 안티바이러스 사용자는 기하급수적으로 늘어나면서 컴퓨터 시스템의 일반적인 구성요소가 되었다. 안티바이러스의 사용 및 보안 중요성도 증가 하였다. 그러나 공격자로부터의 공격을 방어하는 이 안티바이러스 자체에서 발생할 수 있는 보안 취약점이 존재한다. 이를 이용한 보안 위협은 컴퓨터 시스템에 심각한 피해를 줄 수 있다. 본 논문에서는 안티바이러스가 갖는 취약점 분석을 통해 발생 가능한 보안 위협을 정의하며, 분석 결과를 토대로 안티바이러스의 주요 보안 문제 및 대응 방안을 기술한다

I. 서론

안티바이러스(Antivirus)는 바이러스, 웜과 같은 악성 소프트웨어를 탐지, 제거 및 치료하는 컴퓨터 소프트웨어이다. 최근 안티바이러스의 사용이 증가함에 따라 컴퓨터 시스템의 일반적인 구성요소가 되었다. 사용자들은 보안을 위해 사용하는 소프트웨어를 신뢰하기 때문에 그 자체인 안티바이러스 보안에 대한 관심이 부족하다. 이러한 심리 때문에 안티바이러스가 공격의 대상이 되었는 바, 안티바이러스의 취약성을 연구할 필요가 있다.

본 논문의 구성은 다음과 같다. 윈도우 안티바이러스와 관련된 정보를 기반으로 이전에 발생한 취약점을 통해 최근 보안 동향을 살펴본다. 추가로 분석할 취약점 분석도구들에 대해서도 조사하며 이를 이용해 국내 안티바이러스를 분석한다. 발견된 취약점들을 분석하며, 가능한 공격 시나리오를 작성한다. 추가로 예상되는 공격에 대한 대응 방안을 제안한다.

이번 연구를 통해 안티바이러스 보안에 중요성을 한층 강화할 수 있을 것으로 기대한다.

* 한국정보기술원(KITRI) BoB
** 고려대학교 정보홍대학원장
*** 한국정보기술원(KITRI) BoB 멘토

II. 안티바이러스 주요 기능

안티바이러스의 위협은, 여러 가지 요인과 연관되어 있다. 대표적인 요인은 공격표면(Attack Surface)과 연관된 취약점이다.

2.1 스캐너(Scanner)

안티바이러스 스캐너는 파일 및 소프트웨어의 악성 패턴, 행위를 감지하기 위해서 파일 검사를 진행한다. 그 과정에서 다양한 파일 포맷을 지원하며 문서파일, 압축파일, 실행파일, 미디어 파일 등이 대표적이다. 악성검사를 위한

구체적인 데이터 파싱은 파일 데이터에 존재하는 각 포맷의 시그니처나 이름, 패턴[1] 등을 추출한 뒤, 각 파일포맷 별 루틴에서 구분되어 진행된다.

위의 절차를 따르는 악성검사 기능은 크게 “실시간 검사”와 “정밀검사” 의 형태로 동작한다. 실시간 검사는 파일의 생성, 이동, 삭제 등 많은 경우에 대해 동작하므로 드라이버 레벨에서 모든 경우를 처리하며, 검사결과에 대한 짧은 response time이 보다 중요하다. 정밀검사는 UI 프로세스를 통한 정밀검사 등 사용자가 선택한 실행에서만 진행되며, 실시간 검사와는 다르게 검사대상의 포맷 구분을 유저레벨의 dll에서 진행한다. 이후에는 구체적인 파일 데이터 파싱을 진행하여 악성유무를 판단한다.

2.2 업데이트

안티바이러스 업데이트는 실시간으로 악성코드 DB를 업데이트 한다. 악성코드 DB는 신종 바이러스, 악성코드 데이터 내역을 포함하고 있으며, 이 데이터를 기반으로 검사하여 컴퓨터에 있는 파일이 악성코드인지 분류하여 PC를 보호한다.

2.3 클라우드 악성 분석

다양한 보안 위협을 실시간으로 대응하기 위해서는 발견된 악성코드 DB를 통해 검사하여 발견하는 경우도 있지만 탐지하지 못할 수 있다. 이 때 알려지지 않은 위협을 발견했을 경우 해당 행위를 하는 파일을 클라우드 자동 분석 서버로 전송하여 악성 여부를 분석하고 결과를 실시간으로 전달한다.

III. 안티바이러스 취약점 유형 분석

효과적인 분석을 위하여 이전에 발견된 취약점이 다시 발생할 가능성이 높기 때문에 기존 취약점을 분석한다.

취약점 유형 분석 대상은 OPSWAT[2]에서 공개한 윈도우 안티바이러스 시장 점유율 보고서에 따라 상위 10개 회사를 선정하였다.

표 2 Windows Anti-malware Market Share Reports. August 2017

회사명	점유율
AVAST Software a.s.	19.85%
ESET	13.87%
McAfee, Inc.	9.67%
Avira GmbH	7.59%
Safer-Networking Ltd.	7.46%
Malwarebytes Corporation	7.38%
Webroot Inc	5.68%
Qihu 360 Software Co., Ltd	4.95%
Kaspersky Lab	4.91%
Bitdefender	4.24%

3.1 취약점 유형 분석

본 논문은 취약점 유형 분석을 CWE/CVE[3] 기반 데이터로 구성하였다. 최근 보안 동향을 살피기 위해서 최근 5년(2012~2017년도) CVE 자료를 토대로 유형 분석을 하였다.

표 3 취약점 유형표

취약점 유형	개수	비율
Bypass something	73	32%
Code Execution	41	18%
Gain Information	39	17%
XSS	18	8%
DoS	16	7%
Directory Traversal	10	4%
Overflow	7	3%
Sql Injection	8	3%
Gain Privileges	8	3%
CSRF	7	3%
Memory Corruption	4	2%
합계	231	100%

분석 결과 최근 5년 동안 231개의 취약점이 발견되었고 이는 11개의 유형으로 나눌 수 있다.

가장 높은 빈도로 발생한 취약점 유형은 “Bypass something”이다. 이는 안티바이러스가 악성코드를 탐지하지 못하게 우회하는 경우나 예상된 행동을 우회하는 것이다. 이를 이용하여

안티바이러스 프로세스를 무력화하지 않더라도 탐지를 우회하여 공격할 수 있다. 다음으로 공격자가 원하는 임의의 명령을 실행하는 “Code Execution” 와 공격자가 피해자의 민감한 정보를 획득하거나 무결성을 훼손하는 “Gain Information” 취약점 유형 등이 차례로 많은 것을 확인할 수 있다. 분석 결과를 기반으로 발생 가능한 주요 위협 요소를 확인한다.

IV. 안티바이러스 취약점 분석 기법

본 논문에서는 소프트웨어 기반 취약점 분석으로 코드 기반 분석과 퍼징(fuzzing)을 기반으로 하는 분석을 진행하였다. 분석 대상은 국내 안티바이러스 3사를 대상으로 하였다.

4.1 코드 기반 분석

안티바이러스는 대부분은 소스가 공개되지 않으므로 분석 대상에 대한 정보가 부족하여 실제 분석 과정에서 어려운 점이 많다. 그렇기 때문에 역공학(Reverse Engineering)을 통해 취약점을 분석해야 한다. 그렇지만 역공학을 하는 것이 많은 시간이 소요되기 때문에 우선 퍼징 기반 분석을 통해 취약점을 찾고 이 후에 발견한 취약점 중심으로 코드 기반 분석을 하는 방법도 있다.

4.2 퍼징 기반 분석

퍼징 기반 분석은 최근 많이 연구된 소프트웨어 결함 주입 기법 중 하나로 소프트웨어 실행을 통해 발생할 수 있는 결함이나 오류를 분석하기 위한 기법이다. 그 중 퍼징 기법은 무작위 데이터를 반복 입력해서 소프트웨어의 조직적인 실패를 유도함으로써 취약점을 찾아내는 것이다. 본 논문에서는 다양한 포맷의 파일 데이터를 뮤테이트 적용 후 안티바이러스가 읽게 하여 이상현상이 있는지 확인하는 파일 퍼징 방법과 사용자 레벨에서 드라이버에 접근하기 위한 파라미터 값을 뮤테이트하여 입력해 이상현상을 확인하는 드라이버 퍼징 방법을 사용하였다. 이 외에도 배포된 퍼징 Tools[4]을 사용할 수도 있다.

V. 분석 결과

분석 결과 국내 안티바이러스 DoS 취약점과 Memory Corruption 취약점이 발견되었다.

[표 4] 발견된 취약점

취약점 유형	A사	B사	C사
DoS	8	4	5
Overflow	2	1	2
Memory Corruption	0	0	1
합계	10	5	8

본 연구에서 시행한 퍼징 기반 분석으로 안티바이러스 취약점을 발견하였다. 퍼징 기법으로 발견된 크래시들은 다양한 파일 포맷을 파싱하는 과정에서 발생하였으며, 이 파일 포맷들은 hwp, pdf, rar, zip, mime, 7z, exe, cab, gz 포맷 파일이다. 발생한 크래시들을 분석한 결과. 총 23개의 크래시 중에 DoS 취약점이 17개, 정수 오버플로우 취약점 5개, EIP변조 Memory Corruption 취약점 1개가 발견되었다.

VI. 안티바이러스 보안 문제 및 대응 방안

발견된 취약점으로 어떤 공격이 가능한지에 따라 공격 시나리오를 작성한다. 또 이를 대응하기 위한 방법을 제시한다.

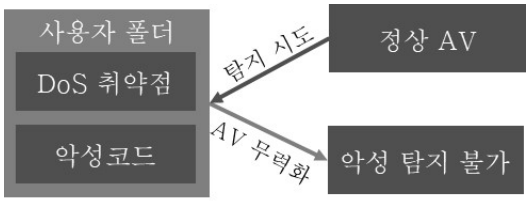
4.1 서비스 거부

발견한 안티바이러스 DoS 취약점은 실시간 검사와 정밀검사에서 수행되는 파일 검사를 위한 파일 파싱 과정에서 발생한다. 이를 이용하면 해당 악성파일을 실행하지 않고도 단순 파일 다운로드나 이동, 복사 행위만으로 DoS 공격이 가능해 안티바이러스 프로세스를 무력화시킬 수 있다.

이러한 공격을 막기 위해서는 안티바이러스의 악성코드DB를 업데이트함으로써 해당 악성파일을 탐지하여 DoS 공격을 대응할 수 있다.

4.2 악성코드 우회

안티바이러스 DoS 취약점을 이용하면 악성코드를 안티바이러스 탐지로부터 벗어나 우회할 수 있다. 공격자의 악성코드와 DoS를 발생시키는 악성코드를 같은 폴더에 위치시키면 안티바이러스 프로세스는 무력화된다. 연구 결과 3개의 안티바이러스 중에 2개의 제품에서 DoS 공격에 의한 검사 프로세스가 무력화 되었을 경우 사용자가 안티바이러스 UI를 통해 무력화된 것을 쉽게 확인하기 어렵다.

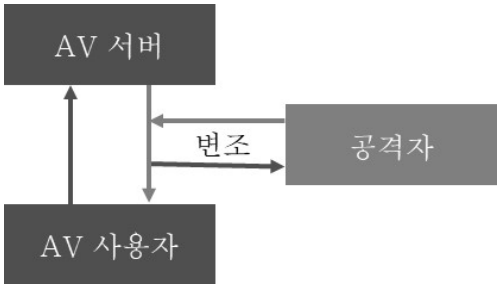


[그림 1] DoS취약점을 이용한 악성코드 우회 공격 시나리오

이것은 프로세스 무력화되었을 경우 안티바이러스는 UI를 통해 사용자가 쉽게 확인할 수 있도록 디자인을 개선할 필요가 있다.

4.3 네트워크 변조

국내 안티바이러스를 분석한 결과 업데이트 시에 HTTP[5] 통신을 사용하는 것을 확인하였다. 이는 안티바이러스 업데이트 중에 인증되지 않은 공격자에 의해 변경되어 보안 위협이 발생될 수 있다. 악의적인 목적을 가진 공격자가 악성코드로 변조하여 공격이 가능하다. 사용자가 다운 받은 업데이트 파일에 악성코드를 포함시킬 것이다.



[그림 2] 네트워크 변조를 이용한 MiTM 공격 시나리오

이것은 안티바이러스 업데이트 중 서버측과 클라이언트측 통신을 기존 HTTP 통신에서 SSL[6] 암호화 통신으로 변경하여 양방향 인증함으로써 MiTM과 같은 중간자 공격에 대응할 수 있다. 또는 네트워크 패킷 통신 후에 패킷을 점검하여 무결성을 확인할 필요가 있다.

VII. 결론

본 논문에서는 안티바이러스에서 발생할 수 있는 취약점을 알아보기 위해 안티바이러스의 주요 기능을 이해하고 이전에 발견된 취약점을 조사했다. 이후 국내 안티바이스가 어떤 취약점을 갖고 있는지 분석하고 이를 통해 공격이 가능한 보안 문제를 제시했으며 이에 상응하는 대응 방안을 제시했다.

식별된 취약점을 통해 안티바이러스가 가지고 있는 보안 위협을 확인할 수 있었으며 대응 방안 연구에 대한 필요성을 제고했다. 이를 기반으로 안티바이러스에서 발생할 수 있는 보안 취약점 및 위협을 대비할 수 있을 것으로 기대된다.

[참고문헌]

[1] 박남열. “우회기법을 이용하는 악성코드 행위기반 탐지 방법”. 2006.

[2] OPSWAT. “Windows Anti-malware Market share report”. http://www.metadefender.com.

[3] NIST. “CVE Statuses in National Vulnerability Database”. http://nvd.nist.gov/vuln.

[4] 박용수. “윈도우 멀티미디어 취약점 분석 방법론 연구”. 한양대학교 산학협력단. 2009.

[5] 김건. “HTTP 프락시를 활용한 인증정보 전달 안전성 분석”. 정보과학연구소. 2012.

[6] Zubair Jeelani. “An Insight of SSL security attacks”. 2013.

[7] Joxean Koret. The Antivirus Hacker’s Handbook. 2015.

기계 학습을 통한 악성코드 연관성 분석 방법론

채진희*, 오동빈**, 정해원**, 김예준*, 이현***, 유정인****, 장원희*****

*순천향대학교 정보보호학과, **경찰대학교 행정학과, ***선린인터넷고등학교,
****가톨릭대학교 미디어기술콘텐츠학부, *****서울호서전문학교 사이버해킹보안과

Methodology of malicious code association analysis through Machine Learning

Jin-Hee Chae*, Dong-Bin Oh**, Hae-Won Jeong**, Ye-Jun Kim*, Hyun Yi***, Won-Hee Jang****, Jung-In Yoo*****

*Dept of Information Security, SoonChunHyang University.

**Dept of Administration, Korean National Police University.

***SunRin Internet High School.

****Dept of Media Technology Contents, Catholic University of Korea.

*****Dept of cyber hacking information security, Seoul Hoseo Occupational Training College.

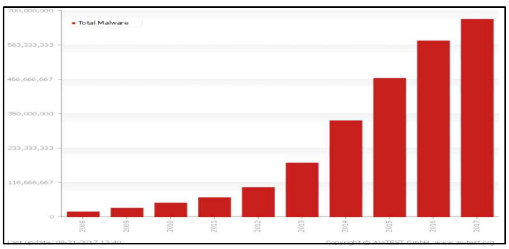
요 약

최근('17) 전 세계에서 발생하는 다양한 사이버 침해사고에서 발견되는 악성코드의 수는 약 7억 개에 달하며, 악성코드의 발생 비율은 가파르게 증가하고 있다[1]. 하지만 이러한 악성코드를 제한된 분석인력으로 모두 분석하는 것은 거의 불가능하기 때문에 변종 악성코드에 대한 정밀한 분석을 우선적으로 수행하여 대응하는 것이 필요하다. 이와 관련하여, 기존의 악성코드 일부 특징(시그니처)을 추출하여 분석에 활용하는 연관성 분석은 수동적이며 비효율적인 악성코드 분석방식이다. 따라서 이러한 분석방식은 수많은 변종 악성코드에 분석 시 적용하기 어렵다는 한계가 존재한다[3]. 본 논문에서는 기존의 악성코드 연관성 분석의 한계점을 보완하기 위해 기계학습(머신러닝)을 이용하여 정확하고 효율적인 악성코드의 연관성 분석 방법론을 제안한다.

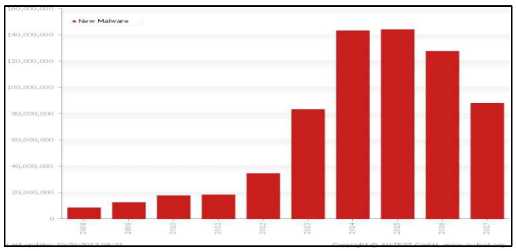
I. 서론

최근 다양한 유형의 사이버 침해사고가 발생함에 따라 이러한 침해사고에서 발견되는 악성코드의 종류와 수도 증가하고 있으며, 정교해지고 있다. 실제, 악성코드 감염 사고는 개인정보 유출, 사생활 침해뿐만 아니라 군사정보 같은 국가 주요정보 유출 및 기반시설에 대한 직접적인 위협을 주는 사례가 늘어나고 있으므로 침해사고에 활용되는 악성코드를 사전에 탐지하여 신속하게 대응하는 것이 매우 중요하다.

이와 관련하여 독일의 민간 보안연구 단체인 "AV-TEST"의 조사 결과에 따르면 [그림1]과 같이 2017년 11월 기준으로 올해('17)까지 발견된 악성코드의 수는 약 7억 개에 달한다[1]. 그리고 [그림2]와 같이 신종 악성코드의 탐지는 2015년까지 급격히 증가하였으나 그 이후에는 감소하고 있는 추세를 보이고 있다. 이것은 2016년부터 탐지되는 악성코드들 중 새로 개발된 신종 악성코드 보다 기존에 발견된 악성코드를 재활용한 변종 악성코드의 출현이 더 많이 늘어나고 있다고 볼 수 있다. 이것은 공격자들이 완전히 새로운 신종 악성코드를 개발하기 보다는 기존에 개발된 악성코드를 재활용한 변종 악성코드를 침해사고에 활용하고 있다고 볼 수 있다. 따라서 이러한 변종 악성코드를 신속하게 탐지하고 효율적으로 대응하기 위해서는 기존에 탐지된 악성코드와 관련된 요소에 대한 구체적인 정형화가 필요하다.



[그림 1] 최근 10년간 발견된 악성코드 수[1]



[그림 2] 최근 10년간 발견된 새로운 악성코드 수[1]

모든 악성코드는 각각의 고유한 특징을 가지고 있다. 이러한 특징을 기반으로 악성코드를 구성하는 각각의 요소를 정형화하고 분류하여 악성코드의 유사성을 정의하는 과정을 악성코드 프로파일링(Malware Profiling)이라고 한다. 악성코드 프로파일링은 각 악성코드가 가지고 있는 개별적인 특징(요소)을 추출하고 개별 악성코드의 공통점을 분석하는 과정이 필요하다. 분석가는 악성코드 분석을 통해 추출된 요소와 각 요소의 공통점을 바탕으로 개별 악성코드 간 연관성과 유사성을 분석한다[2].

기존의 악성코드 연관성 분석 방식은 대부분 분석가에 경험에 기반한 수동적으로 이루어진다. 분석가는 대상 악성코드에서 타 악성코드와는 구분되는 요소인 시그니처(Signature)를 추출하여 연관성 분석에 활용한다. 이러한 방식은 철저히 분석가의 역량에 의존하기 때문에 분석가가 탐지하지 못한 변종 악성코드에 대한 효율적인 연관성 분석이 어렵다는 한계가 존재한다[3].

따라서 이를 보완하기 위해 본 논문에서는 악성코드 바이너리(Binary)가 가지고 있는 다양한 속성(특징)을 기반으로 한 머신러닝(Machine Learning) 악성코드 연관성 분석 방법을 제안하고자 한다.

II. 관련 연구

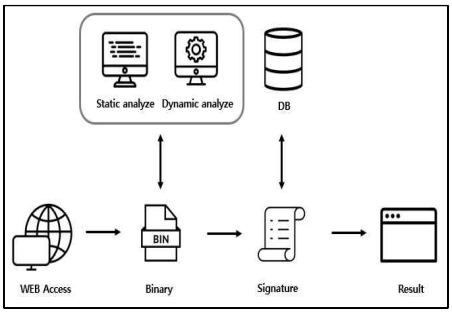
본 장에서는 악성코드 정적 및 동적 분석, 그리고 기존의 악성코드 연관성 분석 방법에 대해 설명한다.

2.1 악성코드 정적 및 동적 분석

악성코드 분석은 바이너리(Binary)를 어떤 방식으로 분석하는지에 따라 정적 분석과 동적 분석으로 나눠진다. 정적 분석은 바이너리의 실행 없이 역공학(Reverse Engineering) 기법을 사용하여 PE(Portable Executable) 구조나, 사용된 API(Application Programming Interface)와 같은 정적 속성을 분석하는 것을 말한다. 동적 분석은 바이너리를 실제로 실행하여 시스템에서 어떠한 변화가 발생하는지에 대한 속성 정보를 추출하는 방식을 말한다.[4].

2.2 기존의 악성코드 연관성 분석

아래의 [그림 3]와 같이 기존의 악성코드 연관성 분석은 분석가가 수동적으로 악성코드 내의 시그니처를 제한적으로 추출하여 악성코드 간의 연관성 분석에 활용한다. 이는 악성코드의 다양한 속성이 아닌 일부 특정 부분을 이용하여 악성코드 유사 여부를 분석하므로 탐지 속도가 빠르다는 장점을 가지고 있다[5][6].



[그림 3] 기존의 악성코드 연관성 분석

하지만 기존의 방법은 연관성 분석에 활용하는 악성코드의 특정 부분만 변조되어도 정확한 탐지가 어렵다는 문제점을 가지고 있다. 또한, 분석가의 역량에 기반하여 수동적으로 시그니처를 추출하기 때문에 분석에 대한 효율성이 낮다.

따라서 기존에 존재했던 악성코드뿐만 아니라 일부만 변형된 악성코드나 알려지지 않은 악성코드에 대해서 특정 시그니처 뿐만 아니라 악성코드의 다양한 속성을 자동적으로 추출하여 효율적으로 분석할 수 있도록 하는 방법이 필요하다.

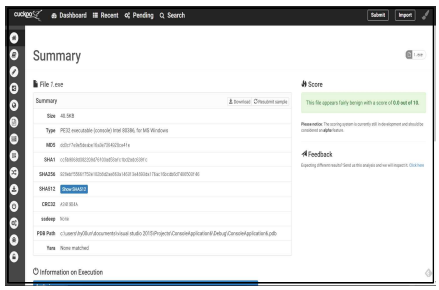
III. 제안방법

본 장에서는 악성코드 행위 기반 분석 방법에 대해서 설명하고 머신러닝 기반 악성코드 연관성 분석 방법에 대해 제안한다.

3.1 악성코드 행위 속성 기반 분석

행위 기반 분석은 악성코드 분석방식 중에서 동적 분석에 해당되며, 악성코드를 실제로 실행하고 동적 데이터를 수집하기 위한 가상 환경인 샌드박스(Sandbox)를 사용한다.

쿠쿠 샌드박스(Cuckoo SandBox)는 대표적인 악성코드 분석을 위한 샌드박스 중 하나이다. 쿠쿠 샌드박스의 분석 결과로는 아래의 [그림 4]과 같이 MD5 해시 값, 악성코드 간 유사도를 측정하는 ssdeep과 같은 바이너리 정보를 나타내는 Summary가 있다. 또한, PDB Path, 컴파일 시간(Compile Time), 섹션(Section) 및 엔트로피(Entropy)와 같은 바이너리 정적 분석 결과를 보여주는 Static Analysis, 악성코드가 외부로부터 받아오는 파일을 알려주는 Dropped files, 네트워크 통신 관련 정보를 보여주는 Network Analysis, 프로세스 트리(Tree)를 통해 사용한 프로세스 및 커맨드(Command) 등과 같은 프로세스 행위에 대해 보여주는 Behavior Summary 등과 같은 기능을 통해 악성코드에 대한 동적 분석 결과를 보여준다[7].



[그림 4] 쿠쿠 샌드박스 실행 화면

3.2 머신러닝 기반 연관성 분석

본 장에서는 기존의 연관성 분석의 한계점을 보완하기 위해 바이너리 자체의 다양한 속성을 기반으로 한 머신러닝 연관성 분석 방법을 제안한다.

3.2.1 피처(Feature) 추출

분석가가 서버에 악성코드를 업로드(Upload)하면, 서버에 구축된 샌드박스를 통해 분석한다. 업로드된 악성코드에 대해 샌드박스 분석결과 추출한 데이터를 바탕으로 분석가는 다시 머신러닝 모델에 사용될 피처를 추출한다.

만약 샌드박스가 탐지할 수 없는 악성코드 내 문자열 값 혹은 인코딩(Encoding) / 디코딩(Decoding) 루틴의 경우에는 야라(YARA)를 활용한다. 야라는 대표적인 시그니처 기반 탐지 도구로서 룰(Rule)을 작성하여 문자열 및 패턴 검색을 지원하는 도구이다[2]. 사전에 작성된 야라 룰을 이용하여 악성코드가 탐지되는지에 대한 여부를 추가적인 피처로 사용할 수 있다. 뿐만 아니라 PDB Path, C&C(Command and Control) 주소, 고정된 시그니처와 구조의 유사도를 판별할 수 있는 바이너리 전체의 퍼지해시(Fuzzyhash) 값, 악성코드의 IAT(Import Address Table) 정보에 기반 한 API 관련 해시인 imphash를 활용한 피처 등, 다양한 속성을 활용할 수 있다.

3.2.2 데이터 처리

이전 과정에서 정적 및 동적 분석으로 추출된 피처를 이용하여 전처리(Preprocess) 과정을 수행한다. 이때 의미 없는 피처를 제거하며, 삭제되는 데이터는 분석가의 판단에 따라 달라진다[8]. 그렇기 때문에 제안하는 방법에서는 크게 6가지의 피처를 지정하여 가중치를 측정 한 후 부여한다. 가중치 측정 기준은 아래의 [표 1]와 같다.

3.2.3 분석 엔진

연관성 분석 과정에서 추출된 피처는 특징의 고유성에 따라 각각 다른 가중치를 가진다. 이때 로지스틱 회귀분석을 사용하여 보다 정확한 가중치 산정 기준에 붙는 계수를 지정할 수 있다. 로지스틱 회귀분석은 분석하고자 하는 대상이 여러 집단으로 나뉘어졌을 경우, 개별 피처가 어느 집단으로 분류할 수 있는지를 분석 및 예측하는 모형에 사용된다[9].

속성	기준
PDB Path	Difflib에 의해 SequenceMatcher 모듈 통해 비교하여 pdb name 우선으로 일치하면 점수 부여
C&C	C&C 주소의 C클래스가 일치하면 점수 부여
Number of Sections	대상 샘플의 섹션(Section) 개수를 비교하여 일치하지 않으면 점수 차감
Yara Rule	야라 룰에 의해 매칭(Maching)할 경우 점수 부여
Fuzzyhash	퍼지해시 값을 구하여 고정된 시그니처와 구조의 유사도를 판별해 ssdeep을 통해 수치화된 점수에 따라 점수 부여
Imphash	imphash 값의 일치 여부를 통해 유사도 판별 후 점수 부여

[표 1] 가중치 측정 기준

IV. 결론

본 논문은 악성코드 연관성 분석에 대한 연구의 필요성을 확인한다. 그리고 기존의 시그니처 기반 연관성 분석 방법을 설명하고, 이에 대한 한계점을 보완하는 머신러닝 기반 연관성 분석 방법론을 제안한다.

현재 악성코드는 다양한 은닉 및 난독화 기법을 사용하고 있으며, 분석 대상이 되는 악성코드의 수는 하루에 약 30만개씩 폭발적으로 증가하고 있다. 따라서 증가하는 악성코드에 대한 신속하고 효율적인 대응을 위해 악성코드의 연관성을 분석하는 작업이 더욱 중요해졌다.

본 논문에서 제시한 머신러닝 기반의 연관성 분석 방법은 악성코드 분석과정에서 사용되는 다양한 속성 값을 종합적, 선택적으로 활용하여 효율적이고 정밀한 비교를 한다는 점에 있어서 빠르고 정확한 연관성 분석이 가능하다. 하지만 다양한 악성코드 속성 요소에 대한 충분한 데이터베이스가 구성되어 있어야하며, 오탐을 최소화하기 위해 다양한 형태 및 속성의 분석 결과도 검토되어야 한다. 또한, 머신러닝에 필요한 피처 추출 시 샌드박스 결과를 바탕으로 분석가가 직접 피처를 구분해야 한다는 과정이 필요하다. 추후에는 이와 같은 문제점들에 대해 보완 및 검증하는 연구가 필요할 것이다.

*본 논문은 2017년도 차세대 보안리더 양성 프로그램(Best of Best)의 지원을 받아 수행된 연구입니다.

[참고문헌]

[1] “AV-TEST", https://www.av-test.org/en/statistics/malware/

[2] 금융보안원 (2017), “2017 사이버 위협 인텔리전스 보고서”

[3] 김종현, 김현주, 김익균 (2015), “악성코드 대응을 위한 사이버게놈 기술동향”, Electonics and Telecommunications Trends vol. 30 No. 5, Oct, 2015, pp. 118-128

[4] CISCO (2017), "Malware Analysis for the Incident Responder", https://blogs.cisco.com/security/malware-analysis-for-the-incident-responder

[5] 하우리 (2009), “악성코드의 악성 행위와 탐지 기법”, http://www.hauri.co.kr/information/issue_view.html?intSeq=93&page=9&article_num=92

[6] 안랩 (2007), “알려지지 않은 악성코드 탐지 기법”, http://noentrypoint.blogspot.kr/2012/08/blog-post_23.html

[7] “Cuckoo Sandbox 결과분석“, http://lureout.tistory.com/24, 2014

[8] Mohammad Mehdi Ahmadian , Hamid Reza Shahriari, "2entFOX: A Framework for High Survivable Ransomwares Detection", ISCISC2016, July-Aug, 2016

[9] 강성욱, “로지스틱 회귀분석(Logistric Regression)", http://sqlmvp.kr/140189702419?Redirect=Log&from=postView, 2013

효율적인 OS Fingerprint Signature에 관한 연구

송진호*, 김용건*, 원유재*
*충남대학교 컴퓨터공학과

A Study on Efficient OS Fingerprint Signature

Jin-Ho Song*, Yong-Gun Kim*, Yoo-Jae Won*
*Division of Computer Science & Engineering, ChungNam University.

요 약

네트워크 패킷을 이용한 OS 판별 기술 OS Fingerprint는 보안이 취약한 디바이스의 정보를 취득할 수 있다. 하지만 OS마다 판별할 수 있는 Signature가 다르기 때문에 OS를 판별하기가 쉽지 않다. 본 논문에서는 네트워크 패킷들을 수집하여 여러 개의 규칙을 만들어 테스트해보고 효율적인 방법을 제안한다.

I. 서론

인터넷이 보편화 되고 스마트 홈, 스마트 시티 등의 IoT(Internet of Things) 서비스들이 증가하고 있다. 이에 따라 취약점을 가지는 디바이스들이 증가하고 있으며, 취약점을 없애기 위하여 보안 기술이 발전하고 있다. 본 논문에서는 취약한 디바이스를 검색할 수 있는 기술인 OS Fingerprint에 대해 연구하고 네트워크 패킷에서 추출할 수 있는 정보에 대하여 제공하도록 한다.

1.1 관련연구

OS Fingerprint 기술은 네트워크 패킷을 이용하여 OS를 판별하는 기술이다. OS Fingerprint에서 사용되는 네트워크 패킷은 표[1]과 같다.

표 1. OS Fingerprint 에서 사용되는 네트워크 패킷

Fields	설명
WWWW	TCP Windows Size
MSS	Maximum Segment Size
TTL	Time to Live
WS	Windows scale
S	Selective ACK
N	No operation
D	Don't Fragment
T	Time stamp
E	EOL(End of Line) Option
Wn	Windows Scaling Option
F	Packet flags
D	Data payload
A	ACK number non-zero
LEN	TCP Packet Length
SS	SYN Packet Length
OOO, QQ	Option set
B	Binary flags
F0n	TCP flags

표[1]에서 보는 것과 같이 많은 데이터를 사용한다는 것을 확인할 수 있었다. 하지만

OS Fingerprint도구마다 모든 데이터를 사용하지 않는다. 표[2]와 같이 도구 마다 사용되는 데이터는 다르며, 또한 데이터가 다르기 때문에 식별하는 OS가 다르게 나타날 수 있다.

표 2. OS Fingerprint 도구 별 사용되는 네트워크 패킷

Tool	Signature
Ettercap	WWWW:MSS:TTL:WS:S:N:D:T:F:LEN
P0f	WWWW:TTL:D:SS:OOO:QQ
NetworkMiner	
SinFP3	B:F0n:WWWW:OOO:MSS:WS:LEN

P0f[1]와 NetworkMiner[2]는 Signature 형식은 같다. 하지만 두 개의 차이점은 DB(DataBase)에서 나타난다. P0f는 순수하게 자신만의 DB를 이용하여 판별하지만 NetworkMiner는 자신이 가지고 있는 DB뿐만 아니라 P0f, Ettercap[3]에서 사용되는 DB도 참조되도록 만들어져 있다.

위의 표와 같이 도구에서 사용되는 Field형식을 알 수 있었다. 다음으로는 OS Fingerprint 도구 마다 OS인식률에 대해 설명하도록 한다.

1.2 OS Fingerprint 별 인식률

OS Fingerprint에서 사용되는 매칭 방법은 SM(Similarity Matching)방식과 EM(Exact Matching) 방식 두 가지가 있다. SM방식은 그림[1]과 같이 네트워크 패킷에서 추출된 데이터를 가지고 유사하게 나타난 부분만을 계산하여 결과가 제일 높게 나온 OS를 출력한다.

FAF0:05B4:80:00:1:1:1:0:A:34:Windows XP
FAF0:05B4:80:00:1:1:1:1:S:40:Windows XP

그림 1. SM(Similarity Matching) 방식

EM(Exact Matching) 방식은 그림[2]와 같이 연속적으로 데이터 결과가 일치하게 나온 OS를 출력하는 방식이다.

65535:128:1:M1460,N,N,T0,N,N,S
64240:128:1:M1460,N,N,T0,N,N,S

그림 2. EM(Exact Matching) 방식

표3. OS Fingerprint 도구 별 인식 비교표

Target	OS		Version
168.188.x.x	Windows XP		SP0
Tool	OS	매칭방법	인식률
Ettercap	Windows XP Pro	SM	80%
P0f	Windows XP SP1+	SM	50%
Network Miner	Windows XP	EM, SM	78%
SinFP3	Windows XP SP0	EM, SM	100%

표3은 Windows XP SP0 버전에서 진행한 결과 내용이다. SinFP3[4]는 100%의 결과로 가장 높게 나왔다. 제일 낮게 나온 도구는 P0f라는 것을 확인할 수 있었다. 이 부분에서의 차이점은 DB에서 확인할 수 있다.

표 4. P0f와 SinFP3의 Signature 비교

Tool	Signature	DB
P0f	64240:128:1:40:M1460,N,W0,N,N,T0,N,N,S:A	64512:128:1:48:M*,N,N,S
SinFP3	B11113 F0x12 W64240 O0204ffff M1460 S0 L4	B11113 F0x12 W64240 O0204ffff M1460 S0 L4

표4는 표3에서 테스트한 P0f와 SinFP3에 대한 Signature결과 데이터이다. 공통적인 부분도 존재하지만 형식 부분에서 다른 것을 확인 할 수 있다. 인식률의 계산법은 그림 [3]과 같이 계산하였다.

$$\text{인식률} = \frac{\text{맞춘 } Field \text{ 개수}}{Signature \text{ 전체 개수}} * 100$$

그림 3. 인식률 계산법

P0f의 Signature수는 14개이다. 하지만 DB에서 매칭 되는 부분은 7개 밖에 되지 않는다. 반면 SinFP3에서는 7개 모두 맞춰지는 것을 확인할 수 있다. 우리는 여기에서 확인할 수 있는 부분은 Signature의 개수가 많아도 결과

가 제대로 나오지 않을 수 있다는 것이다. 이를 고려하여 본 논문에서는 Field들을 이용하여 규칙을 새롭게 만들고 4개의 OS Fingerprint도구와 비교해본다.

1.3 OS Fingerprint 규칙 분석 비교

먼저 표[2]를 확인하면 4개의 도구에서 공통적으로 사용되는 Field들을 확인할 수 있다.

표 5. OS Fingerprint 도구에서 사용되는 공통 Field

Fields	설명
WWWW	TCP Windows Size
MSS	Maximum Segment Size
WS	Windows scale
S	Selective ACK
N	No operation
T	Time stamp
F	Packet flags

4개의 도구에서 공통적으로 사용되는 Field들은 표[5]와 같다. 이를 이용하여 우리는 규칙을 새롭게 구성해볼 수 있다.

표 6. OS Fingerprint 규칙 생성

명칭	Signature	개수
ST1	WWWW,MSS,WS,S,NOP,T,F	7
ST2	WWWW,MSS,S,NOP,T,F	6
ST3	WWWW,MSS,TTL,S,NOP,T,F	7
ST4	WWWW,MSS,TTL,S,NOP,T,F,D	8
ST5	WWWW,MSS,TTL,S,NOP,T,F,D, LEN	9
ST6	WWWW,MSS,TTL,WS,S,NOP,T, F,D	9
ST7	WWWW,MSS,TTL,WS,S,NOP,T,D	8
Ettercap	WWWW,MSS,TTL,WS,S,NOP,T,F,D,LEN	10
P0f	WWWW,MSS,TTL,WS,SS,N,E,T, F,QQ	11
Network Miner		
SinFP3	B,F,WWWW,MSS,WS,S,T	7

ST1(Signature Test 1)은 4개의 도구에서 공통적으로 사용되는 Signature를 뜻한다. ST2에서부터 ST7까지 는 ST1에서 감소시키거나 추가하여 만든 Signature들이다. 우리는 이 Signature와 4개의 도구를 이용하여 인식률 비교해보았다. 테스트 방식은 4개의 도구에서 사용되는 DB와 매칭되는 부분을 인식률 계산법을 이용하여 진행하였다.

1.4 결과

표7. OS 인식률 비교 테이블

Signature	Ettercap	Network Miner	P0f	SinFP3	평균
ST1	85.70%	90%	80%	100%	88.93%
ST2	83.30%	87.50%	75%	100%	86.45%
ST3	85.70%	88.80%	77.70%	100%	88.05%
ST4	87.50%	90%	80%	100%	89.38%
ST5	88.80%	90.90%	81.80%	100%	90.38%
ST6	88.80%	91.60%	83.30%	100%	90.93%
ST7	100%	92.30%	85.71%	100%	94.50%
Ettercap	90%	92.30%	84.60%	100%	91.73%
P0f	100%	95.71%	85.71%	100%	92.86%
NetworkMiner					
SinFP3	83.33%	80%	60%	100%	80.83%

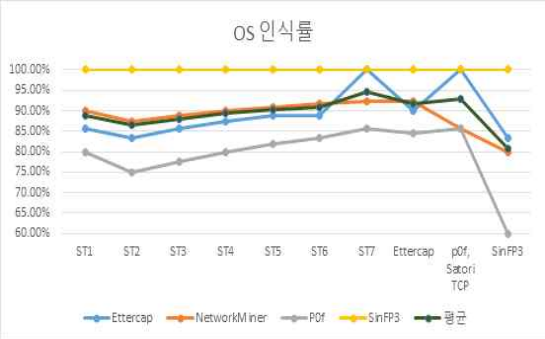


그림 4. OS 인식률 비교 그래프

표6은 ST1에서 ST7과 각 도구 별 인식률 비교 분석 결과이다. 제일 높게 나온 Signature는 Windows Size, Maximum Segment Size, Time to Live 등 총 8개로 구성된 ST7가 나왔다. 낮은 결과로는 ST2로 나왔다. ST2와 ST7의 차이점은 Time to Live의 유무이다. Time to Live는 OS마다 설정되어 있는 값이 다르기 때문에 중요 역할을 담당하고 있다고 생각할 수 있다.

II. 결론

본 논문에서는 네트워크 패킷을 이용하여 OS Fingerprint의 Signature와 인식률 비교 분석에 대하여 정리하였다. OS Fingerprint의 인식률을 높게 구현할 수 있으면, 취약점 디바이스에 사전에 미리 대비 할 수 있는 보안 솔

루션에서 사용할 수 있다. 반대로 공격자 입장에서는 OS의 판별이 높아지면 취약점 디바이스에 더 쉽게 공격할 수 있기 때문에 유의해야하는 점을 가지고 있다. 본 논문은OS Fingerprint와 네트워크 패킷을 이용한 연구에서 유용하게 사용할 수 있을 것이다.

III. Acknowledgement

이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2016-0-00193, IoT 보안 취약점 검색·공유 및 시험 기술 개발)

[참고문헌]

[1]P0f, http://lcamtuf.coredump.cx/p0f3/

[2]NetworkMiner,https://www.netresec.com/?page=NetworkMiner.

[3]Ettercap,https://ettercap.github.io/ettercap/downloads.html,March14,2015

[4]GomoR,https://www.ekoparty.org/archivo/2012/eko8-SinFP3.pdf,2012

IoT System 개발을 위한 Secure Agile Process: MS-SDL과 스크럼 프로세스를 이용한 통합 방법론

오민석*

*고려대학교 컴퓨터정보통신대학원 소프트웨어보안학과

Secure Agile Process for IoT System: Integrared Development Methodology using MS-SDL and SCRUM Process

minseok Oh*

*Software Security, Graduate School of Computer & information Technology,

Korea University.

요 약

최근 4차산업혁명과 더불어 스마트팩토리, 자율자동차, 드론, 스마트헬스 등의 IoT 기반 시스템 및 서비스가 확산되고 있다. 이러한 서비스는 산업 생산성을 향상하고 사용자 삶의 질을 향상 하지만 반면에 보안 공격에 대해 인명의 피해 등 보다 치명적인 결과가 발생할 수 있다. 따라서, IoT시스템과 서비스를 개발하는 전 생애주기(SDLC)에 있어서 보안 약점(Weakness)과 취약점(Vulnerability)을 제거하여 강건한 시스템을 만드는 것은 무엇보다 중요하다. 또한 개발 이후에도 추가 식별되는 보안 이슈에 대한 민첩한 대응과 패치가 무척 중요하며 이를 위해 본 논문에서는 개발 생명주기 전 과정에서 짧은 주기로 요구사항을 개발 통합하여 제품을 개발하는 Agile 방법론 중 최근 가장 각광을 받고 있는 방법론 중에 하나인 SCRUM방법론과 개발 전 생애주기에 보안활동을 정의 하고 적용하는 MS-SDL을 통합하여 보안 이슈에 보다 강건하고 운영 중에도 추가적인 보안 요구사항에 대해 민첩하게 대응 가능한 Secure Agile Process를 정의 하고자 한다.

I. 서론

컴퓨팅 파워의 지속적 성장과 하드웨어 가격 인하로, 인터넷에 그 어떤 것을 연결하더라도 경제적인 부담 없이 가능해졌다. 결국 모든 사물은 스마트해지고 인터넷에 연결되어 통신 능력이 높아 졌으며, 분석능력이 향상되어 새로운 데이터를 활용한 서비스 역시 가능해지고 있다. 이러한 IoT가 확산되면서 자원 활용의 효율성이 증가하고, 생산성이 증가하고, 새로운 비즈니스가 창출되는 등의 긍정적 효과가 생기고 있다. 그러나, Privacy 침해, 해킹/보안의 위협, 통제력의 상실 등의 심각한 문제도 같이 발생하고 있다[1]. 그럼에도 불구하고 IoT시스템의 개발에 관련해서는 최근 화두가 되고 있는 Secure SDLC등의 방법론 등의 적용이 아직 미흡한 상황이다. 또한 IoT 시스템의 이슈는 인명의 침해(자율 자동차, 드론, 스마트 헬스 등의 사례)등으로 이어 질 수 있어 보안 취약점(Vulnerability)이 발견 될 때 운영 중이라 하더라도 이에 대해 빠르게 대응하고 보완 할 수 있는 보다 민첩한 개발 방법론이 필요하다. 이에 착안하여 본 논문에서는 짧은 주기로 반복하여 요구사항을 민첩하게 반영하는 Agile방법론 중 최근 각광받고 있는 SCRUM과 전 SDLC에서 보안활동을 통해 보안 약점과 취약점을 감소시켜 보안적으로 안전한 제품을 개발하는 대표적인 방법론인 MS-SDL을 살펴보고 서로 다른 목표를 가지고 있는 이 두가지 방법론을 통합하여 보다 안전하고, 출시된 제품의 보안 기능에 대해 빠르게 대응 가능한 “Secure Agile Process”를 정의 할 것이다.

1. MS-SDL의 프로세스와 보안 요구사항의 처리 유형

1.1. MS-SDL의 프로세스

MS-SDL은 마이크로소프트사에 의해 개발된 대표적인 Secure-SDLC로 개발생명주기 전 과정에서 보안에 대한 Activity를 정의하고 적용하도록 하고 있으며 그 프로세스는 그림 1과 같다.

표1. MS-SDL의 개발 Process[3]

Process	Core Activity
Pre-SDL Requirements	-Security Training
Phase 1: Requirements	-Security Requirements -Privacy Requirements -Resources
Phase 2: Design	-Secure Design -Threat Modeling
Phase 3: Implementation	-Secure Coding -Security Testing
Phase 4: Verification	-Threat model updates -code review, -Testing -Documentation review and edit
Phase 5: Release	-Final Security Review -privacy review
Post-SDL	-Response

1.2. MS-SDL의 Agile에서의 보안요구사항 처리 방안의 유형

이러한 MS-SDL의 프로세스 상의 보안 요구사항을 Agile Process에서 처리하기 위한 방법은 One time requirements(프로젝트 중 한번만 수행하면 되는 보안 요구사항 및 활동), Every time requirements(매 Sprint 마다 수행해야 하는 보안 요구사항 및 활동), Bucket requirements(전 SDLC를 통해 궁극적으로 최종 제품이 가져야 하는 보안 요구사항 및 활동)로 분류 할 수 있다[5]. 그리고 각 유형별 주요 Activity의 사례는 표2와 같다.

표2.MS-SDL의 보안요구사항의 처리 유형 및 활동 사례[5]

유형	주요 Activity 사례
One time requirements	-보안 정책 정의 -보안 교육 수행 -Coding 가이드라인 정의
Every time requirements (매 스프린트 마다)	-보안 설계 -위협 모델링 -시큐어 코딩 -보안 테스트
Bucket requirements	-Reliability -Resilience

이러한 MS-SDL은 특히 보안 교육을 강조하고 있으며, 설계 과정에서 위협모델링을 통해 위협의 분류와 우선순위를 정의 하고 이를 기반으로 전략적인 보안 대응 가능하도록 개발 프로세스를 진행 한다[4].

2.스크럼 방법론의 프로세스 및 주요 회의

2.1. 스크럼 방법론의 프로세스

스크럼은 예측 가능성과 통제 위험을 최적화하기 위해 반복적이고 점진적으로 접근한다[2]. 2-4주의 짧은 주기로 반복하여 제품을 완성하는 프레임워크이며, SCRUM의 가장 핵심은 Sprint이다. Sprint 는 Sprint Planning, Daily Scrums, 개발 작업, Sprint Review 및 Sprint Retrospective를 포함한다[2][4]. 이를 포함한 스크럼의 전체 프로세스는 그림1과 같다.

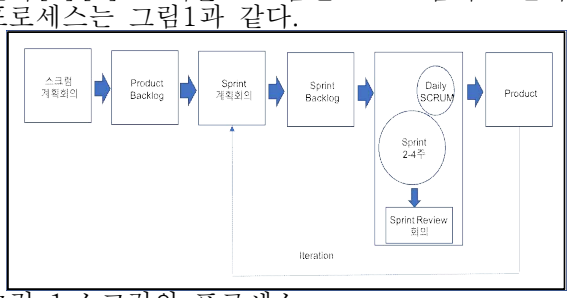


그림 1.스크럼의 프로세스

2.2. 스크럼 방법론의 주요 회의

스크럼 프로세스의 핵심 사항은 전체 시스템의 목표인 Product Backlog와 이를 증분 하여 Sprint Backlog를 정의하고 Sprint를 짧은 주기(2-4주)로 반복하며 지속적인 통합으로 최종 제품을 완성해 나가는 것이다. (Continuous Integration). 이러한 프로세스를 통해 서론에서 설명한 IoT시스템에 대한 빠른 보안 요구사항의 반영과 패치가 효율적이고 효과적으로 가능하다. 또한 스크럼 방법론에서 이러한 프로세스 단계별 Activity를 수행하기 위해 중요한 것은 각 프로세스 단계의 목표나 요구사항의 정의 및 이슈의 체크 등을 다양한 회의를 통해 진행 하는 것이다. 그 회의의 종류는 표3와 같다.

표3.SCRUM의 Meeting 종류

회의 종류	설명
SCRUM 계획 회의	프로젝트의 목표 요구사항 정의
Sprint 계획 회의	Sprint Backlog 정의
Sprint 리뷰 회의	스프린트 데모 수행
Daily meeting	매일 15분 이내 이슈 식별
Retrospective	각 스프린트에 대한 회고 수행

3. MS-SDL과 SCRUM 통합 방안

3.1. MS-SDL과 SCRUM 주요 목표

MS-SDL과 SCRUM은 서로 다른 주요 목표를 가지고 있다. 서로 다른 목표를 가진 이 두가지 방법론을 통합하기 위해서는 두가지 방법론의 목표를 정확히 알고 통합 시 각 방법론의 목표가 희석되지 않도록 하는 것이 중요한다. 먼저 MS-SDL의 주요 목표는 개발 생명주기 전 과정에서 보안 활동을 정의하고 보안 약점과 보안 취약점을 줄이는 것이다. 다음으로 SCRUM방법론의 목표는 짧은 주기의 반복을 통해 요구사항을 빠르고 민첩하게 반영하고 빅뱅 통합의 이슈를 제거하여 품질 높은 소프트웨어를 개발하는 것이다.

3.2. MS-SDL과 SCRUM방법론의 통합 프로세스

위에서 살펴본 봐와 같이 두 방법론의 핵심 목표를 희석하지 않고 활용하여 보안에 강건하고 품질 높은 IoT시스템을 개발하기 위한 방법론을 위해 그림[2]과 같은

프로세스를 통해 두 방법론을 통합한다.

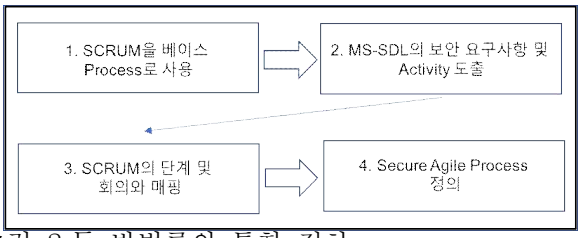


그림 2.두 방법론의 통합 절차

우선 스크럼 방법론을 베이스 방법론으로 정하고 MS-SDL에서 도출한 세가지 유형의 보안 요구 사항과 보안 활동을 스크럼의 각 프로세스 및 주요 회의와 매핑하여 통합된 Secure Agile Process를 정의 한다. 그림[3]는 최종적으로 통합된 Secure Agile Process이다.

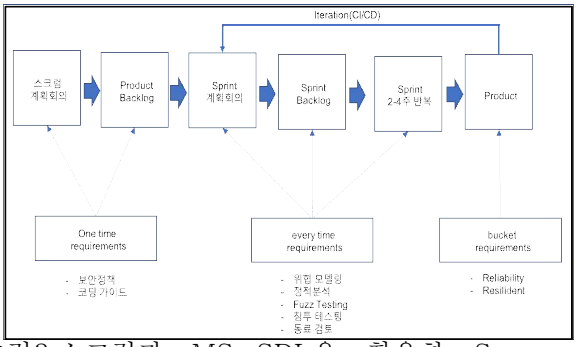


그림3.스크럼과 MS-SDL을 활용한 Secure Agile Process

II. 결론

본 논문에서는 최근 각광을 받는 Agile 방법론인 SCRUM과 대표적인 Secure-SDLC인 MS-SDL를 살펴보고 이 두가지 방법론을 통합하여 보안 요구사항에 대한 민첩한 반영과 패치가 가능한 IoT 시스템 및 서비스를 위한 Secure Agile Process를 정의 하였다. 이렇게 정의된 Secure Agile Process를 통해 보다 강건한 IoT시스템을 개발 가능하며 Release 이후에도 추가 보안 요구사항 및 패치 등의 Agility한 대응이 가능하다.

[참고문헌]

[1]Klaus Schwab, “The Fourth Industrial Revolution”

[2] Jeff Sutherland, Ken Schwaber, “SCRUM GUIDES, http://www.scrumguides.org/index.html, 2016

[3]https://www.microsoft.com/en-us/SDL.

[4] Official ISC2 Guide to the CSSLP CBK Second Edition.

[5] Microsoft Security Development Lifecycle for Agile Development Version 1.0 June 30, 2009

휴리스틱 백신의 탐지기법 및 전망

장서연[1], 김도연[1], 이제원[1], 이윤진[1], 이승연[1], 김지훈[2] 김종성[1,2]

1) 국민대학교 정보보안암호수학과

2) 국민대학교 금융정보보안학과

Seoyeon Jang[1], Doyeon Kim[1], Jewon Lee[1], Yunjin Lee[1],

Seungyeon Lee[1], Jihun Kim[2] Jongsung Kim[1,2]

1) Dept. of Information Security, Cryptology and Mathematics,

Kookmin University.

2) Dept. of Financial Information Security, Kookmin University

요 약

다양한 악성코드들이 생겨나고 있는 오늘날 기존의 백신 프로그램만으로는 더 이상 악성코드를 대응하기 힘들어졌다. 이러한 문제 해결을 위해 인공지능과 백신을 접목시킨 휴리스틱 백신이 사용된다. 휴리스틱 백신은 스캐닝 엔진의 형태에 따라 정적 휴리스틱, 동적 휴리스틱, 네거티브 휴리스틱 기법으로 분류된다. 인공지능으로 악성코드들을 판단함으로서 악성행위를 사전에 방지할 수 있는 휴리스틱 백신으로 인해 더욱 효과적인 보안 서비스를 제공받을 수 있게 되었다. 본 논문에서는 휴리스틱 백신의 정의 및 장단점을 비교하고, 더 나아가 앞으로의 전망에 대하여 서술한다.

I. 서론

현재 랜섬웨어와 같은 악성코드로 해커들이 2016년 동안 벌어들인 수익은 약 8억 5000만 달러에 육박하는 것으로 추정되고 있다[3]. 이러한 악성코드의 피해를 막기 위해서 주로 백신 프로그램을 사용한다. 백신은 컴퓨터 내부를 탐색해 위협요소를 정리해서 일종의 자정작용을 하는 프로그램으로, 대부분의 백신 프로그램은 시그니처 기반(Signature Base)의 진단법을 사용한다. 시그니처 기반의 진단법은 악성 코드로 분류된 파일의 특정 부분 또는 고유한 부분을 이용해 악성코드를 진단하는 방법이다[1]. 그런데 새로운 구조의 다양한 악성코드가 매일 100만 개 이상 출몰하고 있기 때문에 기존의 시그니처 기반의 진단으로는 다양한 악성코드를 진단하는데 무리가 있다[4]. 이러한 문제를 극복하고자 개발한 탐지 기법 중 하나가 휴리스틱(Heuristic) 탐지 기법이다.

휴리스틱이란 불확실하고 복잡한 상황에서 부딪히는 문제를 가능한 한 빨리 풀기 위해 쓰는 직관적인 판단, 경험과 상식에 바탕을 둔 단순하고 즉흥적인 추론을 뜻한다[5]. 즉, 가상의 환경에서 악성코드의 동작을 미리 파악하고 이를 이용해 악성코드를 탐지하는 기술이 휴리스틱 탐지 기법이다. 본 논문에서는 휴리스틱 백신의 정의와 탐지 기법에 대해 정리하고 휴리스틱 백신에 대한 중요성에 대하여 서술한다.

본 논문의 구성은 다음과 같다. 2장에서는 휴리스틱의 정의에 대해, 3장에서는 동적, 정적 휴리스틱 탐지 기법 그리고 네거티브 휴리스틱 탐지 기법에 대해 설명한다. 그리고 4장에서는 휴리스틱 탐지 기법이 갖는 장단점과 단점에 대한 해결방안에 대해 알아본다. 마지막으로 휴리스틱 백신에 대한 앞으로의 전망에 대해 살펴본 후 5장에서 결론을 맺는다.

II. 휴리스틱 백신의 정의

휴리스틱의 사전적 의미는 '시행착오를 겪어 해결법을 찾는다.'이며, 실제로 심리학, 경제학, 철학과 같은 여러 학문에서는 '어림짐작하여 시간 등의 비용을 줄이고 해결법을 찾는 방식'이라는 의미를 갖고 있다. 이런 방식을 이용한 휴리스틱 탐지 기법으로 컴퓨터 내 위협요소를 탐색하는 백신 프로그램이 바로 휴리스틱 백신이다.

휴리스틱 탐지 기법은 휴리스틱 시그니처와 스캔한 파일을 인공지능이 비교해 그 유사도를 측정하거나 가상 공간에서 파일을 미리 실행해봄으로써 악성코드 여부를 판단한다. 휴리스틱 시그니처(Heuristic Signature)란 특이한 코딩 기법이나 의심의 여지가 있는 파일의 행위, 구문, 또는 '특정 폴더에 파일 쓰기', '특정 레지스트리에 있는 키 생성'과 같은 명령어들 등을 나타낸 시그니처를 뜻한다.

III. 휴리스틱 백신의 탐지기법

휴리스틱 탐지 기법은 일반적인 악성 코드가 가지고 있는 특정 폴더에 파일 쓰기와 특정 레지스트리 부분에 키 생성과 같은 명령어(Instruction)들을 스캐닝 엔진(Scanning Engine)에서 시그니처화하여 파일을 검사할 때 이를 사용한다. 그래서 검사 대상 파일이 휴리스틱 시그니처의 비교를 통하여 일반적으로 알려진 악성 코드와 얼마나 높은 유사도를 가지고 있는지를 판단하여 알려지지 않은 새로운 악성 코드를 탐지하는 기법이다. 휴리스틱 탐지 기법은 검사 대상이 되는 파일에 대하여 휴리스틱 검사 시에 스캐닝 엔진이 어떠한 형태를 취하는가에 따라서 다음과 같이 나누어 볼 수 있다.

3.1 동적 휴리스틱 탐지 기법

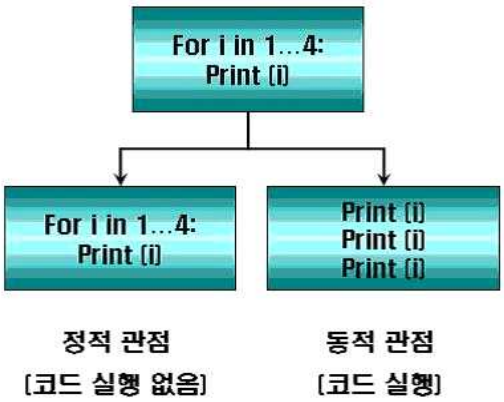
동적 휴리스틱 탐지 기법(Dynamic Heuristic Detection Tech)은 검사 대상이 되는 파일을 가상화된 영역 또는 운영체제와 분리된 특정 공간에서 파일 실행(File Execution)을 통해 수집한 다양한 정보들을 휴리스틱 시그니처와 비교를 통하여 검사를 수행하는 기법이다. 즉 검사 대상이 되는 파일이 어떠한 형식으로든 실행이 되어야 하며 이러한 점으로 인해 런 타임 휴리스틱 탐지(Run-Time Heuristic Detection) 기법이라고도 불리고 있다.

3.2 정적 휴리스틱 탐지 기법

정적 휴리스틱 탐지 기법(Static Heuristic Detection Tech)은 동적 휴리스틱 탐지 기법과는 반대의 개념으로, 파일에 대해서 어떠한 형식의 파일 실행 없이 파일 자체를 그대로 스캐닝 엔진에서 읽어 들여 휴리스틱 시그니처와 비교를 수행하는 탐지 기법이다.

3.3 네거티브 휴리스틱 탐지 기법

앞서 언급한 동적 휴리스틱 탐지 기법, 정적 휴리스틱 탐지 기법과 달리, 스토퍼(Stopper)라고도 하는 네거티브 휴리스틱 탐지 기법(Negative Heuristic Detection Tech)은 악성 코드에서 일반적으로 사용하지 않는 '윈도우 팝업 창 생성'과 같은 명령어들을 탐색하며 악성코드인지를 판단하는 탐지 기법이다 [2]. 즉, 휴리스틱 시그니처와 동일한 명령어들이 존재하지 않을 때 악성코드로 판단하는 기법이다.



[Fig. 1] Static Heuristic and Dynamic Heuristic[1].

IV. 휴리스틱 백신의 장단점

4.1 휴리스틱 백신의 장점

휴리스틱 백신의 장점은 기존에 알려지지 않은 악성코드를 진단할 수 있다는 점이다. 변형된 패턴의 바이러스를 탐지하지 못해 탐지율이 상당히 떨어졌던 기존의 지문 대조 방식[1]과는 달리, 휴리스틱 백신은 데이터베이스에 악성 코드 목록이 올라와 있지 않아도 가상으로 실행해보기 때문에 기존에 알려지지 않은 새로운 악성코드 역시 탐지가 가능하다.

따라서 잠재적인 위협에도 대비할 수 있다. 새로운 악성코드도 진단할 수 있기 때문에 잠재적인 변종 악성코드를 막아낼 가능성이 높다.

4.2 휴리스틱 백신의 단점

휴리스틱 백신은 아직 여러 문제점을 가지고 있는데, 그 첫 번째는 백신의 용량이 커진다는 점이다. 휴리스틱 백신은 검사 대상 파일을 실행하기 위한 가상화 영역을 구성해야 하는데, 이를 위해서 컴퓨터 자원을 많이 소모하게 된다. 백신의 용량이 커지게 되면 사용자가 이용하기에 큰 부담이 될 수 있다.

두 번째는 탐지에 걸리는 시간이 오래 걸린다는 점이다. 보안 분야의 특성상 백신의 실시간 탐지가 중요하다. 그러나 휴리스틱 백신은 일반 백신에 비해

1) 엔진에 적용된 내용과 일치하는 파일의 이름, 규격, 활동 프로세스의 범위를 바이러스 백신이 분석해서 추출하는 기법. 정해진 패턴대로 수행하기 때문에 탐지된 바이러스에 대해서는 치료가 거의 확실하지만 조금만 변형된 바이러스가 나올 경우 탐지할 수 없기 때문에 탐지율이 상당히 떨어진다는 단점이 있음.

탐지를 위한 계산량이 많기 때문에 탐지시간이 오래 걸려 실시간 탐지가 어렵다.

세 번째는 오탐률이 크다는 점이다. 휴리스틱 백신을 사용할 경우, 지문대조 방식을 사용했을 때보다 새로운 악성코드를 탐지할 확률이 높아진다. 그러나 그만큼 정상 파일을 악성코드로 판단하는 오탐이 발생할 확률도 증가한다. 파일을 삭제 및 치료하는 안티 바이러스 소프트웨어 특성상 오탐률이 높을 경우 컴퓨터 시스템 자체에 큰 문제를 유발할 수 있다.

그리고 완전히 새로운 코드로 작성된 악성코드이거나 악성코드가 정상파일과 유사한 행위를 하는 경우 진단이 어렵다는 단점을 가지고 있다.

4.3 휴리스틱 백신 단점의 해결방안

휴리스틱 백신의 탐지속도가 느리다는 점의 해결방안으로 기계학습 알고리즘의 성능을 향상시키는 방법과 더불어 데이터마이닝 알고리즘을 적용하는 방법이 있다.

휴리스틱 백신의 오탐률이 높다는 점을 해결하기 위한 방법으로 기계학습 알고리즘의 분류성능을 향상시키는 방법이 있다. 최근의 조사에 따르면 휴리스틱 악성코드 탐지 기법들에 관련된 여러 방면의 연구들이 진행되고 있음을 알 수 있다[6]. 그 예로 하이파이 머신러닝 기술을 적용하여 실행 전, 실행 중인 파일을 분석하고 센서스 검사, 화이트리스팅 등 '잡음제거' 기능을 활용할 수 있다.

정상파일과 유사한 행위를 하는 악성코드의 경우에는 네거티브 휴리스틱 기법에 대한 연구를 통해 해결해 나갈 수 있다. 그러나 완전히 새로운 코드로 이루어진 악성코드의 경우에는 인공지능이 탐지하기 힘들다. 따라서 새로운 악성코드[를 꾸준히 발견하고, 발견된 악성코드에 대한 머신러닝을 빠르게 구현함으로써 대처할 수 있다.

또 다른 문제인 가상화 영역 구성을 위한 파일들의 크기를 줄여나가는 부분에 있어서는 앞으로의 많은 연구들과 기술의 개발을 필요로 한다.

V. 결론

휴리스틱 백신은 머신러닝 기술과 보안이 접목된 기술로서 스캐닝 엔진의 형태에 따라 탐지 기법들이 분류된다. 본 논문에서는 이러한 휴리스틱 백신의 탐지 기법들을 알아보고, 휴리스틱 백신의 장단점 및 단점의 해결방안에 대해서 서술했다.

휴리스틱 백신은 아직 오탐률이 높다는 단점을 가지고 있지만 머신러닝 분야의 연구가 꾸준히 이루어지고 있음에 따라 성능이 향상될 것으로 기대된다. 또한 다양한 악성코드들이 늘어나고 있음에 따라 머신이 더 많은 학습을 할 수 있는 샘플들이 증가하고 있어 시간이 지날수록 오탐률은 점차 떨어질 것으로 예상된다.

하루에도 무수히 많은 악성코드들이 생겨나고 있는 오늘날에 정해진 패턴으로 악성코드들을 탐지, 차단하는 기존의 백신방법은 더 이상 효율적으로 악성행위를 방어하기가 힘들다. 그렇기에 머신러닝 기술을 활용해 데이터를 학습하는 방법을 이용하고 '패턴'을 사용하지 않는 휴리스틱 백신의 사용은 필수적이게 되었다[7]. 기존의 백신과 휴리스틱 백신이 적절히 조화되어 사용된다면, 서로의 단점을 극복하고 보다 더 안전한 보안환경을 제공하며 더욱 효율적이고 효과적으로 악성코드에 대처할 수 있을 것이다. 그러므로 앞으로 휴리스틱 기능을 탑재한 백신들의 영향력이 더욱 커질 것으로 보인다.

[참고문헌]

[1] 안랩, "파일기반의 탐지기법"
http://www.ahnlab.com/kr/site/securityinfo/secunews/secuNewsView.do?curPage=27&seq=9530

[2] John aycock, "Computer Viruses and Malware", p.69-70 ,
http://academic.naver.com/article.naver?doc_id=116159825

[3] IT world, " "랜섬웨어 공격 지속될 것"…이스트시큐리티, '2017년 예상 보안 이슈 톱 5 발표" ,
http://www.itworld.co.kr/news/102923

[4] 조선비즈, "악성코드, 매일 100만 개 생겨… 전 세계 피해액 한 해 1조원" ,
http://biz.chosun.com/site/data/html_dir/2017/02/06/2017020600027.html#csidxcd0e9e4232c9bd68b5b0abaa62c31ef

[5] 네이버 지식백과, "주먹구구는 얼마나 믿을 수 있나?",
http://terms.naver.com/entry.nhn?docId=3570644&cid=58783&categoryId=58783

[6]Reference: Bazrafshan, Zahra, et al. "A survey on heuristic malware detection techniques." Information and Knowledge Technology (IKT), 2013 5th Conference on. IEEE, 2013,
http://academic.naver.com/article.naver?doc_id=182283587

[7] 아이뉴스, "컴퓨터 백신, 인공지능(AI)으로 달라질까" ,

http://news.inews24.com/php/news_view.php?g_serial=996016&g_menu=020200&rrf=nv

프라이버시를 보호하는 IEC 62304 의료기기 소프트웨어 개발 생명주기*

배경진†, 이동훈‡

고려대학교 정보보호대학원

Protecting Privacy IEC 62304 Medical Device Software Development Lifecycle*

KyungJin Bae†, Dong Hoon Lee‡

Graduate School of Information Security, Korea University.

요 약

IEC 62304는 의료기기 소프트웨어의 기능 안전성을 확보하기 위한 의료기기 소프트웨어 개발 생명 주기에 대한 국제 표준이다. IEC 62304를 준수하여 소프트웨어를 개발하는 경우 높은 수준의 기능 안전을 확보할 수 있다. 그러나 IEC 62304를 준수한 소프트웨어를 사용하더라도 의료기기로부터 발생하는 개인의료정보의 프라이버시를 보장하지 않는다. 개인의료정보는 개인의 민감한 정보를 포함하고 있어 프라이버시가 보호되어야 하며, 개인정보에 대한 노출로 인해 개인에게 직접적인 피해를 입힐 수 있다. 따라서 본 논문에서는 프라이버시 관점에서 IEC 62304 소프트웨어 개발 생명주기가 갖는 문제점을 분석하고, 프라이버시를 보호하는 IEC 62304 소프트웨어 개발 방안를 제안한다.

I. 서론

최근 정보통신기술(ICT)과 의료 산업의 융합으로 유헬스, 로봇 수술 등 의료기기가 점차 지능화됨에 따라 의료기기에서 소프트웨어가 핵심적인 역할을 하고 있다. 의료기기 소프트웨어는 의료기기에 내장된 소프트웨어뿐 만아니라 소프트웨어 자체가 의료기기로 사용된다.

의료기기에서 소프트웨어의 역할이 커짐에 따라 여러 문제점도 증가하고 있다. 의료기기에서 소프트웨어의 결함은 환자의 생명과 직결될 수 있기 때문에 소프트웨어의 기능 안전성은 최우선적으로 고려되어야 한다. 하지만 의료정보는 환자에 의해 작성되는 주관정보, 검사 및 진료 등 에 의한 객관정보, 그리고 의료인의 전문성에 기초하여 작성된 가치판단정보 등으로 포함하는 정보의 종류가 다양하다. 이 중 개인식별성을 지닌 정보에 대해 개인의료정보라 한다[2]. 개인의료정보에는 고유식별번호, 생체정보 등 개인의 민감한 정보를 포함하기 때문에 의료기기 소프트웨어 개발 시 프라이버시 속성 또한 중요하게 고려되어야 한다. 의료기기 소프트웨어에서 개인의료정보의 프라이버시가 보호되지 않는다면 환자 개개인의 개인정보침해가 발생할 수 있다.

국제 표준인 IEC 62304를 준수하여 의료기기 소프트웨어를 개발하는 경우 높은 수준의 기능 안전을 확보할 수 있다. 하지만 기능 안전은 프라이버시와는 다른 속성이며, 의료기기 소프트웨어 개발 시 IEC 62304를 준수하여도 프라이버시를 확보할 수는 없다. 기능 안전이 잘

* 이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2017-0-01863, 함수암호 기법 설계·분석 및 구현기술 연구)

† 주저자, kjna0827@korea.ac.kr

‡ 교신저자, donghlee@korea.ac.kr

갖추어진 소프트웨어라 하더라도 프라이버시가 보호되지 않는다면 환자에게 직접적인 피해를 입힐 수 있다. 원격진료의 경우 개인정보를 탈취하여 임의로 개인의료정보를 생성하는 것은 환자의 생명과도 직결된다[1]. 따라서 의료기기 소프트웨어 개발 생명주기 전체에 걸쳐서 프라이버시를 함께 고려할 필요가 있다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 배경지식으로 IEC 62304와 소프트웨어 터치포인트 방법론을 설명한다. Ⅲ장에서는 프라이버시를 보호하기 위한 비식별화 방법을 소개한다. 그리고 프라이버시 관점에서 IEC 62304가 갖는 문제점을 분석하고, 그에 대한 해결 방안을 제안한다. Ⅳ장에서 결론을 맺으며, 향후 연구 방향을 제시한다.

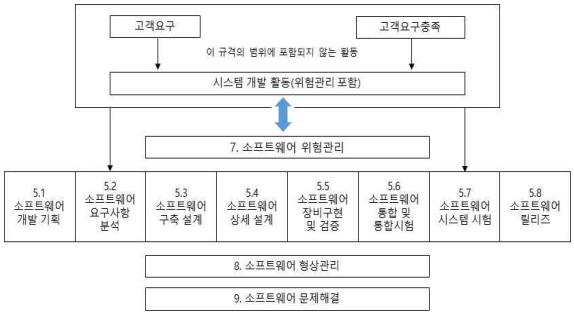
II. 배경지식

본 장에서는 IEC 62304와 제안하는 해결방안에 적용될 소프트웨어 터치포인트(touchpoint) 방법론에 대해 설명한다.

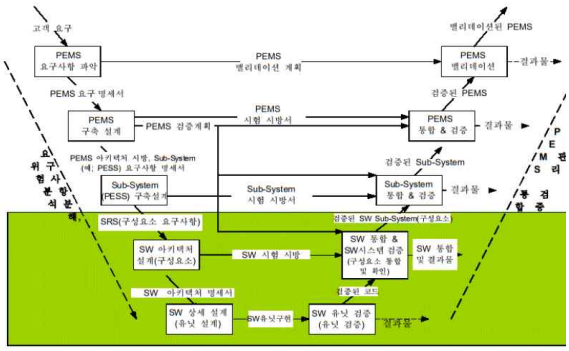
2.1 IEC 62304

IEC 62304는 2006년 5월 9일 JTC 1 SC 62A JWG에서 발간한 의료기기에 적용할 수 있는 소프트웨어 개발 생명주기에 대한 국제 표준이다[7]. 이 표준은 의료기기 소프트웨어의 요구사항의 정의에서부터 제조를 위한 릴리즈(release)에 이르기까지 전체 생명주기에서 의료기기 소프트웨어의 안전한 설계와 운용 및 유지보수에 필요한 활동과 업무를 정의한다. IEC 62304에 정의된 소프트웨어 개발 프로세스 및 활동의 개요는 [그림 1]과 같다[3].

[그림 2]는 PEMS(프로그래밍이 가능한 전기의료기기, Programmable Electrical Medical System)의 개발 과정과 현상을 설명하기 위한 V 모델로서, 소프트웨어 요구사항의 설정에서부터 소프트웨어 단위가 소프트웨어 시스템에 통합할 때까지 적용되는 흐름을 도해한 것이다[4].



[그림 1] 의료기기 소프트웨어 개발 프로세스 및 활동의 개요



[그림 2] V모델의 소프트웨어 개발 생명주기

2.2 소프트웨어 터치포인트 방법론

2004년 Gray McGraw가 제안한 소프트웨어 터치포인트 방법론은 소프트웨어의 보안성을 강화할 수 있는 7가지 터치포인트를 설명한다[6]. 소프트웨어 개발 활동 중에 보안과 관련된 활동을 구분하여 이를 터치포인트라 하며, 기존에 존재하는 소프트웨어 개발 생명주기의 단계마다 생성되는 산출물(artifacts)에 터치포인트를 적용함으로써 보안성을 강화시킬 수 있다. 터치포인트는 유효성(effectiveness)과 중요성에 따라 순위가 매겨져 있으며, [표 1]과 같다.

[표 1] 소프트웨어 보안성 터치포인트

번호	터치포인트
1	정적 분석 도구를 사용한 코드 리뷰(code review)
2	구조적 위험 분석(architectural risk analysis)
3	침투 테스팅(penetration testing)
4	위험 기반의 보안 테스팅(risk-based security testing)

5	악용 사례(abuse cases)
6	보안 요구사항(security requirement)
7	보안 운영(security operation)

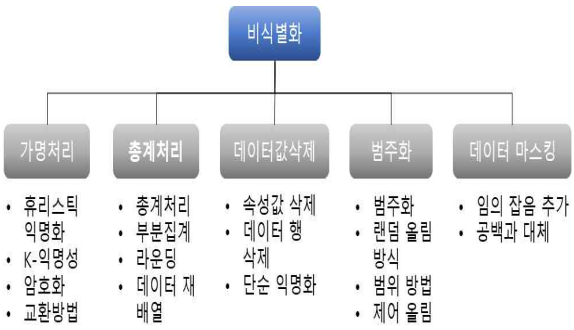
III. IEC 62304의 문제점과 그에 대한 해결방안

본 장에서는 프라이버시를 보호하기 위한 비식별화 방법을 설명한다. 그리고 기존의 IEC 62304를 분석하여 프라이버시 관점에서 문제점을 분석하고, 터치포인트 방법론을 적용하여 프라이버시 속성을 강화하는 IEC 62304 소프트웨어 개발 생명주기를 제안한다.

3.1 프라이버시를 보호하기 위한 비식별화

개인의료정보의 프라이버시를 보호하기 위해서는 정보를 삭제하거나 제 3자 제공 등 활용에 대한 이용자 동의를 받은 경우를 제외하고는 개인의료정보에서 개인식별성이 제거되는 비식별화 과정을 거쳐야한다. 비식별화란 데이터 내에 개인을 식별할 수 있는 정보가 있는 경우, 이의 일부 또는 전부를 삭제, 또는 일부를 속성 정보로 대체 처리하는 등 다양한 방법을 통해 개인을 식별하기 어렵게 만드는 조치를 일컬으며, 현재 개인정보 보호방법 중 최선의 방법으로 여겨지고 있다[5].

비식별화의 주요 기술로는 [그림 3]에서와 같이 크게 5가지 유형으로 분류할 수 있다. 개인정보 중 주요 식별요소를 다른 값으로 대체하여 개인식별을 어렵게 하는 방법인 가명처리, 데이터의 총합 값을 보임으로써 개별 데이터의 값을 보이지 않도록 하는 총계처리, 데이터 공유 및 개방 목적에 따라 데이터세트에 구성된 값 중에 필요없는 값 또는 개인식별에 중요한 값을 삭제하는 데이터 값 삭제, 데이터 값을 범주의 값으로 변환하여 명확한 값을 감추는 범주화, 마지막으로 공개된 정보 등과 결합하여 개인을 식별하는데 기여할 확률이 높은 주요 개인식별자가 보이지 않도록 처리하여 개인을 식별하지 못하게 하는 데이터 마스킹 방법이 있다. 이 중 가명처리를 하는 방법 중 하나로 암호화 방법이 있으며, 이 때 함수암호 기법을 이용하는 경우 클라우드 컴퓨팅에서도 안전한 암호화 및 데이터 가공을 지원받을 수 있다.



[그림 3] 개인정보 비식별화 방법

3.2 기존의 IEC 62304의 문제점과 터치포인트를 적용한 IEC 62304

소프트웨어 개발 기획 활동에서는 개발하려는 소프트웨어 시스템의 범위, 규모 및 소프트웨어 안전에 적합한 개발 활동 진행을 위하여 소프트웨어 개발 계획을 수립하여야 한다. 기존의 IEC 62304에서는 이 단계에서 데이터 비식별화 과정을 계획하고 있지 않다. 소프트웨어 프라이버시 속성을 강화하기 위해서는 소프트웨어 개발 기획 활동에서부터 데이터 비식별화 과정을 기획해야 한다. 어떤 비식별화 방법을 택할지 처리해야할 정보에 따라 선정해야 하며, 이로부터 소프트웨어 요구사항을 수립할 수 있다.

소프트웨어 요구사항이란 소프트웨어가 어떤(what) 기능을, 어떻게(how) 수행할 것 인지에 대하여 설명한 것으로, 물리적 및 논리적인 보안 요구사항 명세하위 단계에서 데이터 비식별화를 위한 요구사항을 명세해야 한다. 기존의 IEC 62304에서는 이에 대해 명세하고 있지 않다. 명세된 요구사항은 소프트웨어 구조 설계와 단위 설계 및 구현을 할 때에도 영향을 미치므로 프라이버시 속성을 강화하기 위해서는 데이터 비식별화를 위한 요구사항을 상세히 명세해야 한다. 요구사항을 작성할 때에는 악용 사례를 활용할 수 있다. 악용 사례는 시스템이 공격을 받을 때의 행동을 기술한 것으로, 어떤 것이 무엇으로부터 얼마 동안 보호되어야 하는지 등

에 대해 구체적으로 기술되어야 한다.

소프트웨어 구조 설계 및 검증 단계에서는 기존의 소프트웨어 안전 요구사항 뿐 만 아니라 비식별화 요구 사항도 함께 실현되도록 소프트웨어 구조 설계 및 구조적 위험 분석을 수행해야 한다. 구조적 위험 분석은 소프트웨어 개발 생명주기의 각 단계에 존재하는 위험을 식별하고 순위화하며, 공격 저항 분석, 모호성 분석(ambiguity analysis), 보안약점 분석(weakness analysis) 등이 있다.

소프트웨어 상세 설계 및 단위구현 단계에서는 소프트웨어 단위가 소프트웨어 비식별화 요구사항과 일치하는지 검증한다. OWASP 2017에 따르면 소프트웨어 단위 설계 및 구현을 위한 고려해야할 사항이 제시되어 있다. 그 중 민감 데이터 노출항목을 비식별화를 위한 요구사항과 함께 검증한다[9]. 임베디드 시스템에서 소프트웨어 구현의 안전성과 호환성, 신뢰성은 MISRA C 코딩 가이드라인을 적용하여 달성할 수 있다[8].

소프트웨어 검증 및 밸리데이션 단계에서는 소프트웨어 단위 테스팅, 통합 테스팅, 시스템 테스팅, 사용자 현장 시험(User Site Testing)을 포함한다. 소프트웨어 단위 테스팅 하위 단계에서는 위험 기반의 테스팅을 수행한다. 이 테스팅은 기능적 테스팅과 분석된 위험에 기반하여 공격자의 입장에서 수행하는 적대적(adversarial) 테스팅 모두를 포함한다. 소프트웨어 통합과 시스템 테스팅 하위 단계에서는 PII(Personally identifiable information) 테스트를 위해 OWASP Testing Guide v4를 참고하여 PII 데이터에 대해 테스팅을 수행해 검증한다[10]. 또한 터치포인트에서 제시하는 침투 테스팅을 시스템 하위 단계에서 수행한다.

소프트웨어 릴리즈 단계에서는 보안 운영 터치포인트를 적용하여 소프트웨어를 릴리즈하기 전에 모든 검증 활동을 만족스럽게 완료하였는지에 대해 지정된 권한자의 승인여부를 확인하여야 한다. 소프트웨어 릴리즈 후 발생하는 피드백은 소프트웨어 유지보수 활동으로 대응한다.

IV. 결론

본 논문에서는 의료기기 소프트웨어 개발 생명주기 국제 표준인 IEC 62304를 프라이버시 관점으로 문제점을 분석하고, 이를 해결하기 위해 개발 생명주기 각 단계마다 터치포인트를 적용하여 프라이버시를 보호하는 의료기기 소프트웨어 개발 생명주기를 제안하였다.

IEC 62304는 의료기기 소프트웨어 개발 생명주기에 대한 표준으로, 개발 생명주기에만 터치포인트를 적용하는 것은 소프트웨어 수준보다 상위 수준에서의 프라이버시를 확보할 수 없다. 의료기기 소프트웨어 안전성을 향상시키기 위해서는 위험관리, 품질관리, SW엔지니어링을 모두 고려해야하는 것처럼 높은 수준의 프라이버시를 제공받기 위해서는 의료기기 관리 표준인 IEC 14971, IEC 13485와 의료기기 제품 규격인 IEC 60601-1, IEC 61010-1에서 각각 시스템 레벨, 아이템 레벨에서의 프라이버시 분석을 수행해야할 것이다. 현재 의료분야는 빅데이터의 활용가치가 큰 분야로 기대되는 만큼 위 문제를 해결하는 것은 향후 좋은 연구주제가 될 것이다.

[참고문헌]

[1] 이희주, 남초이, 윤장호, 윤준섭, 이호진, 김진숙, 김석영, 최재욱, 이경호, “한국형 원격의료체계 기술적 안전성 평가 체계 수립에 대한 연구,” Journal of the Korean Medical Association, 58(12), pp. 1159-1170, 2015년 12월.

[2] 정영철, “의료분야 빅데이터 활용을 위한 개인정보 비식별화 규정 현황과 과제,” 보건복지포럼, 2015년 9월.

[3] 오영배, 정영은, 신석규, “의료기기 소프트웨어 기능 안전성 기술 및 표준화 동향,” TTA 저널 2013(147), 2013.

[4] 식품의약품안전청, 의료기기 소프트웨어 밸리데이션 가이드라인, 2007.

[5] 고 학 수, 최경진, “개인정보의 비식별화 처리가 개인정보보호에 미치는 영향에 관한 연구,” 2015년 12월.

[6] Gary McGraw, Software Security: Building Security In, 2006.

[7] IEC 62304/FDiS(2006) Medical device software- Software life cycle processes.

[8] MISRA, “MISRA C:2012 Amendment 1-Additional security guidelines for MISRA C:2012,” 2016.

[9] OWASP Top 10 - 2017 rcl, http://www.owasp.org

[10] OWASP Testing Guide v4 Table of Contents, https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents

보안 취약점 정보 공유를 위한 STIX 데이터 모델링 및 변환 기술 연구

고은혜*, 김태은*, 김환국*

*한국인터넷진흥원

Study on STIX Data Modeling and Conversion Technology for Sharing Vulnerabilities Information

Eunhye Ko, Taeeun Kim, and Hwankuk Kim

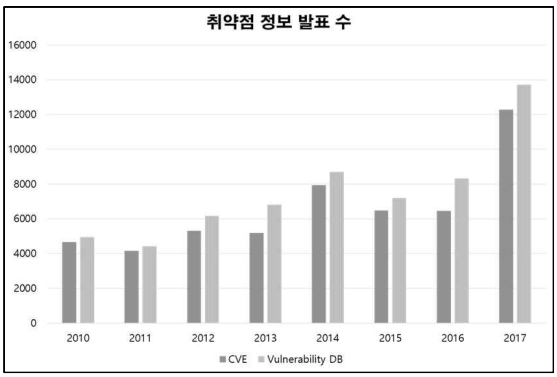
*Korea Internet and Security Agency

요 약

오픈소스 소프트웨어의 취약점 정보가 점점 늘어나고, 알려진 취약점을 악용한 공격을 막기 위해 취약점 관련 사이버 위협 정보 공유의 필요성이 증가하고 있는 추세이다. 이런 취약점 정보를 관리하고 공유하기 위해, 표준화된 사이버 위협 정보 표현 규격인 STIX를 활용하고자 한다. 본 논문에서는 STIX를 활용하기 위한 데이터 모델링과 변환 기술을 제안한다.

I. 서론

최근 다양한 디바이스의 증가와 함께 취약점도 급격하게 발생하고 있다. <그림 1>에서 확인할 수 있는 것처럼, 주요 취약점 정보 발표 사이트인 National Vulnerability Database (NVD)와 Vulnerability Database, Bugtraq 등의 취약점 정보 발표 현황을 살펴보면 취약점이 계속 증가하는 추세임을 확인할 수 있다.



<그림 1> 취약점 정보 발표 현황 [1,2,3]

더욱이 다양한 IoT 디바이스들의 확산으로 IoT 디바이스에서 활용되는 Open Source Software (OSS)의 알려진 취약점을 악용한 공격이 증가할 것으로 예상된다[4]. 또한 포티넷의 조사 보고서에 따르면, 90% 이상의 조직이 발표된 지 3년 이상 된 취약점을 악용한 공격을 당했다고 한다[5]. 이러한 취약점을 악용한 공격을 방지하기 위하여, 알려진 취약점에 대한 대응 및 관련 정보의 빠른 공유가 필요하다.

본 논문에서는 취약점 정보의 신속한 공유를 위해, 국제 표준 규격으로의 변환 및 사용자 정의 방안을 제안하고자 한다. 논문의 구성은 다음과 같다. 2장에서는 위협 정보 공유 관련 국내외 동향 등을 기술한다. 3장에서는 취약점 정보의 모델링을 통해 표준 규격 변환 및 사용자 정의 방법을 제안하였다. 4장에서는 결론과 함께 향후 연구 계획을 서술하였다.

II. 국내외 동향

현재 해외에서는 사이버 위협 정보의 효과적인 공유를 위해 표준화된 규격을 활용하는 추세이다. 표준화된 규격은 Incident Object Description Format (IODEF), Open Indicators of Compromise (OpenIOC) 등이 있다[6,7]. 하지만 본 논문에서는 현재 Organization for the Advancement of Structured Information Standards (OASIS)에서 표준화 작업이 활발하게 진행되고 있는 Structured Threat Information Expression (STIX)에 대하여 다루도록 하겠다[8].

STIX는 사이버 위협 정보 표현을 위한 표준 규격이며, 자동화된 위협 교환과 검색 및 대응 등의 기능을 개선하고자 설계되었다. 이러한 STIX는 IBM X-Force Exchange 등의 여러 플랫폼에서의 도입이 이루어지고 있다.

국내에서는 이와 유사한 형태로 위협 정보 공유 플랫폼인 C-TAS에서 C-TEX라는 표현 규격을 자체적으로 정의하여 정보를 공유하는 데 활용하였다. 최근에는 우리나라에서도 국제적인 움직임에 맞춰, 여러 기관과 보안업계에서 STIX 도입을 추진하고 있다.

III. 취약점 정보 모델링

STIX는 2.0 버전을 기준으로, Observed Data와 Indicator 등 총 12개의 주요 Object로 구성되어있다. 각각의 Object는 하단의 <표 1>에 기술된 것과 같다.

<표 1> STIX Domain Object (SDO)

SDO	Description
Attack Pattern	악의적 사용자가 대상의 훼손을 시도하는 방법을 설명하는 TTP의 한 형식
Campaign	특정 목표군을 대상으로 일정 기간 동안 발생하는 일련의의 악의적 활동 또는 공격을 설명하는 악의적 동작의 그룹
Course of Action	공격을 예방하거나 진행 중인 공격에 대응하기 위해 실행한 작업
Identity	개인, 조직 또는 그룹의 부류 표현
Indicator	수상한 또는 악의적 사이버 활동을 검색하는 데 사용할 수 있는 패턴
Intrusion Set	단일 조직이 지휘한다고 여겨지는 공통 속성을 가진 악의적 동작과 리소스의 그룹화된 집합체
Malware	악성 코드 및 악성 소프트웨어라고도 알려진 TTP(전술, 기법 및 프로시저, Tactics, Techniques, and Procedures)의 일종
Observed Data	시스템과 네트워크에 대해 관측한 정보
Report	위협 행위자, 맬웨어 또는 컨텍스트와 관련 세부정보를 포함한 공격 기법에 대한 설명과 같이 하나 이상의 주제에 초점을 맞춘 위협 인텔리전스의 모음
Threat Actor	악의적인 의도를 가지고 운영된다고 여겨지는 실제 개인, 그룹 또는 조직
Tool	공격을 수행하기 위해 사용할 수 있는 합법적인 소프트웨어
Vulnerability	해커가 시스템 또는 네트워크에 대한 액세스 권한을 얻기 위해 직접 사용할 수 있는 소프트웨어의 실수

본 논문에서는 취약점 정보를 표현하기 위한 “Vulnerability”에 대하여 다루겠다.

3.1 STIX 제공 기본 Object

“Vulnerability” Object는 소프트웨어, 시스템, 네트워크에서의 취약점 정보를 표현하기 위한 항목이며, STIX에서 정의되어있는 형태는 아래 <표 2>와 같다.

<표 2> Vulnerability Object

Property		Description
type	required	STIX Domain Object type ("vulnerability")
id	required	Object의 식별자
created_by_ref	optional	이 개체를 만든 엔터티
created	required	개체의 첫 번째 버전이 만들어진 시간
modified	required	개체의 이 특정 버전이 만들어진 시간
revoked	optional	개체가 철회되었는지 여부
labels	optional	분류의 집합
external_references	optional	STIX 이외의 정보를 참조하는 외부 참조의 목록
name	required	취약점을 식별하는 데 사용되는 이름
description	optional	취약점에 관한 자세한 사항과 컨텍스트를 제공하는 설명

“Vulnerability” Object를 활용하여 CVE와 같은 외부에서 정의된 취약점 정보를 표현할 수도 있고, 아직 정의되지 않은 취약점에 대한 표현도 가능하다. 본 논문에서는 일반적으로 많이 활용되는 CVE 취약점 정보를 기준으로 변환

방법을 제안한다.

CVE 취약점 정보를 “Vulnerability” object의 필수 속성들과 외부 참조 정보를 활용하여 표현하면 아래 <그림 2>와 같이 표현이 가능하다.


```
{
  "type": "vulnerability",
  "id": "vulnerability--e588409c-cc48-11e4-8b9a-34363bd377fe",
  "created": "2017-03-16T00:00:00.000Z",
  "modified": "2017-07-11T00:00:00.000Z",
  "name": "cve-2017-0001",
  "external_references": [{
    "source_name": "cve",
    "url": "https://nvd.nist.gov/vuln/detail/CVE-2017-0001",
    "external_id": "cve-2017-0001"
  }],
}
```


<그림 2> STIX 변환 예(CVE-2017-0001)

3.2 사용자 정의 방법 제안

STIX에서 정의되어있는 “Vulnerability” Object를 이용하더라도, 취약점과 관련된 모든 정보를 표현할 수는 없다. STIX 에서는 이렇게 미리 정의된 Object를 통해 표현할 수 없는 정보는 사용자가 직접 정의하여 활용할 수 있도록 한다.

CVE 취약점은 기본적으로 CVE ID, affected software, problem type, description, CVSS, release date, last modified date 등의 정보를 제공한다. 하지만 STIX에서 정의되어있는 “Vulnerability” Object를 이용한다면, affected software와 problem type, CVSS등의 항목을 표현하기 어렵다. 위의 정보를 표현하고자 한다면 “Vulnerability” Object를 Custom해야 한다. Custom 과정 전에, 각각의 데이터의 특성을 살펴보고, 그에 맞게 Custom을 해야 한다. Affected software는 1개 이상의 데이터가 존재할 수 있으며, problem type은 Common Weakness Enumeration (CWE)의 일련번호를 통해서 표현한다. CVSS와 같은 경우에는 버전 2와 버전 3의 CVSS 점수와, 각각의 점수를 산출하기 위한 요소들도 함께 제공된다. 이런 데이터의 특성을 반영하여 Custom을 진행한 예는 아래 <그림 3>과 같다.


```
"extensions": {
  "x-example-com-affects": [{
    "affected_software": cpe_id
  }],
  "x-example-com-cvss-v2": {
    "vector_string": vector,
    "base_score": base_score,
    "severity": severity,
    "impact_score": impact_subscore,
    "exploitability_score": expoitability_subscore
  }
},
{
  "x-example-com-cvss-v3": {
    "vector_string": vector,
    "base_score": base_score,
    "severity": severity,
    "impact_score": impact_score,
    "exploitability_score": expoitability_score
  }
}
"x_example_cwe": cwe-id
```


<그림 3> STIX “Vulnerability” Custom 예

위의 사용자 정의를 적용하여 다시 CVE 취약점 정보를 표현하면 아래 <그림 4>와 같이 변환할 수 있다.

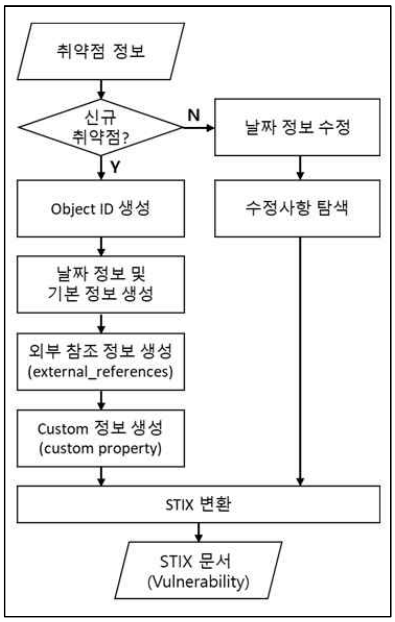

```
{
  "type": "vulnerability",
  "id": "vulnerability--e588409c-cc48-11e4-8b9a-34363bd377fe",
  "created": "2017-03-16T00:00:00.000Z",
  "modified": "2017-07-11T00:00:00.000Z",
  "name": "cve-2017-0001",
  "external_references": [{
    "source_name": "cve",
    "url": "https://nvd.nist.gov/vuln/detail/CVE-2017-0001",
    "external_id": "cve-2017-0001"
  }],
  "extensions": {
    "x-example-com-affects": [{
      "affected_software": "cpe:2.3:o:microsoft:windows_10:-:*:*:*:*:*:*:*"
    },
    {
      "affected_software": "cpe:2.3:o:microsoft:windows_vista:*:sp2:*:*:*:*:*:*"
    },
    {
    }],
    "x-example-com-cvss-v2": {
      "vector_string": "AV:L/AC:L/Au:N/C:C/I:C/A:C",
      "base_score": "7.2",
      "severity": "HIGH",
      "impact_score": "10.0",
      "exploitability_score": "3.9"
    }
  },
  {
    "x-example-com-cvss-v3": {
      "vector_string": "CVSS:3.0/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H",
      "base_score": "7.8",
      "severity": "High",
      "impact_score": "5.9",
      "exploitability_score": "1.8"
    }
  }
  "x_example_cwe": "cwe-264"
}
```


<그림 4> STIX(Custom) 변환 예(CVE-2017-0001)

3.3 취약점 정보의 STIX 변환 기술

CVE 취약점 정보는 1년에 10,000개 이상 발표되며, 지속적인 수정이 발생한다. 또한 CVE 이외의 취약점 DB도 존재하기 때문에, 사람이 하나하나 STIX 형태의 문서로 작성할 수 없다. 따라서 본 장에서는 알려진 취약점을 관리하기 위해 STIX로 변환하는 기술을 제안하고자 한다.

먼저 취약점 정보를 STIX 형태로 변환하기 위해서는 해당 취약점이 이미 변환된 적이 있는지 확인하는 과정이 필요하다. 확인 후, 새로운 취약점 정보라면 Object를 새로 생성하고, 이미 변환한 취약점 정보라면 수정사항을 찾아 해당 부분을 변경해줘야 한다. 이를 간단한 순서도로 표현하면 아래 <그림 5>와 같다.



<그림 5> 취약점 정보 STIX 변환 기술

IV. 결론 및 향후 연구

최근 다양한 취약점이 증가하고, 이를 악용한 공격이 전 세계적으로 이루어지고 있는데, 이에 대응하기 위해서는 즉각적인 사이버 위협 정보의 공유가 필수적이다.

본 논문에서는 취약점 정보의 즉각적인 공유 및 대응을 위해 STIX 표준 규격을 활용한 모델링 및 사용자 정의 방안을 제안하였다. 현재 STIX는 2.0 버전이 발표되어있고, 2.1 버전의 작업이 진행중이다. 향후 2.1 버전에서의 수정사항에 대한 반영 및 국가 CERT 등의 다양한 플랫폼에서의 정보 공유를 위한 연구를 지속할 예정이다.

ACKNOWLEDGMENT

이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2016-0-00193, IoT 보안 취약점 검색·공유 및 시험 기술 개발)

[참고문헌]

[1] Browse Vulnerabilities By Date, CVE Details, Retrieved from http://www.cvedetails.com/browse-by-date.php.

[2] VulDB Archive, Vulnerability Database, Retrieved from https://vuldb.com/?archive.

[3] Y.Younan, 25 Years of Vulnerabilities: 1988 - 2012. Sourcefire Vulnerability Research Team, 2013.

[4] K.Zunnurhain, Vulnerabilities with Internet of Things, In Proceedings of the International Conference on Security and Management, pp. 83, January, 2016.

[5] Fortinet, Threat Landscape Report Q2 2017, August, 2017.

[6] E.Ko, T.Kim, and H.Kim, A Study on the Management Structure of Vulnerability Information in IoT Environment, In Advanced Multimedia and Ubiquitous Engineering, pp. 247-251, 2017.

[7] G.Farnham and K.Leune, Tools and standards for cyber threat intelligence projects, SANS Institute, 2013.

[8] C.Johnson, L.Badger and D.Waltermire, Guide to Cyber Threat Information Sharing, NIST Special Publication 800-150, October, 2014.

2017 한국정보보호학회 동계학술대회

포스터발표 트랙 2

온라인 마켓플레이스를 위한 이상거래탐지시스템(FDS) 분석 및 향후 전망

장창훈

고려대학교 컴퓨터정보통신대학원 소프트웨어공학과

Analysis and Future Prospects of Fault Detection System (FDS) for Online Marketplace

Chang-Hoon, Jang

Software Engineering, Korea University Graduate School of Computer & Information Technology

요 약

오픈마켓, 소셜커머스, 종합몰 및 특화된 중소형몰 등으로 형성된 온라인 마켓플레이스에서 발생될 수 있는 이상거래탐지 및 이상거래방지를 위하여 자동화된 Fault Detection System(이하, FDS)의 필요성이 갈수록 커져가고 있다.
이 논문에서는 현재 사용중인 FDS 의 현황 및 구조에 대하여 파악하고 향후 전망에 대해 예측한다

I. 서론

온라인 마켓플레이스는 현재 G마켓, 옥션, 11번가등의 오픈마켓과 쿠팡, 티몬, 위메프를 중심의 소셜커머스 그리고 롯데닷컴, GSShop, SSG.com, 현대몰등 대형 종합몰 및 전문화된 중소형몰 형태로 주로 구성이 되어 있다. 스마트폰을 중심으로 한 모바일 디바이스의 급성장으로 트래픽 데이터가 기하급수적으로 늘어나고 있으며 매출 성장율에 있어서도 오프라인 시장에 비해서 매년 지속적으로 성장을 하고 있다. 거래량이 증가하는 만큼 비정상적인 트래픽과 부정거래 행위, 금융 사기등도 동반 상승하고 있는 상황에서 다양한 거래 데이터에 대한 실시간 모니터링, 탐지 및 신속한 대처가 더욱 절실한 상황이다. 이에 본 논문에서는 온라인 마켓플레이스에서 구축에 있어서 필수 조건이 되고 있는 Fault Detection System 또는 Fraud Detection System(이하 FDS)인 이상(異常)거래탐지 시스템의 현황을 분석하고 향후 전망을 통해서 판매자와 구매자 그리고 온라인 마켓플레이스 업체 모두가 신뢰할 수 있는 온라인 마켓플레이스에서의 FDS 구축방안을 제시하고자 한다.

1.1 온라인 마켓플레이스의 FDS 현황

온라인 마켓플레이스에서의 FDS는 갈수록 지능화 되어 가는 카드 부당거래 탐지와 개인정보유출 방지 및 각종 피싱(Phishing), 파밍(Pharming)을 유도하는 악성코드등 사이트 전반에 대한 실시간성 모니터링을 통해서 자동으로 탐지 및 방제하기 위하여 대형 사이트 중심으로 도입이 되고 있으며, 또한 PC에서 모바일 디바이스로의 사용자 환경이 변경됨에 따라 그동안 PC에 기본적으로 설치 하였던 Active-X를 통한 공인인증서 보안의 복잡한 방식에서 벗어나 사전 인증등을 통해 절차를 간소화한 간편결제 방식이 호응을 얻고 있다. 이러한 단순화된 과정에서 발생할 수 있는 비정상 거래 데이터, 금융사기 방지를 위한 결제시스템이 필요성은 더욱 강조되고 있으며, 대형 사이트를 위주로 FDS 운영 중이거나 구축 중이다.

[국내 모바일 간편결제서비스 유형 및 종류(2016.6월 기준)]



* 이번 조사에서 '모바일 간편결제서비스'는 '모바일 기기'를 통해 '결제'가 가능한 서비스를 의미함.

발췌) 한국소비자원, 2016

주요 국내 간편결제 서비스

업종	기업명	서비스명	특징
이동통신업	LG유플러스	페이나우	온·오프라인 결용, 휴대전화 번호 입력 등 비밀번호 외에 다양한 인증 기능 보유
	KT	클립	오프라인 전용, BC카드와 제휴 맞춤형 통합 혜택 서비스 제공, 10월 중 NFC 전용 리더기 결제 가능
	SK플래닛 (SKT자회사, 11번가)	시럽페이	온라인 전용, 국내 모든 신용카드와 사용 제휴
유통	신세계그룹	SSG페이	온·오프라인 결용, 바코드 스캔 가능
	롯데그룹	엘페이(가칭)	1만1000개 이상의 가맹점을 가진 인프라 활용 예정
	현대백화점	미정	미정
온라인 쇼핑몰	이베이코리아 (G마켓, 옥션)	스마일페이	온라인 전용, 본인 카드 정보 입력 후 부여받는 1회성 비밀번호 사용. 정보 제화 이체 서비스 도입
	인터파크	엘로페이	온라인 전용, 하루 30만 원 이내 결제 금액에 한해 직불 결제 가능
인터넷	NHN엔터테인먼트	페이코	온·오프라인 결용, NFC 전용 리더기
결제 대행(PG)	KG이니시스	K페이	온·오프라인 결용, 총 20여만 개 가맹점에서 사용 가능

발췌) 한경비즈니스, 2015/09

한국소비자원과 한경비즈니스의 자료에 따르면 금융권, 통신사 및 온라인 마켓플레이스 중심으로 다양한 방식으로 운영중인 간편결제서비스는 기존의 공인인증서에 비해서 훨씬 쉽고 간편한 반면, 사이트마다 약간씩 차이가 있는 만큼 각각의 서비스에서 발생할 수 있는 부정거래 방지를 위하여 사이버 보안의 강화가 더욱 절실하고 이를 연계한 FDS를 구축 하여 운영 중이거나 현재도 새로 구축 중이다.

부정사용 유형별Trends - Global

• CNP Fraud 는 부정사용 중 가장 큰 비중을 차지하며 비중은 `08년 40% 에서 `13년 57% 로 증가, 규모는 767million$ (7천억원) 에 달함



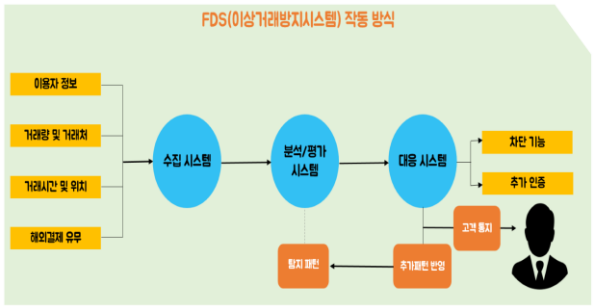
CNP: Card Not Present(비대면 거래)
출처: VISA Korea (TC40 client fraud reporting and operating certificates submission)

발췌) 금융보안연구원 FDS 세미나 자료

이와같이 TC40 Client Fraud Reporting And Operating Certificates Submission(Visa Korea) 에서 발표된 자료를 보면 부정사용 유형별 Trends의 Global 사례를 보면 무카드거래(Card Not Present, CNP) 즉, 가맹점과 결제자와 얼굴을 마주하지 않는 인터넷 거래에서 부정거래가 급격히 증가하고 있음을 볼수 있으며 그 거래현황은 매년 증가하고 있다.

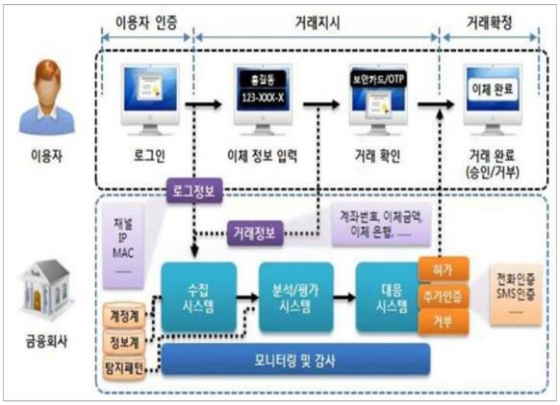
간편결제 서비스가 증가하고 이러한 비대면 결제 방식이 증가하는 인터넷 환경에서 이루어지는 거래 패턴은 기존의 오프라인 방식과는 다르고, 과거데이터 만의 단순 분석으로는 대처가 어려운 상황으로 FDS 구축은 더욱 중요하다고 볼수 있다.

1.2 온라인 마켓플레이스의 FDS 구조



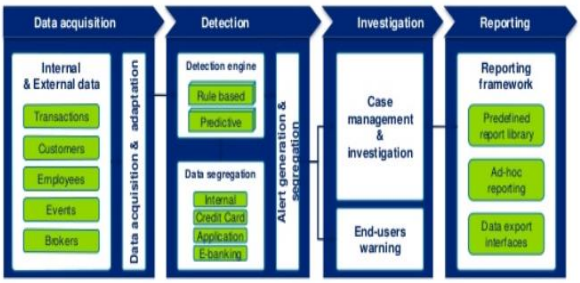
발췌) 미디어자몽, 2016/02

온라인 마켓플레이스의 결제서비스에서 적용중인 FDS는 은행, 카드사등의 기반 시스템과 연계를 통해서 거래데이터를 분석하여 이상거래를 찾아내는 시스템이다. 이러한 비정상적인 데이터의 범주에는 동일 시간대 거래 데이터의 위치(IP등)가 판이하게 다르다던지, 급격히 변화된 거래 시간대등 비정상적인 데이터 패턴을 분석하여 예방을 위하여 주로 적용하고 있다.



발췌) 이상금융거래 탐지시스템 기술가이드

이러한 FDS는 주로 평소 사용자의 구매패턴을 저장하고 분석하는데, 이를 통해 수집, 분석, 평가, 모니터링 및 감사등의 주된 기능을 수행한다. 이를 위하여 IP, MAC Address 사용자 환경에 대한 제반 정보수집과 로그인시 ID시 회원정보와 구매시 결제정보등을 분석하여 이상거래유무를 판단하게 되며, 이를 분석하기 위한 방대한 양의 빅데이터 저장과 분석력이 중요하다.



발췌) FDS 모형 개발 프로세스

그러므로, FDS를 구축하기 위한 기본적인 프로세스는 다음과 같은 영역으로 구분 할 수 있다.

(1) 정보 수집 단계
- 웹로그, 인증정보, 거래정보, 위변조
- 배송지, 거래시간 및 거래이력정보

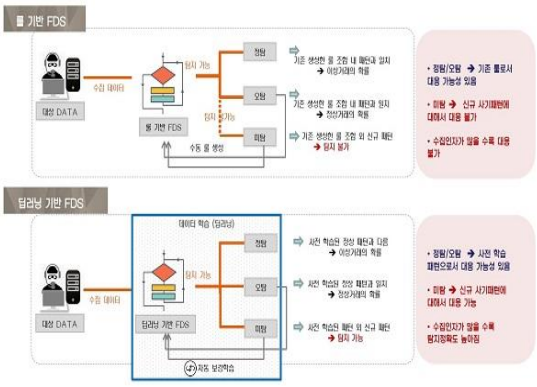
(2) 분석 및 탐지 단계
- 실시간 수집, 패턴매칭, 암호화
- RuleSet매칭, 이벤트 연관 분석
- 기타 시나리오 분석

(3) 모니터링/감사
- 실시간 모니터링 및 통계분석
- 감사 및 권한 관리

(4) 대응 단계
- 정상 거래, 추가 인증

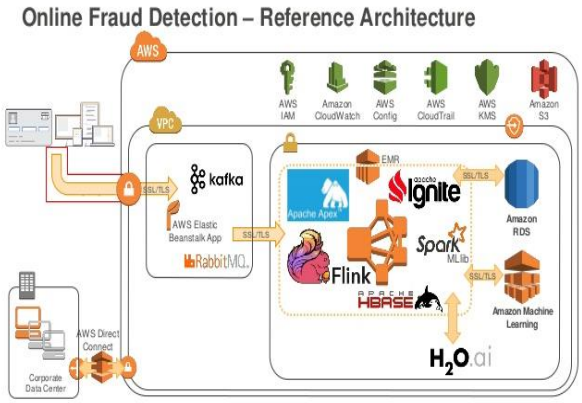
(5) 프로 파일링
- 거래 형태, 이용 습관

이상탐지를 정확히 하기 한 핵심 요소기술은 사이트 에 유입되는 트래픽과 사용자의 행동데이터에 대한 실시간 수집과 이렇게 축적된 빅데이터성의 방대한 데이터에 대하여 실시간에 준하는 분석(프로파일링)을 통해서 사용자의 거래 형태및 이용 습관을 탐지하여 이상 징후에 대한 정확한 예측으로 사전 방지하는 것이다



발췌) 데이터넷, 2016/06

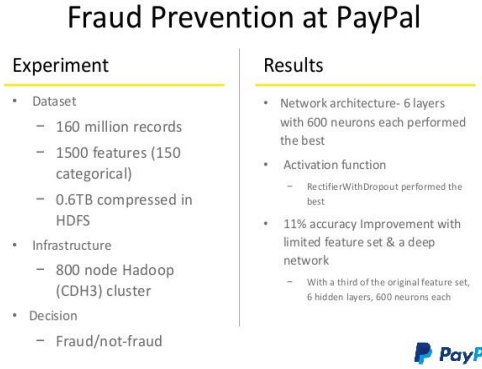
그중에서도 FDS에서의 핵심 기술은 머신러닝, 딥러닝 알고리즘을 적용한 인공지능 기법이 각광을 받고 있는데, 기존에 구축된 RuleSet을 기반으로 온라인마켓플레이스의 데이터 패턴을 학습하고 비정상적인 패턴을 찾아내어 사고를 미연에 방지고하자 함이다. 반면에 이러판 패턴 조차도 방대한 경우에 대비하여 사전에 생성하지 않고, 자동으로 생성 할 수 있는 딥러닝을 통한 기법을 적용하고 있다.



발췌) 아마존 AWS에 적용된 FDS

일례로 아마존의 Cloud System인 AWS에서 사용중인 FDS에서는 룰기반과 딥러닝기반의 다양한 기법을 통해서 온라인에서 발생하는 수많은 데이터에 대한 이상거래를 탐지하고 있다.

1.3 온라인 마켓플레이스의 FDS 전망



발췌) 페이팔의 FDS

머니투데이방송의 TechM 제35호 기사에 따르면 페이팔의 경우 비정상 거래 데이터를 추적하고 이를 구조화 패턴을 적용하고 있으며 2013년 처음 도입시 백여개에 불과한 패턴이 2016년에는 수천개에 이를 정도로 축적이 되고 있으며 이를 위하여 룰베이스드 기반의 머신러닝과 딥러닝등을 이용한 신경망 기법을 적용하고 있다.

금융권의 이상금융거래 탐지시스템(FDS) 고도화 로드맵

단계	1단계 (2014년): FDS 도입단계	2단계 (2015년): FDS 확대단계	3단계 (2016년): 금융권 공동대응단계
수집 시스템	단말기 접속정보 수집	단말기 접속정보 및 금융거래정보 수집, 과거패턴정보 구축	단말기 접속정보 및 금융거래정보 수집, 과거패턴정보 구축
분석/평가 시스템	오용 탐지(black list)	오용거래 및 이상거래 탐지	오용거래 및 이상거래 탐지(고도화), 금융회사간 정보 공유
대응 시스템	차단조치, 추가 인증수단 마련	차단/지연 조치, 추가 인증 조치, 실시간 처리	차단/지연 조치, 추가 인증 조치, 실시간 처리
모니터링 및 감사		모니터링 및 상담전문인력 확보	모니터링 및 상담전문인력 확대, 탐사 결과 감사기능 구현, 관련법규 개정

발췌) 금융감독원, IT조선-2015/03

금융감독원에서 발표한 자료를 기반으로 금융권의 FDS 고도화 로드맵에 따르면 FDS 도입 후 확대 단계를 거쳐서 어느 특정 사이트에 국한된 부분이 아닌 관련 사이트 공동의 대응 단계 구축을 목표로 하고 있는데, 이는 온라인 마켓플레이스에서도 동일하다 할 수 있다.

이에 따라 수많은 사이트에서 Buyer와 Seller의 신뢰성 있는 거래를 위하여 결제시스템 위주로 구축하는 FDS는 그 외에도 개인정보보호와 스팸, 피싱등 방지를 위한 영역에도 확장적용 중이다.

II. 결론

온라인 마켓 플레이스의 디바이스 환경이 PC에서 모바일로 변경이 되고 있는 만큼, 이를 이용한 트래픽 데이터, 금융거래 데이터 등에도 많은 변화가 있고, 또 급격하게 증가하고 있는 실시간성 데이터를 추적, 분석 및 대처하기 위해서는 단순한 패턴 매칭이나 블랙리스트/화이트리스트를 통한 방지등으로는 빅데이터 처리는 불가능한 상황이다. 이에 따라 머신러닝, 신경망등을 이용한 알고리즘을 통한 FDS 시스템 적용으로 부정거래 데이터 차단뿐아니라 예측을 통한 사전방지도 반드시 적용이 되어야 한다.

또한 FDS의 제반 탐지패턴을 우회할 수 있는 각종 해킹 기법들도 계속 발전이 될 것으로 예상되는 관계로 FDS 구축으로만 완료되는 부분이 아니라 온라인 마켓플레이스의 모든 행동 데이터(User Behavior Data)에 대한 피드백 데이터에 대한 적용과 이에 따른 머신러닝, 딥러닝등의 알고리즘 보강이 지속적이고 반복적으로 이루어져야 한다.

[참고문헌]

[1] 한경비즈니스 기사, 2015/09
[2] 금융보안연구원 FDS 자료, 2014
[3] 미디어자몽, 2016/02
[4] 이상금융거래 탐지시스템 가이드, 2016
[5] ACFE Eurpoean FDS, 2014/32
[6] 데이터넷, 2016/06
[7] 아마존 FDS, 2016
[8] TechM 제35호 기사, 2016/03
[9] 페이팔의 FDS
[10] IT조선 금융감독원 자료, 2015/03

Usable Security관점에서 본 포털사이트 로그인API의 취약점

문성훈*, 이예진*, 윤진희*, 이재원*, 김호엽**

*대구대학교 정보보호 영재교육원, **대구대학교 지도교수

Vulnerability in Portal site Login API from Perspective of Usable Security

Seong-Hun Mun*, Ye-jin Lee*, Jin-hui Yoon*, Jae-won Lee*, Ho-Youp Kim**

*Information security institute for the gifted, Daegu University, **Daegu University

요 약

최근 사용자의 편의성을 높이기 위한 방법의 일환으로, 특정 사이트의 로그인을 통해 여러 다른 서비스에 로그인을 할 수 있는 기능인 로그인API를 많은 사이트에서 지원하고 있다. 그러나 API를 제공하는 원 사이트 이용 중 패킷이 탈취당한다면, 패킷의 내용 중 로그인 세션유지를 위한 쿠키 값을 악용하여 별도의 로그인 과정 없이 API를 제공하는 원 사이트인 포털 사이트 로그인은 물론이며 이 API를 사용하는 가맹 사이트까지 로그인이 가능해지므로 2차 피해가 발생할 수 있다. 본 논문에서는 위에서 언급한 취약점으로 인해 발생할 수 있는 피해상황을 가정하고, 이를 방어할 수 있는 개선 방안을 제시하는 것을 목적으로 한다.

Ⅰ.서론

Usable Security는 서비스의 보안이 강화됨에 따라 사용자의 편의성이 감소하게 되는 것을 보완하기 위해 등장한 새로운 개발방식을 의미한다. 간편 로그인, 간편 결제 서비스가 그 대표적인 예라고 할 수 있다.

인터넷의 발달에 따라 수많은 서비스가 생겨나면서, 사용자들은 사이트별로 불편한 회원가입을 해야 했고 이로 인해 개인적으로는 아이디와 패스워드 관리에 어려움을 겪고 있다.

해킹으로 인해 웹 서비스 이용자의 개인정보 유출사고가 빈번하게 발생하면서, 통합 로그인 서비스는 핵심 보안 기술로 자리 잡게 되었다.

별도의 회원가입 없이, 유명 포털사이트의 아이디를 통해 로그인하는 것은 사용자 편의성과 사이트 보안이라는 숙제를 해결한 것으로 인식되어 웹 서비스 제공자들 사이에서 빠르게 확산되고 있다. 고객식별과 마케팅을 위한 최소한의 정보만 공유하기 때문에 개인정보의 과다 보유 부담을 덜 수 있다는 장점도 여기에 한 몫 더하고 있다.

사용자 입장에서도 편리성과 유명 포털사이트의 보안기술을 그대로 이용할 수 있다는 심리적 안정감으로 인해 로그인API를 통한 간편 로그인을 선호하지 않을 이유가 없는 것이다.

그러나 포털사이트를 사용하는 동안 서버와 PC 혹은 서버와 스마트폰 간에 주고받는 패킷이 탈취되어 해당 포털사이트의 로그인API를 사용하는 서비스 사이트 전체가 노출될 수 있

는 취약성이 존재한다면 말이 달라진다.

본 연구를 진행하면서 실험해본 결과 해당 포털사이트는 로그인 유지를 위해 쿠키를 사용하고 있으며 동일 쿠키를 이용하여 다른 컴퓨터에서 로그인이 가능하다는 점이 발견되었다. 쿠키로 로그인 후에는 정상 로그인과 같은 권한으로 포털사이트의 다양한 메뉴를 이용할 수 있음은 물론 로그인API 기능을 제공하는 가맹 사이트로도 접속하여 전 기능을 사용할 수 있는 문제점이 존재하였다.

최근 IP보안 기능이 추가되면서 C클래스 IP 영역을 벗어난 원 거리에서 동시 접속할 경우 포털메일 사용 등의 권한이 제한되지만 이 경우에도 포털의 카페 이용과 로그인API로 가맹점들의 접속이 가능하다.

더욱이 ARP Spoofing공격과 같은 중간자 공격 기법 등으로 쿠키가 탈취된 경우, 동일 IP영역 내이기 때문에 권한 제한까지 발생하지 않게 된다. ARP Spoofing으로 인한 피해 사례로, 2016년 특정 호스팅사가 ARP Spoofing 공격을 당해 호스팅 서비스를 받는 다수의 웹 사이트에서 파밍용 악성코드가 유포되었으며, 많은 사람들이 이용하는 웹 하드와 광고배너 사이트에서도 파밍용 악성코드가 뿌려지는 피해가 있었다[1].

이에 본 연구에서는 로그인API 사용 시 쿠키로 로그인이 가능함으로 인해 발생할 수 있는 문제점을 실험을 통해 증명하고, 로그인API의 문제점을 보완할 수 있는 방안을 제시하는 것을 목표로 한다.

Ⅱ. 연구지식

2.1 로그인API

로그인API는 점유율과 신뢰성이 높은 SNS, 전자상거래, 포털사이트 등의 아이디를 이용하여 타 사이트의 서비스를 이용할 수 있게 만든 서비스이다. 이미 가입되어 있는 대형 사이트의 아이디를 이용하여 사이트마다 별도의 아이디와 패스워드를 만들 필요가 없으며, 공신력 있는 사이트에서 제공하는 서비스이기에 보안의 신뢰성 또한 높다고 인식되고 있다.

2.2 ARP Spoofing

ARP Spoofing은 근거리 통신망(LAN)내에서 주소결정 프로토콜(ARP)의 허점을 이용하여 자신의 MAC주소를 다른 사용자의 MAC주소인 것처럼 속여 정보를 가로채는 일종의 중간자 공격이다[2].

ARP 프로토콜은 논리적인 IP주소(3계층)를 물리적인 MAC주소(2계층)로 바꾸어주는 주소 해석 프로토콜로, 공격자는 sniffing 대상자의 MAC주소로 위장하여 패킷을 가로채 훔쳐보거나 패킷을 변조 한 후 전송한다.

본 논문에서는 ARP Spoofing을 상황을 가정하여 패킷에 존재하는 로그인 쿠키를 가로채고, 쿠키에 존재하는 대상자의 정보로 로그인API를 지원하는 사이트들에 로그인하는 실험을 통해 문제점을 지적하고자 한다.

Ⅲ. 서비스 취약점

3.1 로그인API 구현비교

본 논문에서는 로그인API의 취약점을 분석하기 위하여 대표적인 2종류의 로그인API 구현방식을 찾아 비교하였다.

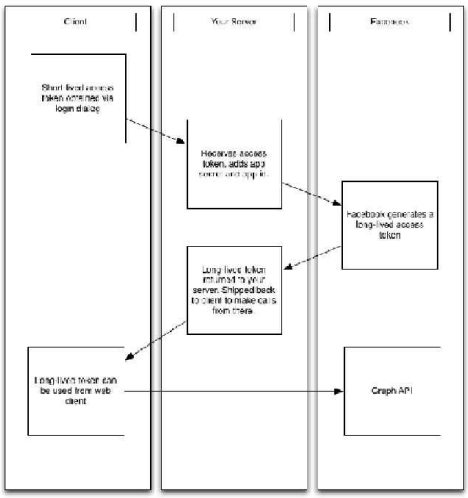


그림 1. SNS 로그인API 구현

먼저, SNS를 통한 로그인API의 경우, 롱텀 토큰이 아니면 로그인 시 사용된 토큰이 오래 유지되지 않도록 구현하였다. 또한 요청 액션마다 토큰을 분리하여 발급하고, 로그인 시도마다 암호화된 토큰값을 변경, 확인하도록 구현되었다.



그림 2. N포털 사이트 로그인API 구현

N포털의 경우, 하나의 access-token으로 복수 사이트에 접속가능하게 하여 사용자 편의성을 극대화 하였다.

탈취한 패킷의 쿠키를 이용하여 다른 PC에서 로그인이 가능하며, 피해 PC에서 로그아웃을 하더라도 포털의 일부 기능과 로그인API로 접속한 사이트들의 세션은 계속 유지된다. 이와 관련한 취약점을 테스트하기 위해 실습을 진행하였다.

3.2 실습

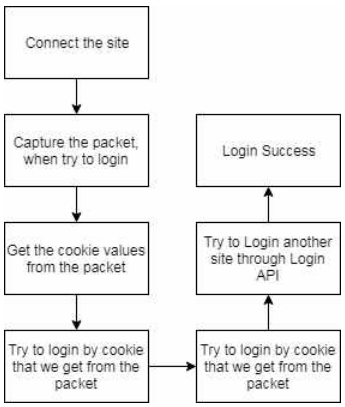


그림 3. 실습 순서도

순서	내용
1	PC1에서 N포털에 로그인 한다.
2	WireShark라는 패킷 캡처 프로그램을 이용하여 패킷을 캡처한다.
3	WireShark를 이용하여 캡처된 패킷에서 로그인 쿠키 값을 찾는다.
4	획득한 쿠키를 사용하여 PC2에서 N포털 메인페이지 등에 로그인을 시도한다. --> 성공 확인
5	획득한 쿠키를 사용하여 PC2에서 N포털 로그인API를 사용하는 가맹점인 M외식업체에 로그인을 시도한다. --> 성공 확인
6	PC1에서 N포털 로그아웃을 한다.
7	그럼에도 불구하고 PC2에서는 N포털의 카페 등을 로그인상태로 이용가능하다는 것을 확인한다.
8	PC2에서 접속한 N포털 로그인API 가맹점인 M외식업체는 PC1의 로그아웃에도 불구하고 로그인 세션이 끊어지지 않고 계속 유지되는 것을 확인한다.
9	[순서2]에서 캡쳐해 두었던 패킷으로 새로운 PC3에서 로그인API 가맹점인 M외식업체에 로그인을 시도한다. --> 성공 확인

표 1. 실습 순서

요약하자면, 테스트를 진행한 N포털에서는 로그인 승인을 위해 access-token을 발행하고 쿠키에 그 정보를 포함시킨다. 간단한 쿠키 인젝션을 통해 쿠키를 변조하지 않고 N포털 사이트에 로그인할 수 있었으며, 이는 N포털의 로그인API를 사용하는 곳까지 세션이 서로 공유되어 회원이 가입하고 있는 전체 사이트에 접속 가능하므로 보안 조치가 필요하다.

본 연구에서 제시하는 보안 취약성은 다음과 같다.

1. N포털 사용 중 탈취된 쿠키를 악용하여 다른 컴퓨터에서 N포털에 로그인이 가능하다.

2. 원 사용자가 N포털에서 로그아웃을 하더라도 N포털의 로그인API로 접속한 사이트의 세션은 그대로 유지되어 피해를 초래할 수 있다.

3. 원 사용자가 로그아웃을 하지 않고 브라우저를 종료 후 다른 곳에서 로그인 하더라도 탈취한 기존의 access-token은 유효하여 탈취한 쿠키를 이용하여 N포털의 전체 기능 사용은물론 로그인API를 사용하는 가맹점까지 사용 가능하다.

4. 로그아웃을 하지 않고 브라우저를 종료

했던 세션의 access-token은 유령 상태로 계속 사용할 수 있다는 점에서 심각한 보안취약점으로 작용한다.

5. 로그아웃 처리된 access-token으로도 N포털의 내부 cafe사이트를 사용할 수 있는 등의 권한이 남아 있어 개인 사생활이 무방비로 노출되어 있다.

6. 로그아웃 처리된 access-token으로도 N포털의 로그인API를 사용하는 사이트에 접속하여 기능을 사용할 수 있다.

Ⅳ. 결론

개방형 또는 공공 네트워크 사용 중 특정 사이트에 로그인 할 경우 패킷이 남에게 보여질 수 있는 취약점이 존재하는 것은 주지의 사실이다. 이 취약점을 악용하는 ARP Spoofing과 같은 중간자 공격에 누구든 노출될 수 있다.

본 연구에서는 ARP Spoofing을 통해 N포털 이용자의 access-token 정보가 포함된 쿠키를 탈취하여 N포털의 로그인API를 지원하는 가맹점을 악의적으로 이용할 수 있다는 것을 실험하였다.

위에서 언급한 취약점을 악용하여 N포털의 로그인API를 사용하는 가맹점을 대상으로 무차별적 로그인 테스트를 실시할 경우 사이트별 가입여부와 사용 내역이 노출되어 개인의 프라이버시가 심각하게 훼손될 수 있다.

이 경우 타인의 사이트를 주인처럼 사용할 수 있다는 우려와 함께 사용자의 정보 유출이 가능하다는 점에서 편의성을 위한 기능이 오히려 독이 될 수 있다는 점을 여실히 보여준다.

Usable Security 관점에서 보안만을 강조할 수 없는 이유는 스마트폰과 노트북의 이동 중 사용과 로그인API를 통한 여러 가맹점 동시 이용에 대한 편의제공도 고려해야하기 때문이다. 스마트폰의 Wi-Fi기능을 이용하여 접속한 경우 사용자의 이동경로에 따라 IP가 지속적으로 변경되어 IP인증을 강화할 경우 로그인 유지에 어려움이 있는 한계가 존재하며 로그인API를 사용하는 여러 가맹점마다 별도의 access-token을 발행할 경우 매 사이트마다 로그인을 다시 해야 하는 번거로움이 발생한다.

본 연구는 로그인API 제공자의 보안 취약점으로 인해 로그인API를 사용하고 있는 전체 서비스가 취약해 질 수 있다는 점을 지적하고 다음과 같은 개선 방안과 의견을 제시한다.

1. 먼저 쿠키만을 사용하여 로그인이 되지 않도록 개선해야 한다.

2. 발급한 access-token은 로그아웃 혹은 세션 타임아웃 되지 않은 유효한 토큰인지 확인해야 한다.

3. 동일한 access-token이 이종 브라우저 혹은 버전이 다른 동일 브라우저에서 복수로 사용될 경우 탈취된 토큰인지 원 사용자의 토큰인지 구별할 수 있게 해야 한다(로그인했던 IP와 현재 IP의 강화된 비교, 로그인API를 사용하는 가맹점별 차별화된 access-token의 발급 및 검증은 사용자의 로그인 횟수를 크게 늘려 불편을 초래하므로 개선방안에는 미포함).

4. 패킷 캡처 후 변조하는 것을 막기 위해 SSL을 적용해야 한다(SSL 적용을 통해 패킷을 암호화하더라도 쿠키는 SSL과 상관없이 탈취될 수 있고 탈취된 쿠키는 로그인 등에 악용될 수 있다는 점에서 SSL 적용은 최소한의 조치이며 충분조건은 아님).

[참고문헌]

[1] 김경애, 「호스팅사, ARP스푸핑 공격 당해 악성코드 유포 外」, 『보안뉴스』, 2016.04.06.

[2] KISA (2007), ARP Spoofing 공격 분석 및 대책

[그림 1] Facebook user authentication in Java web application

[그림 2] 네이버 개발자 센터

아파트(공동주택)에서의 개인정보보호 방안 연구

한충희*, 박순태**, 이상준***
*전남대학교 정보보안협동과정
**한국인터넷진흥원 정보보안팀
***전남대학교 경영학부

Study of Measure about the Privacy in Apartment

ChoongHee Han*, SoonTai Park**, Sang-Joon Lee***
*Division of Interdisciplinary Program of information security, Chonnam National University
**Korea Internet & Security AgencySchool of Business Administration
***Chonnam National University

요 약

주민번호, 고유식별정보, 바이오정보는 개인정보보호의 핵심 대상으로 개인정보보호법과 정부의 각종 고시, 기준 등을 통해 기본적으로 보호되어야 할 주요 대상으로 명시되고 있다. 정부부처나 지자체, 공공기관, 민간기업 등에서는 각종 교육, 평가, 점검 등을 통해 개인정보를 침해할 수 있는 여지를 상당부분 개선하고 있는 중이다. 그러나 아파트에서는 다양한 목적과 이유로 인해 너무 쉽게 무시되거나 침해되는 사례들이 많이 발생하고 있다. 본 논문에서는 아파트에서 이루어지는 개인정보 침해 위험 요인들을 살펴보고 아파트에 적용 가능한 개인정보 처리기준들과 벌칙들을 검토 한다. 이를 토대로 아파트에서의 개인정보보호 향상 방안을 검토하면서 그 대책으로 아파트 개인정보보호 자문센터 설립 등 주요 방법들을 제안하고자 한다.

I. 서론

2014년 1월 카드 3사 개인정보 유출사고 이후 개인정보보호법이 개정되어 원칙적으로 주민등록번호 처리가 금지되고, 주민등록번호 사고 시 5억원 이하 과징금 부과가 가능하게 되었다. 또한 개인정보보호법 위반 시 대표자 또는 책임자 징계 권고가 가능하게 되었다.

2014년 8월 강화된 개인정보보호법 시행 이후 정부부처, 공공기관, 일반 기업들에서는 많은 변화가 일어났다. 법 이전에는 꼭 필요하지 않아도 수집하던 주민등록번호를 수집 처리하지 않는 방식으로 변화된 것이다. 각각의 홈페이지 회원가입 시 불필요한 개인정보는 최대한 배제하고 수집하게 되었다. 신입직원 채용을 위한 입사지원서에도 주민등록번호는 물론 배제되었으며 최근에는 출신학교와 같은 개인정보들도 요구하지 않는 방식으로 개선되었다.

그러나 아파트에서의 개인정보보호 실태는 열악한 상태이다. 개인정보보호에 대한 의식부재와 관련 지식의 부족으로 적절한 개인정보의 수집과 관리가 이루어지지 않는 실정이다.

II. 관련 연구

위와 관련한 선행 연구를 살펴보면, 아파트에 대한 홈오토메이션 관련 기술 도입에 대한 논문은 많으나 아파트에서의 개인정보보호 향상 관련 연구는 미흡한 상태이다.

유사한 연구로는 2001년 '가정정보보호가 우선인 사이버아파트'를 통해 아파트 랜관리를 겸하는 웹마스터의 중요성을 주장하였다[1]. 2013년에 '공동주택 거주자의 홈오토메이션 이용에 관한 연구'를 통해 전자도어락, 비디오폰, CCTV의 도입 실태를 조사하였다. 이 논문에서는 설문조사를 통해 아파트 공간에 대한 CCTV의 도입 확대를

제안하였다[2]. 2017년 '생체인식 기술 기반 개인인증 수단에 따른 사용자 인식'연구는 개인인증에 따른 사용자 의도를 조사 분석하였다[3].

이에 본 논문에서는 혁신도시의 신규 입주 아파트에서 개인정보 침해 위험 요인들을 살펴본다. 또한 아파트 분야에 적용 가능한 개인정보 수집 및 이용 관련 처리기준과 벌칙들을 검토한다. 끝으로 아파트 개인정보보호 자문센터 육성 등 아파트 개인정보보호 강화방안을 제안한 후 결론과 향후 연구에 대해 기술하고자 한다.

III. 아파트 개인정보보호 강화방안

1. 아파트의 개인정보 주요 침해 위험 요인

1.1. 바이오정보 수집에 의한 개인정보 침해 위험

2010년 인구주택 총 조사에 의하면 국내 공동주택은 약 1042만호로 전체 주거유형 중 71%를 차지하며 공동주택 중 아파트 거주비율이 58.4%로 나타나고 있다[4]. 최근에는 건설 업체별로 아파트 입주민들의 마음을 사로잡기 위한 각양각색의 첨단 보안 서비스를 제공하고 있다[5]. 그러나 대체 가능한 방법이 충분히 있음에도 불구하고 지문, 홍채 등의 정보만을 이용해서 출입통제를 하겠다는 아파트들은 개인정보보호법 제16조(개인정보의 수집제한)을 위반할 소지가 있다.

1.2. CCTV에 의한 개인 정보 침해 위험

또한, 아파트 단지에 CCTV가 폭발적으로 설치되고 있다. 2008년 경찰청 경찰백서에 따르면 아파트 지하주차장 등에 약 26만 여개의 CCTV가 설치 운영되고 있다[6]. 또한 2014년의 통계청 자료를 살펴보면 전국적으로 약 800만개의 CCTV가 설치 운영되고 있다. 시설안전 등의 사유로 CCTV를 설치 운영하고자 할 경우에는 안내판 설치, 관리책임자 지정, 임의조작 금지, 녹음기능 사용금지 등의 엄격한 절차와 기준을 따라야 한다. 그러나 현실은 이와 같은 절차와 기준은 간과된 채 진행되고 있다.

1.3. 홈페이지 가입에 의한 개인정보 침해 위험

아파트 입주민 홈페이지를 개설하고 입주민에 대해 의무적으로 가입하도록 하고 있다. 입주민들의 회원가입 과정에서 무분별한 개인정보 수집이 이루어질 가능성이 높아 개인정보보호법 제15조(개인정보의 수집 이용)와 22조(동의를 받는 방법)을 위반할 가능성이 크다. 또한 개인정보가 사적으로 이용될 위험도 증가한다.

2. 아파트 개인정보처리 기준 및 주요 벌칙

2.1. 처리기준 위반 시 벌칙 사항

개인정보보호법은 개인정보를 수집 처리하는 기준을 명확하게 제시한다. 민감정보, 고유식별정보, 주민등록번호, 영상정보에 대한 개인정보보호법의 처리기준과 벌칙 및 과태료 부과기준은 다음 표와 같다.

〈표 1〉 주요 개인정보 처리기준 위반 및 벌칙 사항

구분	처리기준	벌칙사항
민감정보 고유식별정보	법령 또는 별도의 동의	5년/5000만원 이하 징역/벌금
주민등록번호	법령 또는 급박한 상황시	5년/5000만원 이하 징역/벌금
영상정보기기	법령 또는 범죄예방/수사 시설안전/화재예방 교통단속/분석	3000만원 이하 과태료

2.2. 안전성 확보조치의무와 주요 벌칙 사항

개인정보보호법은 개인정보처리자에게 정보주체의 개인정보가 분실, 도난, 유출, 변조 또는 훼손되지 않도록 관리적, 물리적, 기술적 안전성 확보 조치를 요구하고 있다. 관리적 조치사항은 개인정보보호책임자의 지정, 개인정보취급자 교육 등에 대한 내부관리계획을 수립하는 것이다.

물리적 조치사항은 관리용 단말기에 대한 안전조치를 통해 임의 조작을 방지하고, 전산실, 자료보관실 등 별도의 물리적 보관 장소를 두고 출입통제 절차를 수립 운영해야 한다는 것이다. 또한, 기술적 조치사항은 안전한 비밀번호 작성규칙, 일정횟수 이상 잘못 입력한 경우 또는 일정시간 이상 경과된 경우 접근 제한 조치 등이 있다. 위에서 살펴본 물리적, 관리적, 기술적 조치사항들은 다음 표와 같다[7].

〈표 2〉 개인정보처리 관련 안전조치 기준

구분	주요 내용
관리적 조치	내부관리계획의 수립 및 운영 - 개인정보보호책임자 지정 등
물리적 조치	관리용 단말기 안전조치 물리적 보관장소 출입 통제조치

	보조저장매체의 반출입 통제조치 재해재난 대비 안전조치
기술적 조치	업무 담당자 접근권한 관리 안전한 비밀번호 작성규칙 적용 일정 횟수/시간 경과 후 접근제한 접속 IP 제한, 분석 및 대응 안전한 인증 및 접속 수단 제공 개인정보 및 비밀번호 암호화 접속 기록의 보관 악성 프로그램 방지 등

안전성 확보조치 의무 관련 주요 벌칙들을 표로 정리하면 다음과 같다.

〈표 3〉 안전성 확보조치 관련 주요 벌칙

구분	주요 내용
3000만원 이하 과태료	암호화 조치를 취하지 않은 자 안전성 확보조치 미이행한 자 영상정보처리기 설치기준 위반자
1000만원 이하 과태료	미파기 개인정보 별도 보존 위반 영상정보처리기 안내판 미설치 개인정보처리방침 미수립 개인정보관리책임자 미지정
2년이하의 징역 또는 2000만원 이하의 벌금	안전성 확보조치를 하지 않아 개인정보를 분실, 도난, 유출, 위조, 변조, 훼손 당한자

3. 아파트 개인정보보호 강화 관련 방법들

3.1. 아파트 개인정보보호 적용의 제약사항

정부부처와 공공기관, 민간기업 들은 법 개정 이전보다 상당한 수준 개인정보보호가 강화되었다. 그러나 아파트에서는 행정력 부족, 정보주체의 권리의식 부족 등 다양한 이유로 법 적용이 미흡한 실정이다.

3.2. 아파트 개인정보보호 자율규제 촉진 지원

개인정보처리는 법적인 강제만으로는 개인정보보호법의 목적을 달성하기 어렵다. 우리 일상에서 이루어지는 개인정보에 대한 보호를 위해서는 개인정보처리자와 정보주체들의 자발적인 노력이 반드시 필요하고 정부에서는 이러한 노력을 지속하고 촉진할 수 있도록 지원하여야 한다.

3.3. 아파트 개인정보보호 교육, 홍보 강화

아파트 개인정보보호 인식개선과 함께 개인정보보호법 자체에 대한 교육 및 홍보활동이 이루어지도록 해야 한다. 특히 입주자대표회의의 각 위원들에게 의무적으로 개인정보보호법과 관련된 적절한 교육과정을 이수하도록 해야 한다. 아파트 입주자대표회의라는 개인정보처리자가 스스로 개인정보보호에 대한 필요성을 인식하도록 하여 아파트 입주민들의 소중한 개인정보가 과도하게 수집되거나 남용되는 사례가 발생되지 않도록 하는데 매우 중요하다.

3.4. 아파트 개인정보보호 인증마크 시행

개인정보인증마크는 개인정보를 적법하고 안전하게 처리하는 개인정보처리자에게 부여되는 안전마크의 일종이다. 인증마크를 획득한 경우 일정 수준 이상의 안전성과 신뢰성을 확보했다고 볼 수 있다. 개인정보보호협회에서 시행중인 개인정보보호우수사이트마크(ePrivacy마크)와 인터넷 사이트안전마크(i-Safe마크)를 적극 활용할 필요가 있다[8]. 아파트 입주자대표회의가 운영하는 개인정보처리시스템들에 대해 인증마크를 받을 수 있도록 유도해야 한다.

3.5. 아파트 관리규약 준칙 개선

아파트 입주민에게 절대적으로 영향을 주는 아파트 관리규약의 출발점은 지자체별로 정하여 배포하는 '공동주택 관리규약 준칙'이다. 현재 지자체별로 샘플 양식으로 배포하고 아파트 단지별로 가감해서 제정 및 운영하는 방식을 통일된 방식으로 개선해야 한다.

최근에 신규 입주한 아파트 관리규약을 검토한 결과 입주자대표회의에 과도한 집행권한을 부여하고 있는 것으로 판단된다. 또한 개인정보보호 조항들은 형식적으로 삽입되어 있는 것으로 판단된다.

이러한 문제들을 개선하기 위해서는 먼저 관련 부처인 국토부가 일괄적으로 작성 배부하는 방식으로 개선해야 한다. 국토부가 작성하는 '공동주택 관리규약 준칙'에 대해 행자부의 의견을 반영하여 준칙을 제정 배포해야 할 것이다. 또한 형식적으로 삽입되어 있는 개인정보보호 관련 조항들을 상세하게 추가해야 할 것이다. 주요 개인정보 처리기준, 처리기준 위반 시 벌칙과 과태료, 안전성 확보조치 기준과 위반 시 벌칙들을 상세하게 추가해야 한다.

3.6. 아파트 개인정보보호 자문 센터 설립

최근에 신규 입주한 아파트의 실제 사례를 통해 아파트 개인정보보호에 대한 허술한 인식 수준을 확인할 수 있다. 과도하게 수집된 개인정보를

시스템 관리업체가 철저히 암호화하여 관리해 줄 것이며, 혹시라도 개인정보 유출사고 발생 시 책임을 지고 해결할 것이라는 안이한 생각 속에서 총체적인 보안 위기의식의 부재를 살펴볼 수 있었다.

이러한 문제점을 개선하고 아파트 개인정보보호를 강화하기 위해서는 전문 자문센터가 반드시 필요하다. 아파트 개인정보보호 자문센터를 통해 교육, 홍보, 인증, 평가, 점검 등의 행정력을 점진적으로 확대시켜 아파트 개인정보보호를 강화해야 할 것이다.

IV. 결론 및 향후 연구 방향

개인정보는 목적에 필요한 범위 내에서 최소한의 개인정보만을 수집해야 한다. 또한 정보주체들의 개인정보 침해나 남용이 발생되지 않아야 한다.

아파트 입주자대표회의에 대한 교육, 홍보, 자문 등의 체계적인 업무가 가능하도록 아파트 개인정보보호 자문센터 설립 운영방안에 대한 세부 연구가 필요하다. 또한 최근 홈오토메이션 등 최신 IT기술을 도입한 혁신도시들의 새 아파트들에 대한 추가적인 조사가 필요할 것이다.

이러한 연구 활동을 통해 점차적으로 아파트에서의 개인정보보호가 향상되기를 기대할 수 있을 것이다.

[참고문헌]

[1] 가정정보보호가 우선인 사이버 아파트(김연수, KISA, 2001년)

[2] 공동주택 거주자의 홈오토메이션 시스템 이용에 관한 연구(박창선, 한국엔터테인먼트산업학회, 2013년)

[3] 생체인식 기술기반 개인인증 수단에 따른 사용자 인식(육모세, 2017년)

[4] 공동주택 증축 리모델링의 평면변화에 관한 연구(한수빈, 전남대학교 건축공학과, 석사학위논문, 2017.02)

[5] 아파트 보안 마케팅 강화 시큐리티 아파트가 뜬다(최보윤, 주택과 사람들, 주택협회, 2007년)

[6] 네트워크 기반의 CCTV 증가에 따른 영상정보 보호 관리 기법(이지훈, 석사학위논문, 중앙대학교, 2011년)

[7] 개인정보의 안전성 확보조치 기준 해설서(행자부, KISA, 2017.1월)

[8] 전자상거래에서의 소비자보호에 관한 법·제도적 연구(권상로, 한국콘텐츠학회, 2009년)

〈著者紹介〉



한 충 희 (ChoongHee Han)

1996년 : 동국대학교 컴퓨터공학(학사)
2002년 : 동국대학교 정보보호학과(이학석사)
2017년~현재 : 전남대학교 대학원 정보보안협동과정 박사과정
<관심분야> 개인정보보호 및 주요정보통신기반시설 보호대책 등



박 순 태(SoonTai Park)

1992년 : 단국대학교 전자계산학과 학사
1998년 8월 : 국민대학교 정보과학대학원 정보통신학과 석사
2010년 8월 : 전남대학교 대학원 정보보안협동과정 박사
1994년 7월~1999년 9월 : 육군 전산장교
2000년 4월~현재 : 한국인터넷진흥원 정보보안팀 팀장
<관심분야> 정보보호, 정보보증, IT보안성 평가, 정보보호 인력 양성, 정보통신 기반보호, 무선랜 보안



이 상 준(Sang-Joon Lee)

1991년 : 전남대학교 전산통계학과(이학사)
1993년 : 전남대학교 전산통계학과(이학석사)
1999년 : 전남대학교 전산통계학과(이학박사)
2007년~현재 : 전남대학교 경영학부 교수
<관심분야> 경영정보시스템, 전자상거래, 정보보호 등

라즈베리파이를 활용한 패킷 스니핑(Sniffing) 공격 구현

문대정† , 고민우† , 김인호† , 신인철‡ 1)

† 목포대학교 정보보호영재교육원

‡ 목포대학교 정보보호학과

An Implementation of Packet Sniffing Attacks to WLAN using Raspberry Pi

Dae-Jeong Mun, Min-Woo Ko, In-Ho Kim, Incheol Shin

† Institute of Information Security Education for the Gifted at MNU

‡ Department of Information Security Engineering, Mokpo National University

요 약

최근 카페 문화의 확산과 공공장소에서의 통신사 와이파이(WiFi) 증가로 많은 사람들이 무선 랜(WLAN Wireless Local Area Network)을 사용하고 있는 반면, 취약점을 이용한 해킹 사례 또한 증가하고 있다. 본 논문에서는 라즈베리파이(raspberry pi)를 통해 WiFi 네트워크를 대상으로 하는 스니핑공격 시나리오를 설계하고 구현하는 예를 보인다. 이를 통해 다양한 공공장소에서 활용되고 있는 무선 랜의 취약점을 활용한 정보 탈취의 용이성을 보이며, 이에 대한 대응책 또한 제시한다.

Ⅰ. 서 론

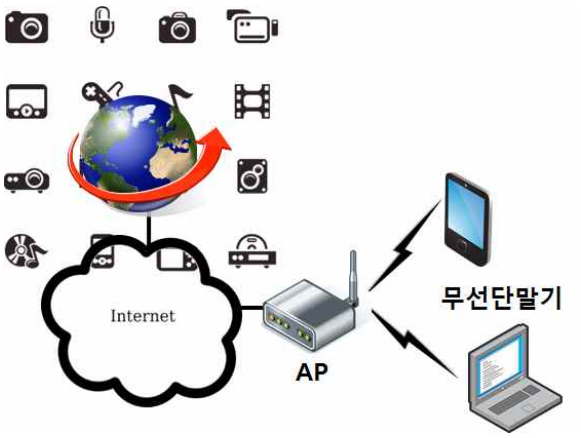
무선 네트워크(Wireless Networks)는 기존의 유선 네트워크(Wired Networks)가 요구하는 전송매체의 설치에서 발생하는 한계점을 극복하고 다양한 지형적 환경에 설치되어 통신메시지를 전달한다. 또한, 이동성(Mobility)을 지원하기 때문에 현재 다양한 형태의 통신기기 사이의 상호연동을 가능케 하고 있다. 이 같은 무선 네트워크의 장점은 유선 통신과 달리 주파수를 기반으로 전자기파를 자유공간에 뿌려주는 기술로 인해 얻어진다. 하지만, 이 같은 장점은 기존의 유선 네트워크에 없었던 재밍(Jamming), 웜홀(Wormhole), 무선통신 도청(Eaves Dropping) 등과 같은 새로운 형태의 보안상 문제점을 야기 시켰다.[6][7]

본 논문에서는 간단한 사회 공학적 공격 기법을 기반으로 하는 스니핑공격(Sniffing Attack)을 통해 무선통신을 가로채는 가상의 공격 시나리오를 설계하고 이를 실험해 보았다. 해당 공격 시나리오는 무선 네트워크에 접속하기 위한 암호가 설정된 무선 랜을 사용하는 카페를 대상으로 하였으며, 이 같은 일반적인 네트워크 환경에 카페 고객들이 인증 없이 사용 가능한 무선 AP(Access Point) 기능을 구현한 라즈베리파이를 설치한 뒤 해당 사용자들이 보내는 메시지를 가로채 중요정보를 탈취하도록 시나리오를 설계하였다.

1) 교신저자 (ishin@mokpo.ac.kr)

Ⅱ. 관련 연구

2.1. 무선랜 구성



〔일반적인 무선 네트워크 구조〕

일반적으로 쓰이고 있는 WLAN은 인터넷과 연결된 AP, 무선통신을 위한 랜카드가 장착된 무선 단말기 등의 물리적인 구성요소와 해당 네트워크를 구분하기 위한 32바이트의 고유 식별자인 SSID(Service Set Identifier)와 사용자 인증/암호화와 관련된 기술적인 요소로 구성되어 있다. [2]

2.2. 무선랜 보안 취약점

⊙ 물리적인 취약점 [4]

인터넷과의 연계를 수행하는 AP는 무선네트워크에서 가장 중요한 역할을 하고 있으나, 외부에 노출된 형태로 위치하기 때문에 '도난 및 파손', '구성설정 초기화', '전원 차단', 'LAN 차단'등의 문제가 발생할 수 있다.

⊙ 기술적인 취약점

앞서 언급한 바와 같이 무선랜은 공기를 전송매체로 사용하기 때문에 유선네트워크와 달리 불법적인 도청이나, 무선전파 전송 장비에 대한 서비스 거부 공격에 매우 취약하다. 본 논문에서는, 일반적으로 사용되고 있는 무선랜에 대한 스니핑공격 시나리오를 설계하고 구현함으로써, 이 같은 기술적인 취약점에 대한 연구를 수행하였다.

⊙ 관리적인 취약점

무선랜을 운영하는 대부분의 기관에서는 AP 장비와 연계되는 단말기들에 대한 관리가 부족하기 때문에 일부 단말기들의 보안상 문제로 인하여 전체 네트워크가 공격자의 표적이 될 수 있다. 또한, 전파 출력에 대한 세심한 관리가 없다면, WiFi 통신 신호에 대한 전파 간섭으로 인한 가용성 문제가 발생할 수 있다.

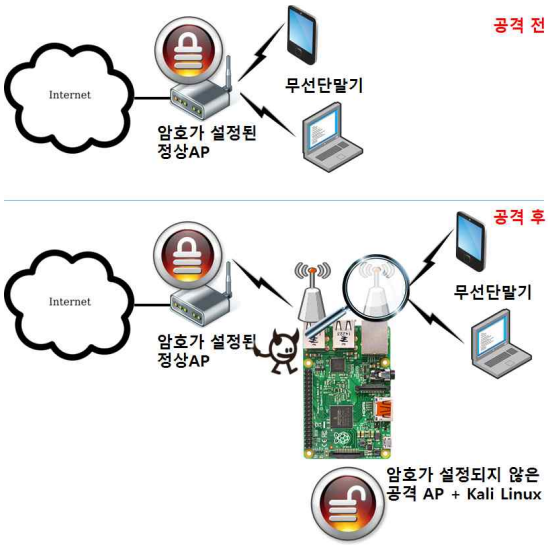
2.3. 라즈베리파이

영국의 라즈베리파이 재단이 학교와 개발도상국에서 기초 컴퓨터 과학 교육을 위해 개발한 싱글 보드 컴퓨터로서 다양한 소프트웨어 개발이나 간단한 하드웨어 개발을 위한 실험용 보드로 사용되고 있다. 본 연구에서는 이 같은 실험용 보드를 이용하여 보안에 취약한 네트워크를 공격하고 중요 데이터를 탈취할 수 있는 가능성을 실험해 보았다.

기존의 라즈베리파이는 라즈비안이라는 운영체제가 설치되어 있으나, 본 연구에서는 칼리 리눅스(Kali Linux)라는 정보 보안을 테스트하기 위한 오픈 소스 리눅스 배포판을 사용하여 패킷 스니핑 공격을 구현하였다. [2]

Ⅲ. 무선랜 스니핑 공격 시나리오

3.1. 라즈베리파이 기반 스니핑 공격 시나리오



〔라즈베리파이 기반 스니핑 공격 구조〕

① 라즈베리파이에 Kali Linux를 설치한다.
② 라즈베리파이에 무선랜카드를 두 개 설치한 뒤, AP기능을 구현한다.
③ 하나의 랜카드는 기존의 정상적으로 설치된 AP와 암호를 통해 연결하고 다른 하나는 다른 무선단말기에 AP서비스를 제공하도록 기능을 설정한다.
④ 라즈베리파이에 구현된 AP의 네트워크 명칭을 공격 대상 네트워크의 이름과 비슷하게 설정하고, 암호를 설정하지 않음으로써, 다른 무선 단말기기들의 연결을 유도한다.
⑤ 공격용 AP를 통해 전달되는 메시지들을 Kali Linux 내 설치된 Wireshark를 통해 수집한 뒤 분석하여 중요정보를 식별 및 탈취한다.
⑥ 혹은 tcpdump 명령어를 통해 전달되는 메시지들을 수집/분석하여 중요정보를 식별하거나 탈취한다. [5]
* tcpdump는 주어진 조건식을 만족하는 네트워크 인터페이스를 거치는 패킷들의 헤더를 출력해 주는 프로그램이다.

3.2. 라즈베리파이 AP구현

라즈베리파이에 무선랜카드를 활용하여 AP를 구현하였다. [3]

① sudo apt-get install libnl
libnl은 netlink 프로토콜 기반의 리눅스 커널 인터페이스를 지원하는 라이브러리로서, hostapd가 kernel과의 통신을 지원한다.
② sudo apt-get install iw
hostapd 정상 작동 여부 및 인터페이스 관리를 위한 프로그램이다.
③ sudo apt-get install libssl-dev
hostapd 데몬 컴파일을 위한 라이브러리이다.
④ sudo apt-get install dnsmasq hostapd
dnsmsq는 1000개 이하의 단말기를 위한 소규모 네트워크에서 활용 가능한 간단한 DHCP (Dynamic Host Configuration Protocol) / DNS (Domain Name Server) 서버이며, hostapd는 인증기능을 포함한 AP 서비스를 지원하는 데몬이다.
⑤ hostapd.conf 파일 편집을 통해 SSID 명을 공격대상 네트워크와 비슷하게 설정함으로써, 다른 단말기들의 접속을 유인한다.
⑥ 다른 단말로부터 무선랜카드로 수집된 메시지를 기존의 정상적인 AP로 패킷을 전달하기 위한 네트워크 브릿지(Bridge)를 설정한다.

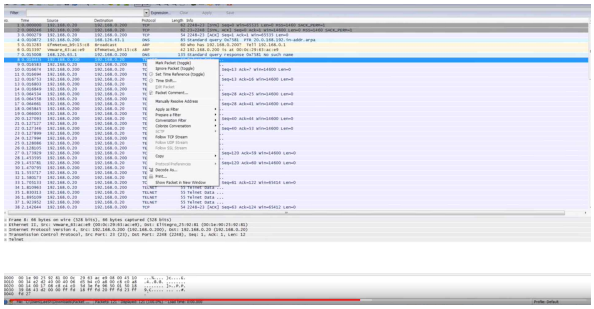
```
> brctl addbr 브릿지명
> brctl addbr 브릿지명 wlan0
> brctl addbr 브릿지명 wlan1
```

⑦ hostapd -dd hostapd.conf
공격용 라즈베리파이를 통해 AP기능을 실행시킴으로서, 네트워크 메시지 스니핑을 수행한다.

3.3. 패킷 스니핑

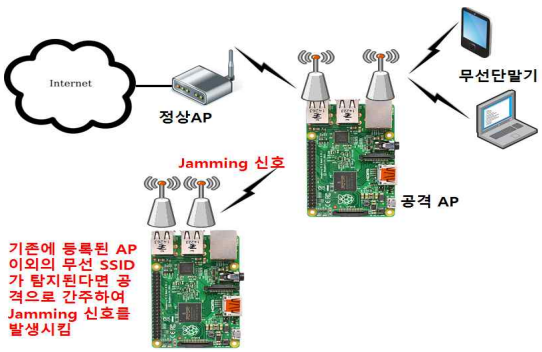
패킷 스니핑을 위한 AP기능을 라즈베리파이에 구현한 뒤, 해당 기기 내 Kali Linux에 설치된 와이어샤크 프로그램을 활용하여, 통신 메시

지를 수집하고 분석한다.



〔칼리리눅스 내 Wireshark를 통해 수집된 메시지〕

Ⅳ. 대응방법



〔무선 네트워크 대상 패킷스니핑 공격 대응방법〕

이 같은 패킷 스니핑공격에 대응하기 위해 본 연구에서는 등록되지 않은 SSID가 발견되었을 경우 해당 네트워크에 무선단말기들이 접속하지 못하도록 재밍신호를 발생시키는 방법을 제안한다.

Ⅴ. 결 론

본 논문에서는 시중에서 쉽게 구입이 가능한 라즈베리파이를 이용하여 공공장소에서 운영되고 있는 무선네트워크를 대상으로 하는 패킷 스니핑 공격을 구현하였다. 본 논문을 통해 무선 단말기를 사용하고 있는 사용자들이 암호가 설정되지 않는 네트워크로의 접속을 유도하여 패킷을 수집하여 분석 및 주요정보 탈취를 수행하였다. 해당 공격은 간단한 사회공학적 기법과 더불어 시나리오로 개발 및 구현되어 무선 네트워크에 대한 취약점을 활용한 공격의 위협을 설명하였다. 또한 해당 공격의 문제점을 파악하여 해결책을 설명함으로써 향후 연구방향을 제시하였다.

[참 고 문 헌]

[1] 한국인터넷진흥원, “알기쉬운 공중 무선랜 보안 안내서”

[2] https://docs.kali.org/kali-on-arm/install-kali-linux-arm-raspberry-pi

[3]https://frillip.com/using-your-raspberry-pi-3-as-a-wifi-access-point-with-hostapd/,”

[4] Silverhwak, “무선랜 취약점 분석”

[5] http://www.tcpdump.org/ tcpdump_man.html

[6] SANS, “Understanding Wireless Attacks and Detection”

[7] Mallikarjun Hangargi, “Denial of Service Attacks in Wireless Networks”

<저자소개>

문 대 정 (Daejeong Moon)
목포마리아회고등학교

고 민 우 (Minwoo Ko)
목포홍일고등학교

김 인 호 (Inho Kim)
목포문태고등학교



신 인 철 (Incheol Shin)
2010년 8월 : University of Florida 박사 졸업
2014년 3월~현재 : 국립목포대학교 정보보호학과 조교수
관심분야 : 컴퓨터이론, 네트워크, 정보보호

금융 정보보호 관리체계(F-ISMS) 분석을 통한 기업별 정보보호 관리체계(ISMS) 연구의 필요성

주영국

고려대학교 컴퓨터정보통신대학원

The necessity of studying Information Security Management System(ISMS) by enterprise through Finance-Information Security Management System(F-ISMS) analysis

Young-Guk Ju

graduate school of Computer & Information Technology,
Korea University.

요 약

고도화 되고 있는 사이버테러의 발생으로 정보보호의 중요성이 대두되고 있으며, 정보보호를 위해 정보보호 관리체계(ISMS) 인증 제도를 운영하고 있다. 정보보호 관리체계(ISMS) 인증은 일원화된 점검 항목으로 인증을 수행하고 있어 점차 확대되고 있는 의무대상자에 적용하기는 기업 특성에 따라 어려움이 있다. 따라서 금융보안원에서 제시한 금융 정보보호 관리체계(F-ISMS)를 분석하여 효과적인 정보보호를 위한 기업 유형별 정보보호 관리체계(ISMS) 인증 점검 항목의 연구가 필요하다.

I. 서론

2013년 3.20 테러(MBC, KBS, 농협 등 전산마비), 6.25 테러(개인정보 유출), 2016년 7월 인터파크 개인정보 유출, 2017년 6월 인터넷나야나 랜섬웨어 감염 사태 등의 크고 작은 사이버테러가 발생했으며. 악성코드 감염, APT(지능형 지속공격) 공격, 랜섬웨어 등 다양한 방식과 개인정보유출, 전산마비, 금전 등 다양한 목적을 가지고 사이버테러가 발생했다. 향후에도 사이버테러가 지속될 것이며, 과거보다 고도화된 사이버테러가 발생할 것으로 예상됨에 따라 정보보안의 중요성이 부각되고 있다.

현재 국내에서는 기업·기관의 정보보호를 위해 정보보호 관리체계(ISMS) 인증, 개인정보관리체계(PIMS), 정보보호등급제. 클라우드 보안인증제 등 다양한 인증 제도를 시행하고 있다. 하지만 인증의 경우 일관된 기준을 가지고 기업을 평하기 때문에 모든 사업유형의 정보보안을 평가하기에는 한계가 있다. 따라서 기업 유형에 따라 기업 간의 관점차이를 이해하고 효율적인 정보보호를 위한 점검 기준에 대한 연구가 필요하다.

II. 관련 연구

본 논문에서는 국내에서 정보보호를 위해 구축운영하고 있는 정보보호 관리체계(ISMS) 인증제도와 금융보안원에서 금융 분야의 정보보호 관리체계(ISMS)를 위해 발표한 금융 정보보호 관리체계(F-ISMS)를 비교분석하도록 한다.

2.1 정보보호 관리체계(ISMS) 인증의 이해

2001년 7월 국내 기업·기관이 스스로 정보보호 관리체계(ISMS)를 구축·운영하는데 활용할 수 있도록 관리체계 모델을 개발하고, 정보통신망법(정보통신망 이용촉진 및 정보보호 등에 관한 법률) 개정을 통하여 정보보호 관리체계 인증 제도를 도입했으며, 2012년 2월 정보통신방법 개정(2013년 2월 시행)을 통해 인증을 의무

적으로 받도록 하고 있다. 정보보호 관리체계(ISMS)의 인증체계는 [그림 1]과 같으며[1], 정보통신망법 제47조제2항에서 정의하는 정보보호 관리체계(ISMS) 의무 대상자는 아래 [표 1]과 같다[2].



[그림 1] 정보보호 관리체계(ISMS) 인증체계

[표 1] 정보보호 관리체계(ISMS) 의무대상자 기준

구분	의무대상자 기준
ISP (정보통신망 서비스)	「전기통신사업법」 제6조제1항에 따른 허가를 받은 자로서 서울특별시 및 모든 광역시에서 정보통신망 서비스를 제공하는 자
IDC (집적정보 통신시설)	정보통신망법 제46조에 따른 집적정보통신시설 사업자
다음의 조건 중 하나라도 해당하는 자	연간 매출액 또는 세입이 1,500억 원 이상인 자 중에서 다음에 해당되는 경우 - 「의료법」 제3조의4에 따른 상급종합병원 - 직전연도 12월 31일 기준으로 재학생 수가 1만 명 이상인 「고등교육법」 제2조에 따른 학교
	정보통신서비스 부문 전년도(법인인 경우에는 전 사업연도를 말한다) 매출액이 100억 원 이상인 자
	전년도 직전 3개월간 정보통신서비스 일일평균 이용자 수가 100만 명 이상인 자

정보보호 관리체계(ISMS) 인증은 2002년부터 시행하여 2017년 10월까지 총 701건의 인증서가 발급되었으며, 그 중 유지되고 있는 인증서는 480건이다. 인증을 취득 및 유지하기 위해서는 최초심사, 사후심사, 갱신심사를 받아야한다. 정보보호 관리체계(ISMS) 인증 심사 설명과 주기는 [그림 2]와 같다[3].



[그림 2] 정보보호 관리체계(ISMS) 인증 심사 주기

- 최초심사 : 정보보호 관리체계(ISMS) 인증 취득을 위한 심사 (범위 변경 등 중요한 변경사항 발생 시에도 최초심사)
- 사후심사 : 정보보호 관리체계(ISMS)를 지속적으로 유지하고 있는지에 대한 심사 (연 1회 이상)
- 갱신심사 : 유효기간(3년) 만료일 이전에 유효기간 연장을 목적으로 하는 심사

정보통신망법을 통해 인증이 의무화되기 전인 2002년부터 2012년까지의 취득한 인증서는 45건으로 전체의 약 9.4%이며, 인증이 의무화된 이후인 2013년부터 2017년 10월까지는 435건으로 전체의 약 90.6%이다[4]. 기업·기관이 스스로 정보보호 관리체계(ISMS)를 구축·운영하는데 활용하도록 개발했으나, 자율적 인증 취득이 활발하게 이루어 지지 않은 것을 알 수 있다. 따라서 정보보호에 대한 중요성이 부각될수록 정보보호 관리체계(ISMS) 인증 의무 대상범위가 확대될 것으로 예상된다.

2.2 금융 정보보호 관리체계(F-ISMS) 인증의 이해

금융회사는 정보통신망법 시행령 제49조제2항제2호, 제3호에 의해 정보보호 관리체계(ISMS) 인증 의무대상에서 제외되었으며, 자율

적으로 인증을 취득하고 있다. 금융보안원 정보보호 관리체계(ISMS) 인증서는 2017년 9월 기준 은행 10개사, 금융투자 17개사, 보험 5개사, 중소서민 9개사, 전자금융업자 2개사, 기타 5개사로 총 48개사에 발급되었다[5]. 인증을 취득, 유지하기 위해서는 정보보호 관리체계(ISMS) 인증과 동일한 주기로 심사를 받아야 한다.

금융보안원에서는 전자금융감독규정, 신용정보업감독규정 등 금융보안 관련 규정 및 표준지침을 바탕으로 금융 분야에 적합한 금융 정보보호 관리체계(F-ISMS) 인증기준 점검항목을 발표했다. 금융 정보보호 관리체계(F-ISMS)는 2017년까지는 기존의 정보보호 관리체계(ISMS)와 금융 정보보호 관리체계(F-ISMS) 중 선택해서 심사를 진행할 수 있으나, 2018년도부터는 정보보호 관리체계(ISMS) 인증을 원하는 금융기업에서는 금융 정보보호 관리체계(F-ISMS) 인증을 취득해야 한다[5].

III. 정보보호 관리체계(ISMS) 비교분석

정보보호 관리체계(ISMS) 인증과 금융 정보보호 관리체계(F-ISMS) 인증의 세부 항목 및 개수와 변경된 항목과 신설된 항목의 수는 아래 [표 2]와 같다[6][7].

[표 2] 정보보호 관리체계(ISMS)와 금융 정보보호 관리체계(F-ISMS) 인증 세부 항목 비교

인증 분야	통제 분야	통제 항목	ISMS 점검 항목	F-ISMS 점검 항목	변경 항목	신설 항목
정보보호 관리과정	정보보호정책 수립 및 범위설정	2	4	4	1	0
	경영진 책임 및 조직 구성	2	4	6	1	2
	위험관리	3	11	11	2	0
	정보보호 대책 구현	2	3	3	0	0
	사후관리	3	6	7	2	1
정보보호 대책	정보보호정책	6	13	14	2	1
	정보보호조직	4	7	10	2	3
	외부자 보안	3	4	14	1	10
	정보자산 분류	3	7	7	2	0
	정보보호 교육	4	10	10	1	0
	인적 보안	5	11	12	4	1
	물리적 보안	9	21	27	4	6
	시스템 개발 보안	10	22	28	7	6
	암호 통제	2	8	12	3	4
	접근 통제	14	46	52	10	6
	운영보안	22	56	83	16	27
	침해사고 관리	7	14	16	1	2
	IT재해복구	3	6	8	3	2
계	18	104	253	324	62	71

정보보호 관리체계(ISMS)와 금융 정보보호 관리체계(F-ISMS) 인증기준 점검 항목을 분석한 결과 2개의 인증분야, 18개 통제 분야, 104개의 통제항목으로 동일하게 구성되어 있으나, 점검항목의 경우 기존의 정보보호 관리체계 점검 항목에서 62개의 항목을 구체화 또는 변경되었으며, 71개의 항목을 신설하여 항목을 구성했다. 금융 정보보호 관리체계(F-ISMS) 인증기준 점검항목의 주요 변경 또는 신설 항목은 아래 [표 3]과 같다.

[표 3] 금융 정보보호 관리체계 점검항목 변경 및 추가 세부 내용

인증 분야	통제 분야	주요 변경 내용
정보보호 관리과정	경영진 책임 및 조직 구성	• 정보보호 예산 및 인력 구성에 대한 기준 점검
	사후 관리	• 최고경영자에게 법적 요구사항 준주여부 검토 결과 보고 절차 점검

인증 분야	통제 분야	주요 변경 내용
정보보호 대책	정보보호 정책	• 정보보호 관리체계 운영을 위해 필요한 문서정보 관리
	정보보호 조직	• 정보보호위원회 심의·의결 사항 점검 • 정보보호 위원회 심의·의결 사항에 대한 보고절차 점검
	외부자 보안	• 전자금융보조업자에 대한 관리 정책 및 절차 점검
	시스템개발보안	• 자체 보안성심의 점검 • 전자금융거래프로그램의 무결성 검증
	암호통제	• 개인정보에 대한 암호화 적용 점검
	운영보안	• 공개서버에 대한 반기 1회 취약점 수행 및 조치 점검 • 전자금융기반시설 취약점 진단 여부 및 보고 절차 점검 • 중요원장 작업에 대한 기록 보존 점검
	IT 재해복구	• 주요 정보 시스템에 대한 이중화 또는 예비 장치 확보 점검

정보보호 관리체계(ISMS)를 전자금융감독규정, 신용정보업감독규정 등의 금융보안 관련 규정과 금융전산표준지침 등 정보보호 관련 지침을 바탕으로 점검항목을 변경 또는 신설하여 금융 정보보호 관리체계(F-ISMS)를 구축할 수 있도록 기준을 개선하였다.

IV. 결론

운영되고 있는 정보보호 관리체계(ISMS) 인증은 의무대상이 확대되고 있으나 일원화된 점검항목을 기준으로 인증을 수행하고 있다. 금융보안원에서 발표한 금융 정보보호 관리체계(F-ISMS)와 같이 기업이 적용 받는 규정 및 지침이 점검항목에 포함되도록 변경 또는 신설이 필요하며, 기업의 특성 또한 점검 항목에 반영되어야한다.

정보통신망법 제47조(2016년 6월 2일 시행)에 따라 의무대상가 된 병원은 의료법, 의료기기법 등의 법률, 대학교는 고등교육법 등의 법률을 반영하여 병원, 대학교 각각에 특화된 점검항목을 도출하는 등 각 기업의 관점의 차이를 줄이고 효율적인 정보보호 관리체계(ISMS)의 연구가 필요하다.

V. 향후 연구 과제

정보보호 관리체계(ISMS) 인증 의무대상자 확대로 추가된 상급종합병원, 대학교의 중요시스템의 특성을 분석하여 Weak Point에 대한 점검 항목을 도출하고 적용받는 법률, 지침, 규정을 분석하여 Compliance 준수 여부를 점검할 수 있는 항목을 도출하는 것을 향후 연구 과제로 한다. 기업별 정보보호 관리체계(ISMS)를 개발은 기업이 정보보호 관리체계(ISMS) 인증 취득 시 활용하여 효율적인 인증 취득과 기업의 정보보안 강화를 기대한다.

[참고문헌]

[1] 한국인터넷진흥원 ISMS 제도소개, https://isms.kisa.or.kr/main/isms/intro/

[2] 한국인터넷진흥원, ISMS 인증제도 안내서, pp. 11-26, 2017.04

[3] 한국인터넷진흥원 ISMS 신청절차, https://isms.kisa.or.kr/main/isms/request/

[4] 한국인터넷진흥원 ISMS 인증서 발급 현황, https://isms.kisa.or.kr/main/isms/notice/

[5] 금융보안원 ISMS 인증, http://www.fsec.or.kr/fsec/subIndex/320.do

[6] ISMS 인증기준 세부점검항목, https://isms.kisa.or.kr/main/isms/notice/

[7] 금융 분야 ISMS 인증기준 점검 항목, http://www.fsec.or.kr/user/bbs/fsec/148/319/bbsDataList.do

안드로이드 환경에서 랜섬웨어 탐지를 위한 랜섬웨어 분석 시스템 연구

박인규*, 곽진**

*아주대학교 컴퓨터공학과 정보보호응용및보증연구실

**아주대학교 사이버보안학과

Research on Ransomware Analysis System for Detection of Ransomware in Android Environment

In-Kyu Park*, Jin Kwak**

*ISAA Lab., Department of Computer Engineering, Ajou University.

**Department of Cyber Security, Ajou University.

요 약

안드로이드 시장이 확대되고 안드로이드 사용자가 증가하면서 개인 업무 영역까지 확대되었다. 이에 따라 안드로이드에 대한 보안은 항상 문제가 되어왔으며, 최근에는 안드로이드에 유포되는 랜섬웨어까지 등장하면서 보안 문제는 더욱 대두되었다. 안드로이드 환경의 애플리케이션은 리패키징을 이용해 악성행위를 수행하는 애플리케이션을 유포하기 용이하기 때문에 안드로이드는 많은 랜섬웨어의 표적이 되어 왔으며, 랜섬웨어는 현재에도 지속적으로 등장하고 있다. 현재까지 등장한 안드로이드 랜섬웨어에 대한 대응방안은 시스템 자원 사용률을 이용한 파일 암호화 탐지나 미끼 파일을 통한 암호화되는 파일에 대한 모니터링 등의 방안이 있지만 이는 파일에 대한 보호가 정상적으로 이루어지지 않을 수 있으며 이미 랜섬웨어에 감염된 이후에 탐지할 확률이 높다. 안드로이드 랜섬웨어로부터 디바이스 정보를 보호하기 위해 파일에 대한 모니터링이나 보호보다는 랜섬웨어에 대한 관리가 필요하다. 본 논문에서는 안드로이드 랜섬웨어를 방지하기 위한 애플리케이션 분석 시스템을 제안한다.

I. 서론

안드로이드 환경에서 사용되는 애플리케이션은 리패키징 기법을 통해 재유포를 할 수 있다. 리패키징은 일반적으로 유포되는 애플리케이션에 소스코드를 일부 수정하여 다시 패키징하는 기법으로 이를 악용할 경우 악성코드를 추가하여 유포할 수 있다. 이러한 이유로 안드로이드에서 악성 애플리케이션이 쉽게 유포될 수 있으며 최근에는 랜섬웨어까지 등장하였다. 랜섬웨어는 일반적으로 사용자의 시스템을 감염시켜 시스템에 존재하는 파일을 암호화한 후 복호화 키 제공을 조건으로 금품을 요구하는 악성코드이다. 하지만 안드로이드 랜섬웨어의 경우 파일을 암호화하는 악성행위도 수행하지만 일반적인 랜섬웨어와는 다르게 사용자의 스마트폰을 잠그는 악성행위도 수행한다. 사용자의 스마트폰을 잠그는 경우는 기존 사용자의 잠금 패턴을 제거한 후 공격자가 임의의 잠금 패턴을 삽입하고 잠금 패턴을 해제할 수 있는 PIN 번호를 초기화하여 사용자가 잠금패턴을 재설정할 수 없도록 한다[1,2].

일반적으로 안드로이드 환경에서 랜섬웨어 탐지를 위한 데이터 백업, 자원 사용률 모니터링, 미끼 파일 등의 방법이 존재한다. 이러한 방법들은 파일 암호화 탐지 실패, 시스템 자원 상승률이 다른 이유로 급격히 상승하거나 랜섬웨어의 암호화 순서에 따라 미끼파일이 첫 번

이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. NRF-2017R1E1A1A01075110).

째로 암호화되지 않는 등 다양한 이유에서 오탐률이 높을 수 있다. 랜섬웨어로부터 데이터를 보호하기 위해서는 데이터 위주의 보호보다는 랜섬웨어가 악성행위를 수행하여 데이터 암호나 스마트폰 잠금을 수행하기 전에 해당 랜섬웨어를 관리할 수 있는 방안이 필요하다.

본 논문에서는 애플리케이션을 관리하기 위해 설치 및 실행 탐지 모듈, 제어 모듈, 분석 모듈, 삭제 모듈을 제안하고 각 모듈에서 수행되어야 할 기능을 설명한다.

본 논문의 구성은 다음과 같다. 2장에서 안드로이드 랜섬웨어에 대한 사전 연구를 진행한다. 3장에서는 안드로이드 랜섬웨어 탐지를 위한 시스템을 제안하고, 마지막으로 4장에서 결론을 맺는다.

II. 관련 연구

2.1 안드로이드 환경의 랜섬웨어

안드로이드 환경에서 유포되는 랜섬웨어의 경우 일반적으로 PC환경에서의 랜섬웨어와는 다르게 데이터 암호화 수행뿐만 아니라 사용자의 안드로이드 디바이스를 잠그는 기능을 수행한다. 데이터 암호화의 경우 AES 암호나 RSA 암호를 통해 사용자의 안드로이드 디바이스에 존재하는 모든 데이터에 대한 암호화를 수행한다. 디바이스 잠금은 사용자가 설정한 기본 패턴에 대한 잠금 대신 공격자가 원하는 패턴의 잠금으로 재설정한다. 본래 화면 잠금을 재설정하더라도 사용자가 기존에 설정해둔 PIN 번호로 초기화할 수 있지만 PIN 번호도 특정 API 사용으로 초기화할 수 있다. PIN 번호 초기화의 경우 안드로이드 7.0 버전부터 보안상의 이유로 초기화 기능을 사용할 수 없게 업데이트 되었지만 초기화만 할 수 없고 PIN 번호가 설정되어 있지 않은 디바이스에 대해 여전히 재설정할 수 있는 기능이 남아있다[2].

안드로이드 환경에서 사용자가 애플리케이션을 설치할 경우 설치 즉시 애플리케이션이 실행되지 않는다. Google Play에서 설치했을 경우 열기 버튼을 눌러 실행하거나 APK 파일로 설치했을 경우에도 단순히 설치만 완료될 뿐 애플리케이션이 실행되지 않는다. 애플리케이션이 실행되지 않을 경우 랜섬웨어는 데이터 암호화나 사용자 안드로이드 디바이스 잠금 기능을 수행할 수 없으므로 위협이 되지 않을 수 있다. 하지만 공격자가 강제로 애플리케이션을 실행시킬 수 있어 단순히 안드로이드 디바이스에 설치만하고 실행하지 않는다고 해서 감염되지 않았다고 할 수 없다[2,3].

2.2 랜섬웨어 탐지 방법

2.2.1 데이터 백업

일반적으로 랜섬웨어가 데이터 암호화를 수행하는 것에 대비하기 위해 랜섬웨어가 데이터에 대한 암호화를 수행하는 즉시 이를 탐지하여 중요 데이터에 대한 백업을 수행하는 것으로 데이터를 보호한다. 일반적으로 시스템 상에서 의심스러운 파일 훼손 행위가 수행될 때나 사용자가 설정한 주기마다 데이터에 대한 백업을 수행한다. 하지만 데이터 백업의 경우 사용자가 소유한 데이터가 많아질수록 데이터 백업에 소요되는 시간이 증가하며 랜섬웨어 악성행위 수행이 늦어질 경우 데이터 백업이 정상적으로 수행되지 않을 수 있다[1,2].

2.2.2 시스템 자원 사용률

시스템 자원 사용률은 랜섬웨어가 사용자 시스템 상에서 데이터를 암호화할 때 시스템 자원을 많이 소모하는 것을 이용한 탐지 기법이다. 시스템 자원 사용률을 지속적으로 모니터링하고 특정 순간에 자원 소모량이 급격히 높아질 때 랜섬웨어가 악성행위를 수행하는 것으로 판단한다. 하지만 이러한 방법의 경우 자원 사용률이 랜섬웨어가 데이터를 암호화할 경우에만 증가하지 않고 다른 이유로 인해 사용률이 높아질 수 있어 오탐률이 높을 수 있다[1,2].

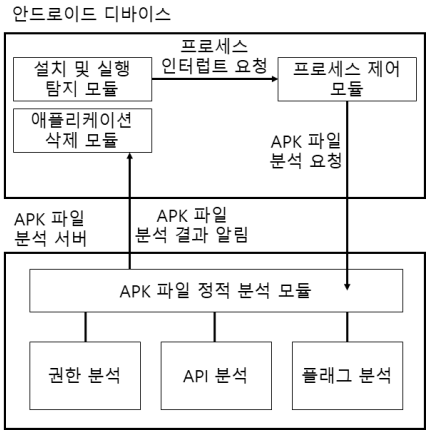
2.2.3 미끼(Decoy) 파일

미끼 파일을 통한 랜섬웨어 탐지는 미끼 파일을 생성해서 미끼 파일이 암호화되는지에 대한 여부를 지속적으로 모니터링해서 랜섬웨어가 데이터를 암호화할 때 미끼 파일을 암호화할 경우 이를 탐지하는 기법이다. 하지만 랜섬

웨어가 미끼 파일을 늦게 암호화 할 경우 그만큼 늦게 탐지가 되고 이미 암호화된 파일에 대해 복구를 수행할 수 없다[2,3].

III. 애플리케이션 분석 방안

랜섬웨어로부터 데이터를 보호하기 위해 데이터를 보호하는 것보다 랜섬웨어의 악성행위 수행을 방지하도록 한다. 제안하는 애플리케이션 관리 시스템은 네 가지 모듈로 구성되어 있다.



[그림 1] 애플리케이션 분석 구성도

3.1 애플리케이션 설치 및 실행 탐지 모듈

3.1.1 애플리케이션 설치 탐지

안드로이드 디바이스의 경우 브로드캐스트 리시버를 통해 디바이스에서 발생하는 다양한 이벤트를 감지할 수 있다. 애플리케이션 설치 탐지에서는 애플리케이션 설치시에 ACTION_PACKAGE_ADDED 액션으로, 업데이트시에 ACTION_PACKAGE_REPLACED 액션으로 설치 및 업데이트 이벤트를 감지한다.

3.1.2 애플리케이션 실행 탐지

2절에서 언급한바와 같이 안드로이드 랜섬웨어가 설치되는 것은 큰 위협이 되지 않을 수 있다. 하지만 랜섬웨어가 실행되면서 큰위협이 된다. 안드로이드는 사용자가 랜섬웨어를 실행하지 않아도 공격자가 특정 명령을 통해 랜섬웨어를 실행시킬 수 있다. 따라서 애플리케이션이 실행될 때 이벤트를 탐지한다. 이벤트 탐지의 경우 AccessibilityService를 통해 실행 이벤트를 탐지하며 설치 탐지 모듈에서 수집한 정보를 통해 설치된 애플리케이션이 실행될 때 이를 탐지한다.

3.2 프로세스 제어 모듈

안드로이드 디바이스는 특정 명령을 통해 애플리케이션을 제어할 수 있다. killBackgroundProcess를 통해 애플리케이션에 대한 제어를 수행할 수 있으며 애플리케이션 실행 탐지 모듈에서 애플리케이션 실행을 탐지할 경우 프로세서 제어 모듈에서 애플리케이션 종료 명령을 수행한다.

3.3 프로세스 정적 분석 모듈

정적 분석을 위해 해당 애플리케이션에 대한 권한, API, 플래그를 분석한다. 분석은 특정 분석 서버를 통해 수행한다.

안드로이드의 경우 애플리케이션에 특정 권한을 부여해야 API나 플래그를 사용할 수 있기 때문에 분석을 수행할 때 권한, API, 플래그 순으로 분석을 진행한다.

3.3.1 권한 분석

권한 분석에서는 랜섬웨어에서 데이터 암호화나 디바이스 잠금을 수행하기 위해 사용되는 권한에 대해 정적 분석을 수행한다. 다음 [표 1]은 랜섬웨어가 악성행위를 수행하기 위해 사용될 수 있는 권한 목록이다.

[표 1] 랜섬웨어가 사용하는 권한

권한	설 명
android.permission.DISABLE_KEYGUARD	- keyguardmanager를 사용하기 위한 권한으로 기본 잠금화면을 제거하고 새로운 잠금 화면을 실행하기 위해 사용
android.permission.WAKE_LOCK	- 안드로이드 디바이스의 화면을 제어하며 공격자가 금품 요구 등의 메시지를 띄우기 위해 사용

3.3.2 API 분석

API 분석에서는 랜섬웨어가 특정 악성 행위를 수행하기 위해 사용하는 API에 대한 분석을 수행한다. 다음 [표 2]는 랜섬웨어가 악성행위를 수행하기 위해 사용될 수 있는 API 목록이다.

[표 2] 랜섬웨어가 사용하는 API

API	설 명
dofinal()	- 랜섬웨어의 기능 중 하나인 데이터 암호화를 수행하기 위해 사용되는 API
disableKeyguard()	- 사용자가 설정한 잠금 화면을 변경하기 위해 사용
resetPassword()	- 잠금 화면을 재설정한 후 이를 해제하지 못하도록 PIN 번호 재설정

3.3.3 플래그 분석

안드로이드에서는 API 대신 플래그를 사용하여 특정 행위를 수행할 수 있다. 따라서 API만 분석을 하는 것이 아닌 특정 플래스에 대해서도 분석을 수행한다. 다음 [표 3]은 랜섬웨어가 악성행위를 수행하기 위해 사용될 수 있는 API 목록이다.

[표 3] 랜섬웨어가 사용할 수 있는 플래그

플래그	설 명
FLAG_SHOW_WHEN_LOCKED	- 디바이스가 잠금상태여도 특정 화면을 보여주기 위해 사용 - 디바이스가 잠금 되더라도 금품 요구에 대한 메시지를 띄우기 위해 사용.
FLAG_DISMISS_KEYGUARD	- keyguard를 해제하기 위해 사용 - 비밀번호로 설정이 되어 있지 않았을 경우에 사용 가능

3.4 애플리케이션 삭제 모듈

애플리케이션에 대한 분석이 종료되면 분석 모듈은 삭제 모듈에 분석 결과를 전송한다. 결과를 전송 받을 경우 삭제 모듈은 애플리케이션에 대한 위험도를 판별하여 삭제 여부를 결정한다. 애플리케이션 삭제는 ACTION_DELETE 액션을 통해 삭제 가능하며 설치 탐지 모듈에서 수집한 패키지명을 통해 삭제한다.

IV. 결론

본 논문에서는 안드로이드 환경에서 발생하는 랜섬웨어로부터 디바이스를 보호하기 위해 애플리케이션 분석 시스템을 제안하였다. 애플리케이션에 대한 분석 수행을 위해 애플리케이션 설치 및 실행 탐지 모듈, 프로세스 제어 모듈, APK 파일 정적 분석 모듈, 애플리케이션 삭제 모듈로 구성하였으며, 랜섬웨어가 사용할 수 있는 기능에 대한 분석 수행을 위해 권한, API, 플래그를 차례로 분석하였다.

향후 본 논문에서 제안한 시스템을 실제 구성하고 안드로이드 환경에서 발생할 수 있는 랜섬웨어에 대한 실질적 분석을 통해 탐지율에 대한 실증연구를 진행할 예정이다.

[참고문헌]

[1] Tian Yang, Yu Yang, Kai Qian, Dan Chia-Tien Lo, Ying Qian, Lixin Tao, “Automated Detection and Analysis for Android Ransomware”, 2015 IEEE 17th International Conference on High Performance Computing and Communications(HPCC), August. 2015.

[2] 윤정무, 조제경, 류재철, “파일 I/O Interval을 이용한 랜섬웨어 공격 차단 방법론”, Journal of The Korea Institute of Information Security & Cryptology VOL.26, NO.3, June. 2016.

[3] 김봉준, 김운수, 이정환, 임신혁, 송상근, 이상준, “안드로이드 플랫폼 기반 프로세스 모니터링을 이용한 랜섬웨어 방지 시스템의 설계 및 구현”, 한국정보과학회 동계학술발표회 논문집, 2015.

정보보안성강화를 위한 공공기관 위협모델링(Threat modeling) 제안

최종현*

*고려대학교 정보보호학과

A proposal for public institution threat modeling for enhancing information security

Jong-Hyun Choi*

*Division of information security, Korea University.

요 약

사이버공격에 대비하여 공공기관은 어플리케이션 개발단계에서 시큐어코딩, 데이터 암호화등의 보안조치를 취한다. 하지만 이 같은 조치로는 갈수록 심각해지는 사이버공격을 막기 역부족이므로 어플리케이션과 기타 하드웨어 인프라 등에 대한 다각적 고려가 필요하다. 다양한 환경을 분석하고 해커로부터의 예상공격을 대비하기 위해서는 위협모델링을 활용해야한다. 이 논문에서는 마이크로소프트사와 EMC사에서 사용 중인 위협모델링 이론을 소개하고 이를 바탕으로 공공기관 위협모델링을 제안한다.

I. 서론

위협모델링(Threat modeling)은 공격자의 관점에서 예상되는 위협(Threat)을 찾아내고 분석하는 프로세스이다[1]. 이를 통해서 소프트웨어가 가질 수 있는 보안취약점(vulnerability)을 미리 분석하고 설계에 반영하여 보안이 강화된 소프트웨어를 만들 수 있다[2]. 이는 제품 출시 단계에 발견되는 결함을 제거하는 것이 나중에 조치하는 것 보다 경제적으로 월등히 이점이 있다는 이론에 비추어 보면 반드시 실행되야 할 프로세스임을 의미한다[3].

또한 위협모델링(Threat modeling)은 비단 소프트웨어 설계에만 국한되지 않고 운영 중인 소프트웨어에 적용할 수 있다. 보안적 관점에서 소프트웨어 아키텍쳐를 분석하고 위협(Threat) 및 보안취약점(vulnerability)을 발견하고 조치할 수 있는데 이러한 점에서 위협모델링은 정보보안 역량이 부족하여 보안취약점을 능동적으로 분석하지 못하고 정부의 보안가이드라인 및 감사기준에만 맞추기에도 급급한 공공기관에 유용한 분석모델이 될 것이다.

이 논문에서는 마이크로소프트사와 EMC사에서 사용되는 위협모델링(Threat modeling)을 소개하고 이를 기반으로 공공기관 위협모델링을 제안하고자 한다.

Ⅱ.마이크로소프트 위협모델링[4]

마이크로소프트 위협모델링의 단계는 다음과 같이 구성된다.

◦ 보안목표설정
◦ 어플리케이션 전반적인 구성
◦ 어플리케이션 상세구성
◦ 위협확인
◦ 취약점확인

보안목표를 먼저 설정하고 4단계(어플리케이션 전반적인구성, 상세구성, 위협확인, 취약점확인)를 수행하며 이러한 4단계는 보안목표를 달성할 때 까지 반복한다.

2.1 보안목표설정(Identify Security Objectives)

◦기밀성, 가용성, 무결성에 기반하여 어플리케이션 보안목표를 설정한다.

2.2 어플리케이션 전반적인 구성(Application Overview)

◦종단간 배포시나리오(Draw the End-to-End Deployment Scenario) : 물리적 구성, 논리적 구성, 주요특징, 주요서비스, 통신방식, 외부의존도 등을 고려하여 전반적인 구성도를 그린다.

◦역할 정의(Identify role) : 사용자의 역할에 따라 권한을 설계한다.

◦주요 시나리오 확인(Identify key Usage Scenarios) : 어플리케이션 주요 동작들을 기술하고 이를 데이터 (Create,Read,Update,Delete)관점에서 분석한다.

◦기술명세정보(Identify Technologies) : 운영체제, 웹서버, 데이터베이스 등 사용되는 기술명세 정보를 확인한다.

◦보안메커니즘(Identify application security mechanisms): 입력값 유효성, 인증, 허가, 민감데이터, 암호화등 보안기술에 해당한다.

2.3 어플리케이션 상세구성(Decompose Your Application)

◦ 신뢰범위설정(Identify Trust boundary) : 신뢰범위는 접근할 수 있는 권한이 변경되는 부분을 말한다. 예를 들면 외부인터넷과 DMZ구간 경계, DMZ와 내부망 경계이다[5].

◦ 데이터흐름조사(Identify data flows) : 데이터의 입력부터 출력까지의 흐름을 조사.한다.

◦ 엔트리포인트조사 (Identify entry points) : 사용자의 입력 값을 받는 항목 조사(관리자 로그인페이지, 웹페이지 포트등)한다.

◦ 출구포인트조사(Identify exit point) : 데이터가 전송되는 연관 시스템이나 데이터가 사용자들에게 제공되는 웹페이지들에 대한 조사한다.

2.4. 위협확인(Identify Threats)

◦ 보편적인 위협에 대해 리스트로 정의한 Cheat Sheet: Web Application Security Frame[6] 를 활용하여 위협을 확인한다.

◦ 사용자들의 행동(USE CASE) 분석을 통한 위협을 확인한다.

◦ 데이터플로우를 통해 위협을 확인한다.

2.5. 취약점확인(Identify Vulnerabilities)

◦확인된 위협을 바탕으로 취약점을 찾고 조치한다.

Ⅲ. EMC 위협 모델링(A Developer-Driven Threat Modeling Process)[7]

EMC에서 소프트웨어개발 설계단계에서 주로 사용되는 위협 모델링 기법이다. 특이한 점은 개발자가 소프트웨어 개발과정에서 발생할 수 있는 위협을 위협라이브러리를 통해서 스스로 판단하고 조치한다는 점이다. EMC 위협모델링은 다음단계로 구성된다.

◦ 데이터 흐름 다이어그램 설계(Dataflow Diagrams and Annotations)

◦ 위협 확인(Identifying Threats)

◦ 리스크 평가(Assessing Risk)

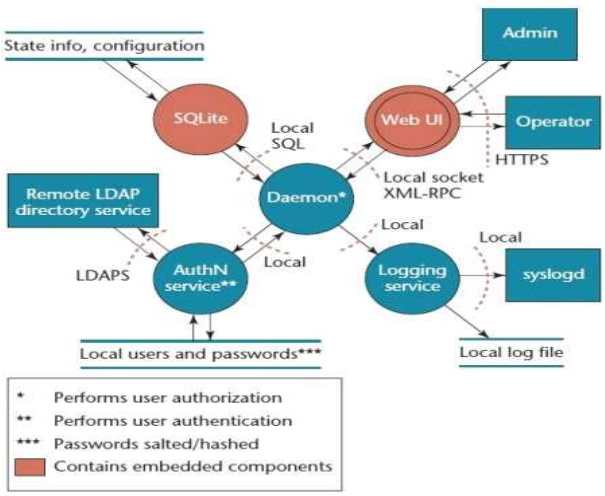
◦ 리스크에 대한 조치(Mitigating Risk)

3.1 데이터 흐름 다이어그램 설계 (Dataflow Diagrams and Annotations)

[표 1] 도형 목록

도형	설명
원	프로세스(시스템)
더블원	복잡한 프로세스
두 줄	데이터 저장소
화살표	데이터흐름
사각형	사용자, 연결된 시스템

[표 1] 도형목록의 도형을 참조하여 정보시스템의 전체적인 구성을 표현하고 보안관련 정보를 추가하여 [그림 1]과 같이 다이어그램을 설계한다.



[그림 1] 데이터 흐름 다이어그램

3.2 위협 확인(Identifying Threat)

개발자가 다이어그램에 표기된 모든 구성을 점검하려면 엄청난 시간과 노력이 필요하므로 현실적인 방법으로 서로 연관된 개체들(interaction)에 대해서만 점검한다. 위협라이브러리에 위협에 대한 구체적인 명세 및 그에 따른 시스템리스크, 조치가이드까지 상세히 설명되어 있기 때문에 개발자는 이를 참조하여 개발단계에서 빠르게 위협에 대해 판단할 수 있다.

3.3 리스크 평가(Assessing Risk)

개발자는 10개의 Yes/no 소프트웨어 보안에 관련된 질의를 받고 이에 대해 기술적 리스크만 평가한다. 질의에 대한 답변에 따라 위협에 대한 항목별 점수가 조정되고 이를 합산하여 소프트웨어 리스크에 대해 정량적으로 평가한다.

3,4 리스크 조치(Mitigating Risk)

개발자는 위협라이브러리를 참고하여 발생된 위협에 대해 보안조치한다.

Ⅳ. 위협모델링 비교

[표 2] 위협모델링 비교

구분	마이크로소프트	EMC
범위	소프트웨어 아키텍처	상호 연관된 요소
방식	시스템 질의	위협라이브러리 참조
전문성	높음	낮음
주체	개발관련자 전체	개발자
비용	고비용	저비용

마이크로소프트사 위협모델링의 경우 높은 수준의 정보보안 활동을 보여주지만 수행함에 따른 비용이 많이 들고 개발관련자들의 높은 기술적 소양을 요구한다. 하지만 EMC의 위협모델링은 개발자가 시스템의 일부요소만을 위협라이브러리를 참고하여 판단하기 때문에 적용이 쉽고 비용이 적게 든다. 하지만 전문성이 요구되는 보안활동을 기대하기 어렵다.

Ⅴ 공공기관 위협모델링

5.1 위협 모델링 단계

마이크로 소프트와 EMC 위협모델링의 특징을 비교하고 공공기관의 정보보안환경을 고려하여 다음과 같은 위협모델링을 고안하였다.

◦ 어플리케이션 아키텍처 구성
◦ 어플리케이션 구체화
◦ 위협 및 취약점 확인
◦ 취약점 조치

5.2 어플리케이션 아키텍처 구성

어플리케이션의 주요 특징 및 구조, 운영환경에 대해 전반적으로 기술한다. 기술 범위로는 하드웨어구성, 시스템의 논리적 구조, 통신 방식, OS, 외부의존도 등에 대한 사항이 이에 해당한다.

5.3 어플리케이션 구체화

정보보안 관련 정보를 구체적으로 기술한다. 데이터 흐름 관련 어떠한 시나리오를 통해 데이터가 저장되고 이에 따른 입출력은 어떠한지 조사한다. 이 과정에서 사용되는 엔티티나 관련된 요소들에 대해 조사한다. 방화벽, DMZ망 등 정보를 통해서 어플리케이션의 신뢰범위를 설정하고 이에 따른 유저의 접속권한을 설정한다.

5.4 위협확인 및 취약점 조치

위협 확인 및 취약점 조치를 하는 가장 기본적인 방법은 체크리스트를 이용하여 적용하는 것이다. 공공기관이 참고할 만한 가이드라인으로 KISA에서 제공하는 전자정부 SW개발◦운영자를 위한 소프트웨어 개발보안가이드가 있다. 소프트웨어 개발 전 과정에서 준수되야 할 보안사항 뿐 아니라 각종 위협상황과 그에 대한 취약점 및 대처방안 등이 아주 상세히 기록되어있다. 하드웨어 및 기반 소프트웨어(OS, DBMS등)에 위협에 대비한 가이드라인으로 KISA에서 제공한 '주요정보통신기반시설 기술적취약점분석●평가방법상세가이드'가 있다. 주요 정보통신 하드웨어 인프라 및 기반소프트웨어(OS, DBMS 등)에 대한 위협상황, 보안취약점 및 조치사항에 대해 상세하게 기록되어 있다.

하지만 단순히 보안 가이드를 충족시켰다고 해서 어플리케이션에 대한 보안성강화를 만족시켰다고 보기는 어렵다. 왜냐하면 공공기관에서 사용되는 보안 가이드라인의 경우 실시간으

로 업데이트가 되지 않아 최신 사이버 공격상황에 대한 대처방법을 제공하지 못하기 때문이다. 또한 전체 공공기관의 공통적인 보안 요소를 기준으로 작성된 것이기 때문에 개별 어플리케이션의 특성을 고려한 보안관리가 어렵다는 점에서 한계가 있다. 그러므로 보안 체크리스트 적용 활동은 최소한의 보안활동이라고 여기고 추가적인 보안활동이 필요하다.

가장 좋은 방법으로 어플리케이션에 대한 구조와 특징을 고려하여 해커의 입장에서의 가상 공격 시나리오를 가정하고 이를 적용해보는 것이다[8]. 이 때 발생할 수 있는 결과를 바탕으로 취약점을 찾고 이를 조치할 수 있다. 하지만 보안전문회사가 아닌 공공기관에서 이 방식을 어플리케이션 전체 구성요소에 적용하기에는 전문성 및 비용적 측면을 고려해 보았을 때 현실적으로 불가능하다. 왜냐하면 대부분의 경우 체크리스트 적용하는 것도 쉽지 않기 때문이다. 따라서 현실성을 고려하여 EMC 위협모델링에서 언급된 개념인 상호동작 하는 요소를 중심으로 대상범위를 좁혀야 한다. 주고받는 데이터의 구간을 기준으로 가상 시나리오를 작성하고 결과를 분석한다. 예를 들어 VPN 경우 사용자가 서비스 접속을 위하여 해외에서 국내에 있는 회사망으로 접속할 때 사용되는데 접근하는 엑세스포인트와 사용자 컴퓨터의 데이터 흐름 구간에 대한 어떤 식으로 해커가 정보를 탈취할 수 있을까? 하는 시나리오를 만들어 결과를 통해 취약점을 찾을 수 있다.

보안취약점이 발생되었을 때 취약점을 제거하는 방식이 가장 이상적이지만 현실을 고려해보면 그렇지 못하다. 예를 들어 취약점 제거에 따른 비용이 막대하게 발생한다던지 타 시스템과에 연계등의 기술적인 문제로 인하여 취약점 제거가 어려울 수 있다. 따라서 어플리케이션의 운영환경, 제거비용 등을 다각적으로 고려하여 취약점조치에 대해 방법을 생각해야 한다.

Ⅶ 결 론

마이크로소프트와 EMC가 사용하는 두 가지의 위협모델링을 설명하였고 이를 근거로 공공기관에서 적용 가능한 공공기관 위협모델링을 만들어보았다. 공공기관 위협모델링은 어플리케이션과 운영환경에 대한 모든 정보를 기반으로 체크리스트 적용과 같은 보안성강화 활동 뿐 아니라 가상의 보안시나리오를 적용하여 그 결과에 따른 취약점조치를 통해 어플리케이션의 전반적인 보안성을 크게 향상 시킬 수 있을 것이다. 특히 정보보안 역량이 부족한 공공기관에서 이를 도입하여 적용한다면 적어도 정보보안 활동에 대한 체계는 최소한의 체계적인 활동을 할 수 있을 것이라고 본다.

추후 연구에서는 공공기관 위협모델링을 실사용중인 시스템에 적용하여 결과에 대한 피드백을 근거로 이론을 상세화 하도록 하겠다.

[참고문헌]

[1]wikipedia “Threat model” https://en.wikipedia.org/wiki/threat_model /(2007.10.17.)

[2] B. W. J.D. Meier, Alex Mackman, “Threat Modeling Web Applications,” Microsoft Corporation, https://msdn.microsoft.com/ko-kr/ms978516, 2005

[3] Gregory Tassey, “The Economic Impacts of Inadequate Infrastructure for Software Testingle,” NIST page 36,2002

[4] B. W. J.D. Meier, Alex Mackman, “How To: Create a Threat Model for a Web Application at Design Time,” Microsoft Corporation, https://msdn.microsoft.com/ko-kr/ms978527, 2005

[5] M. Paul, "Official (ISC)2 Guide to the CSSLP CBK second" CRC Press Taylor & Francis Group, 6000 Broken Sound Parkway NW, Suite 300 Boca Raton, 2014

[6] B. W. J.D. Meier, Alex Mackman, “Cheat Sheet: Web Application Security Frame,” Microsoft Corporation, https://msdn.microsoft.com/ko-kr/ms978518, 2005

[7] D. DHILLON, “Developer-Driven Threat Modeling,” IEEE Journals Mag., vol. 9, no. 4, pages 41 - 47, 2011.

[8] B. Potter and G. McGraw, “Software security testing,” IEEE Security & Privacy , vol. 2, issue 5, page 81- 85, 2004.

랜섬웨어 유포방식과 유형분석*

조형택*, 이동근*, 사경진*, 김광훈*, 염흥열**

*순천향대학교 융합서비스보안학과, **순천향대학교 정보보호학과

Ransomware Distribution Method and Type Analysis

Hyoung-Taek Cho*, Dong-Geun Lee*, Kyeong-Jin Sa*, Kwang-Hun Kim*, Heung Youl Youm**

*Department of Convergence Service Security, **Department of Information Security Engineering, Soonchunhyang University

요 약

랜섬웨어 악성코드는 초기 미국에서 활동하는 악성코드였으나 전 세계적으로 감염자가 급증하여 큰 이슈가 되었다. 랜섬웨어는 주로 사회공학적 기법을 사용하는 이메일, 웹을 통한 Drive-By-Download 방식으로 유포 되었으나 기술의 발전과 공격방식이 증가함에 따라 유포방식 또한 늘어나고 있다. 따라서 본 논문에서는 랜섬웨어의 대표적인 유포방식과 유형을 분석하고 대응방안을 제시한 후 현 기술의 한계점과 앞으로의 발전방향을 제시한다.

I. 서론

요즘 기하급수적으로 컴퓨터 이용자가 증가하고 있고, 다양한 스마트 기기의 발달로 그에 비례하여 전 세계적으로 악성코드의 감염 수가 증가하고 있다. 악성코드들 중 하나인 랜섬웨어는 초기 미국에서 활동하는 악성코드였으나, 최근에는 전 세계적으로 감염자가 급증하고 있다. 초기 랜섬웨어는 E-mail을 이용하는 전형적인 기법이 활용되었지만, 현재는 다양한 기법들로 유포되고 있다.[1]

본 논문의 구성은 다음과 같다. 2장에서 유포방식의 종류를 알아보고 3장 동작방식에서 암호화, 감염 사실 알림, 복구방해, 화면 잠금에 대해 서술한 뒤 4장 대응방안을 제시한 후 5장 결론에서 현재기술의 한계점과 발전 방향을 제시하고 끝을 맺는다.

* 본 연구는 미래창조과학부 및 한국인터넷진흥원의 “고용계약형 정보보호 석사과정 지원사업”의 연구결과로 수행되었음

II. 유포방식

2.1 이메일을 이용한 유포방식

E-mail을 이용한 스팸 메일은 전통적인 악성코드 유포 기법으로, 사용자를 속여 메일에 첨부된 악성코드 파일을 실행하도록 하는 사회공학적 공격 기법을 주로 사용한다. 국내의 경우에는 경품, 배송이나 이벤트 관련 메일로 주로 위장을 하고, 국외의 경우에는 이력서, 금융 안내 그리고 송장으로 위장한다.[2].

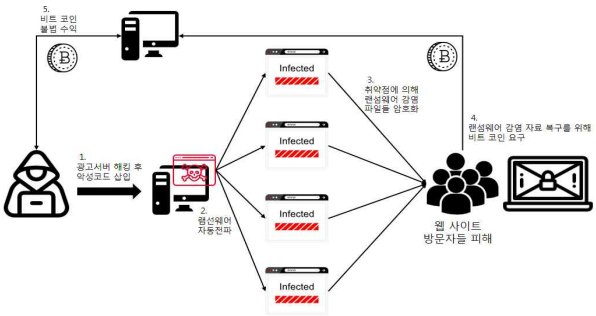
이 기법을 사용하는 대표적인 랜섬웨어로는 록키, 케르베르, 테슬라크립트 등이 있다.

2.2 Drive-By-Download

보안이 취약한 웹사이트를 통해 유포하는 기법으로 웹사이트를 방문만 하여도 감염되기 때문에 불특정 다수의 감염을 이끌어 낼 수 있다.

주요 기법으로는 익스플로잇 킷(Exploit Kit)을 이용하여 취약점이 있는 웹사이트를 찾는 방법과 웹사이트에 등록된 광고를 통해 유포하는 멀버타이징(Malvertising)기법이 있다.[3]

[그림 1]은 공격자가 서버를 해킹하여 유포하는 과정을 나타내고 있다. 대표적인 랜섬웨어는 Cerber, Locky, CryptXXX 등이 존재한다.



[그림 1] 랜섬웨어 유포하는 광고 서버[3]

2.3 서버 APT 공격

패치가 안 된 취약점을 갖고 있는 서버 혹은 보안 관리체계가 취약한 서버를 집중 표적으로 삼아 악성코드를 유포하는 기법이다.[4]

이 기법은 기존의 불특정 다수의 감염을 목적으로 한 무작위 공격 형태에서 APT 공격을 이용한 랜섬웨어 유포로 진화한 점에서 주목받았다. 서버 공격의 경우 서버내의 호스트가 동시에 감염된다는 점에서 위협적인 공격 방식으로 분류 된다.[4] 대표적인 랜섬웨어로는 SAMSAM, Erebus등이 존재한다.

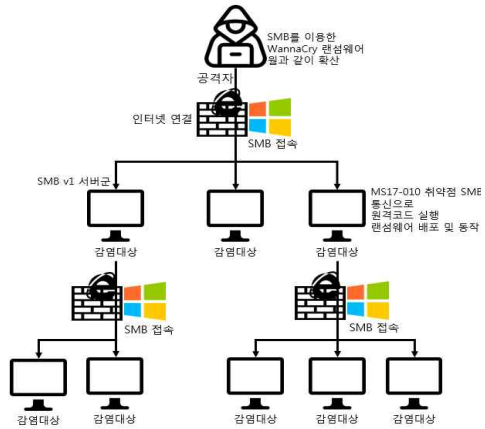
2.4 네트워크형 웜

워너크라이 랜섬웨어가 최초로 이용한 유포 방식으로 윈도우 SMB(Server Message Block) 취약점(CVE-2017-0143)을 통해 악성코드를 유포하는 기법이다.[5]

이 기법은 로컬 IP대역 및 랜덤 IP를 대상으로 SMB 취약점이 존재하는 운영체제 시스템을 공격하여 감염을 시킨다. 기존 유포방식과는 달리 윈도우 보안 업데이트(MS17-010)가 적용 되지 않은 시스템은 별도의 사용자 동작 없이도 인터넷이 연결 되어 있다면 감염 될 수 있다는 특징을 지니고 있다.[6]

2.5 원격 데스크톱 프로토콜(RDP)

사용자나 관리자들이 '1234', 'password'와 같이 쉽게 유추할 수 있는 암호를 설정하게 되면, 사전파일을 이용한 공격에 취약해져 공격자가 쉽게 시스템에 접근한다.[2] 대표적인 랜섬웨어로는 UmbreCrypt,, Kangaroo등이 있다.



[그림 2] 웜 형식의 감염 방식

2.6 업데이트 변조방식

대개 사용자가 어떤 프로그램을 설치할 때 제휴프로그램, 스폰서프로그램 등으로 함께 설치되는 PUP 업데이트 서버를 해킹하여 랜섬웨어를 유포하는 방식이다. 공격자가 해당 PUP의 업데이트 서버를 해킹해 정상 프로그램이 아닌 랜섬웨어가 다운로드 되도록 조작하여 랜섬웨어에 감염되도록 한다. 대표적인 랜섬웨어로는 Allcry 랜섬웨어가 있다.[7]

III. 랜섬웨어 동작방식

3.1 랜섬웨어 암호화 방식

3.1.1 파일내용 암호화 및 확장자 변조

가장 많이 사용되는 방식으로 사용자의 파일을 암호화하여, 파일 열람이나 설치 된 프로그램을 사용하지 못하게 하거나 문서는 열리지만 내용을 읽지 못하도록 암호화 한다.[8]

암호화되는 대상은 주로 문서나 바탕화면의 모든 파일, 시스템파일을 제외한 모든 파일 등 랜섬웨어마다 다르다.[8]

아래의 그림은 감염 가능한 모든 확장자를 나타내고 있다.

.7z	.bkp	.sb	.vpk	.DayZProfile	.pak	.crt	.crw	.wpd	.odc
.rar	.qic	.fos	.tor	.rofl	.big	.cer	.bay	.dxg	.odm
.m4a	.bkf	.mcgame	.psk	.hkx	.unity3d	.der	.sr2	.xf	.odp
.wma	.sidn	.vdf	.rim	.bar	.wotreplay	.x3f	.srf	.dwg	.ods
.avi	.sidd	.ztmp	.w3x	.upk	.xxx	.srw	.arw	.pst	.odt
.wmv	.mddata	.sis	.fsh	.das	.desc	.pef	.3fr	.accdb	
.csv	.itl	.sid	.ntl	.iwi	.py	.ptx	.dng	.mdb	
.d3dbsp	.itdb	.ncf	.arch00	.litemod	.m3u	.r3d	.jpe	.pptm	
.sc2save	.icxs	.menu	.lvl	.asset	.flv	.rw2	.jpg	.pptx	
.sie	.hvpl	.layout	.snx	.forge	.js	.rwl	.cdr	.ppt	
.sum	.hplg	.dmp	.cfr	.ltx	.css	.raw	.indd	.xlk	
.ibank	.hkdb	.blob	.ff	.bsa	.rb	.raf	.ai	.xlsb	
.t13	.mdbackup	.esm	.vpp_pc	.apk	.png	.orf	.eps	.xlsm	
.t12	.syncdb	.001	.lrf	.re4	.jpeg	.nrw	.pdf	.xlsx	
.qdf	.gho	.vtf	.m2	.sav	.txt	.mrwref	.pdd	.xls	
.gdb	.cas	.dazip	.mcmeta	.lbf	.p7c	.mef	.psd	.wps	
.tax	.svg	.fpk	.vfs0	.slm	.p7b	.erf	.dbfv	.docm	
.pkpass	.map	.mlx	.mpqge	.bik	.p12	.kdc	.mdf	.docx	
.bc6	.wmo	.kf	.kdb	.epk	.pfx	.dcr	.wb2	.doc	
.bc7	.itm	.iwd	.db0	.rgss3a	.pem	.cr2	.rtf	.odb	

[그림 3] 감염 대상 확장자

3.1.2 MBR영역 암호화

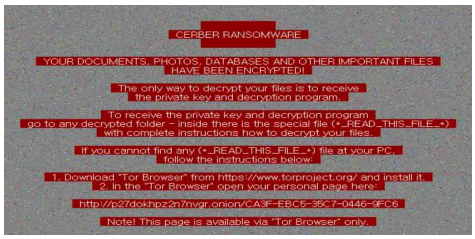
기존 파일만을 암호화시키는 것과는 달리 MBR(Master Boot Record)영역을 암호화한다. MBR영역은 운영체제 부팅에 필요한 정보를 가지고 있는데 이 부분을 암호화시키면 운영체제 부팅자체가 불가능하게 된다.[9]

3.2 감염 사실 알림

공격자는 파일을 암호화한 사실을 감염된 PC의 사용자에게 알리고 복호화를 빌미로 비트코인을 요구하기 위해 경고문을 남기게 된다. 그 방법은 다음과 같다.

3.2.1 바탕화면 변경

사용자가 감염된 사실을 바로 알 수 있도록 바탕화면에 경고문이 나오도록 설정 한다.



[그림 4] 바탕화면 경고문

3.2.2 새 창 띄우기

시작프로그램에 경고문 파일을 등록하여 부팅 될 때 마다 경고문을 띄우며, 경고문을 닫지 못하게 'X'버튼을 비활성화하거나 프로세스를 숨김 형태로 실행한다.



[그림 5] 'X'버튼이 비활성화 된 경고문

3.2.3 소리를 통한 알림

새 창을 띄우거나 바탕화면 변경 이외에도 감염 후 지속적으로 음성으로 감염되었다는 사실과 금전을 지불해야한다고 알려준다.

```
Set SAPI = CreateObject("SAPI.SpVoice")
SAPI.Speak "Attention! Attention! Attention!"
For i = 1 to 10
SAPI.Speak "Your docum"+"ents, photos, databases and other
Next
```

[그림 6] 소리가 나게 하는 VBScript 코드

3.3 복구 방해

3.3.1 볼륨쉐도우

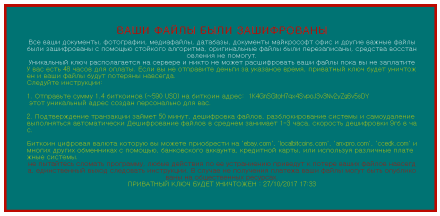
시스템 복원에 사용되는 볼륨 쉐도우(Volume Shadow)를 삭제하기 위해 "vssadmin delete shadow /all" 명령을 수행하여 복원지점을 없애버린다.

3.3.2 덮어쓰기

삭제되는 파일을 복구할 수 없도록 "writefile 함수"를 이용하여 원본파일에 덮어쓰기를 하는데 이렇게 되면 하드디스크의 내용이 새로 쓰여져 원본 데이터 기록이 남지 않게 된다.

3.4 랜섬웨어 화면 잠금

파일 암호화 여부와는 별도로 화면 잠금 비밀번호를 바꾸거나 MBR영역을 수정하여 부팅 잠금을 한다.



[그림 7] 화면 잠금 경고문

3.5 다양한 랜섬웨어별 특징

<표 1>은 본 논문에서 알아본 동작 방식을 이용하는 랜섬웨어를 정리 한 것이다. <표 1>은 변종으로 나온 경우도 포함한 것이다.

IV. 대응방안

가장 흔하게 유포되는 방식인 출처가 불분명한 이메일이나 첨부파일은 클릭하지 말아야 한다. RDP의 비밀번호를 변경하거나, 비활성화를 해야 한다. 응용 및 시스템의 보안 패치를 상시적으로 실시해야 한다. 워너크라이처럼 네트워크형 웜의 경우 SMB도 비활성화를 시키거나 SMB 관련 포트를 차단해야하고, 신뢰할 수 없는 사이트의 접근을 주의해야한다.

또, 본 논문에서 정리한 랜섬웨어 동작방식의 특징을 이용해 랜섬웨어를 실시간 탐지를 하거나 사전에 차단할 수 있을 것이다.

랜섬웨어를 막는 절대적인 보안대책은 없기에 사용자는 중요한 자료를 백업하고, 보안 패치와 꾸준한 관심을 가지는 게 중요하다.

<표 1> 랜섬웨어별 동작 방식

	암호화		알림				복구 방해		화면
	파일	MBR	바탕화면	새 창	소리	Dos	볼륨쉐도우	덮어쓰기	잠금
Lcoky[2]	O	X	O	O	X	X	O	O	X
7EV3N[10]	O	X	X	X	X	X	O	O	O
Petya[2]	X	O	X	X	X	O	O	O	X
Mischa[10]	O	O	O	X	X	O	O	O	X
GoldenEye[2]	O	O	X	X	X	O	O	O	X
Cerber[2]	O	X	O	O	O	X	O	O	X
Wannacry[10]	O	X	O	O	X	X	O	O	X
SamSam[4]	O	X	X	O	X	X	O	O	X
Erebus[10]	O	X	X	O	X	X	O	O	X
CrytoLocker[10]	O	X	O	O	X	X	O	O	X
Hermes[10]	O (확장자명변경x)	X	X	O	X	X	O	O	X
Spora[10]	O	X	O	X	X	X	O	O	X

V. 결론

본 논문에서는 랜섬웨어의 유포방식과 동작 방식에 대해 알아보고 대응방안을 제시하였다. “100% 보안은 존재하지 않는다.” 라는 명제가 존재하듯 보안 위협이 언제 어떻게 발생할지 쉽게 예상 할 수 없다. 따라서 본 논문에서 제시한 대응방안을 준수해야하며, 향후에는 머신러닝을 이용한 랜섬웨어 변종 탐지에 관한 연구로 사전에 피해를 줄일 수 있으면 한다.

[참고문헌]

[1] AVTEST, Totalmalware, 2017, October, 20, 06:21 from https://www.av-test.org/en/statistics/malware

[2] ASEC 대응팀, “최신 랜섬웨어 동향 분석 보고서”, Ahnlab, 2017.01

[3] 송요한, “랜섬웨어 한국을 노린다”, 2017, October, 21, 13:55 from http://www.comworld.co.kr/news/articlePrint.html?idxno=49195

[4] CAITLIN O'HIGGINS, "The Last Key on The Ring - Server Solutions to Ransomware", 2017, October, 21, 15:12 from https://www.trendmicro.com/vmware/server-solutions-to-ransomware/

[5] CVE, CVE-2017-0143, 2017, October, 22, 12:25 from https://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143

[6] Sean Dillon, Dylan Davis, October, 2017, 22, 15:31 from https://www.rapid7.com/db/modules/exploit/windows/smb/ms17_010_eternalblue

[7] 김경애, “추석연휴 노린 올크라이 랜섬웨어, 국내 피해 속출,” October, 2017, 22, 20:31 from http://www.boannews.com/media/view.asp?idx=57314

[8] Xin Luo, Qinyu Liao, Information Systems Security:Awareness Education as the Key to Ransomware Prevention, pp.195-202, September, 2007

[9] 벌새, MBR 변조를 통해 부팅을 방해하는 PETYA 랜섬웨어 주의 (2016.3.28.) 2017, October, 22, 12:25 from http://hummingbird.tistory.com/6313

[10] PCrisk, Virus and Spyware Removal Guides, uninstall instructions, 2017, October, 24, 16:12 from https://www.pcrisk.com/removal-guides/(9733, 10020, 10781, 10809, 10824, 10952, 11258)

SPA(Single Page Application) 기반 웹 보안 취약점 연구

이복음

고려대학교 컴퓨터정보통신대학원 소프트웨어보안학과

SPA (Single Page Application) based web security vulnerability research

Bok-Eum Lee

Korea University Graduate School of Computer & Information Technology

요 약

웹 2.0 시대 이후 웹 어플리케이션 환경에는 많은 변화가 있었다. 격변하는 웹 환경 속에 등장한 것이 SPA(Single Page Application)이다. 모바일을 포함한 멀티브라우저 환경에서 웹 성능 이슈는 꾸준히 거론되었기 때문에 SPA는 트래픽을 감소시키고 사용자에게 더 나은 경험을 제공하여 각광을 받게 되었다. 하지만 그만큼 보안 상 고려해야할 사항이 더 많다. SPA의 특성은 자바스크립트를 이용해서 동적으로 페이지를 변경하면서 동작한다. 자바스크립트 코드는 브라우저에서 실행되기 때문에 사용자 또는 해커가 자바스크립트 코드를 볼 수 있다. 그래서 코드 안에 보안로직이 들어가 있다고 하더라도 로직을 탈취당할 수 있다. 게다가 자바스크립트는 실행 중에 브라우저의 디버깅 기능을 이용해 변수의 값을 모니터링 하거나 변경하여 비즈니스 로직을 위조하는 행위가 가능하다. 그래서 SPA는 클라이언트 보안, 서버 API의 보안, 인증과 세션관리 등 간과할 경우 큰 문제가 될 수 있는 사항이 많다. 본 논문에서 SPA의 보안 취약점 및 해결방안을 제시한다.

I. SPA의 등장 및 개념

1.1 SPA - 단일 페이지 어플리케이션

SPA는 Single Page Application의 약자로 단일 페이지 어플리케이션이라는 뜻이다. SPA는 현재 웹 개발의 트랜드이며 차세대 패러다임으로 각광받고 있다. 전통적인 웹의 동작방식은 요청시마다 서버로부터 렌더링 된 웹 페이지를 브라우저에서 화면에 보여준다. 그러나 전통적인 서버 렌더링 방식은 속도적인 측면에서 풍부한 웹 환경을 표현하기에 역부족이었고 이를 해소하기 위해 캐싱과 압축을 사용하기도 하였다.

SPA는 최초 한번 웹 어플리케이션에 필요한 정적파일들을 로드한 후 각 페이지를 호출 할 때 페이지 전체를 서버에 요청하는 것이 아니라 데이터만 변경해서 사용할 수 있는 웹 어플리케이션을 말한다. SPA 개발을 위한 프레임워크 및 자바스크립트 라이브러리들이 등장하였는데 대표적으로 구글의 AngularJs, 페이스북의 ReactJS를 비롯하여 BackboneJS, Sencha 등이 있다.

1.2 서버 vs 클라이언트 사이드 렌더링[1]

웹 페이지를 화면에 그려주는 것을 렌더링이라고 한다. 전통적인 방식으로는 서버사이드 렌더링이 있는데 서버에서 HTML화면을 생성하여 클라이언트에 보내는 방식이다. JSP, PHP등이 이에 해당한다.

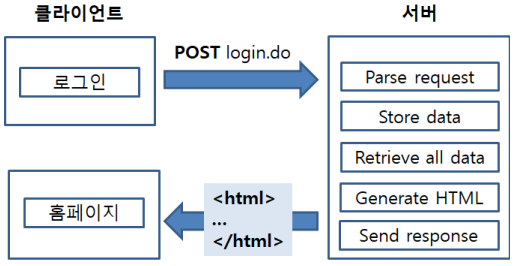


그림 1) 서버사이드 렌더링

SPA의 핵심인 클라이언트사이드 렌더링 방식은 백엔드(BackEnd)와 프론드엔드(FrontEnd)가 완전히 분리된 개념이다. 서버는 요청된 데이터만 제공하고 화면은 클라이언트에서 담당한다.

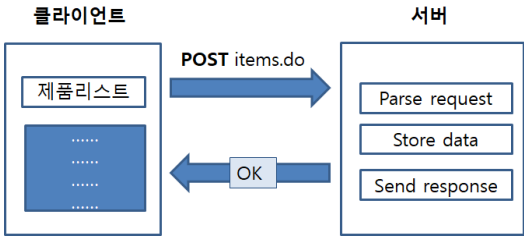


그림 2) 클라이언트 사이드 렌더링

II. 클라이언트 사이드에서 보안 상 고려해야 할 사항

2.1 정적파일 보안 관리[2]

사용자의 브라우저에서 SPA가 실행되기 위해서는 서버에서 HTML, JS, CSS와 같은 정적 파일들을 제공해야 한다. 그러기 위해서는 공개적으로 액세스 할 수 있는 디렉토리에 해당 파일들을 배치해야 한다. 그러나 SPA 실행 파일들은 보안 측면에서 취약할 수 있다. 그렇기 때문에 정적파일의 보안을 위해서는 다음과 같은 사항을 고려해야 한다.

첫 번째로 난독화이다. 난독화는 자바스크립트 코드 자체를 분석하기 어렵게 만드는 과정이다. UglifyJs와 같은 도구를 사용하여 Javascript 코드를 쉽게 난독화 시킬 수 있다. Javascript를 변형하고 리버스 엔지니어링을 어렵게 만든다.

두 번째로 정적파일에 접근(access)할 때 인증(authentication)을 요구하는 것이다. 오직 인증된 사용자만이 중요 로직이 들어있는 자바스크립트 파일을 내려 받을 수 있도록 하여야 한다.

2.2 비즈니스로직은 서버에 구현

노출되지 않아야 할 중요 비즈니스 로직을 자바스크립트 안에 구현하는 것은 보안상 매우 위험하다. SPA의 특성상 서버는 요청된 데이터만 제공하고 화면과 기능은 주로 클라이언트 사이드가 담당한다. 그러나 중요 비즈니스 로직까지 자바스크립트에 구현하면 사용자가 자바스크립트 데이터 위변조를 할 경우, 허용되지 않은 정보가 노출될 수 있다.

2.3 클라이언트의 Session Management

서버에서 세션은 생성되지만 브라우저에서 세션정보가 보관된다. 세션객체가 클라이언트로 전송되면 반드시 보호되어야 한다. 세션을 보호하기 위해서는 첫 번째 클라이언트 사이드에서 세션 데이터 보관 시 최소한의 데이터만 저장하는 방법이 있다.
두 번째 암호화된 토큰을 이용하여 세션객체를 암호화 한다. 최근에 많이 사용되는 암호화 토큰 방법 중 하나가 JWT(JSON Web Token)이다. JWT는 JSON data를 base64 로 암호화하여 세션객체를 보호한다. [3]

2.4 DataStorage 관리

웹 스토리지(Web Storage)란 클라이언트에 소량의 데이터를 저장해 두기 위한 저장 공간이다. 그런데 웹 스토리지를 잘 못 사용할 경우 보안에 위협적이다. 웹 스토리지는 클라이언트 측에서 Script를 이용하여 접근 할 수 있다. 그렇기 때문에 Script의 스토리지 접근에 대해서 검증 작업이 있어야 한다. 예를 들어 공격자는 XSS 공격 등을 이용하여 웹 스토리지에 접근하여 저장된 쿠키, 세션 등의 정보를 탈취 할 수 있다.

뿐만 아니라 호스팅을 공유하는 사이트의 경우, 하나의 Local Storage에 공유된 모든 사이트의 정보가 저장 된다. 공유된 사이트 중 하나의 사이트가 공격 도구로 사용될 경우, 공유된 모든 사이트의 Lcoal Storage 내 정보가 유출될 수 있다. [4]

그렇기 때문에 클라이언트에서 스토리지로 접근할 때 검증 작업은 필수적이다.

2.5 Content Security Policy(CSP)[5]

공격자가 XSS 공격하면 브라우저는 어플리케이션에 속한 스크립트와 악의적으로 주입한 스크립트를 구분하지 못하고 보안취약점을 드러낼 수 있다. 그래서 클라이언트는 서버에서 제공하는 모든 것을 맹목적으로 신뢰하는 대신, CSP를 사용한다. CSP는 신뢰할 수 있는 콘텐츠 소스의 허용 목록을 생성할 수 있게 해주는 Content-Security-Policy HTTP 헤더를 정의하는 것이다. 브라우저에는 정의된 소스로부터 받은 리소스만 실행하거나 렌더링 할 것을 지시한다. CSP는 인라인 script가 실행되는 것을 예방한다. 공격자가 스크립트를 주입하더라도 그 스크립트가 허용 목록과 일치하지 않는다면 실행되지 않는다.

III. 서버사이드에서 보안 상 고려해야 할 사항

3.1 서버의 Session Management

서버는 세션 전체를 통제할 수 있어야 한다. 세션 트래킹이 가능하고 만료된 세션에 대한 컨트롤이 가능해야 한다. 세션상태 정보에 만료일을 표시하고 수신할 때마다 체크함으로써 부실한 세션의 지속적인 사용을 막을 수 있다. 또한 클라이언트사이드에서 세션이 조작될 수 없도록 보호해야 하는데 서버가 세션 데이터를 서명(인증)하고 세션데이터를 수신할 때마다 서명을 확인해야 한다. 또한 클라이언트로 세션 데이터를 보낼 때 세션데이터를 암호화 한다.

3.2 API 인증 (Authentication) [6]

서버는 REST API를 호출한 사람(클라이언트)가 적절한 사용자인지 판단을 해야 한다. 누구나 API를 호출할 수 없고, 인증을 받은 사람만 API 호출을 허용할 수 있어야 한다. 대표적인 인증 방식으로는 API Key 방식과 OAuth 방식, Bi-diretional Certification 방식 등이 있다. 본 논문에서는 각 인증 방식에 대한 자세한 설명은 생략한다.

3.3 SSL을 이용한 네트워크 레벨 암호화

클라이언트가 API를 호출할 때 보호되어야 할 내용이 있을 수 있다. 이 경우, API 메시지를 암호화해야 한다. 메시지를 암호화하는 방법 중에서 가장 쉽고 범용적인 방법은 HTTPS를 이용하는 것이다. 이 방식은 서버 설정만으로도 가능하고, 가장 널리 쓰이는 방식이기 때문에 신뢰할 만하다.

3.4 본문 메시지 암호화

네트워크 레벨의 암호화를 사용할 수 없거나, 또는 네트워크 레벨의 암호화를 신뢰할 수 없는 상황에는 메시지 자체를 암호화 하는 방법을 사용할 수 있다. 암호화하는 방식은 두 가지가 있는데 전체 메시지를 암호화하는 방법과 특정 필드만 암호화하는 방법이 있다. 메시지를 암호화하여 통신하기 위해서는 서버와 클라이언트 암호화 키를 가지고 서로 인코딩과 디코딩하여 메시지를 주고받는다.

3.4 Pre-flight를 활용한 세부적인 CORS 통제[7]

보안의 목적으로, 브라우저는 스크립트 내에서 초기화되는 cross-origin HTTP 요청을 제한한다. 하지만 SPA의 특성상 CORS (cross-origin resource sharing)를 세부적으로 관리할 필요가 있다. 이 때 활용할 수 있는 것이 Pre-flight 호출 방식이다. 이는 XMLHTTPRequest를 특정 URL로 요청하기 전에 먼저 HTTP Options를 호출한다. 그 후 서버는 해당 URL을 접근할 수 있는 Origin URL과 HTTP Method를 반환한다. 최근에는 HTTP Header를 직접 조작하지 않더라도 Spring과 같은 서버 어플리케이션 개발 프레임워크에서 CORS 구현을 지원하고 있다.

IV. 결론

본 논문에서는 SPA기반 웹의 보안 취약점을

분석하고 그에 대한 해결방안을 제시하였다. SPA는 서버사이드 렌더링 방식이 아닌 클라이언트 사이드 렌더링 방식을 사용하기 때문에 브라우저의 특성과 자바스크립트의 코드 보안에 더 많은 주의를 필요로 한다. 뿐만 아니라 세션관리, API 보안 등 자칫 간과하면 취약점이 될 수 있는 보안 이슈를 고려해서 설계해야 한다. 본 논문은 클라이언트 사이드와 서버사이드에서 고려해야 할 보안 요소가 무엇인지 분석하고 그에 대한 해결 방안을 제시하였다.

[참고문헌]

[1] Securing Single Page Applications. Philippe De Ryck - iMinds-DistriNet, KU Leuven.

[2]http://yottaram.com/blog/2014/09/12/protecting-a-javascript-single-page-app-with-laravel/

[3] https://jwt.io/

[4] 한국인터넷 진흥원. 인터넷 사이트 쿠키의 주요 이슈 및 취약점분석. 2014. 08.

[5]https://developers.google.com/web/fundamentals/security/csp/?hl=ko

[6] REST API의 이해와 설계-#3 API 보안 . http://bcho.tistory.com/955

[7] HTTP 접근 제어 (CORS). MDN Web Docs. mozilla

Wi-Fi 신호 변동 기반의 IoT 디바이스 인증 기법

정혜라†, 소재우*

† 서강대학교 정보통신대학원, *서강대학교 전자공학과

IoT Device Authentication Scheme Based on Wi-Fi Signal Variations

Hyera Jeong†, Jaewoo So*

† Graduate School of Information & Technology, Sogang University

*Department of Electronic Engineering, Sogang University

요 약

본 논문은 Wi-Fi를 사용하는 홈 IoT (Internet of Things) 환경에서 IoT 디바이스를 인증하는 기법을 제안한다. Wi-Fi 액세스 포인트와 클라이언트들이 Wi-Fi 접속 비밀번호를 공유하는 PSK (pre-shared key) 환경에서 스마트폰은 입력 장치가 없는 IoT 디바이스로 Wi-Fi 비밀번호를 전송한다. 이 때, 공격자가 스마트폰과 IoT 디바이스 간 무선 트래픽을 스니핑하거나 IoT 디바이스로 위장하면 Wi-Fi 비밀번호가 공격자에게 노출될 수 있는 문제가 있다. 본 논문은 Wi-Fi 수신 신호 세기 상관관계를 기반으로 스마트폰이 IoT 디바이스의 위치를 추정하고, 이를 기반으로 공격자 여부를 판단한 후 암호화된 채널을 통해 Wi-Fi 비밀번호를 전송하는 새로운 IoT 인증 기법을 제안한다.

I. 서론

IoT 디바이스들이 널리 설치되면서 IoT 디바이스 인증 문제가 중요한 이슈로 대두되고 있다[1]. IoT 디바이스 인증 기법에 대해 많은 연구가 진행되었다. 논문 [2]에서는 게이트웨이와 컨트롤러를 이용하여 비대칭키 기반으로 IoT 디바이스를 인증하는 기법이 제안되었다. 논문 [3]에서는 IoT-OAuth 기반의 인증 기법이 제안되었다. 그러나 논문 [2], [3]은 홈 Wi-Fi 환경에서 별도의 입력장치가 없는 IoT 디바이스를 인증하는데 어려운 점이 있다. 홈 Wi-Fi 환경에서 액세스 포인트 (access point, AP)는 무선 클라이언트들과 Wi-Fi 접속 비밀번호를 공유한다. 홈 Wi-Fi 환경에서 IoT 디바이스를 설치할 때, 많은 업체들은 스마트폰을 이용하여 Wi-Fi 비밀번호를 IoT 디바이스에 전송한다. 따라서 공격자는 스마트폰과 IoT 디바이스간 트래픽을 스니핑하거나 IoT 디바이스로 위장하여 Wi-Fi 비밀번호를 알아낼 수 있다. 이러한 문제점을 해결하기 위해서 논문 [4]-[6]은 IoT 디바이스를 움직여서 수신 신호 세기 변화를 발생시키고 이를 기반으로 IoT 디바이스 위치를 판단하여 IoT 디바이스를 인증하는 기법을 제안하였다. 본 논문은 종래 기법들과 달리 별도의 인증 서버가 없으면서 IoT 디바이스를 움직일 필요 없이 중간자 (man in the middle attack, MITM) 공격에 강인한 IoT 디바이스 인증 기법을 제안한다.

* 본 논문의 교신저자는 소재우 임 (이메일: jwso@sogang.ac.kr).
이 논문은 2016년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (2016R1D1A1B03934150).

II. 문제 동기 및 시스템 모델

2.1 문제 동기

홈 Wi-Fi 환경에 설치되는 스마트 전원 스위치, 전등 스위치 등과 같은 IoT 디바이스는 별

도의 입력 장치가 없어 홈 Wi-Fi AP에 접속하기 위해 스마트폰으로부터 Wi-Fi 비밀번호를 전송받아야 한다. 일례로 벨킨에서 판매하는 위모(Wemo) 스마트 전원 스위치를 고려한다. 그림 1에서 위모 IoT 디바이스는 전원이 인가되었을 때 Wi-Fi AP로 동작하며, 스마트폰은 IoT 디바이스에 접속한다. 그리고 위모 IoT 디바이스는 디바이스 ID와 MAC 주소를 스마트폰에 전송하고, 스마트폰은 디바이스 ID와 MAC 주소를 사용하여 Wi-Fi 비밀번호를 암호화하여 IoT 디바이스에 전송한다. 그러나 IoT 디바이스 ID와 MAC 주소는 쉽게 노출되는 정보이어서 공격자는 다른 IoT 디바이스의 펌웨어를 해킹하여 암호화 방식을 찾아낼 수 있다. 따라서 공격자는 도청과 위장 공격을 통해 Wi-Fi 비밀번호를 획득할 수 있다.

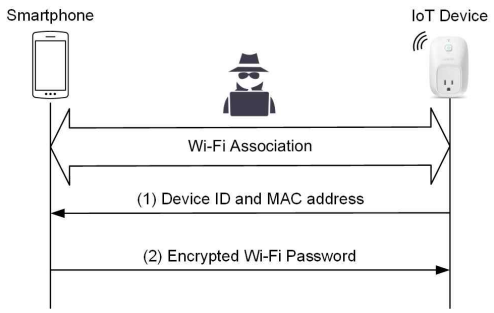


그림 1. Wi-Fi 비밀번호 스니핑

2.2 시스템 모델

본 논문은 Wi-Fi 접속 비밀번호를 공유하는 홈 Wi-Fi 환경에서 별도의 입력장치가 없는 IoT 디바이스를 설치하는 경우를 고려한다. 스마트폰은 Wi-Fi 비밀번호를 IoT 디바이스에 전송하여 IoT 디바이스가 홈 Wi-Fi AP에 접속할 수 있도록 하여야 한다. 공격자는 무선 스니핑과 IoT 디바이스로 위장하여 Wi-Fi 비밀번호를 알아내려고 한다.

III. 제안하는 IoT 디바이스 인증 기법

3.1 제안하는 IoT 디바이스 거리 추정 기법

무선 채널에서 지역 확산, 쉐도우 페이딩 등의 채널 파라미터들은 공간적으로 상관관계가 있다[7], [8]. 무선 채널 상관관계를 기반으로 근접 거리를 추정하는 연구가 진행되었다[9], [10].

Wi-Fi 채널 n에서 AP m으로부터 수신하는 신호의 세기를 $h_{m,n}$이라하고, $h_{m,n}$의 평균과 분산을 각각 $\mu_{m,n}$와 $\sigma_{m,n}$로 표현한다. 그러면 다음과 같이 변환된 신호 세기 $x_{m,n}$는 평균이 0이고 분산이 1인 정규분포 $N(0, 1)$을 따른다[8].

$$x_{m,n}=\frac{h_{m,n}}{10\sigma_{m,n}}-\frac{\mu_{m,n}}{\sigma_{m,n}}. \quad (1)$$

여기서, $h_{m,n}$의 단위는 데시벨이다. AP는 동작 중에 Wi-Fi 무선 채널을 변경하지 않으므로 무선 채널을 나타내는 첨자 n을 생략하도록 한다.

그림 2와 같이 스마트폰 및 IoT 디바이스 주변에 여러 개의 Wi-Fi AP들이 존재할 수 있다. 스마트폰과 IoT 디바이스는 그림 3의 순서도에 따라 N개의 무선랜 채널들에 대해 변환된 수신 신호 세기들의 집합 ($\boldsymbol{X}_{phone}$과 $\boldsymbol{Y}_{device}$)을 각각 구축한다.

$$X_{phone}=\left\{x_1,x_2,\cdots x_{M_1}\right\} \quad (2)$$

$$Y_{device}=\left\{y_1,y_2,\cdots,y_{M_2}\right\}. \quad (3)$$

여기서, M_1은 스마트폰이 관측한 AP들의 개수이고, M_2는 IoT 디바이스가 관측한 AP들의 개수이다.

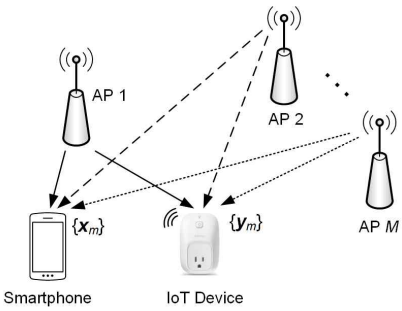


그림 2. Wi-Fi 신호 세기 측정

AP m의 무선 채널에서 스마트폰이 수집한 신호 세기들을 x_m이라 하고, IoT 디바이스가 수집한 신호 세기들을 y_m 이라 하자. 그러면 스마트폰과 IoT 디바이스 간의 거리 d와 (x_m, y_m)의 공분산 사이의 관계는 다음과 같다[7].

$$\rho_m=E(x_m y_m)=e^{\left(-\frac{d}{d_D}\right)}. \quad (4)$$

여기서 d_D는 무선 채널 환경에 따라 결정되는 상수이다[8]. 수식 (4)로부터 스마트폰과 IoT 디바이스 간의 거리는 다음과 같이 추정할 수 있다.

$$\tilde{d} = \frac{1}{M}\sum_{m=1}^{M} \ln \rho_m^{-d_{ref}}. \quad (5)$$

여기서, M은 $\min(M_1, M_2)$ 이다. 수식 (5)에서 계산된 거리가 사전에 정의된 δ 보다 작으면 스마트폰은 IoT 디바이스가 근접 거리에 있다고 판단하고, 그렇지 않으면 IoT 디바이스가 근접 거리에 없다고 판단한다.

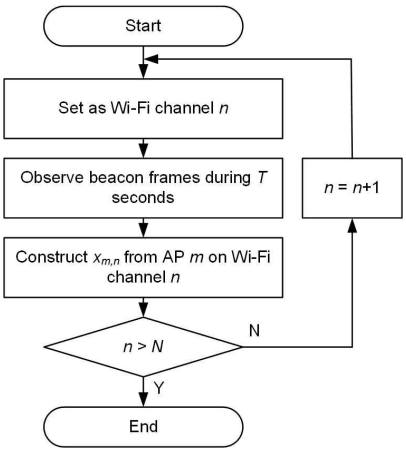


그림 3. Wi-Fi 신호 측정 순서도

3.2 제안하는 IoT 디바이스 인증 프로토콜

제안하는 IoT 디바이스 인증 프로토콜 과정은 MITM 공격을 피하기 위해 SSL (secure sockets layers) 과정과 유사하다[11]. 그림 4는 제안하는 IoT 디바이스 인증 과정을 보여주고 있다.

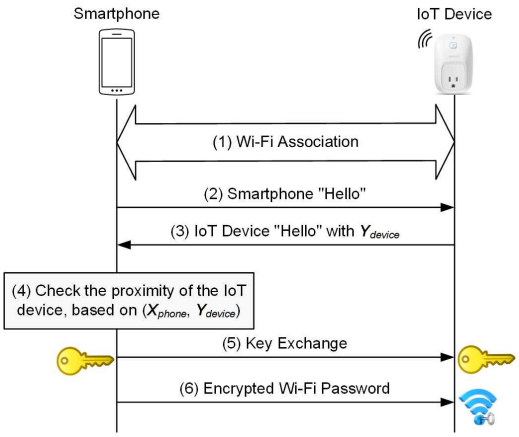


그림 4. 제안하는 IoT 디바이스 인증 과정

- **STEP 1 (Wi-Fi 접속):** IoT 디바이스는 전원이 인가되었을 때 Wi-Fi AP로 동작하면서 그림 4와 같이 주변 Wi-Fi 신호를 측정하여 변환된 수신 신호 세기들의 집합 $\boldsymbol{Y}_{device}$를 구축한다. 스마트폰은 IoT 디바이스 설정 앱을 실행할 때, 그림 4와 같이 주변 Wi-Fi 신호를 측정하여 변환된 수신 신호 세기들의 집합 $\boldsymbol{X}_{phone}$를 구축한다. 그리고 스마트폰은 Wi-Fi AP로 동작하는 IoT 디바이스에 접속한다.
- **STEP 2 (스마트폰 Hello):** 스마트폰은 IoT 디바이스에게 "암호화 방식"을 전송한다.
- **STEP 3 (IoT 디바이스 Hello):** IoT 디바이스는 스마트폰에게 선호하는 암호화방식, 측정한 신호 세기 집합 $\boldsymbol{Y}_{device}$, 랜덤하게 생성한 공인키를 전달한다.
- **STEP 4 (근접 거리 판단):** 스마트폰은 수식 (5)를 사용하여 계산된 추정거리 $\tilde{d}$를 사용하여 IoT 디바이스가 근접 거리에 있는지 판단한다. IoT 디바이스가 근접 거리에 있다고 판단하면 STEP 5 이후 과정을 진행하고, 그렇지 않으면 인증 과정을 종료한다.
- **STEP 5 (키 교환):** 스마트폰은 STEP 3에서 수신한 공인키를 사용하여 암호화된 세션키를 IoT 디바이스에 전달한다. IoT 디바이스는 개인키를 사용하여 수신한 데이터를 복호화하여 세션키를 추출한다.
- **STEP 6 (Wi-Fi 비밀번호 전송):** 스마트폰은 암호화된 세션 채널을 사용하여 Wi-Fi 접속 비밀번호를 IoT 디바이스에 전송한다.

IV. 제안하는 IoT 디바이스 인증 프로토콜 분석 및 토의

제안하는 IoT 디바이스 인증 프로토콜의 보안을 아래와 같은 관점에서 분석하고 토의한다.

- **무선 도청:** 공인키/세션키가 공개되지 않는 이상 암호화된 세션 채널을 사용하여 전송하는 Wi-Fi 접속 비밀번호를 엿들을 수 없다.
- **IoT 디바이스 위장:** 스마트폰과 IoT 디바

이스 간의 통신에서 공격자는 IoT 디바이스로 위장할 수 있다. 만약, 공격자가 IoT 디바이스와 동일한 SSID와 MAC를 사용하고 높은 전송 전력을 사용하는 AP로 동작하면, 스마트폰은 공격자의 AP에 Wi-Fi 접속을 한다. 따라서 종래의 기법에서는 공격자의 위장 공격에 취약하여 스마트폰이 Wi-Fi 비밀번호를 공격자에게 전송할 수 있는 문제가 있다. 그러나 제안하는 기법에서는 근접 거리를 판단하여 IoT 디바이스를 검증하므로 공격자의 위장 공격을 방어할 수 있다.

V. 결론

본 논문은 스마트폰이 IoT 디바이스를 검증하는 IoT 환경에서 구현 가능한 IoT 디바이스 인증 기법을 제안하였다. 제안하는 인증 기법은 스마트폰이 수신하는 Wi-Fi 신호 세기들과 IoT 디바이스가 수신하는 신호 세기들 간의 공분산을 기반으로 근접 거리를 판단한다. 제안하는 인증 기법은 공격자의 도청과 IoT 디바이스 위장 공격을 효율적으로 방어할 수 있다.

[참고문헌]

[1] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, “A survey on security and privacy issues in Internet-of-Things,” *IEEE Internet of Things J.,* vol. 4, no. 5, pp. 1250-1258, Oct. 2017.

[2] O. Salman, et al., “Identity-based authentication scheme for the Internet of Things,” in *Proc. IEEE ISCC,* pp. 27-30, Aug. 2016.

[3] S. Cirani, et al., “IoT-OAS: An OAuth-based authentication service architecure for secure services in IoT scenarios,” *IEEE Sensors J.,* vol. 15, no. 2, pp. 1224-1234, Feb. 2015.

[4] R. Mayrhofer and H. Gellersen, “Shake well before use: Intuitive and secure pairing of mobile devices,” *IEEE Trans. Mobile Computing,* vol. 8, no. 6, pp. 792-806, Jun. 2009.

[5] I. Ahmed, et al., “Checksum gestures: continuous gestures as an out-of-band channel for secure pairing,” in *Proc. UBICOMP,* pp. 391-401, Sep. 2015.

[6] J. Zhang, Z. Wang, Z. Yang, and Q. Zhang, “Proximity based IoT device authentication,” in *Proc. IEEE INFOCOM,* pp. 1-9, 2017.

[7] W. Lee, J. Kim, and S.-W. Choi, “New D2D peer discovery scheme based on spatial correlation of wireless channel,” *IEEE Trans. Veh. Technol.,* vol. 65, no. 12, pp. 10120-10125, Dec. 2016.

[8] A. Algans, K. I. Pedersen, and P. E. Mogensen, “Experimental analysis of the joint statistical properties of azimuth spread, delay spread, and shadow fading,” *IEEE J. Select. Areas Commun.,* vol. 20, no. 3, pp. 523-531, Apr. 2002.

[9] Y. Agata, J. Hong, and T. Ohtsuki, “Room-level proximity detection based on RSS of dual-band Wi-Fi signals,” in *Proc. IEEE ICC,* pp. 1-6, 2016.

[10] H. Shafagh and A. Hithnawi, “Come closer - proximity-based authentication for the Internet of Things,” in *Proc. MobiCom,* pp. 421-424, 2014.

[11] A. Freier, et al., “The secure sockets layer (SSL) protocol version 3.0,” IETF RFC 6101, Aug. 2011.

금융회사의 전사적 보안협업을 위한 핵심성공요인

박지윤

중앙대학교 산업융합보안학과

A Study on Critical Success Factors for Security Collaboration Focused on Financial Companies

Ji-Yun Park

*Division of information security, Chung-ang University.

요 약

금융보안의 패러다임이 타율적 규제에서 자율 규제로, 사전규제에서 사후규제로 변화됨에 따라 금융회사 스스로의 자율보안체계 확립을 위한 역량 강화가 요구되고 있다. 따라서 금융회사 스스로 전사적 정보보호 활동을 해 나가는 과정에서 갈등 요인을 최소화하는 동시에 효과적인 정보보호 거버넌스의 구현을 위한 보안협업의 중요성과 필요성이 더욱 중요해졌다. 하지만 보안협업에 대한 선행연구들은 기술적 초점에 맞추어져 있으며 금융회사를 대상으로 한 연구는 이루어지지 않았다. 본 논문은 기존의 협업 방법론에 기반하여, 금융회사의 속성을 반영한 보안협업에 있어 핵심성공요인이 무엇인지 대하여 연구하고 핵심성공요인을 도출하였다.

I. 서론

정보통신기술의 발전과 모바일 기기의 활용이 가속화되면서 기존 금융권의 환경도 급속하게 변화되고 있다. 모바일 기기를 활용한 이용자수의 증가와 데이터, 클라우드, IoT와 같은 IT신기술과 금융서비스와의 융합은 사용자들에 있어 편의성을 가져다주었으나 동시에 여러 공격 유형을 통해 대규모 금융보안 사고가 지속적으로 발생되고 있음에 따라 정보 유출의 위험도 기존보다 더욱 증가했다.

금융당국은 보안성 심의 제도와 각종 설치형 보안 모듈 의무사용, 액티브X 설치 등 높은 보안수준을 요구하며 사전규제를 실시하였으나 급속도로 발전하는 IT금융 신기술의 반영과 금융시장변화를 따라갈 수 없었고 오히려 국내 핀테크 산업을 가로막는 장벽으로 작용함은 물론 사용자들에 있어서도 불편을 증가시키는 결과를 초래하였다.

이에 금융권에서는 보안성 심의제도 폐지 등 금융보안의 규제 패러다임을 타율적 규제에서 자율규제로, 사전규제에서 사후규제로 전환하였다. 따라서, 금융회사는 변화된 금융시장 환경에 맞추어 유연하고 선제적인 대응 체계 마련이 필요하다.

기존의 타율적 규제방식에 익숙한 금융회사가 소비자들의 금융자산보호수준을 하락시키지 않으면서 체계적이고 효과적인 정보보호 활동을 통해 금융소비자들을 더욱 안전하게 보호하기 위해서는 금융회사의 자율보안 역량 강화가 필요하며, 무엇보다 기업 내의 다양한 이해관계자간의 상호 협조와 역량 강화가 우선적으로 뒷받침되어야 한다.

따라서 본 연구는 금융회사의 목표와 속성을 반영하여 효과적인 전사적 보안협업을 위한 핵심성공요인을 도출하고자 한다.

II. 관련 연구

2.1 협업의 개념

협업(collaboration)은 사회적, 학문적으로 다양한 용어로 대체되거나 혹은 혼재되어 사용되어 왔다[1]. 협업에 대한 정의를 명확하게 하는 것은 어려우며 일반적으로 협업과 유사한 개념인 협력과 협조, 조정 등의 인접 개념들을 통하여 이해가 가능하다. 따라서 협업에 대한 이해를 위하여 이러한 유사의미에 대한 연구와 정의를 알아보고자 한다.

협력(coordination)에 대한 Himmelman(2002)의 개념정의를 살펴보면 공통의 목적 달성을 위한 활동 및 상호이득을 위한 정보교환활동이라 정의했다. Singh(2005)[5]은 공통의 목적과 연관된 업무를 추구하여 행동하는 것이라 말했다.

협조(cooperation)에 대한 Gulati(2012)[6]의 개념정의를 살펴보면 동의된 목표에 관한 공동의 추구라 정의할 수 있다. Laudon(2012)[7]는 협조란 협력과 같은 의미로, 공통의 목표를 달성하기 위해 함께 행동하는 것으로 정의하였다.

협업은 서로 다른 영역에서 공동의 작업을 촉진시키는 프로세스이다[2]. Gulati는 협조와 협동 간의 균형을 찾아가는 활동을 협업이라 정의하였다. 협업은 공유된 목표에서부터 의사결정을 위한 프로세스 까지 조직 내 혹은 조직 간의 관계가 긴밀해질수록 단계가 높아진다고 할 수 있다[3]

2.2 협업에 영향을 미치는 요인

Dean[8]은 조직 내 협업을 촉진시키는 요소들에 대한 양적연구를 바탕으로 갈등관리, 리더십, 문화, 신뢰성의 4가지 주요요인을 제시하였다.

Meng[9]은 제조, 소매, 공급 등의 산업군 내에서의 전문가 인터뷰를 통해 파트너 관계 간에 영향을 주는 요인을 도출하였다. Meng이 제시한 요인들은 총 8가지로 자원공유, 위험할당, 신뢰성, 문화, 상호목적, 공동업무, 문제해결, 의사소통, 성과측정, 지속적 개선을 말한다.

Winer[10]는 공동의 목표달성을 위한 이해관계자간의 관계설정 및 협업과 관련한 역할 결정, 설계, 평가, 성공요인의 연구를 통해 협업의 성공요인을 프로세스, 의사소통, 자원, 목적, 환경, 구성원의 특징으로 정의하였다.

이외에도 이상곤, Horwath, Groenewegen 등의[11,03,12] 연구들을 통한 협업 요인으로는 신뢰성, 의사소통의 정도, 인센티브, 정보공유성, 의존성, 전문인력, 최고경영자의 의지, 동기부여 등의 요인들이 있다.

III. 금융회사의 보안협업을 위한 핵심성공요인 도출

3.1 핵심성공요인에 대한 선행연구

핵심성공요인(Critical Success Factors)이라는 개념은 D. Ronald Daniel[13]에 의해서 1961년에 처음 소개되었으며, 조직의 목표 달성에 영향을 미치는 내적/ 외적 요인으로 광범위하게 사용하고 있다. D. Ronald Daniel은 핵심성공요인을 통하여 공동의 목표를 효과적이고 효율적으로 달성할 수 있으며 최고경영층의 의사결정을 지원하는 정보로써 활용될 수 있다고 주장하였다.

James Dobbins와 Richard Donnelly는 핵심성공요인의 기능으로 이해관계자의 주요 관심사항 식별, 전략 계획 개발, 프로젝트의 생명주기 및 실패 원인 식별, 신뢰성 평가, 비즈니스 기회 및 위협 식별, 인적 자원의 생산성 평가를 제시하였다.

John F. Rockart와 Christine V. Bullen[15]은 핵심성공요인과 조직의 목표 간의 관계를 아래 그림과 같이 계층적 구조를 통하여 제시하였다. 핵심성공요인을 통하여 목표 달성을 위해 필요한 활동들이 식별가능해지며, 모호한 목표들이 최종적으로는 측정가능해지고 성과측정을 통해 목표 달성 여부에 대한 측정도 가능하게 된다. 이러한 계층적 구조는 조직의 내적, 외적 환경에 따라 다양하게 나타날 수 있다.

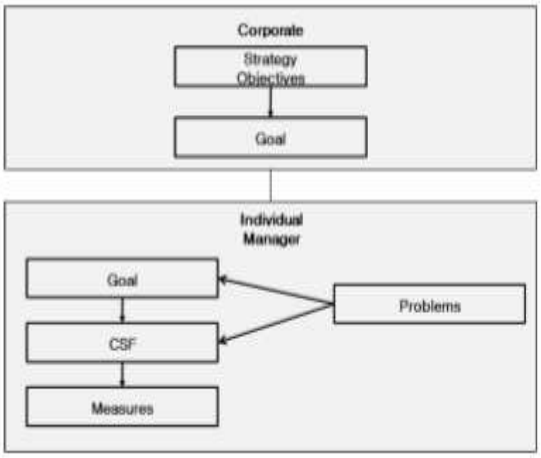


그림 1. 핵심성공요인과 목표의 계층적 관계

3.2 협업 모형

Ansell&Gash[16]는 공공기관과 비정부조직 이해관계자 간의 협업사례 137개를 메타분석을 통해 귀납적으로 결론을 도출하여 아래 그림과 같이 협력적 거버넌스 모형을 제시하였다. 이 모형은 공공기관과 비정부조직 사이뿐만 아니라 다양한 종류의 협업의 경우에도 적용할 수 있어 연구목적에 맞게 변형 가능하다는 장점이 있다[4].

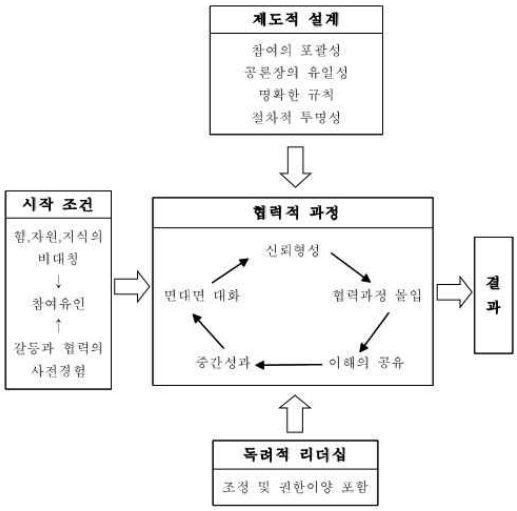


그림 3. Ansell&Gash의 협력적 거버넌스 모형

3.3 금융회사의 보안협업을 위한 핵심성공요인 도출 과정

핵심성공요인의 도출을 위해서는 성공사례 조사를 통하여 요인을 도출하는 것이 일반적이나 아직 국내 금융회사의 경우 관련 사례나 연구가 부족하고 보안협업에 대한 인식이 미흡하여 일반적인 접근이 어렵다. 따라서 델파이기법을 활용하여 연구를 진행하고자 한다. 델파이기법은 현재의 상태에 관한 일반화 된 자료가 미흡한 경우 다수의 전문가의 의견이라는 집단적인 사고를 통해 체계적으로 접근하는 분석방법이다.

총 3회에 걸친 전문가 인터뷰를 통하여 아래와 같은 핵심성공요인 도출 모형과 핵심성공요인들을 수립하였다.

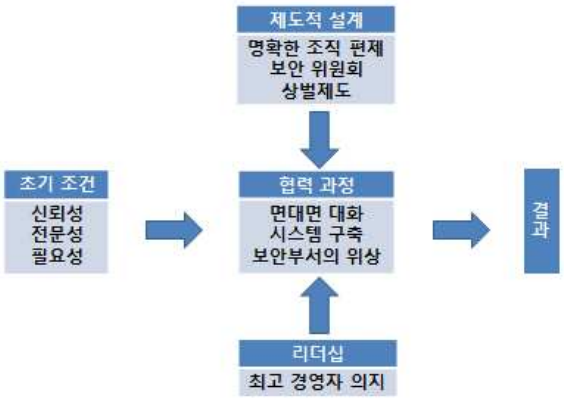


그림 3. 핵심성공요인 도출 모형

보안협업을 위한 초기 조건 성공요인:

1. 조직 내 보안 협업 주체 간의 신뢰 구축
2. 보안 협업 주체의 높은 인력 전문성
3. 보안협업에 대한 필요성 및 인식 제고

보안협업을 위한 제도적 설계 성공요인:

1. 정보보호의 주요한 의사결정에 기반 한 권한과 책임의 명확화
2. 보안위원회의 활동
3. 보안협업에 관한 명확한 상벌제도의 정립

보안협업 과정에서의 성공요인:

1. 이해관계자와의 의사소통 채널의 존재를 통한 원활한 커뮤니케이션

2. 보안협업을 위한 기술적 도구 구축

3. 정보보호 문화 형성을 통한 보안부서의 위상 제고

보안협업을 위한 리더쉽 요인:

정보보호를 비즈니스 이슈로 전환하여 최고경영자의 보안협업에 대한 의지 강화

IV.결론 및 향후 연구 과제

본 논문에서는 협업에 대한 개념과 협업 요인을 바탕으로 금융회사의 보안 전문가 인터뷰를 통하여 금융회사의 보안협업을 위한 핵심성공요인을 도출하였다. 도출된 핵심성공요인은 금융회사에서의 보안협업을 수행하기 위한 방향 제시 및 계획을 수립할 경우에 있어 유용한 자료로 활용될 수 있을 것이다.

한편, 본 논문에서 도출된 보안협업모형은 실질적인 적용사례를 통하여 도출된 것이 아닌 기존 선행연구들을 바탕으로 전문가 인터뷰를 통하여 고려된 것이기 때문에 실제 금융회사 종사자들을 대상으로 설문을 통해 적용 가능성과 중요성을 검증해야 할 필요가 있다.

[참고문헌]

[01] 구본선, 주민자치회를 통한 협력적 거버넌스 정착에 관한 연구 서울대학교 행정학과 석사학위논문, 2016

[02] 도정민, 보안협업을 위한 핵심성공요인, 중앙대학교 석사학위논문, 2014

[03] 진철구, ICT 공급망내 이해관계자간 협업을 위한 핵심성공요인에 관한 연구, 한국경영정보학회, Vol.2016 No.11, 2016

[04] Ansell, C., & Gash, A. , Collaborative Governance in Theory and Practice, Journal of Public Administration Research and Theory , 18(4): 543-571, 2008

[05] B. Singh, Collaboration and competition tensions in multi-unit organizations facing volatile business environments, Case Western Reserve Univ. 2005

[06] Christine V. Bullen, John F. Rockart, A Primer on Critical Success Factors, Center for Information Systems Research, 1981

[07]D. Ronald Daniel, Management Information Cricis, Harvard Business review, 1961

[08] Groenewegen L., Factors influencing collaboration within a partnership of large infra-structural projects in the Netherlands, TUDelft, 2013

[09] K.S Dean, Strategies and Benefits of Fostering Intra-Organizational Collaboration, 2010

[10] Laudon, K. C., & Laudon, J. P. Essentials of management information systems Pearson Upper Saddle River, 2011

[11] M.B. winer and K.L. Ray, Collaboration handbook, Amherst H. Wilder Foundation, 1994

[12] P. Danese, Towards a contingency theory of collaborative planning initiatives in supply networks, Int J Prod Res, 2011

[13] R. Gulati, F. Wohlgezogen and P.Zhelyazkov, The two facts of collaboration: Cooperation and coordination in strategic alliances, The academy of management naals, pp. 531-583, 2012

[14] S G Lee & S B Cho, A Study on the Success Factors of On-line Collaboration for Manufacturing Innovation Using AHP Analysis, KITS, 2011

[15] X. Meng, M.Sun and M. jones, Maturity model for supply chain relationships in construction, J. Manage. Eng., 2011

개인정보보호를 위한 효율적인 암호키 교체방안 연구

이선주

고려대학교 컴퓨터정보통신대학원 소프트웨어 보안학과

Study on Replacement of Cryptographic Key for Personal Information Protection

Sun-Ju Lee

Software Security Department

Graduate School of Computer & Information Technology, Korea University

요 약

2011년 개인정보보호법 제정에 따라 기업 및 공공기관은 보유하고 있는 개인정보에 대하여 자체 개발 또는 암호화 솔루션을 이용하여 개인정에 대해 암호화를 수행하였다. 보안의 안정성을 위해 암호화를 위해 사용한 암호 키는 사용 유효기간을 최소 2년에서 최대 5년 동안 사용을 권고하고 있어 정기적인 교체가 필요한 상황이다.

본 연구에서는 개인정보 암호화를 위한 방법과 암호 알고리즘별 유효기간 권고에 따라 암호키 관리시스템의 내부 아키텍처에 대해 연구하였다. 이에, 암호키 관리시스템의 암호 키를 보호하고 있는 마스터키 변경을 통해 암호키 변경 시 모든 개인정보에 대해 재암호화를 해야 하는 문제를 고려한 효과적인 키 교체 방안을 제시하고자 한다.

I. 서론

과거 개인정보 대량 유출 사고가 빈번히 발생함에 따라 사회 전반적으로 개인정보에 대한 관리의 중요성이 대두되었다. 이를 계기로 정부에서는 2011년 개인정보보호법을 제정하여 유예기간을 거쳐 시행을 하게 되었다.

개인정보보호법 등에 의거하여, 기업 및 공공기관은 필수적으로 개인정보를 암호화 하여야 한다.

이에 각 기업 및 공공기관에서는 보유하고 있는 개인정보에 대해 자체개발 및 국내•외 암호화 솔루션을 활용하여 암호화를 진행하였다. 하지만 암호화만으로는 개인정보의 안전성을 보장할 수 없다. 암호화된 정보의 안전성은 암호 알고리즘뿐만 아니라 암호 키 관리와도 연관이 있다고 할 수 있다.

만약 암호 키에 대한 관리 소홀로 암호 키가 유출된다면 암호화를 통한 정보보호는 의미가 없다. 그렇기 때문에 암호키를 안전하게 관리하는 것은 매우 중요하며 정보보호관리체계(ISMS; Information Security Management System) 인증제도, PCI DSS(Payment Card Industry Data Security Standard) 등에서는 암호 키 관리에 대한 요구 조건을 따로 명시하고 있다. 또한 미국 표준기술연구소(NIST; National Institute of Standards and Technology)에서는 암호 알고리즘 키별로 사용 유효기간을 정해 일정 기간이 지나면 변경을 권하고 있는 상황이다. 그렇지만 국내는 개인정보보호법 제정이후 많은 기업 및 공공기관에서 개인정보에 대해 암호화를 진행하였지만, 현재까지 암호화 솔루션의 기술적인 한계나 기업의 IT환경의 복잡성으로 인해 제1 금융권의 경우에도 일부데이터에 대해서만 암호화를 진행한 상황이다.

본 논문에서는 지금까지 많은 기업이 개인정보에 대해 암호화를 수행하면서 사용한 키가 미국 NIST에서 권고한 유효기간을 지난 경우도 있으며, 앞으로 도래하는 상황에서 보안성을 유지하면서 암호 키를 교체할 수 있는 현실적인 방안을 제시하고자 한다.

II. 암호화 관련 국내 법 규정

암호화와 관련된 법률 및 규정은 개인정보보호법, 정보통신망법, 전자금융감독규정, 신용정보업감독규정 등이 있다. 이들 법률 및 규정의 암호화 관련 내용들

을 간단히 살펴보면 다음 [표 1]과 같다.

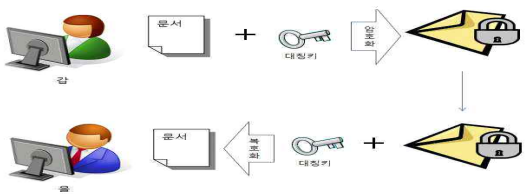
구분	내용
개인정보 보호법	고유식별정보를 처리하는 경우에 그 주민등록번호를 포함한 고유식별정보가 분실•도난•유출•변조 또는 훼손도지 아니하도록 암호화 등 안전성 확보에 필요한 조치를 하도록 함
정보통신망법	개인정보의 보호조치로 기술적•관리적 조치를 하여야 하며, 그 기준으로 개인정보를 안전하게 저장•전송할 수 있는 암호화 기술 등을 이용한 보안조치를 하도록 함
전자금융 감독규정	DMZ구간 내에 거래로그를 관리하기 위한 경우 암호화하여 저장•관리하고, 내부사용자 및 이용자 비밀번호를 암호화하여 보관하도록 함
신용정보업 감독규정	신용정보회사등은 패스워드, 생체정보 등 본인임을 인증하는 정보에 대해서는 복호화되지 않도록 일방향 암호화하여 저장하여야 함

[표 1]암호화 관련 법률

III. 암호화의 종류

1. 대칭키 암호화

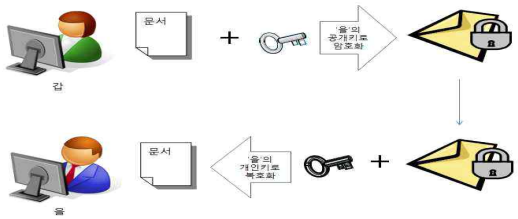
대칭키 암호화 방식은 [그림 1]과 같이 전송하고자 하는 평문을 암호화하고 복호화하는데 동일한 키를 사용하는 방식으로 공개키 암호화 방식에 비해 속도가 빠르지만, 정보 교환 당사자 간에 동일한 키를 공유로 인해 관리상의 어려움이 있다. 대표적인 암호 알고리즘으로 국내 SEED, ARIA, 미국의 3DES, AES 등이 있다.



[그림 1]대칭키 암호화 방식

2. 공개키 암호화

공개키 암호화 방식은 [그림 2]와 같이 평문을 암•복호화하는데 서로 다른 키를 사용하는 방식으로 비대칭키 암호화 방식이라고도 한다. 공개키 암호화 방식은 대칭키 암호화 방식에 비해 느리기 때문에 작은 크기의 데이터 암호화에 많이 사용되고 있다. 대표적인 암호 알고리즘으로는 국내에는 KCDSA, 국외에서는 RSA, ECC 등이 있다.



[그림 2]공개키 암호화 방식

3. 일방향(해쉬함수) 암호화

일방향 암호화 방식은 해쉬함수를 이용하여 암호화된 값을 생성하며 복호화 되지 않는 방식이다. 그렇기 때문에 비밀번호와 같이 복호화가 필요 없지만 입력 값의 정확성 검증이 필요한 경우에 사용하고 있다. 대표적인 해쉬함수로 SHA-2, RIPEMD-160 등이 있다.

IV. 암호키 관리시스템

현재 개인정보를 암호화하기 위한 상용솔루션을 구분하면, 컬럼암호화 방식과 블록 암호화 방식으로 나눌수 있다. 컬럼 암호화 방식은 DB의 Table에 저장되어 있는 데이터에 대해 컬럼단위로 암호화 할 수 있는 방식으로 DBMS에 암•복호화 모듈을 탑재하는 Plug-In방식과 암•복호화 함수를 제공하는 API방식이 있다. 또한 블록암호화 방식은 TDE(Transparent Data Encryption)방식과 파일암호화 방식으로 나눌수 있다. TDE방식은 응용 프로그램의 수정 없이 DB 내부에서 컬럼, 테이블스페이스 레벨의 암호화를 하는 방식이며, 파일암호화 방식은 서버에 에이전트 설치를 통해 암호화를 지원하는 방식이라고 할 수 있다.

이런한 암호화 방식은 암호 키 및 키와 연관된 정보를 관리하는 암호 키관리시스템(KMS; Key Management System)과 보안 함수(예, 암호 알고리즘, 키 생성 함수)를 사용하는 하드웨어, 소프트웨어, 펌웨어가 있는 암호모듈로 구분할 수 있다. 또한 개인정보를 저장하고 있는 서버(UNIX, Windows 등)에서 키관리시스템으로 부터 전달받은 키를 이용하여 실질적인 암•복호화를 수행하는 함수(에이전트, 커널모듈, API 등)가 있다. [그림 3]은 키관리시스템, 암호모듈, 서버간의 관계를 보여준다.



[그림 3]암호 키 생성 및 전송 과정

[그림 3]에서 암호 키를 생성하는 암호모듈은 보안성을 강화하기 위해 하드웨어화 하여 사용하는 경우도 있는데 이를 보안토큰(HSM; Hardware Security Module)이라 하는데, 전자서명생성키 등 비밀정보를 안전하게 저장·보관하기 위하여 키 생성, 전자서명 생성 등이 기기 내부에서 처리되도록 구현된 하드웨어 기기를 말한다.

V. 암호 알고리즘 및 사용 유효기간

정보보호제품 및 시스템에 암호 알고리즘을 탑재·적용하는 경우, 알고리즘의 종류나 키 길이에 따른 유효기간 등을 고려하여 해당 시스템의 안전성 수준을 만족할 수 있도록 선택해야 한다. 이를 위해 미국 일본, 유럽 등에서는 암호 알고리즘 및 키 길이에 따른 유효기간에 대한 가이드라인을 제시하고 있다. 그러나 국내의 경우, 「암호 알고리즘 키 길이별 유효기간」 기준이 없어 가이드라인이 시급한 실정이다.

1. 보안 강도별 암호 알고리즘 비교

암호 알고리즘 및 키 길이 선택 시, 암호 알고리즘의 안전성 유지기간과 보안강도별 암호 알고리즘 키 길이 비교표(표 2), 암호키의 사용 유효기간을 기반으로 암호 알고리즘 및 키 길이를 선택하도록 권장한다.

보안강도'	대칭키	공개키	해쉬함수	안전성 유지기간
80비트	80	1024	80	2010년까지
112비트	112	2048	112	2011년~2030년까지 (최대20년)
128비트	128	3072	128	2030년 이후 (최대30년)
192비트	192	7680	192	
256비트	256	15360	256	

[표 2]보안강도별 암호 알고리즘 비교표

만약 2009년에 80비트 보안강도의 암호 알고리즘을 적용하였을 경우, 2010년 이후에는 80비트 보안 강도의 암호 알고리즘이 안전하지 않으므로 이를 계속 사용하는 것은 권장하지 않고 있다.

즉, 112비트 이상의 보안강도를 제공하는 암호 알고리즘으로 암호화 또는 재해쉬를 적용하여 안전성을 강화하도록 권장하고 있다.

2. 암호키 사용 유효기간

암호 알고리즘 키 사용 유효기간이란 암호키를 사용할 수 있는 기간을 말하는 것으로서, 암호 알고리즘에 사용되는 키의 사용 유효기간은 암호키를 사용하는 기간과 암호문을 해석하기 위해 사용하는 복호키를 사용하는 기간을 말한다. [표 3]은 미국 표준기술연구소(NIST)의 키 관리 권고안을 기반으로 암호키의 사용 유효기간을 제시한다.

키 종류		사용 유효기간	
		송신자	수신자
대칭키	비밀키	최대 2년	최대 5년
공개키	암호화 공개키	최대 2년	
	복호화 개인키	최대 2년	

[표 3]암호키 사용 유효기간(NIST 권고안)

VI. 결론

본 논문에서는 미국 표준기술연구소에서 권고한 암호키 사용 유효기간 권고안에 맞게 보안성을 충족하기 위해서는 키 사용 유효기간 만료전에 기존에 사용한 암호 키를 교체해 주어야 한다.

하지만, 암호 키를 교체하는 것은 기존에 암호화를 했던 모든 개인정보에 대해 재암호화를 의미하는 것으로서 서비스 가용성 및 비용측면에서 기업 및 공공기관에게 부담이 되는게 현실이다.

이에, 현재까지 기업 및 공공기관에서 개인정보에 대한 암호화시에 사용한 키의 길이에 따른 보안강도와 사용 유효기간을 고려하여 효율적인 암호 키 교체 방안을 제시하고자 한다.

「3. 암호 키 관리시스템」에서 키관리시스템 및암호모듈 그리고 암호 키를 사용하는 서버간의 관계에 대해 설명을 하였다. [그림 4]에서 키관리 시스템은 암호모듈에서 생성한 암호 키를 보관하고 있다. 만약 생성하여 보관 중인 키가 유출되는 경우 개인정보와 같은 민감한 정보를 모두 복호화할 수 있기 때문에 보관하고 있는 암·복호 키에 대해 내부적으로 재암호화하여 보관하도록 설계되어 있다. 이때 사용한 암호 키를 마스터키라고 부른다.

[그림 4]키관리시스템 내부 구조

결론적으로, 암호 키 사용 유효기간이 지난 경우 암호화를 위해 사용한 암·복호 키는 그대로 유지한



상태에서 마스터키 교체를 통해 키 교체와 같은 보안성 측면에서 효과를 볼 수 있다. 이 경우 기업 및 공공기관에서 운영중인 시스템의 서비스 중단이나 별도의 비용이 발생하지 않는다.

이와 같이 기업 및 공공기관은 마스터키 교체를 통해 기존에 암호화했던 개인정보에 대해 재암호화를 하지 않고서도 보안성을 유지할 수 있는 현실적인 방안이라고 할수 있겠다.

[참고문헌]

[1]금융보안연구원, “DB암호화 기술 가이드”, 2014.12

[2]KISA, “기업 및 기관의 IT정보자산 보호를 위한 암호정책 수립 기준 설명서”, 2010.1

[3]KISA, “개인정보 암호화 조치 안내서”, 2012.10

[4]KISA, “암호이용 안내서”, 2013.12

[5]KISA, “암호 알고리즘 및 키 길이 이용 안내서”, 2013.01

[6]KISA, “암호기술 구현 안내서”, 2013.12

[7]KISA, “암호 키 관리 안내서”, 2014.12

[8]김두현, “개인정보보호의 최근 동향과 보호방안”, 2014.11

[9]한국인터넷자율정책기구, “KISO Journal Vol.14 - 개인정보보호 관련 규제입법 동향”,2014

서버로그 상관분석을 통한 보안 침해사고 탐지

윤인재*

*고려대학교 컴퓨터정보통신대학원

The security breach detection through server log correlation analysis

In-Jae Yun*

*Graduate School of Computer & Information Technology, Korea University.

요 약

인터넷 통신이 발전하면서 네트워크 구간에서의 패킷 전송 속도도 빨라졌으며 서비스 구간에서 사용되는 프로토콜(Protocol) 또한 많이 변화되어 가고 있다. 평문 통신인 HTTP(Hyper Text Transfer Protocol), FTP(File Transfer Protocol)는 HTTPS(Hyper Text Transfer Protocol Security)와 SFTP(Secure File Transfer Protocol)로 변화되어 가는 것처럼, 네트워크 구간에서는 많은 프로토콜이 암호화 통신으로 변경되고 있다. 이에 따라 본 논문에서는 현재의 평문기반의 네트워크 패킷에서 패턴 기반 침해사고를 탐지하는 네트워크 침입탐지 시스템의 현 상황에 대해 문제점을 확인하고, 호스트기반 서버로그(감사로그)를 수집하여 SIEM(Security Information and Event Management)을 활용하여 악의적인 침입시도 및 침해사고 탐지가 가능 하다는 것을 제시한다.

I. 서론

현재 네트워크 구간에서 침입시도 및 침해사고를 탐지하는 시스템으로는 침입탐지 시스템(IDS: Intrusion Detection System), 침입방지 시스템(IPS: Instrusion Prevention System), 웹 방화벽(Web Application Firewall) 등이 있으며 현재도 많은 기관 및 기업에서 사용하고 있다.

네트워크 보안 시스템은 네트워크 패킷을 스위치 또는 미러 탭을 통하여 패킷 미러링(Mirroring) 받거나 네트워크 구간에 인라인(In-Line) 모드로 설치되어 네트워크에서 흐르는 패킷들의 정보를 바탕으로 사전에 정의 된 탐지 패턴을 통해 악의적인 침입시도 및 침해사고를 탐지하는 방법을 사용한다.

침입탐지 시스템은 네트워크 전송 패킷이 평문 일 경우 탐지하며 악의적인 침입시도로는 웹 취약점 스캔, SQL Injection, XSS(Cross-Site Scripting), RFI(Remote File Inclusion), LFI(Local File Inclusion), Brute Force, Port Scan 등이 있다. 하지만 침입탐지 시스템은 악의적인 침입시도 및 침해사고를 탐지하기 위해 사전에 정의 된 Snort[1] 기반의 패턴 매칭 탐지 기법을 사용하고 있으며 탐지 된 내용을 토대로 분석된다.

인터넷 통신이 발전함에 따라 네트워크 통신에서 패킷이 암호화 되어 전송되는 비중도 늘고 있어 네트워크 패킷을 통한 악의적인 침입시도 및 침해사고를 탐지하기 어려워지고 있다. 또한, 침입탐지 시스템의 Snort 기반의 패턴 매칭 탐지 기법은 시간이 흐를수록 보안위협을 탐지하지 못하는 경우가 늘고 있어 이에 대한 대응을 위해 본 논문에서는 호스트 기반의 서버로그를 활용하여 상관분석을 통한 악의적인 침입시도 및 침해사고 탐지 기법을 확인한다.

2장에서는 관련연구로서 SIEM을 통한 서버로그 수집 방법에 대해 설명하며, 3장에서는 수집된 서버로그를 예를 들어 상관분석을 통한 공격위협을 탐지 및 모니터링하는 방법에 대해 설명하고, 4장에서는 결론을 맺는다.

II. 관련 연구

2장 관련 연구에서는 침입탐지 시스템의 현황과 본 논문에서 말하고자 하는 서버 운영체제(OS: Operating System)별 로그와 웹 로그를 SIEM으로 로그 수집(Log Collection) 및 수집된 로그를 파싱(Parsing)하고 파싱 된 로그를 통해 상관분석을 통한 악의적인 침입시도 및 침해사고 탐지방법에 대한 연구 현황을 기술한다.

2.1 침입탐지 시스템 현황

현재 침입탐지 시스템은 네트워크로부터 전달 받은 패킷을 바탕으로 사전에 정의된 탐지 패턴을 기준으로 알려진 침입시도 및 침해사고를 탐지한다. 탐지패턴 생성은 Snort 룰 기반으로 하며 Snort 기반의 다양한 탐지규칙을 통해 탐지 및 알람을 전달하며 보안 담당자는 알람된 탐지내용을 토대로 평문기반의 문자열을 분석한다. Snort룰은 네트워크 패킷이 평문일 경우 탐지가 가능하나 암호화 패킷은 탐지하지 못하는 상황이 계속 발생되며 증가하고 있다.

2.2 침입탐지 시스템과 서버로그의 차이점

침입탐지 시스템에서 발생되는 보안 이벤트는 네트워크 패킷에서 패턴 매칭 기반 위협 탐지로 패턴 매칭이 되면 담당자에게 알람을 주는 방식이다. 단순 문자열 기준으로 탐지하다보니 악의적인 공격이 아닌 정상적인 인터넷 통신인 경우에도 탐지 및 알람을 전송 한다.

서버로그 및 웹 로그는 서버에서 발생되는 실제 행위에 대해 로그를 남기며, 침입탐지 시스템처럼 정상적인 통신을 침입시도 및 침해사고를 탐지하지는 않으나, 실제적으로 이루어지는 모든 행위를 감사로그로 저장하므로 침입탐지 시스템의 이벤트 보다 더 직관적이고 정확도가 높다. 그러나 정상적인 행위까지 모두 로그로 남기다보니 침입탐지 시스템의 탐지 이벤트보다 많아 상관분석을 통해 전체 감사로그에서 악의적인 침입시도 및 침해사고 탐지해야 한다.

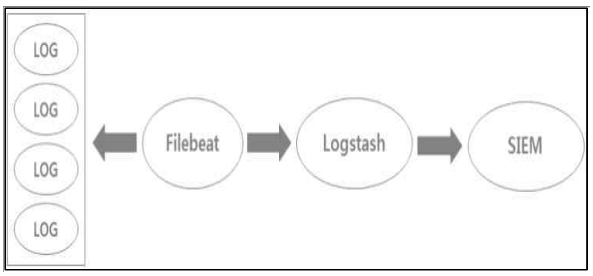
2.3 로그 수집 (서버로그, 웹 로그)

[표 1]은 서버 운영체제 별로 서버로그를 통해 침입시도 및 침해사고를 탐지하기 위해 필요한 로그 정보이다.

[표 1] 서버로그 및 웹 로그 정보

구분	로그	로그 정보
윈도우	응용 프로그램	날짜, 시간, 사용자, 컴퓨터, 이벤트ID, 원본, 종류, 범주 등
	시스템	
	보안	
리눅스	사용자	접속정보, 로그인 성공/실패, 인증 로그, 데몬 로그 등
	시스템	
웹	웹 로그	날짜, 시간, IP, 접속 페이지, user-agent, 메소드, 응답코드 등

서버로그를 수집하기 위해서는 서버별로 수집에 필요한 에이전트(Agent) 설치가 필요하며 오픈소스 기반의 Filebeat를 활용하여 수집 진행한다. Filebeat는 일반적인 형식의 로그 데이터들을 수집, 파싱 그리고 시각화를 단일 명령으로 처리가 가능한 내부 모듈(Apache, Nginx, System 및 MySQL) 들을 제공한다[2].



[그림 1] 서버로그 수집 시스템 구성

[그림 1] 과 같은 시스템 구성으로 서버로그를 SIEM 으로 로그 수집 및 상관분석 방법을 3장에 기술한다.

III. 상관분석 구현

시스템에 문제가 발생했을 경우와 악의적인 침입시도로 인해 침해사고가 발생했을 경우 보안담당자는 로그분석을 진행해야 한다. 해킹의 흔적과 기록을 확인하기 위해서 보안담당자는 서버로그를 분석 및 의존하여, 누가(ID), 어디서(IP Address), 어떻게(Port) 들어와서 어떤 취약

점을 통해 불법적으로 침입을 했는지, 또는 정보 훼손 및 변경을 했는지에 대한 내용은 모두 서버로그 분석을 통해 이루어진다[3].

3.1 웹 로그 상관분석 구현

웹 로그는 웹 서버에 대한 모든 접근 기록을 기록한 데이터로 접속한 IP Address, 접근 시간, 접근 방법, 대상 URL, 프로토콜, 응답코드, 전송 바이트 수 등을 저장한다. SIEM으로 수집된 웹 로그를 통해 아래 [표 2]와 같이 상관분석이 가능하다[4].

네트워크 구간에서 암호화 통신을 하더라고 웹 로그에서는 평문으로 로그가 저장되므로 유일 값으로 접속한 IP Address를 잡고 공격 유형별로 메소드, 응답코드, URL 등을 AND 조건(모두 참일 때 발생)일 경우 알람을 발생 시킨다.

- 예: 메소드="PUT"&응답코드="201"&URL= "*.txt"
- 예: 메소드="GET"&응답코드="200"&URL="*cmd.exe"

[표 2] 웹 로그 상관분석 예

구분	메소드	응답코드	URL(예)
파일 업로드	PUT	201	/alert.txt
명령어 실행	GET	200	/test7.asp?path=/c:\windows\system32**cmd.exe**
서버 취약점	GET POST	500	/test7.asp?|4**66|800 40e37**|[Microsoft][ODBC_SQL_Server_Driver][SQL_Server]개체_이름_'a_job'이(가)_잘못되었습니다.

3.2 윈도우 로그 상관분석 구현

윈도우 서버는 시스템, 보안, 응용프로그램 및 서비스 로그 등이 있다. 윈도우 서버는 독자적인 프로토콜을 채택하고 있으며, 이벤트 뷰어를 통해 로그 검색 및 다운로드가 가능하며 SIEM으로 로그 전송 후 상관분석이 가능하다[5].

윈도우 서버 응용프로그램 로그로 예를 들면 서버IP나 호스트명으로 유일 값으로 잡고 [표 3]과 같이 2가지 이상의 값을 AND 조건 일 경우 알람이 가능하며 침입탐지 시스템보다 직관적이고 빠르게 확인 가능하다.

- 예: 원본="AntiVirus"&이벤트ID="51"
- 예: 이벤트ID="4625"&메시지="*winlogon.exe*"
- 예: 이벤트ID="4688"&메시지="*cmd.exe*"

[표 3] 윈도우 응용프로그램 로그 상관분석 예

원본	이벤트 ID	메시지
AntiVirus	51	보안 위험 요소 발견! Backdoor.Winnti 파일 위치: c:\windows\system32\wmtsk.dll.vir 검사 유형: 수동 검사. 수행 작업: 재시작 필요.
EventSystem	4625	Account For Which Logon Failed: Security ID: S-1-0-0 Account Name: hacker Account Domain: AAA097 Failure Information: Failure Reason: Unknown user name or bad password. Status: 0xc000006d Sub Status: 0xc000006a Process Information: Caller Process ID: 0x2694 Caller Process Name: **C:\Windows\System32\winlogon.exe** Network Information: Workstation Name: WYNA097 **Source Network Address:** 89.27.154.18 Source Port: 58541
EventSystem	4688	Security ID: S-1-5-21-1630367190-4051936831-3185179968-1001 Account Name: Hacker Account Domain: INJAE-APP Logon ID: 0x1517e1c2 Process Information: New Process ID: 0x1ef8 New Process Name: C:\Windows\SysWOW64**cmd.exe** Token Elevation Type: TokenElevationTypeLimited (3) Creator Process ID: 0x2850 **Process Command Line: C:\windows\system32\cmd.exe /S /D /c" dir C:\windows\system32 /a**

3.3 리눅스 로그 상관분석 구현

리눅스 로그파일은 일반적으로 /var/log/에 시스템의 모든 로그를 기록 및 관리하고 있다. 로그파일은 일반적으로 syslog.conf 설정에 따라 달라지며 boot.log는 부팅 및 각종 서비스 시작 및 중지, cron은 정기적인 작업, message의 경우 커널 에러, 리부팅 메시지, 로그인 실패, secure는 시스템의 접속에 관한 로그파일로 언제, 누가, 어디에서 접속했는가에 대한 로그를 기록하고 있다[6].

[표 4]와 같이 리눅스 로그도 서버IP나 호스트 이름이 키가 되며 로그 내용 값을 활용하여 상관분석을 진행한다.

- 예: log="*root*"&log="*wget*"&log="*http*"
- 예: log="*sshd*&log="*root*"&log="Failed passwd*"

[표 4] 리눅스 로그 상관분석 예

로그파일	log
message	Oct 12 02:17:50 aaahost root: [1] PID=10887, PWD=/root, CMD= 622 **wget http://179.185.16.20/log**
secure	Oct 17 07:43:35 localhost **sshd**[21324]: **Failed password for root** from **179.185.16.20** port 27025 ssh2

3.4 임계치(Threshold) 기반 상관분석 구현

서버로그(웹, 윈도우, 리눅스)는 인증 실패와 같이 불법적인 로그인 시도에 대해서도 로그로 기록한다. 그러나 로그인 과정 발생하는 모든 로그인 실패 로그는 모두 모니터링이 어려우며 정상적인 로그인 과정에서도 많이 발생한다. 이에 따라, 인증 실패와 같은 로그는 임계치 기반으로 실패 횟수에 따른 모니터링 및 분석이 필요하며, 임계치 이상으로 로그가 발생될 경우에는 Brute Force 공격으로 판단하고 IP차단 및 불필요한 계정 접근을 차단해야 한다.

웹 로그에서는 Admin Page 및 불법적인 로그인 시도, 윈도우는 원격 데스트톱(RDP: Remote Desktop Protocol), 리눅스는SSH(Secure Shell)와 같은 접근 시도 등의 로그를 예로 들 수 있다. 지속적인 불법시도는 임계치 기반으로 모니터링이 가능하며 과탐을 방지 할 수 있다.

IV. 결론

본 논문에서는 빠른 속도로 발전하는 인터넷 환경과 더불어 네트워크 패킷 암호화에 따른 악의적인 침입시도 및 침해사고를 네트워크 단에서 탐지가 힘들어 지고 있어 서버로그를 활용한 탐지 방안에 대해 예를 들어 제시하였다.

침입탐지 시스템이 암호화된 네트워크 구간이라고 모든 것을 탐지하지 못하는 것은 아니다. 그러나 탐지할 수 없는 경우가 지속적으로 발생하고 증가하고 있으므로, 서버로그를 활용하여 침입시도 및 침해사고에 대해 모니터링 및 분석하는 것이 가능하다는 것을 보였다. 또한, 신규 보안위협에 대해 모니터링하려면 지속적으로 침입시도 및 침해사고에 대해 분석 및 상관분석룰을 만들어야 한다.

본 논문에서 제시한 상관분석을 활용하여 분석 범위를 넓히고 지속적인 침입시도에 대한 모니터링을 강화한다면 보호하고자 하는 자산과 정보에 대해 보다 안전하게 관리하고 운영하는데 큰 도움이 될 수 있을 것이다.

[참고문헌]

[1] Snort (오픈소스 침입탐지 시스템) (https://www.snort.org/documents)

[2] Filebeat (경량 로그 수집기) (https://www.elastic.co/kr/products/beats/filebeat)

[3] 임성화,이도현,김점구,“리눅스 기반 침입 방지를 위한 로그 분석 방법 연구”,한국융합보안학회,제15권 2호,3,2015.03.

[4] 김석기,안정용,한경수,“웹 로그(WEB LOG) 데이터 분석 방법에 관한 연구”, 한국통계학회,제 14권 2호,2,2001

[5] 김완집, 염흥열,“이기종 로그에 대한 통합관리와 IT 컴플라이언스 준수”, 한국정보보호학회,제 20권 5호,3,2010.10

[6] 임성화,이동현,김점구,“리눅스 기반 침입 방지를 위한 로그 분석 방법 연구”, 한국융합보안학회,제 15권 2호,2,2015.03

SW Security 표준의 CWE 영향도 분석

이헌기*, 최진영

*고려대학교 정보보호대학원 정보보호학과

Analysis of CWE Technical Impact for SW Security Standards

Lee Heon Ki*, Jin-young Choi

*Division of Information Security, Korea University.

요 약

SW 보안(Software Security) 표준을 채택하면 소프트웨어의 신뢰성 및 보안성을 향상시킬 수 있다. 국내에서 많이 사용하고 있는 SW 보안 표준은 행안부의 소프트웨어 개발보안 가이드, OWASP Top 10, CWE/SANS Top 25이다. 본 논문에서는 국내외 SW 보안 표준들의 특징과 관계를 기술하고, CWE의 기술적 영향도 관점에서 각 SW 보안 표준의 보안 취약점 규칙들을 분석하여 국내 SW 보안 표준의 보안 강화의 필요성을 제안한다.

I. 서론

보안 침해의 약 75% 이상이 응용 소프트웨어와 관련돼 있고[1], 최근 발생되는 인터넷상 공격시도는 소프트웨어 보안 취약점을 악용하는 것으로, 특히 외부에 공개되어 불특정 다수를 대상으로 사용자 정보를 처리하는 웹 애플리케이션의 취약점으로 인해 중요정보가 유출되는 침해사고가 빈번하게 발생되고 있다. 이러한 안전하지 않은 소프트웨어는 금융, 의료, 국방, 에너지 및 기타 중요한 인프라를 약화시키고 있기 때문에 더욱 빠르고 정확하게 발견하는 것이 중요하다.

최근 새로운 기술(클라우드, 컨테이너 및 API 포함)과 오픈 소스의 폭발적 증가로 어플리케이션 및 API를 위한 위협 요소가 끊임없이 변화하고 있다. 이러한 변화에 대응하기 위해, 보안 위험뿐만 아니라 발생 빈도에 우선순위를 두고 OWASP Top10(2017년)[2]은 보안 기술을 주기적으로 업데이트하고 있으나, 행안부의 소프트웨어 개발 보안 가이드(2013년)[3]는 그렇지 못하고 있다.

본 논문에서는 국내외 3 가지 SW 보안 표준 특징과 관계, 그리고 각 SW 보안 표준과 관련된 CWE의 기술적 영향도 관점에서 정보보호 서비스 유형과 보안 취약점의 어플리케이션 영역을 식별하여 국내 소프트웨어 개발보안 가이드의 보안 강화를 제시한다.

II. SW 보안 표준의 특징

국내적으로 소프트웨어 보안성 코딩 가이드는 대표적인 3 가지 SW 보안 표준을 주로 사용하며, 각각에 대해 특징을 간략하게 설명한다.

(1) 소프트웨어 개발보안 가이드[3][4]

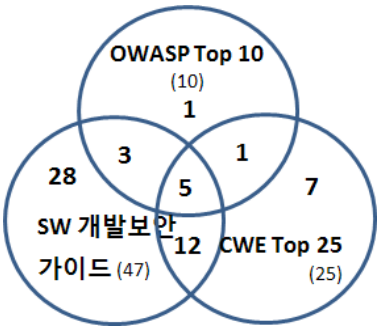
전자정부서비스 보안을 강화하기 위해 2012년 12월부터 시큐어 코딩에 대한 법규가 제정 및 시행되고 있으며, 2013년에 개정하여 지금까지 소프트웨어 보안약점 47 가지 항목의 기준을 사용하고 있다[3]. 한편 2017년에 발간된 소프트웨어 개발보안 가이드[4]는 2013년 가이드의 예제 등을 보완하였다.

(2) OWASP Top 10[2]

가장 심각한 웹 어플리케이션의 보안 위험 요소를 10 가지 항목으로 선정하여 2004년, 2007년, 2010년, 2013년, 2017년과 같이 주기적으로 업데이트하고 있다.

(3) CWE/SANS Top 25[5][6]

SW의 심각한 취약점으로 이어질 수 있는 가장 광범위하고 치명적인 오류들을 25 가지 항목으로 선정하여 2010년, 2011년에 발표하였다.



[그림 1] SW 보안 표준 관계

그림 1과 같이 3 가지 SW 보안 표준들의 특징은 아래와 같이 웹과 API관련 보안 취약점의 5 가지 항목들을 공통 표준으로 채택하고 있다.

(1) CWE-79 Cross-site Scripting
(2) CWE-306 Missing Authentication for Critical Function
(2) CWE-307 Improper Restriction of Excessive Authentication Attempts
(2) CWE-352 Cross-Site Request Forgery
(3) CWE-676 Use of Potentially Dangerous Function (API)

또한 행안부의 소프트웨어 개발보안 가이드[3][4]는 웹 보안 취약점인 OWASP Top 10[2]과 SW 오류인 CWE Top 25[5]에 대해 각각 80%, 68%의 국제 SW 보안 표준들을 채택하는 특징을 갖고 있다.

III. 각 SW 보안 표준에 대한 CWE의 기술적 영향도 분석

SW 보안 표준의 전체적인 목적은 소프트웨어의 보안성을 높이는 것이다. 각 보안 취약점 규칙의 침해 결과인 정보보호 서비스 유형과 이에 관련된 CWE 기술적 영향도의 분석은[7] 표 1과 같이 보안 취약점의 어플리케이션 영역을 식별하는데 사용되고, 또한 이러한 이론적 근거로 MITRE CWE[7]에서는 보안 약점의 카테고리 표준으로 Research Concepts(연구 개념)과 Development Concepts(개발 개념)의 CWE 사전 보기(Dictionary View)를 제공하고 있다. 예를 들어, CWE-442(Web Problem)[7]의 CWE 카테코리는 기밀성, 접근 제어, 부인 방지의 정보보호 서비스 유형인 웹 보안 취약점 규칙들의 집합으로 정의하고 있다. 또한 CWE-254(Security Features)[7]의 CWE 카테코리는 인증. 접근 제어, 기밀성, 암호화, 권한 관리의 정보보호 서비스 유형인 정보보호의 보안 취약점 규칙들의 집합으로 정의하고 있다.

[표 1] CWE 기술적 영향도별 보안 취약점 공격 영역

정보보호서비스	CWE 기술적 영향도	어플리케이션 영역
기밀성 (Confidentiality)	• Read application data • Read memory	Code Component
무결성 (Integrity)	• Execute unauthorized code or commands • Modify application data • Modify memory • Varies by context	Code/Component Component Component Code
가용성 (Availability)	• DoS: crash/exit/restart • DoS: Resource consumption (CPU/Memory)	Code Component
접근제어 (Access Control)	• Bypass protection mechanism • Gain privileges	Code Code

표 1과 같이, CWE 기술적 영향도의 요소와 관련한 어플리케이션 영역의 식별은 보안 취약점의 공격 영역으로 각 보안 취약점 규칙들의 보안 강화에 대한 우선순위 기준의 지침을 제공할 수 있다. Code 단위의 보안 취약점 노출은 치명적인 코드 오류로 1차 공격의 코드 취약점으로 자동화 공격 대상이지만, Component 단위의 보안 취약점 노출은 논리 흐름에 대한 수동적 분석인 논리적 공격 대상으로 Code 단위 보다 보안 강화의 우선순위가 상대적으로 낮은 2차 공격의 Component 취약점이다. 이러한 논리 흐름의 공격은 공격자가 수동으로 Component의 로직 구조 또는 어플리케이션 계층을 분석하여 공격해야 하는 약점이 있다. 예를 들어, 행안부의 소프트웨어 개발 보안 가이

드[3][4]만이 SW 보안 표준으로 제공하는 CWE-113 (HTTP Response Splitting) 보안 취약점은 표 1과 같이 무결성의 정보보호 서비스 유형과 관련된 “Modify application data”의 CWE 기술적 영향도와 매핑이 되고, 이와 관련된 어플리케이션 영역에서 Component 단위의 보안 취약점을 노출하고 있어, HTTP Response Splitting 공격은 아래와 같이 항상 3 가지 구성이 최소한 필요하다[8][9].

(1) HTTP 응답분할을 가능하게 하는 보안 허점이 있는 Web Server
(2) Target - 공격자를 대신하여 웹 서버와 상호 작용하는 Entity
(3) Attacker -웹 브라우저에서 악의적인 코드 주입으로 공격 시도

[표 2] SW 보안 표준별 CWE 기술적 영향도 매핑

SW 보안 표준 / CWE 기술적 영향도	SW 개발 보안가이드 (2017)	OWASP Top 10 (2017)	CWE Top 25 (2011)
DoS: crash/exit/restart	3	0	2
DoS: Resource consumption (CPU/Memory)	2	0	0
Read memory	1	0	1
Modify memory	1	0	0
Execute unauthorized code or commands	6	2	6
Read application data	13	3	6
Modify application data	4	0	0
Bypass protection mechanism	10	2	6
Gain privileges	5	2	3
Varies by context	2	1	1
#Rule of SW Security Standards	47	10	25

표 2에서 각 SW 보안 표준의 CWE 기술적 영향도를 분석한 결과, 국제 SW 보안 표준들과 매핑되지 않고 오직 행안부의 소프트웨어 개발보안 가이드[3][4]만이 제공하는 3 가지 유형의 CWE 기술적 영향도는 표 1과 같이 모두 Component 단위의 보안 취약점을 노출하는 속성이며, 이와 관련된 총 7 항목의 보안 취약점들은 아래와 같다.

(1) DoS: Resource consumption (CPU/Memory)
- CWE-404 (Improper Resource Shutdown or Release)
- CWE-674 (Uncontrolled Recursion)

(2) Modify memory
- CWE-416 (Use After Free)

(3) Modify application data
- CWE- 113 (HTTP Response Splitting)
- CWE-367 (TOCTOU Race Condition)
- CWE-495 (Private Array-Typed Field Returned From A Public Method)
- CWE-496 (Public Data Assigned to Private Array-Typed Field)

위와 같이 행안부의 소프트웨어 개발보안 가이드[3][4]만이 제공하는 Component 단위의 7 가지 보안 취약점들은 국제 SW 보안 표준처럼 공격 가능성의 우선순위가 높은 Code 단위의 보안 취약점으로 보안을 강화할 필요가 있다. 예를 들어, 그림 1의 SW 보안 표준 관계에서 행안부 소프트웨어 개발보안 가이드의 차집합인 Code 단위의 9 가지 보안 취약점들을 선정하는 것이다. Code 단위의 9 가지 보안 취약점에 대한 각 SW 보안 표준별 카테고리는 아래와 같다.

(1) OWASP Top 10
- A1: CWE-77

(2) CWE/SANS Top 25
- Risk Resource Management: CWE-120, 131,
- Porous Defenses: CWE-250, 311, 798, 862, 863

(3) OWASP Top 10과 CWE/SANS Top 25의 교집합
- A9/Risk Resource Management: CWE-829

IV. 결론

어플리케이션 및 API를 위한 위협 요소가 끊임없이 변화하고 있다. 이러한 변화의 핵심 요

소는 새로운 기술(클라우드, 컨테이너 및 API 포함)의 빠른 수용, 오픈 소스 컨포넌트와 같은 third-party 라이브러리와 프레임워크의 폭발적 증가 등이 있다. 이러한 요인으로 인해 어플리케이션과 API를 분석하기가 더 어려워지고, 이를 통해 보안 위험 요소가 크게 변경될 수 있다.

이러한 보안 위협 요소의 변화에 대응하기 위해, 웹 보안 취약점인 OWASP Top 10[2]과 SW 오류인 CWE Top 25[5]는 오직 Code 단위 보안 취약점들의 보안 기술을 주기적으로 제공하는 것처럼 행안부의 소프트웨어 개발 보안 가이드[3][4]도 Component 단위의 보안 취약점들을 공격 가능성이 쉬운 1차 공격의 Code 단위 보안 취약점들로 개정할 필요가 있다. 추가적으로 웹 보안 취약점과 SW 오류를 모두 포함하고 있는 현재의 행안부의 소프트웨어 개발 보안 가이드[3][4] 역시 웹 보안 취약점인 OWASP Top 10[2]과 SW 오류인 CWE Top 25[5]의 국제 SW 보안 표준 특성처럼 웹과 SW 오류를 구분한 소프트웨어 개발 보안 가이드의 개발도 필요하다.

마지막으로 보안 취약점 표준은 보안을 강화할 수 있는 소프트웨어를 개발하는데 도움이 되는 유일한 보안성 코딩 가이드라는 것을 기억하는 것이 중요하다.

[참고문헌]

[1] Gartner, "Now is the time for security at application level", http://www.gartner.com, December, 2005.

[2] OWASP, OWASP Top 10 Application Security Risks - 2017, September 8, 2017, https://www.owasp.org/index.php/Top_10_2017-Top_10

[3] 행정안전부, “전자정부 SW개발.운영자를 위한 소프트웨어 개발보안 가이드”, 2013. 11

[4] 행정안전부, “전자정부 SW개발.운영자를 위한 소프트웨어 개발보안 가이드”, 2017. 1

[5] MITRE, 2011 CWE/SANS Top 25 Most Dangerous Software Errors, September 13, 2011, http://cwe.mitre.org/top25

[6] Microsoft, The Microsoft SDL and the CWE/SANS Top 25, 2009

[7] MITRE, CWE Version 2.11, May 18, 2017, http://cwe.mitre.org/index.html

[8] Web Application Security Consortium, WASC THREAT CLASSIFICATION Version 2.0, 2010, http://projects.webappsec.org/w/page/13246978/Threat%20Classification

[9] Web Application Security Consortium, Threat Classification Version 1.0, 1994, www.webappsec.org

악성코드가 호출하는 API 시퀀스를 이용한 유사 악성코드 클러스터링 방안 연구

유대훈, 최보민, 강홍구, 박준형*

*한국인터넷진흥원

A Study on the Malware Clustering Based on its API Call Sequence

Yoo Dae Hoon, Choi Bo Min, Kang Hong Koo, Park Jun Hyung*

*Internet & Security Agency

요 약

급증하는 악성코드의 대부분은 기존 악성코드의 변종이다. 따라서 제한된 인력으로 급증하는 악성코드에 대응하기 위해서는 유입되는 악성코드간의 관계 분석을 통해 악성코드의 특성을 자동으로 식별할 수 있어야 한다. 하지만 대량의 악성코드를 전수 비교하여 유사한 악성코드를 식별하는 것은 현실적으로 불가능하다. 본 논문에서는 이를 해결하기 위해 악성코드의 행위정보를 전처리하여 정해진 길이의 벡터를 생성하고, K-means 알고리즘을 이용하여 이를 클러스터링함으로써 행위가 유사한 악성코드들을 효율적으로 분류하는 방안을 제시한다.

I. 서론

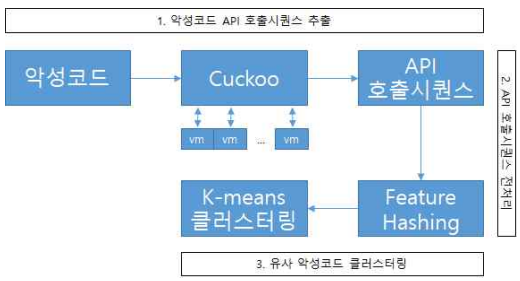
2017년 하루 평균 190만개 악성코드가 발생하는 등 악성코드가 급증하고 있다[1]. 대부분의 악성코드는 기존 악성코드의 변종이며, 이를 미리 자동으로 식별할 수 있다면 제한된 대응인력을 효율적으로 활용할 수 있을 것이다. 이를 위해 본 논문에서는 대량의 악성코드들을 대상으로 유사한 악성코드를 자동으로 클러스터링하는 기법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 제안하는 기법에 대해 상세히 설명하고, 3장에서는 제안하는 기법을 이용한 클러스터링 실험 결과를 제시한다. 마지막 4장에서는 결론과 향후 연구방향에 대해 이야기한다.

II. 본론

2.1 개요

(그림 1)은 본 논문에서 제안하는 기법을 나타낸 흐름도이다.



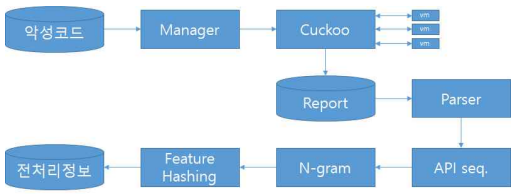
(그림 1) 악성코드 분류 흐름도

먼저 Cuckoo 샌드박스를 이용하여 악성코드의 행위정보를 생성하고, 이 중 첫 번째 프로세스의 API 호출 시퀀스를 추출한다. 다음으로 API 호출 시퀀스에 2-gram과 Feature Hashing 기법을 적용하여 악성코드 별로 정해진 길이의 행위정보 벡터를 생성한다. 마지막으로 악성코드들의 행위정보 벡터에 K-means 클러스터링 알고리즘을 반

복 적용하여 유사한 악성코드를 클러스터링한다.

2.2 API 호출 시퀀스 전처리

악성코드의 가장 대표적 행위정보는 해당 악성코드가 호출 API의 목록이다. 악성코드는 하나이상의 프로세스를 생성하며, 각 프로세스는 수개에서 수십만개 이상의 서로 다른 개수의 API를 호출한다. 본 논문에서는 오픈소스 동적 분석도구 및 샌드박스인 Cuckoo와 KVM을 이용하여 악성코드의 행위정보 보고서를 생성하고, 각 악성코드가 첫 번째로 호출하는 프로세스의 API 호출 시퀀스를 추출하여 클러스터링에 사용한다.

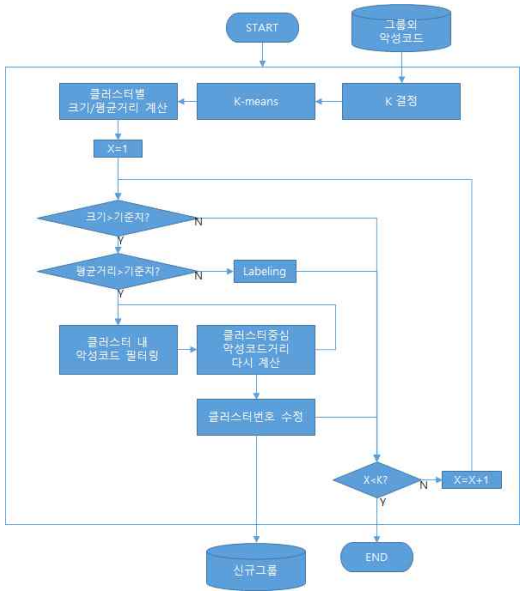


(그림 2) 악성코드 별 API 호출 시퀀스 전처리

위와 같이 추출한 악성코드 별 API 호출 시퀀스는 악성코드의 종류와 가상머신환경에 따라 길이가 천차만별이므로 그대로 비교하기 어렵다. 따라서 본 논문에서는 N-gram 기법과 Feature hashing 기법을 이용하여 길이가 상이한 API 호출 시퀀스를 동일한 길이의 정수 벡터로 변환한다. N-gram 기법은 임의의 길이인 시퀀스의 통계적 특성을 나타내기 위한 방법으로 자연어 처리등에 주로 사용된다. Feature hashing 기법은 데이터의 인덱스를 해싱하여 차원을 축소하는 방법이다. 본 논문에서는 악성코드의 API 호출 시퀀스를 2-gram으로 나타내고, 이를 scikit-learn의 feature hashing 함수를 사용하여 길이가 1024인 벡터로 압축한다.

2.3 유사 악성코드 클러스터링

클러스터링 알고리즘은 여러 객체들을 서로 유사한 것들끼리 묶어내는 방법을 말한다. 가장 널리 사용되는 클러스터링 알고리즘 중 하나인 K-means 클러스터링 알고리즘은 동작이 직관적이고, 속도가 빠르다는 장점이 있으나, 지정하는 군집의 갯수에 따라 군집정확도의 차이가 크다는 단점이 있다. 악성코드 변종의 가짓수를 미리 알 수 없으므로, 본 논문에서는 K-means 클러스터링 알고리즘을 반복하여 적용함으로써 클러스터의 갯수를 자동을 정하도록 한다.



(그림 3) 유사 악성코드 클러스터링

본 논문에서 제안하는 유사 악성코드 분류방안은 클러스터링을 반복하여 적용하며, 밀집도가 높고 크기가 큰 클러스터에 속하는 악성코드들을 유사하다고 판단한다. 이 때, 클러스터 내 악성코드 중 클러스터 중심과 유사도가 떨어지는 악성코드를 필터링함으로써 분류결과의 정확도를 높이고자 한다.

III. 실험 및 결과

본 절에서는 본 논문에서 제안한 알고리즘을 이용하여 행위가 유사한 악성코드를 분류하고 그 결과를 분석한다. 실험에 사용된 악성코드는 2017년 상반기 수집된 악성코드 24,845건으로 샘플의 타입은 아래 표와 같다.

[표 1] 분류샘플 파일타입

파일타입	exe_64bit	exe_32bit	pdf
악성코드수	8,913	11,855	4,077

3가지 파일타입의 악성코드 24,845건을 유사

한 행위에 따라 분류한 결과, 전체 악성코드의 65.34%가 367개의 클러스터로 분류되었으며, 클러스터들의 평균유사도 평균은 99.77%였다. 자세한 분류결과는 [표 2]와 같다.

[표 2] 악성코드 분류결과

평균유사도	클러스터 수	악성코드 수	악성코드 비율
1.000	27	1017	4.09%
0.999 이상	119	5707	22.97%
0.998 이상	65	4888	19.67%
0.997 이상	39	1583	6.37%
0.996 이상	45	1501	6.04%
0.995 이상	28	581	2.34%
0.994 이상	17	425	1.71%
0.993 이상	8	217	0.87%
0.992 이상	7	151	0.61%
0.991 이상	5	67	0.27%
0.990 이상	3	40	0.16%
0.980 이상	4	57	0.23%
클러스터외	1	8611	34.66%
합계	368	24845	65.34%

IV. 결론

본 논문에서는 악성코드의 API 호출시퀀스를 N-gram 기법과 Feature Hashing 기법으로 전처리하여 악성코드 별 행위벡터를 생성하고, 이를 K-means 알고리즘을 이용하여 클러스터링 함으로써 행위가 유사한 악성코드를 분류하는 기법을 제안하였다. 2017년에 수집된 악성코드 약 25,000건을 대상으로 제안한 알고리즘을 적용해본 결과, 전체의 65.34%를 평균 유사도 99.77%로 분류할 수 있었다. 본 연구는 클러스터링알고리즘을 이용하여 행위가 유사한 악성코드를 높은 유사도로 묶어낼 수 있는 가능성을 보였다는 점에서 의의가 있다.

향후 연구에서는 악성코드가 호출하는 다양한 프로세스를 대상으로 행위정보 수집하는 등 전처리과정을 개선하고, 악성코드간 유사행위 분류과정을 개선하여 분류성능을 높일 예정이다.

ACKNOWLEDGMENT

이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2016-0-00081, 악성코드 全 생명주기 통합 프로파일링 및 공격그룹 식별 기술 개발)

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No.2016-0-00081, The Development of Integrated Malware Life-cycle Profiling and Attack-group Identification Technology)

[참고문헌]

[1] AV-TEST, www.av-test.org, 2017

[2] Lee, T., Choi, B., Shin, Y. et al., Automatic malware mutant detection and group classification based on the n-gram and clustering coefficient, J Supercomput (2015), 2015

[3] 최보민, 강홍구, 이태진, 악성코드 API 정보를 이용한 변조유형 예측방법 연구, 한국통신학회 2015년도 동계종합학술발표회, 2015

사이버위협에 대한 소프트웨어 보안강화 방안 연구

이길호

고려대학교 컴퓨터정보통신대학원 소프트웨어보안학과

Study on strengthening software security against cyber threats

Kil-Ho Lee

SoftwareSecurity, Korea University Graduate School of Computer & Information Technology.

요 약

21세기 현대사회를 살아가는 최대 이슈는 당연 4차산업혁명과 빅데이터, 클라우드, 로봇, 인공지능(AI), 사물인터넷(Internet of Thing) 등 TV나, 신문, 인터넷, 기타 모든 매체에서 핵심적으로 다루는 필수 용어일 것이다.

그러나 이러한 新 성장동력의 핵심 주제속에 간과하고 있는 사이버 범죄자들의 지능화, 고도화, 전문화된 방법으로 소프트웨어의 취약점을 이용한 정보유출(금융정보 등)로 경제적인 이득을 취하는 등의 사이버범죄에 대한 대응방안은 속수무책이다.

이러한 사이버보안을 위협하는 소프트웨어의 취약점 및 보안약점을 극복하여 좀 더 안전하고 신뢰할 수 있는 정보보안과 정보유출을 최소화 할 수 있는 보안강화 방법을 분석하여 발전 방향을 제안한다.

I. 서론

최근 2017년 5월 12일 윈도우 SMB취약점을 이용하여 전 세계의 기업과 병원, 공공기관 등 대상을 가리지 않고 감염시켜 피해를 확산시켰던 일명 멀웨어(malware: 악성프로그램)라고 불리우는 워너크라이(WannaCry)랜섬웨어는 다행히 우리나라의 경우 주말이 겹쳐서 피해가 적었지만 사이버범죄자(악의적인 공격자)들의 사이버보안위협은 매우 심각하다.

이와같이 사이버공격에 대한 사이버 피해는 더욱 더 증가할 것이며, 사이버보안을 위협하는 소프트웨어 취약점 및 보안약점에 대한 극복을 위한 대응방안은 전 세계의 공통관심사가 아닐 수 없다. 이렇게 우리는 하루가 다르게 발전하는 현대시대를 살아가고 있지만 보다 발전된 형태의 소프트웨어 취약점 및 약점을 극복할 수 있는 좀 더 체계적인 방법으로 신기술 보안강화 대응이 필요할 것이다.

II. 소프트웨어 개발 보안 검토

2.1 개요 및 배경

소프트웨어 개발 단계(Software Development Life Cycle : SDLC)에서 보안 약점(Weakness)을 완전히 제거한다는 것은 사실상 불가능하지만 최대한 보안약점을 극복한 소프트웨어를 개발한다는 것이 주목적일 것이다.

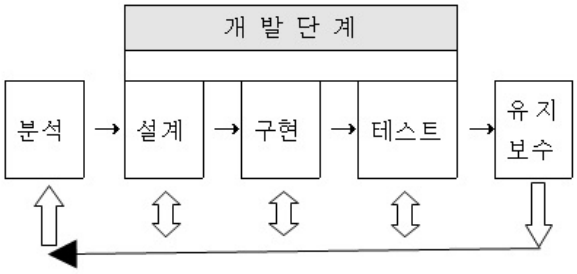


그림1. 소프트웨어 생명주기 모형

2.2 CERT C

CERT C는 미국 정부에서 지원하는 카네기 멜론대학 내의 Software Engineering Institute 에서 개발 하였고 89개의 필수 규칙과 132개의 권장 규칙이 있으며 제조분야뿐만 아니라 C언어로 작성된 모든 소프트웨어에도 적용할 수 있다. 그리고 소프트웨어오류, 보안 결함, 소프트웨어 취약점으로 이어질 수 있는 C언어의 일반적인 에러 및 오류를 열거하고 어떻게 안전한 코드를 작성 할 수 있는지를 제안하며 호환 가능하지 않은 코드를 구별하고 있다.

개발 단계별로 코딩 규칙에 대한 테스트 문서 작성이나 검사를 함으로써 소프트웨어 보안 약점을 사전에 제거할 수 있어 보안 취약점 감소 효과를 기대 할 수 있다[1].

2.3 2017년 OWASP Top 10

응용프로그램 보안 위험의 중요성에 대한 인식을 기반으로 보안취약점 업데이트를 지속해 온 2017년 OWASP Top 10 버전은 2013년 이후 지속적으로 업데이트되어 왔으며 불안전한 소프트웨어는 국내 금융, 의료, 국방 등 주요 기반시설을 더욱 더 취약한 상태로 만들고 있다.

[표1] OWASF Top10(2013 버전과 2017버전 비교

OWASF Top 2013		OWASF Top 2017	
A1	인젝션	A1	인젝션
A2	인증 및 세션관리 취약점	A2	인증 및 세션관리 취약점
A3	크로스 사이트 스크립팅 (XSS)	A3	크로스 사이트 스크립팅 (XSS)
A4	취약한 직접 개체 참조	A4	취약한 접근제어
A5	보안설정 오류	A5	보안설정 오류
A6	민감데이터 노출	A6	민감데이터 노출
A7	기능 수준의 접근통제 누락	A7	공격 방어 취약점(신규)
A8	크로스사이트 요청변조 (CSRF)	A8	크로스사이트 요청변조 (CSRF)
A9	알려진 취약점 있는 컴포넌트 사용	A9	알려진 취약점 있는 컴포넌트 사용
A10	검증되지 않은 리다이렉트 포워드	A10	취약점 API(신규)

OWASF Top 10 2013년 버전 VS 2017년도 버전 변경사항을 고려해보면

1) 2013 A4(취약한 직접 개체 참조)와 2013 A7(기능수준의 접근통제 누락)을 2017 A4 (취약한 접근통제)로 통합하고

2) 2017 A7에 공격 방어 취약점을 신설하였다.

3) 2013 A10(검증되지 않은 리다이렉션과 포워드)가 예상과 달리 문제가 발생되지 않아 최종항목에서 삭제하고 보안강화를 위하여 2017 A10에 (취약한 API)를 신설하였다[2].

특히, 웹 서버, 웹 어플리케이션 서버 등 웹관련 SW 자체의 보안취약점을 이용한 공격은 침입차단시스템 등 보안장비로 대응하기가 쉽지 않다.

따라서 공공기관이나 기업에서는 관련 응용프로그램 보안위협을 신속하고 정확하게 대처해야 하며 사소한 보안문제를 간과해서는 안될 것이다.

Ⅲ. 관련 연구

3.1 CWE(Common Weakness Enumeration, 美NIST의 SW 일반적인 보안약점 목록)

미 국토 안보부 국가 사이버 보안부서(National Cyber Security Division)의 지원을 받는 MITRE(연구 개발 센터를 운영하는 비영리 회사)는 CWE 웹 사이트를 유지 관리하여 상위25개 소프트웨어 오류에 대한 자세한 설명과 오류에 대한 완화 및 회피에 대한 권위있는 지침을 제시하며, 악용가능한 취약점으로 이어질 수 있는 700 개 이상의 추가 소프트웨어 오류, 설계 오류 및 아키텍처 오류에 대한 데이터도 포함하고 있다[3].

최근에도 계속 업그레이드 되어 발표되고 있지만 소스코드의 버그 또는 보안 약점은 지속적으로 발생하고 있다.

3.2 CVE(Common Vulnerabilities Exposures, 美NIST의 SW 일반적인 보안취약점 노출)

공개된 사이버 보안 취약점에 대한 공통 식별자

목록으로 고유한 소프트웨어 취약점에 대한 정보를 토론하거나 공유 할 때 신뢰를 보장하고 도구 평가를 위한 기준을 제공하며, 사이버 보안 자동화를 위한 데이터 교환을 가능하게 한다[4][5].

최근 특정의 SW 보안취약점을 대상으로 지능화된 방법을 통해 지속적으로 공격하기 때문에 이에 대한 근본적인 대책이 필요하다.

3.3 개발시 시큐어 코딩 적용

시큐어 코딩(Secure Coding)은 소프트웨어 개발 과정에서 프로그램 입력값에 대한 검증누락 또는 데이터의 잘못된 형식지정 등으로 예측하지 못한 일이 발생되는 논리적 오류와 개발자의 실수로 인해 발생할 수 있는 보안취약점들을 최소화하여 보안성 향상을 목적으로 한다[6].

[표2] 신호처리모듈의 CERT C 시큐어 코딩 규칙

LDRA 코드	필수 규칙
LDRA 331S	unsigned형의 모든 상수에는 접미사 'U'를 사용하여야 한다. (Literal value requires a U suffix)

[표3] 접미사 U의 올바른 사용법

```
… u8a + 5       /*규칙 준수 되지 않음*/
… u8a + 5U      /*규칙 준수*/
… u9a + (unit8_t)5  /*규칙 준수*/
```

[표4] 신호 처리 모듈 예제

부적절한 코드

```
Unsigned char PaketDecoder(unsigned char ptnum)
{
Volatile unsigned char ReturnCmd;
ReturnCmd = 0;
…
}
```

해결 후

```
Unsigned char PaketDecoder(unsigned char ptnum)
{
Volatile unsigned char ReturnCmd;
ReturnCmd=0U;
…
}
```

[표3]은 바른 접미사 사용법에 대해 설명하고 있으며, [표4]를 통해 신호 처리 모듈에서 위반된 사례를 보여준다. 감시 시스템의 신호 처리 모듈에서는 unsigned형을 많이 사용하였지만 접미사 U를 사용한 곳은 없다.

그러므로 접미사를 사용하여 의도하지 않는 값이 나올 가능성을 사전에 차단하여 감시시스템에서 정확한 값을 전달하여 위험을 미리 방지하는 것이 목적이다[7].

3.4 네트워크 망분리에 대한 고찰

사이버공격에 의한 업무중단 사태를 방지하기 위하여 보안성이 우수한 업무망과 인터넷망의 **물리적인 망분리**를 추진하여 급증하는 정보보안 및 개인정보 유출 사고로부터 개인정보에 대한 보호기반 마련과 공공기관 및 기업의 정보보호 수준을 향상시킬 수 있으며, 해당기관 및 기업의 중요자료 유출의 근본적 차단을 통해 정보보안 보안성 강화를 도모해야 할 것이다.

또한 보안대책으로 사용자 PC의 최신 운영체제의 보안업데이트와 백신 프로그램 사용, 악성코드와 연결되는 IP주소차단 등을 강구해야 한다.

Ⅳ. 정보보호관리체계(ISMS)와 연계 검토

정보보호관리체계란 기업이 외부위협으로부터 자사의 정보자산을 보호하기 위해 수립·관리·운영하는 종합적인 체계로써 예상하지 못한 위기상황에서 비즈니스 안정성을 유지하고 정보자산을 적절하게 보호하기 위한 경영활동의 일부이다[8].

<표5> ISMS 인증 절차[8]

인증절차	내용
① 준비	ISMS 구축
	ISMS는 운영
	인증신청
② 심사	심사준비
	인증심사
	보완조치
	조치확인
	심사 결과보고서 작성
③ 인증	인증위원회 심의 준비
	인증위원회 심의 및 인증서 교부

최근 주요 기업에서 발생한 대부분의 해킹사고는 내부관리 소홀 및 정보보호에 대한 경영진의 관심 부족 등이 원인이라 할 수 있다[8].

따라서 소프트웨어 **보안관리 강화 해결방안**으로 기업의 정보자산 및 정보보호 향상을 위해서는 **정보보호관리체계(ISMS)시스템 구축시** <표5> ① **ISMS 준비단계**에서 소프트웨어 개발시 **시큐어 코딩으로 개발한다는 명시적 항목을 추가**해서 정보보호 관리체계(ISMS)를 구축해야 할 것이다.

V. 결론

본 논문에서는 최근 심각하게 대두되고 있는 애플리케이션 소프트웨어 보안영역에서 그 문제를 파악해 볼 수 있다.

20년전 아리안 로켓재난, 유럽항공우주국이 로켓을 발사한 후 수십 초 만에 파괴된 원인을 살펴보면 사소한 소스코드의 결함때문이었다.

공공기관 및 기업 등이 정보보호관리체계(ISMS) 구축시 ISMS 준비단계에서 소프트웨어의 취약점 및 보안약점을 소프트웨어 개발 시 시큐어 코딩을 적용 및 개발한다는 항목을 만들어서 체계적인 시스템을 구축한다고 해도 사이버범죄자들의 지속적인 위협 공격은 계속될 것이다.

또한 소프트웨어의 취약점이 개발자만의 문제가 아니라 주변 환경이 더 문제인 것 같다.

왜냐하면 보안업체 베라코드(Veracode)는 “많은 업체들이 애플리케이션 보안 문제 때문에 보안사고를 겪는다”고 한다. 그렇다면 무엇이 문제일까? “조직 전체적인 보안훈련을 제대로 하지 않는 것이 첫 번째 문제이고, 사업 운영 팀들이 개발 환경에서 취약점을 파악하고 처리하는 것에 큰 의의를 두지 않는 게 두 번째 문제입니다.”[9]

따라서 정보시스템 웹서버가 10년전에 구축한 환경이라면 정보시스템을 정보보호관리체계(ISMS) 구축시 시큐어코딩을 적용한 새로운 웹서버로 교체 구축한다면 대형사고 및 소프트웨어 결함의 위협으로부터 조금이나마 벗어날 수 있기 때문이다.

우리나라 대부분 기업의 CEO 나 임원 및 최고책임자의 지속적인 정보보안 교육의 중요성을 인식시키고 정보보호 및 정보보안은 그 조직을 상징하는 의미임과 동시에 그 조직의 생명이라는 생각을 갖게하여 효율적인 정보보안 강화를 해야 할 것이다.

이제는 급속하게 새로운 패러다임으로 변해가는 사물인터넷(Iot : Internet of Thing)시대에 정보보호 및 정보보안만이 회사의 총체적인 자원(가장 중요한 귀중품)이라는 인식을 갖고 위험관리와 정보보안은 내가 갖고있는 USB나 외장하드에 있는 개인정보만 중요한 것이 아니라 바로 조직 자체에 존재하고 있는 살아있는 개인정보보호와 정보보안에 가장 중요한 역할을 할 때 라고 사료된다.

[참고문헌]

[1]로버트 C, 현동성 역. 시코드 저,“버그 없는 안전환 소프트웨어를 위한 CERT C프로그래밍,” 에이콘, pp.26, 2010

[2]http://www.certit.kr/?mod=document&uid=97&page_id=249

[3]https://cwe.mitre.org/about/index.html

[4]https://cve.mitre.org/about/

[5]Gartner, "Now is the time for security at Application Level", 2006, 12.

[6] Junesung choi, Wooje Kim, Wonhyung Park, Kwangho Kook, Defense SW Secure Coding Application Method for Cyberwarfare Focused on the warfare System Embedded SW Application Level, Journal of Korea Association of Defense Industry Studies, vol. 19, no. 2, pp. 91-103, 2012

[7]Herbert Schildt 저, 유해영/우진운 역, “알기 쉽게 해설한 C” Second Edition, 이한 출판사,p150, July 30, 2003.

[8]KISA ISMS Certification Scheme guide p 11, p 24, 2016.3.

[9]http://www.boannews.com/media/view.asp?idx=57625&page=1&kind=1&search=title&find=%BE%D6%C7%C3%B8%AE%C4%C9%C0%CC%BC%C7+%BA%B8%BE%C8+%B9%AE%C1%A6%2C+%C7%D8%B0%E1%B5%C9+%B1%E2%B9%CC

국가사이버안보법안 타당성에 관한 연구: 사이버위협정보 공유를 중심으로

정해원*, 김예준**, 채진희**, 이현***, 유정인****, 장원희*****

*경찰대학교 행정학과, **순천향대학교 정보보호학과, ***선린인터넷고등학교,
****가톨릭대학교 미디어기술콘텐츠학부, *****서울호서전문학교 사이버해킹보안과

A Study on the Feasibility of the National Cyber Security Law: Focusing on Cyber Threat Information Sharing

Jin-Hee Chae*, Dong-Bin Oh**, Hae-Won Jeong**, Ye-Jun Kim*, Hyun Yi***, Won-Hee Jang****, Jung-In Yoo*****

*Dept of Administration, Korean National Police University.

**Dept of Information Security, SoonChunHyang University.

***SunRin Internet High School.

****Dept of Media Technology Contents, Catholic University of Korea.

*****Dept of cyber hacking information security, Seoul Hoseo Occupational Training College.

요 약

오늘날 사이버위협은 국가안보 영역에서 비전통적 안보위협요인 중 가장 심각한 것으로 인식되고 있다. 전 세계 주요 국가들은 민간과 정부의 영역을 구별하지 않는 사이버공격으로 인해 심각한 국가적 피해를 입고 있다[1]. 이러한 사이버위협에 대응하기 위해 최근 유관집단 간 협력이 중요시되고 있다[4]. 우리나라는 19대 국회부터 '사이버위협정보 공유에 관한 법률안', '국가사이버안보에 관한 법률안', '국가사이버안보법안' 을 발의하고 사이버위협정보 공유센터를 구축하는 등 민•관 사이버위협정보 교류를 위한 국가차원의 노력을 이어왔다. 하지만 실질적인 정보의 공유는 이루어지지 않았고 여전히 사이버위협정보는 파편화되어 효과적으로 쓰이지 못하고 있다. 사이버위협정보 공유의 필요성에 정부와 학계는 모두 공감하고 있으나 국가사이버안보법안에 관해서는 여전히 상반된 시각이 존재한다. 본 논문은 이와 같은 국가사이버안보법안에 대한 찬반 의견을 최근까지의 관련 법률안을 바탕으로 비판적으로 살펴보며 개선방향을 제언한다.

I. 서론

1.1 사이버위협 현황

오늘날 사이버위협은 국가안보 영역에서 비전통적 안보위협요인 중 가장 심각한 것으로 인식되고 있다. 이러한 사이버위협은 사경제 영역은 물론 공공 영역까지 무차별적으로 공격대상으로 하고 있다. 에너지, 수자원, 교통 등 대다수 국민의 일상생활과 밀접한 관련이 있고 국가적인 중요성을 지닌 주요 국가기반시설이 치명적인 사이버공격을 받을 경우 사회적 혼란은 물론 막대한 피해가 우려되는 상황이다[1]. 더불어 해커, 핵티비스트, 테러리스트 등과 같은 이른바 비국가 행위자들뿐만 아니라 국가차원의 지원을 받는 집단들도 등장하여 사이버안보의 문제가 전통안보의 문제만큼이나 중요한

국가안보의 사안으로 주목받고 있다[2].

전 세계 주요 국가들은 최근 민간과 정부의 영역을 구별하지 않는 사이버공격으로 인해 심각한 국가적 피해를 입었다. 미국에서는 물리적 테러위협을 동반한 해킹 및 기밀정보유출 사건인 소니 해킹사건이 발생했다. NATO는 민간에 직접적인 불편과 피해를 발생시킨 에스토니아 해킹사건 및 러시아로부터 물리적인 군사작전과 결합하여 사이버공격 피해를 입은 그루지야 사건을 경험하였다[3]. 특히 북한의 대남 사이버 공격을 지속적으로 받고 있는 우리나라는 2009년 7.7 디도스 사건, 2013년 3.20 사이버테러 사건, 2014년 한국수력원자력 해킹 사건, 2016년 국방망 중추신경인 국방데이터통합센터 해킹 사건 등 상당한 사회적 혼란과 피해를 초래한 사건을 경험하였다. 최근에는 사드 배치 보복과 관련한 중국발 보복성 해킹 우려까지 직면하여 위협행위 주체의 다양성과 그 피해의 범위가 날로 확대되고 있는 실정이다[2].

1.2 사이버위협정보 공유의 필요성

심각해지는 사이버 공격으로부터 대응하기 위해 유관집단 간 협력이 중요시 되고 있다. 최근 사이버위협은 복수의 대상을 목표로 하는 동시다발적 공격의 형태를 취하고 있는데, 이는 기업 및 기관이 단독으로 피해를 줄이기 쉽지 않기 때문이다. 더불어 한번 사용된 신종 공격기법은 향후에 다시 쓰이는 경우가 많다. 따라서 유사한 공격 기법에 의한 피해 사례를 줄이기 위해, 사이버위협정보 공유가 주목을 받고 있다. 특히 학계에서는 국가안보문제를 해결하기 위한 업계와 정부의 다양한 수준 간 정보공유 메커니즘의 필요성을 강조한다. 사이버 보안정보를 공유하는 것은 공유하지 않았을 때보다 저비용을 보안 수준을 높일 수 있으며 공유 주체의 범위가 확대될수록 더 가치가 있다는 사실은 여러 연구를 통해 검증되었다[4].

정부의 사이버위협 정보를 공유하기 위한 제도적 논의는 2011년부터 행자부•미래부 등이 각 기관별로 사이버위협정보 공유센터를 구축하는 등 활발하게 제기되어 왔다. 그러나 개별적이고 단편적인 시스템 구축으로 실질적 개선은 아직 미진한 상황이다. 통합된 운영체계나 구축되지 않았고, 참여기관의 의지가 부족하며, 인센티브 부여 등 유인책이 미흡하고, 프라이버시 보장 등 개인정보 보호와 안보•국익과의 법익 고려 등 현실적인 문제가 해소되지 않는 등 미비점이 지적되고 있다[5]. 한편으로 지난 19대 국회부터 지금까지 사이버안보 체계를 개선하고자 하는 법률안들이 다수 발의되었다. 법안들은 주로 현행법상 공공과 민간을 포괄할 수 있는 사이버안보법이 존재하지 않는다는 문제의식에 입각해 있으며, 또한 이 법안들은 사이버 위협정보 공유의 법적 강제에 관한 내용을 담고 있다[2].

본 논문에서는 19대 국회에서 발의된 '사이버위협정보 공유에 관한 법률안'(이철우 의원 대표발의), 20대 국회에서 발의된 '국가 사이버안보에 관한 법률안'(이철우 의원 대표발의), '국가사이버안보법안'(정부 제출)의 주요내용을 소개하고 사이버 위협정보 공유관련 법안 타당성에 대해 비판적으로 살펴보고 개선방향을 제언할 것이다.

II. 법안의 주요내용

2.1 현행 사이버안보 법체계

법안의 구체적인 내용에 들어가기에 앞서 우리나라의 사이버안보법 체계를 살펴볼 필요가 있다. 우리나라 사이버안보법 체계는 크게 공공과 민간으로 나뉜다. 대통령훈령인 「국가사이버안전관리규정」을 중심으로 한 공공부문 규율(국가정보원 총괄)이 존재하며, 「정보통신망 이용촉진 및 정보보호 등에 관한 법률(이하 '정보통신망법')」을 중심으로 한 민간부문 규율(미래창조과학부 총괄)이 나머지 한 축을 차지한다. 이는 법리적 측면에서 자율이 우선되어야 할 민간부문과 보다 강화된 정부 개입의 영역인 공공부문을 구분하기 위함이라 이해된다. 더불어 공공과 민간을 불문하고 주요정보통신기반시설에 대해서는 「정보통신기반보호법」을 근거로 강화된 정보보안 규율을 적용하고 있다. 「정보통신기반보호법」의 경우에는 전자적 침해

행위에 대한 주요정보통신기반시설의 보호를 법의 목적에서 규정하고 있는데 비해, 「국가사이버안전관리규정」의 경우에는 사이버 공격에 대한 국가정보통신망을 보호한다는 구체적인 내용을 담고 있어 서로 상이한 보호대상 및 적용범위를 규정한다[6].

「국가사이버안전관리규정」은 중앙행정기관의 장, 지방자치단체의 장 및 공공기관의 장이 국가정보통신망에 대한 사이버 공격의 계획 또는 공격사실, 사이버안전에 위협을 초래할 수 있는 정보를 입수한 경우에는 지체 없이 그 사실을 국가안보실장 및 국가정보원장에게 통보하도록 정하고 있다(제10조). 「정보통신기반보호법」은 금융·통신 등 분야별 정보통신기반시설을 보호하기 위한 정보공유·분석센터를 구축하는 것을 장려하는 차원에서 재정적·기술적 지원을 할 수 있다고 정하였다. 정보공유·분석센터는 취약점 및 침해요인과 그 대응방안에 관한 정보를 제공하거나, 침해사고가 발생하는 경우 실시간 경보·분석체계를 운영하는 데 사용된다(제16조).

	관할 영역	관할부처
정보통신망법	민간 정보통신망 일반	미래창조과학부
정보통신기반보호법	민간 기반시설	미래창조과학부
	공공 기반시설	국가정보원
국가사이버안전관리규정	공공 정보통신망	국가정보원

[표 1] 현행 사이버안보 법체계

2.2 관련 법률안의 주요내용

2.2.1 사이버위협정보 공유에 관한 법률안 (이철우 의원 대표발의)

본 법안은 국정원장이 국가안보실장, 미래창조과학부장관 등과 협의하여 범정부 차원에서 사이버위협정보를 공유하기 위한 방법과 절차를 마련하도록 한다(안 제4조). 국가의 주요 정보와 정보통신망을 관리하는 기관(이해 "사이버위협정보 공유기관")은 사이버위협정보를 수집하고 상호 공유해야 한다고 정한다(안 제4조). 사이버위협정보 공유기관의 범위는 다음의 각 호와 같이 정하였다. 「대한민국헌법」, 「정부조직법」 및 그 밖의 법령에 따라 설치된 국가기관과 지방자치단체 및 「국가정보화 기본법」 제3조 제10호에 따른 공공기관(제2항 1호), 「정보통신기반 보호법」 제5조제1항에 따른 주요정보통신기반시설을 관리하는 기관(제2항 2호), 「정보통신망 이용촉진 및 정보보호 등에 관한 법률」 제46조 제1항에 따른 집적정보통신시설 사업자 및 같은 법 제47조의4 제2항에 따른 주요정보통신서비스 제공자(제2항 3호), 「전자금융거래법」 제2조제21호에 따른 전자금융기반시설을 운영하는 기관이나 단체 또는 사업자(제2항 4호), 「산업기술의 유출방지 및 보호에 관한 법률」 제9조에 따른 국가 핵심기술을 보유한 기업체나 연구기관(제2항 5호). 사이버위협정보 공유를 효율적으로 수행하기 위하여 국정원장 소속으로 사이버위협정보 공유센터가 설치·운영된다(안 제5조). 국정원장은 법무부장관 등 국가기관 및 전문가가 참여하는 협의회를 구성하여 사이버위협 정보의 남용방지 대책을 수립하여야 한다(안 제7조). 법률안에서 정한 사이버위협 정보 공유기관 이외에 사이버위협정보를 보유한 사람은 공유센터의 장에게 신고하거나, 공유센터의 장이 해당자에게 사이버위협정보의 제공을 요청할 수 있게 하였다(안 제8조). 동법안은 2015.5.19. 제안되었고 위원회 심사를 거쳤으나 2016.5.29. 임기만료 폐기되었다.

2.2.2 국가 사이버안보에 관한 법률안 (이철우 의원 대표발의)

본 법안은 사이버안보에 관한 중요한 사항을 심의하기 위하여 대통령 소속하에 국가사이버안보정책조정회의를 두었다(안 제4조). 그리고 국가차원의 종합적이고 체계적인 사이버안보 업무 수행을 위하여 국가정보원장 소속으로 국가사이버안보센터 또한 두고 있다(안 제6조). 국가정보원장은 사이버안보업무의 효율적이고 체계적인 추진을 위하여 사이버안보 기본계획을 수립하고 이에 따라 시행계획을 작성하여 책임기관의 장에게 배포해야 한다(안 제7조). 책임기관의 장은 사이버공격 정보를 탐지·분석하여 즉시 대응할 수 있는 보안관제센터를

구축•운영하거나 다른 기관이 구축•운영하는 보안관제센터에 그 업무를 위탁하여야 한다(안 제10조). 또한 책임기관의 장은 사이버위협정보를 다른 책임기관의 장 및 국가정보원장에게 제공하여야 하며 국가정보원장은 국가차원의 사이버위협정보의 효율적인 공유 및 관리를 위하여 국가사이버위협정보공유센터를 구축•운영할 수 있도록 하였다(안 제11조). 책임기관의 장은 사이버공격으로 인한 사고가 발생한 때에는 신속히 사고조사를 실시하고 그 결과를 중앙 행정기관 등의 장 및 국가정보원장에 통보하도록 하는 근거를 마련해 두었다(안 제12조). 본 법안은 2016.5.30. 제안되었고 위원회 심사 중에 있다.

2.2.3 국가사이버안보법안 (정부 제출)

본 법안은 사이버안보와 관련된 국가의 정책 및 전략 수립에 관한 사항 등을 심의하기 위하여 대통령 소속으로 국가사이버안보위원회를 두되, 위원장은 국가안보실장으로, 위원은 국회•법원•헌법재판소•중앙선거관리위원회의 행정사무를 처리하는 기관 및 중앙행정기관의 차관급 공무원 중 대통령령으로 정하는 사람과 사이버안보에 관하여 전문적인 지식과 경험을 갖춘 사람 중에서 국가안보실장이 임명하거나 위촉하도록 하였다(안 제5조). 국가기관•지방자치단체 및 국가적으로 중요한 기술을 보유•관리하는 기관 등은 '책임기관'으로서 소관 사이버공간을 안전하게 보호하는 책임을 지도록 하고, 국가정보원장은 책임기관을 지원하기 위한 기술적 역량이 있는 기관 또는 단체를 지원기관으로 지정할 수 있도록 하였다(안 제6조, 제7조). 사이버위협정보의 공유와 관련해서는 국가정보원장 소속으로 사이버위협정보공유센터를 두고, 책임기관의 장은 소관 사이버위협정보를 사이버위협정보 공유센터의 장에게 제공하도록 하며, 사이버위협정보 공유센터의 장은 위협정보를 공유하는 경우 국민의 권리가 침해되지 아니하도록 기술적•관리적 및 물리적 보호조치를 마련하도록 하였다(안 제12조).

III. 국가사이버안보법 제정을 둘러싼 찬반론

3.1 입법 필요성을 긍정하는 입장

일반적으로 제기되어 온 입법 필요성은 다음과 같다. 즉, 우리의 국가적 대응 활동은 공공・민간 부문이 제각각 분리, 독립 대응하고 있어 광범위한 사이버공격에 효율적인 대처가 어렵고, 공공부문은 대통령훈령인 「국가사이버안전관리규정」에 근거하고 있어, 행정기관 이외 민간분야 및 입법・사법기관은 적용범위에서 제외되고, 민간 부문은 사이버공격 예방 및 대응을 위한 법률 미흡으로 사이버공격 징후를 실시간 탐지・차단하거나 신속한 사고 대응에 한계가 있다는 것이다. 주요정보통신기반시설에 대해서는 주요정보통신기반시설법에 의거 구무총리실장에게 권한을 부여하고, 그 외의 공공기관이 운영하는 정보통신망에 대해서는 훈령에 따라 적용하므로 법적 부조화가 발생하고 있다[10]. 이에 정부와 민간이 함께 협력하여 국가차원의 체계적이고 일원화된 대응 체계를 구축하고, 이를 통해 사이버공격을 사전에 탐지하여 사이버위기 발생가능성을 조기에 차단하며, 위기 발생 시 국가의 역량을 결집하여 신속히 대응 할 수 있도록 하기 위한 입법이 필요하다는 주장이다. 특히 최근 안보영역에서 사이버안보가 가지는 특수성과 중요성을 반증하는 주요국의 강화된 사이버안보 입법례를 거론하며 북한발 사이버공격에 상시적으로 노출되어 있고, 높은 수준의 정보통신기반 시설을 갖춘 우리나라에게 사이버안보를 대응하기 위한 공공과 민간을 모두 아우르는 입법적 기반이 필요함을 역설한다[2].

한국은 '3.20 사이버테러'와 '6.25 사이버공격'을 계기로 16개의 관계부처 합동으로 수립한 「 국가 사이버안보 종합대책」의 일환으로 "사이버위협 정보 분석 • 공유시스템(C-TAS)"과 범정부 로그분석시스템(nSIMS)"을 구축했다. 이 시스템들은 각가 한국인터넷진흥원과 정부통합전산센터에서 운영되었다. 엄청난 양의 데이터를 수집하여 분석하는 nSIMS는 빅데이터

활용의 우수사례로 뽑히며 많은 기대를 모았지만 시스템 구축 완료 후 같은 해 발생한 한국수력원자력에 대한 사이버공격 발생 시 아무런 효과도 발휘하지 못한 것 아니냐는 비판이 있었다[10]. 사이버안보법 입법을 찬성하는 측은 이러한 시스템의 실패가 민간부문의 정보공유 의무 공백으로 인한 것이라 주장하며 보다 포괄적인 사이버안보 대응체계 및 법률의 필요성을 강조한다[7].

3.2 입법 필요성을 부정하는 입장

입법 필요성을 부정하는 측은 법체계적인 관점에서 현행 법제로도 사이버안보 대응을 위한 제도적 근거로 부족함이 없다는 점을 지적한다. 현재 국내에는 민간 영역의 사이버보안을 규율하는 「정보통신망이용촉진 및 정보보호 등에 관한 법률」, 국가기반시설의 사이버 보안을 위한 「정보통신기반 보호법」 및 「전자금융거래법」, 「국가정보화기본법」이 존재한다. 이들 법안의 주된 내용은 공공 영역의 사이버 보안을 규율하는 국가사이버안전관리규정(대통령훈령)에 있는 사항과 크게 다르지 않아 입법적인 실효성이 없다는 것이다. 기존 법률을 통해 이미 수행되고 있는 사항을 중복규정 할 필요성이 없고, 기존 법제의 개념 정의조차 통일되어 있지 않은 상황에서 법률을 새로 제정하는 것은 중복으로 인한 비효율과 혼란을 가중시킬 뿐이라고 지적한다.

보다 핵심적인 반대 측의 논거는 국정원의 사이버보안 권한 강화로 인한 민간 사찰과 감시확대이다. 법안상 사이버안보는 국가안보보다 넓은 개념으로 사이버분야의 안보를 명분으로 하여 국정원이 민간에 대한 감시권한을 확대하는 것이며4, 사이버안보는 국정원법 제3조에서 규정하고 있는 국정원의 직무범위를 벗어난 것이기 때문에 허용될 수 없다는 입장이다. 따라서 국가정보원이 동법안에 따라 1차적인 권한을 보유하는 것이 타당한가에 대한 의문, 아울러 국정원 컨트롤타워에 대한 견제구조(정부내 타 사이버공격 대응기관, 국회, 법원)가 미흡하다는 점, '사이버위협정보 공유센터'를 국가정보원 소속으로 하는 것은 적절하지 않다는 지적이 제기된다. 특히 이철우 의원안을 포함한 이전의 법안들이 사이버위협 정보공유를 의무화하고 있었는데, 이를 통하여 정보기관이 민간에 대한 감시권한을 확대하는 수단으로 이용될 수 있다는 우려도 제기한다[2].

IV. 합리적 논의를 위한 제언

국가사이버안보법 제정 찬반양측 모두 민관 사이버위협 정보공유의 필요성에 대해서는 공감하고 있다. 정보공유의 주체는 크게 정보기관, 수사기관, 정책부처, 정보보안업체, 기업, 시민 등으로 볼 수 있는데, 그간 정보기관은 국가안보라는 이유로, 수사기관은 수사비밀이라는 이유로 정보공유에 소극적이었다. 정책부처는 피해정보는 충분히 확보할 수 있었으나 범죄자 정보는 부족했다. 정보보안업체는 고객의 피해사실을 공유할 수 없었을 것이고, 피해기업은 경제적 손실, 법적 책임추궁, 브랜드가치 하락 등을 이유로 정보공유에 방어적 태도로 대응하였다. 각자의 여건에 의해 파편화된 정보는 효과적이지 못하였고 전체적인 대응을 어렵게 해왔다[11].

자발적인 형태의 민•관 사이버위협정보의 교류가 가장 이상적이지만, 실제로 자발적인 공유의 형태는 일어나기 쉽지 않고 정보공유의 시대적 요구가 크기 때문에 법률로써 의무를 규정하고자 하는 주장이 힘을 얻게 되었다. 생각건대 법률로써 공유의 목적, 주체, 대상, 범위 등 세부사항을 구체적으로 정하여야 관련자 모두의 권익을 보호할 수 있기에 법률을 통해 사이버위협정보 공유의무를 규정하는 방향은 옳다. 사이버안보와 관련하여 제기된 일련의 입법적 시도들을 둘러싼 논의의 주된 쟁점은 공유시스템 담당기관과 입법의 방향이다.

4.1 사이버위협정보 공유체계 담당기관

공유체계 주관부서와 관련하여 사이버안보 추진체계의 정책총괄기능과 실무총괄기능의 구성방법이 주된 논점이다. 정책총괄기능과 실무총괄기능을 하나의 기관에 집중시키는 형태는

기능의 직접에 따른 효율성이 높은 반면에 권한이 집중되어 제어가 사실상 불가능한 상황이 우려된다. 두 기능의 담당기관을 분리시키는 형태라면 정책총괄기능을 국무총리실보다는 대통령실이 담당하도록 하여 컨트롤 타워의 위상을 승격하고 부처 간 업무 조정 및 통제를 용이하게 하는 것이 상당한 타당성을 지닌다고 할 수 있다[8].

정책총괄기능과 실무총괄기능의 분리여부를 막론하고 실무총괄기능은 반드시 기술적 역량을 갖춘 기관이 수행하여야 한다. 따라서 청와대가 실무총괄기능을 담당하기에는 현실적으로 무리가 있다고 본다. 한편 단지 기술적 역량의 보유에 그치는 것이 아니라 다른 유관기관과의 협력을 능동적으로 주도할 수 있는 지위에 있고 안보 문제를 적절히 다룰 수 있는 기관이 적합하다고 할 수 있다[8]. 이와 관련하여 경찰청을 실무총괄기능을 담당하도록 하는 바를 제언한다. 경찰청은 사이버침해에 의한 일반범죄의 예방 및 수사를 위해 1995년 해커수사대의 발족을 시작으로 1997년 컴퓨터범죄수사대, 1999년 사이버범죄수사대, 2000년 8얼 사이버테러 대응센터(CTRC)에 이르기까지 사이버범죄 수사조직을 이끌어왔다. 현재 사이버테러 대응센터는 「경찰청과 그 소속기관 직제 시행규칙」 제9조 제1항에 의해 수사국에 설치되며, 사이버테러의 탐지·추적수사 및 경보 등 조치, 사이버테러관련 수사기법의 연구·개발 및 국제경찰기구 등과의 협력, 사이버범죄의 수사 및 지도 및 디지털매체 등 증거분석 업무 등의 사무를 수행하고 있다[9]. 경찰청을 충분한 기술적 역량, 능동적 유관기관 협력관계, 안보사안 대응경험을 바탕으로 사이버위협 정보공유의 실무담당기관으로 적합하다.

4.2 입법의 방향

사이버위협정보 공유의무를 정하는 방법은 기존 ‘정보통신망법’ 등의 관련 법안에 규정을 삽입하는 것과 새로운 입법을 통하여 공유의 의무를 강조하는 방법이 있다. 현재 ‘정보통신망법’은 정보통신망, 기반시설 등의 보호를 위한 정부 및 사업자의 사후 조치를 중점으로 규정되어 있어, 일반 이용자 컴퓨터 자체의 보안 허점을 이용한 공격에는 법 적용상 한계가 있는 실정이다. 따라서 ‘정보통신망법’에 사이버위협정보 관련 규정을 삽입하는 경우, 이와 동시에 기존 법체계의 방향을 포괄적으로 개정해야 한다[11].

‘사이버위협정보 공유에 관한 법률안’, ‘국가사이버안보법’ 등의 입법적 시도는 모두 새로운 법률로 정보교류의 의무를 규정하는 방법의 예시이다. 다만 국가사이버안보법의 경우 제정의 목적이 사이버위협정보 공유에만 그치지 않기 때문에 공유의 의무를 강조하는 의미가 퇴색된다. 사이버위협정보 공유체계의 필요성은 이미 전세계적으로 크게 강조되고 있으며, 국가사이버안보법 내의 타 규정들에 관한 논란으로 인하여 사이버위협정보 공유의무를 정하는 법이 늦게 제정되어서는 안 될 것이다. 국회는 ‘사이버위협정보 공유에 관한 법률안’과 같은 구체적 입법을 통하여 사회적 요구에 신속히 부응해야 한다.

V. 결론

국가사이버안보법안에 관해 여전히 상반된 시각이 존재하지만, 사이버위협정보 공유의 필요성은 모두가 인식하고 있다. 19대 국회부터 관련 법률안이 여럿 논의되었으나 실현되지 못했다. 최근 전세계 국가들에 대한 사이버공격으로 인한 상시적인 안보위협은 커져가고 있고, 특히 우리나라는 북한의 지난 사이버공격을 고려할 때 하루속히 관련 법제도를 확보해야 한다. 사이버위협정보 공유의 의무를 규정하는 법률은 기존 법체계와의 조응을 고려하여 법적 공백없이 치밀하게 제정되어야 한다. 공유체계 정책총괄 담당기관은 기존 학계의 의견과 마찬가지로 청와대로 하되, 실무총괄기능의 경우 국가정보원에서 경찰청을 전환하는 것도 전향적으로 검토할 필요가 있다. 민 • 관이 협력하여 사이버위협에 대응하고자 하는 논의가 국회, 정부기관, 기업 등 관련 이해관계자 모두를 중심으로 다시금 활성화되기를 바라며, 이를 통해 새로운 사이버위협정보 공유체계를 구현해 다

가오는 사이버위협에 우리나라가 효과적으로 대응할 수 있기를 기대한다.

[참고문헌]

[1] 박상돈, & 김인중. (2013). “사이버안보 추진체계의 제도적개선과제 연구”. *융합보안논문지, 13*(4), 3-10.

[2] 김도승. (2017). “국가사이버안보법 제정 논란의 시시비비”. *한국인터넷자율정책기구저널, 26*, 10-25.

[3] 정준현. (2016). “북한의 사이버공격에 대한 국가총력적 대응체계를 위한 법제방향”. *성균관법학, 28*, 23-51.

[4] 김하영. (2017). “*국내 사이버위협 정보공유의 활성화에 영향을 미치는 요인*”. 석사학위논문, 충북대학교 일반대학원, 충청북도.

[5] 윤오준. (2017). “*사이버위협 정보공유 시스템 확산에 영향을 미치는 핵심요인에 관한 연구*”. 박사학위논문, 숭실대학교 대학원, 서울.

[6] 이연수, 이수연, 윤석구, & 전재성. (2008). “주요국의 사이버 안전관련법 조직체계 비교 및 발전방안 연구”. *국가정보연구, 1*(2), 35-116.

[7] 김지선. (2016). “*한국과 미국의 사이버위협 정보 공유 입법안 비교연구*”. 석사학위논문, 고려대학교 대학원, 서울.

[8] 박상돈, & 김소정. (2013). “사이버안보법 제정을 위한 국내 사이버안보 법률안 연구”. *융합보안논문지, 13*(6), 91-98.

[9] 강수진. (2013). “*국가 사이버범죄 대응전략 설계*”. 고려대학교 산학협력단. 경찰청.

[10] 권오국, & 석재왕. (2016). “주요국의 사이버테러 대응체계와 시사점 분석: 미국・영국・독일 사례의 비교를 중심으로”. *한국경호경비학회지, 49*, 185-214.

[11] 최정호. (2015). “*사이버 위협정보 활용방안 연구*”. 한국해양대학교 산학협력단. 경찰청.

IOT 기반의 블록체인패러다임과 보안

송호령*, 성종현*, 김창훈**

*대구대학교 정보보호영재교육원, **대구대학교 지도교수

IOT based block chain paradigms and security

Jong-Hyeon Sung*, Ho-Ryeong Song*

Daegu University

*Daegu University Information security institute for the gifted

**Daegu University tutor

요 약

IoT(사물인터넷)는 각종 사물에 센서와 통신 기능을 내장해 인터넷에 연결하는 기술을 뜻한다. 최근 이 IoT 기술을 적용한 제품들은 다양하게 개발, 제작되어지고 있다. 하지만 현재 다수의 IoT 제품들이 해킹의 위험에 노출되어있고 그에 따른 침해사고들이 일어나고 있다. 본 논문에서는 IoT 보안의 방안으로 블록체인 인증 프로토콜과 함께 그에 따른 IoT 분산형 컴퓨팅 플랫폼의 적용을 제안한다.

I. 서론

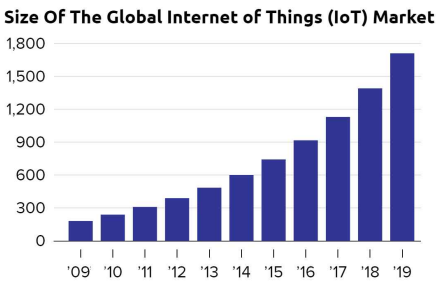


그림 1. IoT시장 성장 동향(출처 ; HKExnews)

IoT 산업은 4차 산업 혁명과 함께 아주 빠른 속도로 성장하고 있다. 그림 1에서 볼 수 있듯이 2017년 현재까지 기하급수적인 성장이 이뤄지고 있었으며, 2019년까지 가파른 성장을 할 것으로 예측된다.

IoT 기기의 예로는 우리가 알고 있는 스마트 가전제품부터 신체의 정보를 실시간으로 모니터링 할 수 있는 웨어러블 기기, 자율 주행 자동차 거기에 최근에는 의료 쪽에서 환자의 치료 목적으로까지 사용되고 있다. 스마트폰 하나로 설정부터 조작, 모니터링을 할 수 있을 정도로 접근성이 매우 높다. 이러한 높은 접근성은 한편으로는 편리할 수 있지만, 보안 체계를 제대로 갖추지 못한 높은 접근성은 오히려 해커들의 표적이 될 수 있다. 기존의 IT 분야에 있는 보안 취약점들이 IoT 환경에 그대로 존재한다.

Worm과 Virus, Dos 및 DDos 공격, 비인가된 사용자의 접근, 기밀성/무결성 공격 등으로 IoT 디바이스와 통신/네트워크가 대상이 될 수 있다. 기존의 존재하지 않던 새로운 취약점들이 발생할 수도 있으므로 IoT가 상용화되기 위해서는 강력한 보안이 반드시 필요하다. 이 글에서는 취약한 IoT 보안의 대안으로 블록체인을 기반으로 분산 컴퓨팅을 이용한 클라우드 플랫폼 도입 시 한계점과 해당 문제를 해결할 분산 플랫폼을 제안할 것이다.

II.본론

[1][2][3] 최근에 차세대 보안 체계로 주목받고 있는 블록체인은 가상화폐인 비트코인에 처음 사용되었다. 블록체인 기술을 이용한 비트코인은 거래를 중개하는 중개 기관이 존재하는 기존의 금융거래 서비스와는 달리 네트워크 내 참여자들이 거래의 무결성, 유효성을 검사한다.

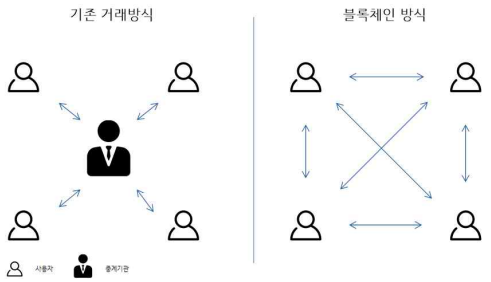


그림 2. 기존거래방식과 블록체인 방식 비교

이 검증을 하는 하기 위해서는 블록이라는 것이 필요하다. 블록은 일정한 시간 동안의 거래 내역들을 담고 있다. 그리고 블록은 이 앞에 생성되었던 블록들을 검증하면서 체인과 같은 구조를 이루게 되는데 이를 블록체인이라고 부른다.

이러한 블록체인의 거래구조는 거래를 조작하기 위해서는 네트워크 참여자들의 블록을 과반수 그 이상 조작해야 하기 때문에 사실상 해킹이 불가능에 가까운 보안체계라고 할 수 있다. 그래서 비트코인과 같은 가상화폐에 이용되고 있는데 본 연구에서는 코인이 아닌 인증서로서의 역할로 이용하는 것이다.

이러한 블록체인을 IoT 보안에 적용하기 위해서는 인증서 발급을 위한 채굴작업에서 요구하는 컴퓨팅 자원을 충족할 필요가 있다. IoT 기기들은 대부분 컴퓨팅 자원을 데이터 수집, 분석 및 학습을 하는 데 사용되기 때문에 인증을 위해 개별 IoT 기기들로 채굴작업을 하는 것은 각 기기에 부담으로 작용하는 한계가 발생한다.

그에 따른 해결방법으로 클라우드 컴퓨팅 기반 플랫폼을 구축해서 채굴작업을 할 때 필요한 컴퓨팅 자원을 Faas나 Saas 같은 클라우드 서비스를 통하여 지원받을 수 있다. 이 경우 문제없이 해당 기가에서 인증서 발급을 위한 채굴작업을 할 수 있다.[4]

하지만 클라우드 컴퓨팅을 사용할 경우에는 초기 인프라 구축에 필요한 비용도 문제지만 IoT기기가 증가할 시 스케일 업을 해야하며 이에 따른 비용이 증가된다. Faas가 최소한의 로드를 담당하더라도 해당 플랫폼에 수많은 기기의 로드가 몰릴 가능성이 있으며 로드밸런싱에서 추가적인 비용이 소요된다.

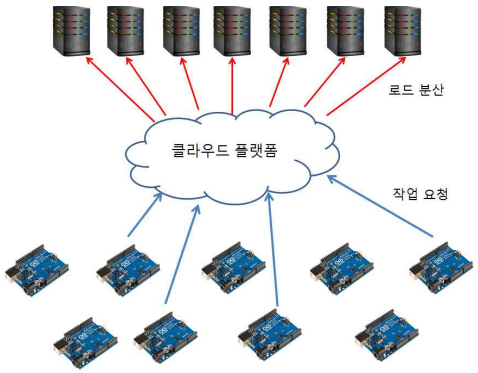


그림 3. 클라우드 플랫폼을 이용한 블록체인 인증작업

그러므로 중앙식 클라우드 플랫폼 도입의 한계점들을 해결하기 위해 본 연구에서는 분산형 플랫폼 도입을 제안한다. 분산형 플랫폼은 1)플랫폼 엔진에 연산을 요청하면 2) 엔진의 로드벨런서는 최대한 많은 기기에 연산요청을 보내서 3)IoT 기기 중 프로세서의 가용성이 크며, 적재된 작업량이 적은 디바이스를 다수 확보한다. 4)그 이후 다수 확보된 디바이스를 이용하여 분산컴퓨팅으로 채굴작업을 진행한다. 5) 채굴 연산에 필요한 컴퓨팅 자원을 작업을 하지 않는 남는 IoT기기를 이용하므로 별도의 인프라 비용이 적으며. 확장성이 뛰어나다.[5]

하지만 이 또한 해결해야 할 점들이 존재한다. 초기 IoT기기들의 개수가 적을 때, 로드가 특정기기에 집중될 수 있다는 것이며, 분산 컴퓨팅 시, 연산을 통해 키를 얻는 기기가 사용자가 원한 기기가 아닌 불특정 기기이며 이때 키에 대한 복제가능성이 존재한다.

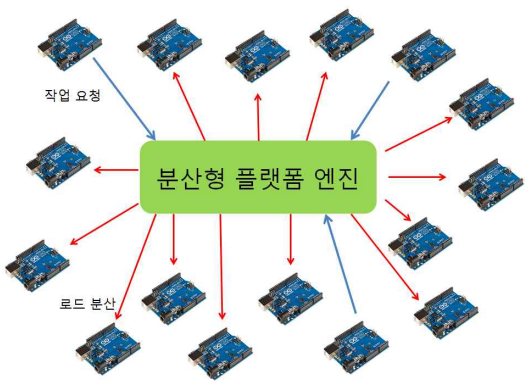


그림 4. 분산형 플랫폼을 이용한 블록체인 인증작업

III. 결론

분산형 컴퓨팅 플랫폼의 한계 중 IoT기기 제한 문제는 시장이 확장될수록 해결이 가능하다. 또한 두 번째 문제의 해결방법은 플랫폼에서 기기의 투명성과 은닉성을 보장하는 정책을 도입하는 것이다. 연산을 요청받은 기기는 어떠한 기기가 요청한 지 알 수 없게 하고, 요청한 기기는 어떠한 기기가 연산을 통하여 키를 발급한지 모르게 하는 것이다. 또한 플랫폼에서 각 연산환경은 컨테이너화를 하여 독립적인 상태로 만든다.

분산형 플랫폼 도입의 장점은 추가 인프라 비용이 발생되지 않으며 IoT 기기가 증가되면 증가될수록 더욱 성능이 좋아진다는 것이다. 그리고 소수의 IoT 기기들로 좌우되지 않고 자유로운 환경이 될 수 있게 도와주며 컴퓨팅 자원이 한쪽에 몰리는 것을 방지한다. 그러므로 몇 개의 IoT 기기들에 문제가 생기더라도 심한 마비가 생기지 않게 될 것이다. 또한 분산형 플랫폼이 가지고 있는 한계는 대부분 해결 가능한 기술이 존재하고 있기 때문에 추가적인 연구로 해결이 가능할 것이다.

이러한 이점을 근거로 IoT 보안에 블록체인 인증 프로토콜 적용과 분산형 컴퓨팅 플랫폼 도입 시, 표준화가 이루어진다면 각 IoT기기 생산 기업들은 별도의 보안을 위한 비용이 들지 않으며, 유저들은 지속적으로 발생하는 크래커의 위협으로부터 조금 더 안전하게 이용이 가능할 것이다.

[참고문헌]

[1] 박성준 (2017). 블록체인패러다임과 핀테크 보안. 한국통신학회지(정보와통신), 34(3), 23-28

[2] 유성민 (2017). '블록체인 기반 사물인터넷 서비스' 산업 전망. 한국정보기술학회지, 15(1), 15-20.

[3] 이동영, 박지우, 이준하, 이상록, 박수용 (2017). 블록체인 핵심 기술과 국내외 동향. 정보과학회지, 35(6), 22-28.

[4] 성순화. (2017). 클라우드 컴퓨팅에서 안전한 사물인터넷 데이터를 위한 키 관리. 정보보호학회논문지, 27(2), 353-360.

[5] 한승욱, 손희석, 김병오, 한선영, & 이동만. (2012). Personal Genie: 스마트 IoT 를 구성하는 이질적인 객체간의 자발적 상호작용을 위한 상황인지 미들웨어 시스템. 정보과학회논문지: 시스템 및 이론, 39(4), 270-276.

그림 1. IoT시장 성장 동향 (출처 ; HKExnews)

초음파 공격을 고려한 인공지능 스피커 보안위협 및 대응방안 분석

최근영

고려대학교 소프트웨어보안학과

Analysis of Smart Speaker Threat and Countermeasures considering Ultrasonic Attack

Geunyeong Choi

Graduate School of Computer & Information Technology, Korea University.

요 약

음성대화인식 기술의 발달로 인공지능 스피커 제품 및 서비스가 다양하게 출시되고 있다. 인공지능 스피커는 일상생활에서 사용자 음성을 통해 정보를 수집하고 명령을 분석하여 스스로의 기능을 수행하거나 다른 기기로 명령을 전달할 수 있다. 그러나 사람의 귀에 들리지 않는 초음파를 이용한 Dolphin Attack으로 가상개인비서가 해킹될 수 있음이 밝혀져 음성인식 기능을 사용하는 인공지능 스피커 또한 보안 문제가 발생할 수 있다. 따라서, 인공지능 스피커에서 발생 가능한 보안 문제를 파악하여 대응방안을 찾는 노력이 필요하다. 이 논문에서 인공지능 스피커에서 발생 가능한 보안위협 및 대응방안을 분석하여 기술하였다.

I. 서론

음성대화인식 기술이 발달하면서 인공지능 스피커 제품 및 서비스가 다양하게 출시되고 있다. 아마존 에코, 구글 홈, SKT NUGU 등 인공지능 스피커를 개발하는 회사와 개발 제품이 늘어나면서 시장이 빠르게 성장하고 있다. 인공지능 스피커의 세계 시장 규모는 연평균 42.3% 성장해 2020년에는 21억 달러를 기록할 것으로 전망되고 있다.[1]

인공지능 스피커는 사용자의 음성을 인식하여 정보를 수집하고 사용자의 명령을 수행하거나 다른 기기와 통신하여 명령을 전달할 수 있다. 인공지능 스피커가 사용자의 명령을 인식하기 위해서는 '트리거 워드(trigger word)'를 먼저 인식해야 한다. '트리거 워드'를 통해 작동이 시작되지 않는 이상 데이터 처리 능력이 없기 때문에 개인정보 등 관련된 보안문제가 발생 가능한지에 대한 논의가 계속해서 있어 왔다.

그러나 Dolphin Attack이라 불리우는 초음파 공격을 통해 사용자가 알아들을 수 없게 트리거 워드를 실행할 수 있음이 밝혀져 인공지능 스피커가 악용될 수 있는 가능성이 확인됐다.[2]

이 논문에서는 국내외 인공지능 스피커 제품과 서비스를 알아보고 인공지능 스피커에서 발생 가능한 보안위협과 대응방안을 분석하여 기술한다.

II. 관련 연구

2.1 인공지능 스피커

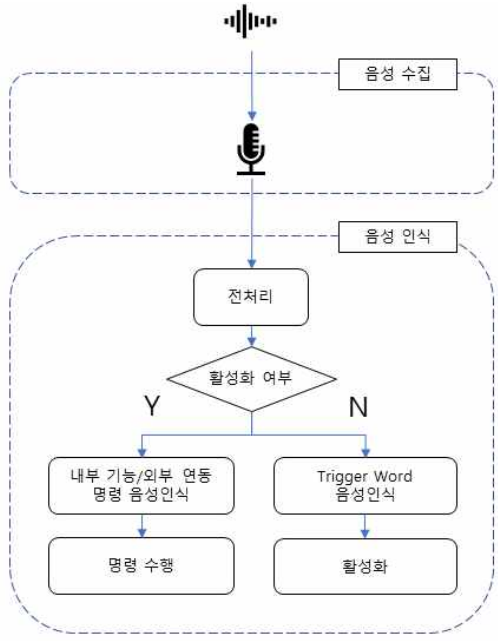
인공지능 스피커는 국외 뿐만 아니라 국내에서도 다양한 제품이 만들어지고 있다. 그 중에서 아마존 에코가 가장 많이 사용되고 있는 제품으로 외부에서 알렉사의 기능을 접근 및 사용할 수 있도록 스킬 방식을 제공하여 타 기업 및 개인 개발자들이 알렉사 기능을 고도화하거나 알렉사 기반의 자체 음성인식 기기를 개발할 수 있도록 하고 있다.

표 1. 국내외 주요 인공지능 스피커

구분	명칭 (회사)	주요 기능	Trigger Words
국외	에코 (아마존)	아마존 주요 서비스 연결	Alexa
	구글 홈 (구글)	구글 서비스 연결	OK Google, Hey Google
	인보크 (MS)	스카이프 무료 통화	Hey, Cortana
국내	NUGU (SKT)	쇼핑, 교통 서비스	아리아
	GiGA Genie (KT)	음악, 영상통화	기가지니, 지니야 등

인공지능 스피커의 동작은 크게 음성 수집과 음성 인식으로 나뉘며 마이크를 통해 수집된 음성은 인공지능 스피커의 활성화 여부에 따라 Trigger Word를 인식하는 절차 또는 명령을 수행하는 절차로 이행된다.

그림 1. 인공지능 스피커 동작 원리



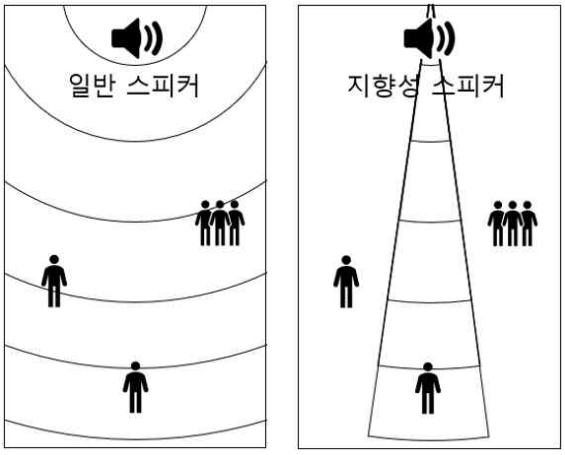
2.2 초음파 공격

Dolphin Attack은 사람의 귀에 들리지 않는 초음파를 이용해 음성인식 명령어를 전송하는 공격 기법이다. 이 공격 기법은 소유자가 공격이 이루어지고 있음을 알 수 없어 공격당한 사실을 공격이 완료된 후 알게 된다.[2]

초음파를 이용해 소유자가 알 수 없도록 인공지능 스피커의 Trigger Word를 송신하여 인공지능 스피커를 활성화할 수 있고 활성화된 후에는 인공지능 스피커의 명령어를 이용해 다양한 공격을 수행할 수 있다.[2]

그리고 지향성 스피커를 이용할 경우, 원거리에서 특정 범위 또는 방향에만 소리가 전달될 수 있도록 조준해 초음파를 송신할 수 있다. 이 경우, 건물 밖에서 공격을 시도할 수 있고 대상이 다수일 때 정밀하게 공격할 수 있다.[3]

그림 2. 지향성 초음파 스피커 특징



III. 보안위협 및 대응방법 분석

3.1 보안위협 분석

초음파 공격으로 인해 소유자가 인지하지 못한 상태에서 인공지능 스피커로 접근 및 악의적인 명령 전달이 가능하다. 그리고 지향성 초음파 스피커를 이용할 경우, 빅데이터를 통해 학습된 합성 음성 또는 녹음된 음성을 송신하여 인공지능 스피커를 소유자로 속이는 것이 가능하다. 이처럼 인증 부재 또는 인증 우회를 통해 공격자는 인공지능 스피커에 접근하여 악의적인 명령을 수행할 수 있고 악성코드가 있는 URL에 접근하도록 유도하여 악성코드를 다운로드 및 실행하도록 하는 등의 추가적인 공격을 연결할 수 있다.

스마트 홈과 인공지능 스피커가 연결되어 있을 때 공격자가 메모/알림의 TTS 기능을 이용하여 소유자에게 잘못된 방문자 정보를 전달할 경우, 소유자가 스스로 공격자에게 문을 열어주는 사회공학적 공격도 가능하다.

표 2. 인공지능 스피커 보안 위협 분석

유형	시나리오	결과
침입	- 초음파 공격을 이용해 소유자가 인지하지 못하도록 접근 - 합성음성/녹음 음성을 전송하여 소유자로 위장	다른 공격 연결
악용	- 인공지능 스피커에 악의적인 명령 전달	시스템 악용
감시/도청	- 인공지능 스피커에서 공격자의 휴대폰으로 전화를 연결하여 도청 - 지속적/반복적으로 중요 정보를 문자, 메일로 발송하여 정보 유출	개인 정보 유출
차단	- 인공지능 스피커에 연속으로 명령을 전달하여 소유자의 접근 및 사용 차단	서비스 거부
사회공학	- TTS 기능으로 잘못된 정보를 제공하여 소유자를 혼동하게 함	소유자 조종

3.2 대응방법

초음파 공격을 이용한 인공지능 스피커에 대한 공격을 방지하기 위해서는 대응방법을 크게 인공지능 스피커를 보완하는 방법과 보다 안전한 인공지능 스피커 사용법을 제안하는 것으로 나눌 수 있다.

인공지능 스피커는 Trigger Word를 제외하고 사용자 인증, 접근 제어 등이 없어 명령을 수행할 때 제한을 받지 않으므로 특히 인증에 대한 보완이 필요하다.[4] 인공지능 스피커는 사용자의 일상생활에서 사용됨을 고려할 때, 일상생활에서 사용하는 사용자의 휴대폰, 웨어러블 장비 등을 이용한 다중 인증으로 인증부재에 대한 취약점을 보완 가능하다. 그리고 인공지능 스피커의 음성인식 기능을 고려할 때, 인공지능 스피커가 소유자만 알 수 있는 임의의 질문을 소유자에게 하고 소유자는 그에 대한 대답을 하는 Challenge-Response 방식의 인증도 적용 가능하다.

사용자는 인공지능 스피커를 이용하지 않을 경우, 인공지능 스피커를 종료하는 것이 안전하나 일상생활에서 즉각적으로 이용하기 위한 목적에 다소 적합하지 않으므로 Trigger Word를 변경하거나 불필요한 명령어를 차단하도록 설정할 수 있어야 한다.[5]

표 3. 인공지능 스피커 보안위협 대응 방법

구분	대응 방법	위협
인공지능 스피커	음성 수집시 가청 주파수 대역을 제외한 주파수를 필터링	침입
	주요 기능 수행 전 사용자가 수행 내용을 확인하는 사용자 Interaction 추가	악용, 감시/도청, 사회공학
	Multifactor 사용자 인증	침입
	Challenge-Response 기법의 사용자 식별/인증 도입	침입
	인공지능 스피커에서 연동 가능한 프로그램/서비스에 대한 white list 방식 적용	악용, 감시/도청
사용 방법	사용하지 않을 때는 인공지능 스피커를 종료/비활성화	전체
	Trigger Word를 변경	침입
	인공지능 스피커에서 제공하는 기본 차단 설정을 이용해 불필요한 명령어를 비활성화	악용

IV. 결론

인공지능 스피커는 스마트홈, IoT 시대에 허브 역할을 할 수 있는 중요 요소로 인식되고 있다. 이러한 중요성을 고려하여 본 논문에서는 발생 가능한 위협을 알아보고 대응방안을 도출함으로써 인공지능 스피커로 인해 발생할 수 있는 사고를 미연에 예방하고 위험을 감소시키는데 도움을 줄 수 있을 것으로 생각한다.

또한 인공지능 스피커는 IoT 시대에서 IoT 기기들과 연동하는 허브 역할을 담당하고 있는 만큼 새로운 IoT 기기들과의 상호 작용으로 인해 발생할 수 있는 위협과 대응방안에 대해서도 계속해서 연구가 필요할 것으로 생각한다.

[참고문헌]

[1] 강효진. 인공지능(AI 스피커) 플랫폼의 발전과 디지털콘텐츠산업과의 연계방안. 정보통신산업진흥원. (2017.9.5.).

[2] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, Wenyuan Xu. DolphinAttack: Inaudible Voice Commands. ACM Conference on Computer and Communications Security (CCS) 2017.

[3] 이학수, 문원규. 압전 필름을 이용한 초지향성 스피커 기술 연구. 한국음향학회 학술대회, 515-518. (2004).

[4] 김규동, 김현성. AI 가상비서의 특징 및 보안 현황 분석. 대한전자공학회 학술대회, 486-488. (2017).

[5] ELLA WILLS. Hackers can spy on you and take over your smart devices by controlling Siri, Alexa, and Google Now using inaudible ultrasound commands . Daily Mail Online. (2017.9.7). http://www.dailymail.co.uk/sciencetech/article-4861136/Voice-assistants-vulnerable-silent-voice-control-attack.html

기계학습을 이용한 네트워크 전장정보 수집
(Collecting Network Field Information using Machine Learning)

한규석*, 윤지원*

고려대학교 정보보호대학원 정보보호학과

Kyu soak Han*, Jiwon Yoon*

Graduate School of Information Security, Korea University.*

요 약 최근 IoT(Internet Of Things) 및 ICT(Information and Communications Technologies) 기술이 집약된 다양한 시스템이 개발로 인해 다양한 기기들이 네트워크에 연결되면서 각기 다른 리소스와 기능을 갖는 기기들에 따른 다양한 운영체제가 등장하게 되었다. 해킹 보안에 대한 관심이 높아지면서 각 기기들에 대해 탑재된 운영체제의 취약점 분석과 이에 따른 실제적인 공격 기법에 대한 연구가 진행되었으며, 이에 따라 기기의 운영체제의 종류와 세부적인 버전 그리고 중점적으로 강화되어 있는 기능(API)이 보안에 있어 중요한 정보로 대두되고 있다. 사이버 전에서 이러한 정보 수집을 관제하는 것은 사이버 위협의 전초이기 때문에 이러한 스캐닝을 네트워크 트래픽 관제하는 방안은 끊임없이 연구되어 왔으며, 이 관제를 피하기 위해 정보수집자는 은밀하게 포트 정보를 수집하기 위한 대책을 맞서 준비하는 실정이다. 본 논문에서는 네트워크 관제시스템에서 중요하게 여기지 않는 네트워크 기본 명령어들과 기계학습을 이용한 분석을 통해서 상대에 대한 정보를 얻어 낼 수 있는 스캐닝에 대해서 다루게 된다.

키워드 : 운영체제 탐지, 기계학습, 네트워크, 네트워크 관제, 네트워크 명령어

Abstract Recently, various systems based on Internet of Things (IOT) and ICT (Information and Communications Technologies) technology have been developed, and various devices are connected to a network, and various operating systems according to devices having different resources and functions have appeared. As the interest in hacking security increased, researches on the vulnerability analysis of the operating system installed on each device and the actual attack technique were carried out. Accordingly, the type and detailed version of the operating system of the device, Function (API) is emerging as important information in security. Since the control of this information gathering at the cyber warfare is the beginning of the cyber threat, the way to control the network traffic of this scanning has been studied constantly. In order to avoid this control, the information collectors prepare countermeasures to secretly collect port information . In this paper, we deal with scanning which can acquire information about opponent through network basic commands which are not important in network control system.

Keyword : Operating system detection, machine learning, network, Network control, Network commands

1. 서 론

전 세계적으로 사이버 공격이 점차 고도화되고 있는 상황에서 해킹 사건들이 끊임없이 발생하면서, 보안장비를 회피하여 사이버 전장망의 정보를 수집하는 것은 항상 거론되고 있는 기술이다. 성능이 좋은 여러 툴이나 명령어들이 있지만, 이는 보안관제에 의해서 막히거나 파악이 되어 버리는 경우가 많다. 이에 보안장비를 회피해 전장정보를 수집하는 기술과, 이를 관제 하려는 기술은 창과 방패처럼 발전하고 있는 추세이다.

대상이 되는 기기들의 보안적인 취약점은 아주 작은 정보에도 노출 될 수 있다. 예를 들면 기기의 운영체제나, 기기의 운영체제가 되는 기본적인 기술들, 세부 버전, 성능들, 또한 통신 프로토콜 가지 모든 부분에서 취약점이 노출되어 관리자의 목적과 다른 의도를 갖는 인터넷 유저에게 이용 될 수 있다. 이런 취약점의 노출은 우리가 알고 있는 채널과 그에 대한 기기의 반응을 통해서도 이루어 질수 있지만, 예상치 못한 곳에서도 도출 될 수 있기에, 부채널(side channel) 공격이나 covert channel에 대한 연구도 활발히 진행되는 추세이다.

이런 최근의 흐름에 따라, 평범하다고 여겨지는 기기의

동작들에 대해서 기계학습 기법을 적용하여, 사이버 기기의 보안 취약점을 분석하기 위한 정보수집을 진행하려한다.

2. OS Fingerprinting

사람의 지문이 고유한 식별 도구로 사용되는 것과 마찬가지로 운영체제는 패킷(packet)에 대한 지문으로 네트워크에서 고유하게 식별될 수 있는데, 패킷 지문은 다양한 운영체제의 통신 프로토콜 구현의 차이점 때문이다. 호스트가 네트워크를 통해 보내는 패킷의 특정 프로토콜 플래그(flag), 옵션, 패킷 필드 및 데이터의 초기 값을 분석하여 호스트에 설치된 운영체제를 판별할 수 있다.

2.1 OS Fingerprinting 전략

운영체제 핑거프린팅은 시스템에 대한 특정 정보를 식별하는 것을 의미하는데, TCP SYN 패킷의 TCP/IP 헤더를 이용하여 패킷을 생성하는 운영체제를 식별한다. IP는 통신을 가능하게하기 위해서 사용자에게 논리 주소(logical address)를 할당하는데 사용되며, TCP는 신뢰할 수 있는 패킷을 전달한다.

운영체제에 대한 핑거프린팅은 일반적인 보안 연구자들에게도 중요한 의미를 갖는다. 우리 주변의 기기들은 동일한 운영체제를 사용하지 않는데, 그 필요성이나 사용장소에 따라서 원하는 API가 강조되어 특정 기능이 강화되거나 그 기기의 퍼포먼스에 따라서 그 크기가 작거나 혹은 풀 패키지 운영체제일 수 있다. 이렇게 다양한 운영체제는 shodand이나 CVE, CVS, CVSS 등을 통해 각각이 갖고 있는 보안 취약점(vulnerability)과 이를 직접 코드로 만들어 공략할 수 있는 침해코드(exploit)가 공개되어있는 실정이다.

운영체제를 판별하기 위한 값들로는 MSS(Maximum Segment Size), TTL(Time To Live), Sliding Window Size, TCP Options, Packet Length, IP ID가 있는데, 본 논문에서는 운영체제를 식별하기 위한 중요한 3가지 요소에 대해서만 설명한다.

2.1.1 TTL (Time-To-Live)

원격 호스트로부터 전송받은 패킷의 IP Header에 설정된 TTL을 분석한다. 물론 전송받은 패킷에 최초의 TTL 정보가 들어있지는 않지만, 보통의 OS들이 사용하는 TTL 기본값과 현재 TTL값 간의 차를 고려하면 원격 호스트의 OS를 추정할 수 있다. 각 OS별 기본 TTL 값은 다음과 같다.

Operating System	Initial TTL
Linux(kernel 2.4 and 2.6)	64
Google's customized Linux	64
FreeBSD	64
Windows XP	128
Windows 7, Vista	128
Cisco Router (IOS 12.4)	255

예를 들어 전송받은 패킷의 TTL이 110일 때, 이 패킷은 windows 또는 Cisco에서 왔음을 알 수 있다. 또, 이가 만약 windows에서 온 패킷이라고 가정했을 때, 이 패킷은 128-110=18개의 홉을 거쳐 왔다는 사실도 알 수 있다. TTL을 홉의 개수보다 한 개 적은, 즉 여기서는 17으로 설정한 traceroute를 이용하면 원격 호스트에 직접 접근하지 않아도 호스트 경로에 대한 정보를 얻을 수 있다.

2.1.2 Window size (WS)

이 필드는 수신 컴퓨터가 받는 데이터의 수를 의미한다. 연결이 설정되고 데이터가 전송되면 수신 버퍼를 차지하는 양에 따라서 패킷이 얼마의 window size를 사용하고 있는지, window size가 얼마나 자주 변하는지 분석하면 원격 호스트의 OS를 추정할 수 있다. 각 OS별 기본 window size는 다음과 같다.

Operating System	TCP window size
Linux(kernel 2.4 and 2.6)	5840
Google's customized Linux	5720
FreeBSD	65535
Windows XP	65535
Windows 7, Vista	8192
Cisco Router (IOS 12.4)	4128

또 Linux, FreeBSD, Solaris는 한 세션동안 같은 window size를 유지하지만, Cisco와 Windows는 계속 그 값이 변한다.

2.1.3 Packet length

전체 길이 필드는 패킷의 길이를 나타내며 TCP 헤더, IP 헤더 및 페이로드(payload)를 포함한다. 이 필드는 수동 운영체제 핑거프린팅에서 중요한 역할을 한다.

SYN 패킷의 경우에 총 길이는 운영체제에 따라 다르며, 각 운영체제는 자체 TCP NOP(No Operation) 옵션을 설정하여 각각의 운영체제의 총 패킷의 길이를 고유하게 할 수 있다. 모든 운영체제는 최소 하나의 TCP 옵션필드를 사용하므로 TCP SYN 패킷 길이는 44바이트 이상이어야 한다.

2.2 Active/Passive OS Fingerprinting

운영체제 핑거프린팅 기법에 따라 활성 운영체제 핑거프린팅과 수동 운영체제 핑거프린팅으로 구분될 수 있다. 이 절에서는 각각의 대표적인 tool에 대해 설명하도록 한다.

2.2.1 Active OS Fingerprinting

Active OS Fingerprinting은 패킷을 원격 호스트로 전송하고 해당 응답을 분석하는 프로세스로 이를 통해

스캐너는 보다 정확한 결과를 얻을 수 있다. 전통적인 접근법은 여러 개의 합법적인 패킷을 탐색할 때 네트워크의 TCP/IP 스택 동작을 검사하는 방법이다.

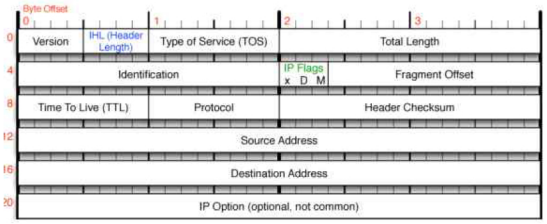


그림 1 IPv4 Header Diagram

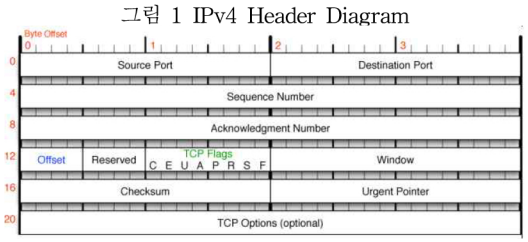


그림 2 TCP Header

1) Nmap

Nmap(network mapper)의 가장 잘 알려진 기증 중 하나는 TCP/IP stack fingerprinting을 사용하는 원격 운영체제 탐지로, 일련의 TCP 및 UDP 패킷을 원격 호스트에 보내고 응답의 모든 비트를 검사하도록 한다. TCP ISN 샘플링, TCP 옵션 지원 및 주문, IP ID 샘플링 및 초기 윈도우 크기 검사와 같은 여러 가지 테스트를 수행한 뒤 nmap-os-db와 결과를 비교하며 알려진 2600개 이상의 OS fingerprinting 데이터베이스를 검색하고 일치하는 항목이 있으면, OS 세부 정보를 인쇄하도록 하며, 다음과 같은 기능을 제공한다.

- 호스트 탐지: 네트워크상의 컴퓨터들을 확인
- 포트 스캔: 하나 혹은 그 이상의 컴퓨터들에 열린 포트를 나열
- 버전 탐지: 응용 프로그램의 이름과 버전 확인
- 운영체제 탐지: 원격으로 운영체제와 네트워크 장치의 하드웨어 특성을 확인

2) QueSO

TCP/IP fingerprinting을 사용하여 원격으로 탐지할 수 있는 최초의 광범위한 도구로, 이는 별도의 파일에 저장된 Fingerprinting을 사용한 첫 tool이다. ambiguous packet을 이용하여 탐지를 하였으며, 이 모호성 때문에 TCP/IP와 다른 동작으로 발생하였으므로 탐지가 가능하도록 하였다.

2.2.2 Passive OS Fingerprinting

Passive OS Fingerprinting은 active OS Fingerprinting과 같은 개념이지만, 다른 방법으로 구현된다. 이는 원격 호스트에게 직접 query를 날리는 대신, sniffer를 이용해 원격 호스트에서 온 패킷만을 캡쳐하여 사용한다. 네트워크의 트래픽을 발생시키지 않기 때문에 passive라 불리며, 캡쳐 된 패킷 분석을 통해 원격 호스트의 OS를 추정할 수 있다. 이때의 분석은 모든 OS의 IP stack이 각각의 다른 특성을 갖는다는 것에 기초한다.

passive하게 패킷을 분석하기 위해서는 수집한 패킷의 몇몇 signature를 추출하여 사전에 구축되어 있는 signature 데이터베이스와 비교해야 한다. 대부분의 signature는 TCP/IP 패킷의 헤더에 위치하며, passive fingerprinting은 각 패킷의 헤더 분석을 통해 이뤄진다. 패킷의 헤더 정보는 SSH나 HTTPS와 같은 암호화된 프로토콜 상에서도 얻어질 수 있기 때문에 이 방법은 매우 유용하다. 아래에서는 대표적인 passive fingerprinting의 도구들을 설명한다.

1) siphon

siphon은 2000년에 “Siphon Project”란 이름으로 PoC(proof-of-concept) 단계로 개발된 도구이다. 이 도구는 TCP의 SYN 패킷이나 ACK 패킷에서 세 가지의 signature 정보를 추출하여 분석하며, 이 signature들은 아래와 같다.

signature	설명
TTL	데이터 그램이 인터넷 시스템에 남아있을 수 있는 최대 시간 분석
Window size	패킷이 얼마의 윈도우 사이즈를 사용하는지, 그리고 윈도우 사이즈가 얼마나 자주 변하는지 분석
DF(Do not Fragment flag)	DF 플래그를 사용하는지 여부를 분석

표 4 siphon의 signature
(Table 4. signature of siphon)

2) p0f

p0f는 2000년에 Michal Zalewski가 처음 개발한 도구이다. p0f는 TCP의 SYN 패킷만을 사용하며, siphon이 사용하는 세 가지의 signature 이외에도 다섯 가지의 signature를 추가로 분석한다. 이 추가적인 signature는 주로 TCP 연결을 시작할 때의 SYN에만 설정된 TCP 옵션과 관련이 있으며, 이는 아래와 같다.

signature	설명
MSS(Max Segment Size Option)	CP가 헤더를 제외하고 실을 수 있는 최대 데이터 크기
WSO(WIndow cale Option flag)	더 큰 window size의 window를 사용하기 위해 TCP 확장 옵션을 사용하고 있음을 나타내는 플래그
SACK(Selective Acknowledgments Option flag)	통신 중에 여러 세그먼트 중 손실된 세그먼트만 선택적으로 확인응답하는 TCP 옵션을 사용하고 있음을 나타내는 플래그
NOP(No Operation Option flag)	4바이트 단위의 배수로 옵션 사이들 패딩하기 위한 옵션 플래그
PS(Packet Size)	패킷의 바이트 사이즈

표 5 p0f의 signature
(Table 4. signature of p0f)

3. 기계학습을 이용한 정보 수집 기법 제안

높은 정확도로 관심을 모았던 ANN은 학습에 시간이 너무 오래 걸려서 비현실적이라 여겨졌다. ANN의 가장 큰 문제점은 모델이 학습 데이터에 너무 맞춰서 학습되는 과적합(over-fitting)이 발생한다는 점이었다. 이렇게 외면받던 ANN이 다시 각광받기 시작한 것은 ANN의 문제점을 극복한 Deep learning 기법이 제시된 이후부터였다.

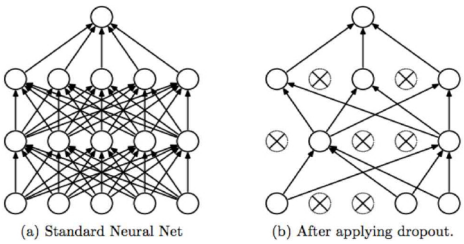


그림 4. Deep Learning의 dropout 알고리즘
(Fig.3 dropout algorithm of Deep Learning)

4. 실험

이 논문에서 제안하는 실험은 기초적인 OS detection이다. 우리가 감지하려는 컴퓨터 2대를 각각 다른 운영체제를 설치한다. 이 실험에서는 Linux, window10을 설치하였다. 이 각각의 기기에 기본적인 네트워크 명령어인 ping을 보내어 그 응답을 모으게 된다.

아래의 그림은 실험의 기초가 되는 트레이닝을 이야기한다. 우선적으로 매번 날아오게 되는 ping에 대한 응답에 대해서 비트를 무작위로 추출한다. 단, 시작하는 비트는 무작위이지만, 그 이후에 추출되는 비트는 순차적으로 추출하게 된다. 그리고 이를 기계학습에 적용하여 트레이닝을 시키게 된다. 그렇게, 답을 알고 트레이닝을 하게되면 다른 추출에 대해서도 os X라는 결과가 추출되게 된다. 이를 이용해서 이 실험에서는 osX, Y이렇게 2개의 답을 비교해본다.

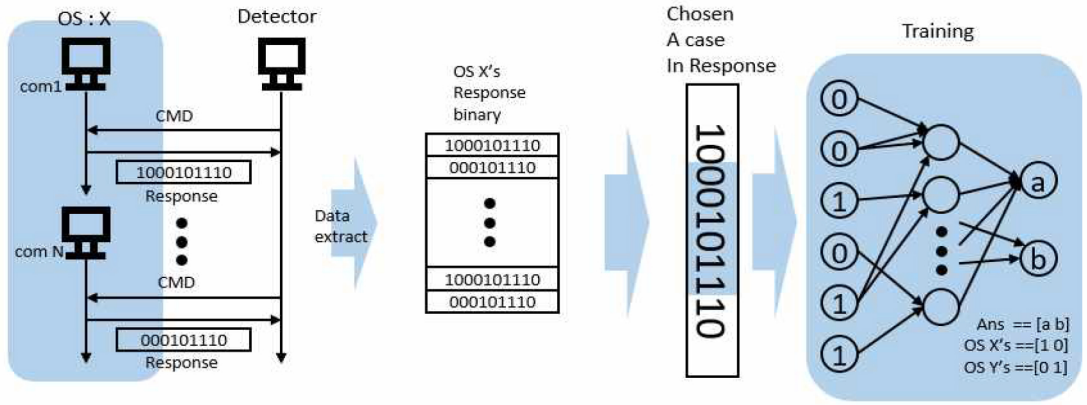


그림 3. OS detection을 위한 기계 학습 training

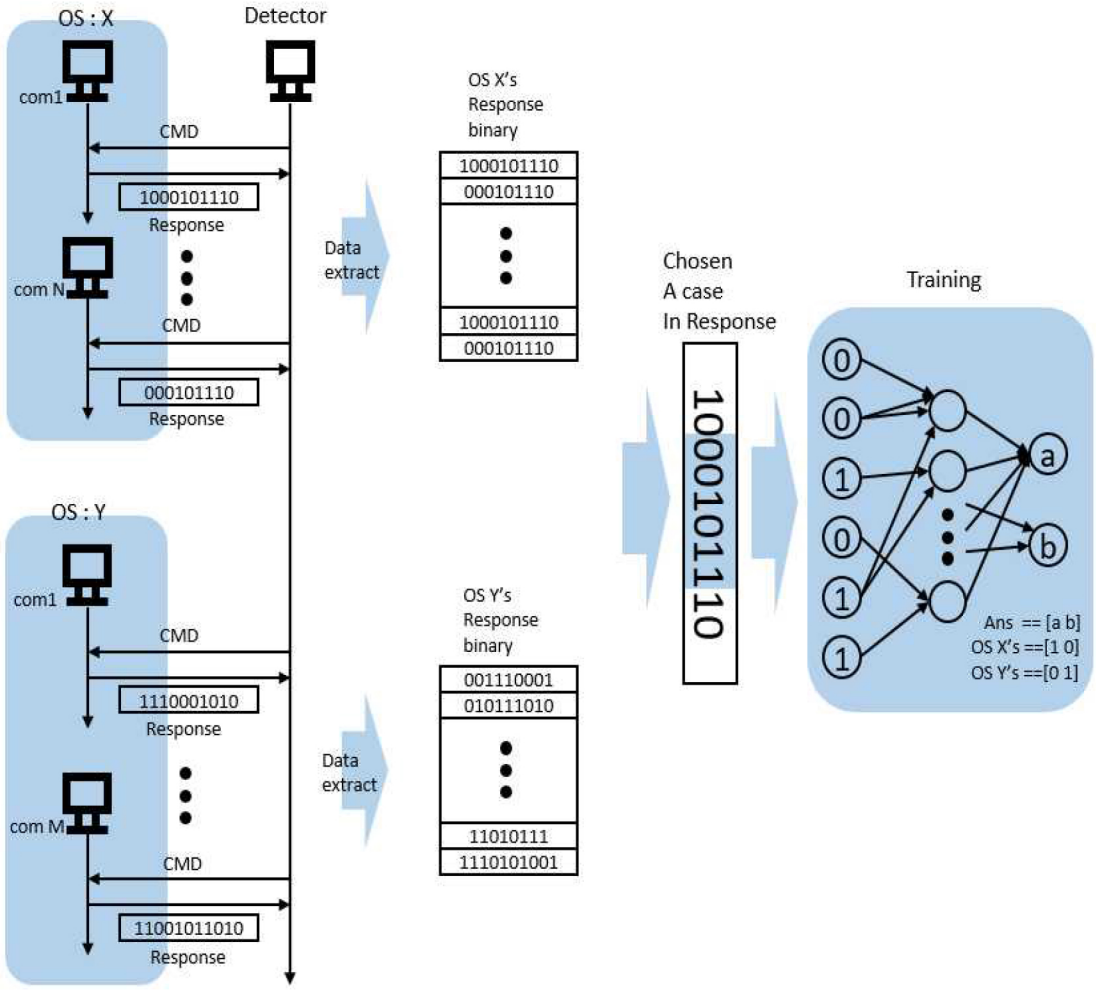


그림 5 실험을 위한 개요도 : 서로 다른 OS detection을 위한 기계 학습 training

4.1 네트워크 명령어를 통한 정보 수집

2개의 운영체제는 Linux와 window10을 사용했다. 명령어는 ping으로 했으며, 이에 대한 응답을 옆의 표6의 코드를 통해서 수집했다

- os_addr.txt에서 대상 IP 주소 읽음어 옴
- tshark로 저장된 패킷 분석
- 수집된 Linux 수: 28
- 수집된 Window 수: 19

pcap2hex.py
데이터 수집 모듈
from subprocess import call, check_output, STDOUT import pdb os_name = 'linux' # OS 이름 snt = "tshark -r data/{}_dump.pcapng -T fields -e data -Y 'ip.src=={}'; exit 0" f = open('{}_addr.txt'.format(os_name),'r') # tshark를 통한 hex dump fw = open('{}_output2.txt'.format(os_name),'w') # 결과 읽어오기 cnt = 0

```
rls = f.readlines()
for line in rls:
    cmd = snt.format(os_name,line.strip())
    print cmd
    tmp = check_output(cmd, stderr=STDOUT,
shell=True) # command line 결과 읽어오기
    fw.write(''.join(tmp.split())+'\n')
    print cnt
    cnt+=float(1/len(rls)*100)
f.close()
fw.close()
```

표 6 pcap2hex.py
(Table 2. pcap2hex.py)

4.2. 기계학습을 이용한 명령어 응답 분석

4.1 절에서 수집한 데이터를 바탕으로 우리는 기계학습을 통해서 답을 알고 있는 상태로 그림 5와 같이 Training을 하였다. 그리고 임의의 운영체제의 ping에 대한 응답에 대해서 그 응답을 보낸 OS가 과연 무엇인지 맞추게 해보았다.

- Neural Network를 이용한 Classification
- 4-fold cross-validation (# of test-set : 47)
- 전체 학습횟수 2000
- 교차 검증 횟수 100

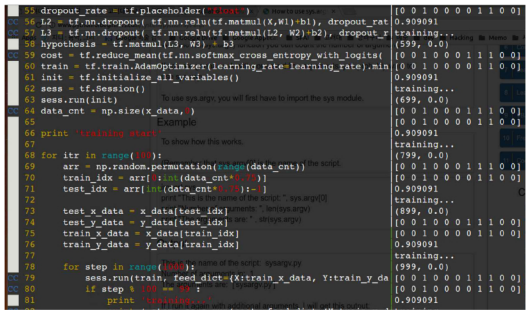


그림 6 실험 결과
(Fig. 4 result of experiment)

그 결과 97%의 경우에 해당 OS의 종류를 맞춘 것을 알 수 있었다.

5. 결론

위 실험의 결과에서 볼수 있듯이, 단순한 명령어에 대한 응답안에서 추출된 비트열이라 할지라도, 이를 기계학습 기법을 이용해 트레이닝 및 분류를 시켜, 어떤 운영체제인지 구분하는 것이 가능하다. 이 기법을 더욱이 고도화 시켜, 더 놓은 탐지 율과, 기기의 다른 특징들도 구분짓는 시스템의 구축이 가능할 것으로 보여진다. 또한 이런 시스템은 기본적인 네트워크 명령어를 기반으로 전장망 정보 수집이 진행되기 때문에, 보안관제에 탐지되지 않을 것으로 기대된다.

이번 논문에는 운영체제를 맞추는 것에 실험을 진행했으나, 해당 기기가 갖고 있는 중요 API에 따른 응답, 주요 사용 용도에 따른 응답의 차이와 같은 좀더 자세한 규격이나 용도에 대한 정보 수집이 가능하게 Training시켜보는 것이 앞으로의 목표이다.

더 나아가 기기의 다른 특징들을 구분 짓게 하고, 이를 공개된 취약점 정보와 연동해 네트워크 취약점 도출에 큰 기여를 할 것으로 보여진다.

참 고 문 헌

[1] http://www.netage.com/orgscope/tool/hyperbolic.html

[2] Kang, Min Suk, Soo Bum Lee, and Virgil D. Gligor. "The crossfire attack." Security and Privacy (SP), 2013 IEEE Symposium on. IEEE, 2013.

[3] Kumar, Pushpender, and Narottam Chand. "Clustering in Wireless Multimedia Sensor Networks." Journal of Sensor Technology 2013 (2013).

[4] Ng, Andrew Y., Michael I. Jordan, and Yair Weiss. "On spectral clustering: Analysis and an algorithm." Advances in neural information processing systems 2 (2002): 849-856.

[5] Von Luxburg, Ulrike. "A tutorial on spectral clustering." Statistics and computing 17.4 (2007): 395-416.

[6] Schölkopf, Bernhard, Alexander Smola, and Klaus-Robert Müller. "Nonlinear component analysis as a kernel eigenvalue problem." Neural computation 10.5 (1998): 1299-1319.

[7] Opsahl, Tore, Filip Agneessens, and John Skvoretz. "Node centrality in weighted networks: Generalizing degree and shortest paths." Social networks 32.3 (2010): 245-251.

[8] Borgatti, Stephen P. "Centrality and AIDS." Connections 18.1 (1995): 112-114.

[9] Jackson, Matthew O. Social and economic networks. Vol. 3. Princeton: Princeton university press, 2008.

[10] Barthelemy, Marc. "Betweenness centrality in large complex networks." The European Physical Journal B-Condensed Matter and Complex Systems 38.2 (2004): 163-168.

IoT 환경에서 전자정부 시스템 활용을 위한 행정안전부 C 시큐어코딩 가이드 보완점 연구: SEI CERT C Coding Standard 2016 적용

백승태*

*고려대학교 컴퓨터학과

Research on improving C Secure coding Guideline of MOIS to serve e-Government system in IoT: Applying SEI CERT C Coding Standard 2016

Seung-Tae Baek*

*Department of Computer Science and Engineering, Korea University.

요 약

행정안전부에서 C 시큐어코딩 가이드를 제정하여 보안성을 확보한 시스템을 구축하기 위한 노력이 이루어지는 가운데, 최근 IoT 환경에서 전자정부 시스템을 제공하기 위한 필요성이 제기되고 있다. 본 논문에서는 행정안전부 C 시큐어코딩 가이드와 SEI CERT C Coding Standard 2016의 비교 분석을 진행하며, 행정안전부 C 시큐어코딩 가이드에서 잘못 제시된 안전한 코드예제를 지적하고 보완점을 제시한다. 이를 통해, IoT 환경에서 안전성, 신뢰성, 보안성을 확보한 전자정부 시스템을 제공하기 위한 임베디드 시스템 개발 관점에서 향후 C 시큐어코딩 가이드라인 개정에 대한 올바른 방향을 모색한다.

I. 서론

전자정부 관련 정보화사업에 있어서 시큐어코딩의 적용은 필수적인 상황이다. 행정안전부는 '행정기관 및 공공기관 정보시스템 구축•운영 지침'에 따라 전자정부 관련 정보화사업 수행시에 안전한 SW 개발을 위한 목적으로, C 시큐어코딩 가이드를 제정하여 배포하였다.[1]

한편 행정안전부는 2016년 5월 '전자정부 2020 기본계획'을 발표하였다. 정부는 '전자정부 2020 기본계획'에서 새로운 전자정부 추진 방향의 핵심수단을 PC•인터넷 중심에서 지능정보기술 중심으로 전환한다고 밝혔으며, 그 중 사물인터넷(IoT)기술이 포함되었다.[2]

IoT 디바이스에서 전자정부 시스템을 제공하기 위해서는 임베디드 시스템 개발에 있어 C 언어 기반의 시큐어코딩이 필수적이다. 그러나 행정안전부에서 제공하는 C 시큐어코딩 가이드는 정보시스템 위주의 보안약점에 따른 예제를 제공하기 때문에 추가적인 개선이 필요하다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로서 행정안전부 C 시큐어코딩 가이드, SEI CERT C Coding Standard 2016을 소개한다. 3장에서는 행정안전부 C 시큐어코딩 가이드와 SEI CERT C Coding Standard 2016의 비교를 통해 IoT환경에서의 전자정부 시스템 활용을 위한 보완점을 제시한다. 4장에서는 결론으로서 본 논문을 마무리한다.

Ⅱ. 관련연구

2.1 행정안전부 C 시큐어코딩 가이드

전자정부 SW 개발 • 운영자를 위해 행정안전부에서는 2011년 6월 C 시큐어코딩 가이드 1판을 개발하였다. 2011년 9월에 보안약점 내용 및 예제코드를 수정하여 개정된 2판이 개발되었고, 2012년 6월에 행정안전부의 'SW 개발 보안 가이드'가 개정되어 이에 따른 보안약점 항목 및 내용 변경으로 2012년 9월 개정된 3판이 개발되어 현재까지 사용되고 있다. C 기반의 정보시스템 개발시 고려할 보안약점 유형을

7가지로 분류해 각 유형에 속하는 보안약점 총 58개와, 그에 해당하는 예제를 제공한다.[1]

2.2 SEI CERT C Coding Standard 2016

CMU(Carnegie Mellon University)의 SEI(Software Engineering Institute) 소속 CERT(Computer Emergency Response Team)에서는 안전성, 신뢰성, 보안성을 확보한 시스템을 개발하기 위한 목표를 가지고 SEI CERT C Coding Standard 2016을 배포하였다. 개발 관점에서 총 14가지의 주제를 제시하고, 각 주제에 속하는 코딩 룰을 제시한다. 제시된 코딩 룰은 총 99개 이며각각 의 코딩 룰 에 속 하는 안전하지 않 은 코드와 안전한 코드의 예를 제시한다. 그리고 각 코딩 룰에 대해 심각성(Severity), 악용가능성(Likelyhood), 수정 비용(Remedation costs)과 우선순위(Prority), 단계(Level)에 따른 등급 평 가로 위험평가를 제시한다. 이에 더 해, 안전하지 않 은 코드가 어떤 유형의 보안약점을 포함하는지 CWE(Common Weakness Enumeration) 번호를 제공한다.[3]

Ⅲ. 비교 분석 및 보완점 제시

3.1 비교 분석

행정안전부 C 시큐어코딩 가이드와 SEI CERT C Coding Standard 2016의 특징을 비교하면 아래 [표1]과 같이 나타낼 수 있다.

[표1]. 행정안전부 C 시큐어 코딩 가이드와 SEI CERT C Coding Standard 2016의 비교 [1][3]

항목	행정안전부 C 시큐어코딩 가이드	SEI CERT C Coding Standard 2016
안전성 표준	X	O
신뢰성 표준	X	O
보안성 표준	X	O
국제 표준	X	X
적용분야	전자정부 정보시스템	모든 종류의 C 소프트웨어
대상	개발자 발주자	개발자
서술방식	보안약점 → 안전한 코딩기법 → 예제	코딩 룰 → 예제 → 보안약점

[표1]에서 확인한 바와 같이, 행정안전부 C 시큐어코딩 가이드는 보안약점을 유형화하고 그에 따른 예제를 제공한다. 가이드로서 제시된 58가지의 보안약점에 추가적인 보안약점을 제거하는 것 은 개발자들이 꼭 해야하는 일이다 . 하지만 이러한 서술방식은 실 제 개발자들에게 가이드에서 제공하는 58개의 보안약점만 해결하면 된다는 인식을 불 러일으킬 수 있다또 한 제시된 보안약점 외 에 다른 보안약점이 취약점으로 발전되어 피 해가 발생할 경우에 면책 의 근거가 될 수 있다.

반면, CERT C Coding Standard 2016의 서술방식은 코딩 룰을 설 정하고 그에 따른 예제를 제공하며, 보안성을 확보하지 못한 코드 예제가 포함하는 보안약점, 및 정의되지 않은 행위(Undefined Behavior)와 명세되지 않 은 행위(Unspecified Behavior)를 제공한다. C 개발 관점에서 직접 적인 코딩 룰 로서 근본적인 원리제시를 한다는 점에서 행정안전부 C 시큐어코딩 가이드와 차 이가 있다.

이 차이는 정수 오버플로우 보안약점을 예시로 들어 설명될 수 있다. 아래 [표2]와 [표3]은 각각 정수 오버플로우의 서술 방식을 나타낸다.

[표2]. 행정안전부 C 시큐어코딩 가이드의 정수 오버플로우 서술[1]

보안약점	정수 오버플로우
안전한 코딩기법	메모리할당 함수에서는 할당량을 표시하는 파라미터를 unsigned로 전달해야 한다.
예제	부호있는 정수를 malloc()에 전달
	크기 계산 없 이 배열 인덱스로 사용

[표3]. CERT C Coding Standard 2016의 정수 오버플로우 서술[3]

코딩 룰	내용		
INT32-C	부호있는 정수의 오버플로우가 발생하지 않도록 하라.		
예제	덧셈	뺄셈	곱셈
	나눗셈	나머지	음수 단항 연산자
보안약점	CWE-129, CWE-190		

[표2]와 [표3]에서 볼 수 있듯, 행정안전부 C 시큐어코딩 가이드는 안전한 코딩기법 항목에서 CERT C Coding Standard 2016과는 달리, 근본적인 원칙 대신 특정 상황에서 고려할 내용을 서술하고 있으며 예제 또 한 특 정 상황에서의 활용례를 제시하고 있다. 반면 CERT C Coding Standard는 코딩 룰로 써 근본적인 원칙을 제시하고원칙 이 위반될 수 있는 기본적인 상황에 대한 예제를 제시한다.

그런데 행정안전부 C 시큐어코딩 가이드의 정확한 코딩 룰 없는 서술방식 때문에 안전한 코드 예제마저 원칙을 어겨 제시되었다. 한 가지 사례로 [그림1]의 행정안전부 C 시큐어코딩 가이드에서 제시된 정수 오버플로우의 안전한 코드를 들 수 있다.

[그림1]. 행정안전부 C 시큐어코딩 가이드의 정수 오버플로우 안전한 코드[1]

```
1: #include <stdlib.h>
2: void* intAlloc(int size, int reserve)
3: {
4: void *rptr;
5: unsigned s;
6: size += reserve;
7: s=size* sizeof(int);
8: if(s<0)
9: return NULL;
10: rptr = malloc(s);
11: if (rptr == NULL)
12: exit(1);
13: return rptr;
14: }
```

[표2]의 안전한 코딩기법에 따라 [그림1]에서 unsigned형 변수 s를 선언하고 6번째 줄에서 s=size*sizeof(int)를 계산한다. 하지만 intAlloc 함수의 인자로 받는 size는 int 타입이기 때문에 만약 코드의 6번째 줄을 연산한 결과로 size 값이 음수가 나온다면, s 또한 굉장히 큰 값이 될 수 있다. 물론 s는 unsigned형으로서 항상 양수이기 때문에 코드의 8번, 9번째 줄은 절대 실행되지 않을 것이다. 이렇게 s가 굉장히 큰 값이 된 상태로 malloc함수의 인자로 전달되기 때문에 가이드에서 제시한 보안약점인 정수 오버플로우를 막을 수도 없을 뿐만 아니라, 신뢰성 있는 시스템을 개발함에 있어 큰 문제로 작용한다.

[그림2]는 [그림1]의 결함을 수정하여 완벽한 시큐어코딩을 적용한 결과이다. 코드 변경은 6~9번째 줄과 11번째~15번째 줄이 새롭게 수정된 코드이다. int형 인자 size와 값reserve 의 외부 의존성을 없애고 10번째 줄이 실행되기 이전에 덧셈 정수 오버플로우 여부를 확인한다. 이후 [그림1]의 8~9번째 줄을 [그림2]의 11~12번째 줄로 수정하였다. 그리고 s를 계산하기 전에 곱셈 오버플로우 여부를 확인한다. 모든 예외처리 과정을 거친 후에 비로소 malloc() 함수에 값을 전달한다.

따라서 행정안전부 C 시큐어코딩 가이드에 대한 전반적인 보완이 필요하다.

3.2 보완점 제시

현행 행정안전부 C 시큐어코딩 가이드는 코딩 룰을 제시하지 않으며, 안전한 코드 예제마저 보안약점을 해결할 수 없는 코드로 작성되어 있다. 개정되는 행정안전부 C 시큐어코딩 가이드는 유형화 된 보안약점에 따른 코드 예제 대신, CERT C Coding Standard 2016처럼 원칙적인 코딩 룰과 함께, 코딩 룰이 위반되어 보안약점을 발생시키는 근본적인 예제를 제공해야 한다. 임베디드 시스템 개발에서는 포인터와 메모리 직접 접근을 활용하는 C언어의 사용이 필수적이며 정밀한 개발이 전제되어야 하기 때문에 근본적인 예제가 응용 가능성이 높고, 향후 IoT 환경에서 전자정부를 제공하기 위한 임베디드 시스템의 안전성, 신뢰성, 보안성을 확보할 수 있다.

[그림2]. [그림1]의 결함을 수정한 결과

```
1: #include <stdlib.h>
2: #include <limits.h>
3: void* intAlloc(int size, int reserve){
4:   void *rptr;
5:   unsigned s;
6:   if(((reserve>0)&&(size>INT_MAX-reserve)) ||
7:     ((reserve<0)&&(size<INT_MIN-reserve))){
8:     /*Handle Error*/
9:   }
10:  size += reserve;
11:  if(size < 0)
12:    return NULL;
13:  if(size > UINT_MAX/sizeof(int)){
14:    /*Handle Error*/
15:  }
16:  s=size* sizeof(int);
17:  rptr = malloc(s);
18:  if (rptr == NULL)
19:    exit(1);
20:  return rptr;
21: }
```

Ⅳ. 결론

본 연구에서는 IoT 환경에서 전자정부 시스템 활용을 위해서 행정안전부 C 시큐어코딩 가이드를 적용하는 것이 안전성을 확보함에 있어 부족함을 확인하고, CERT C Coding Standard 2016을 적용하여 C 시큐어코딩 가이드의 개선이 필요함을 역설하고, 보완점을 제시하였다.

[참고문헌]

[1] 행정안전부, 한국인터넷진흥원, C 시큐어코딩 가이드, 2012.

[2] 행정안전부, 전자정부2020기본계획, 2016.

[3] SEI, SEI CERT C Coding Standard 2016 Edition - Ruls for Developing Safe, Reliable, and Secure Systems, 2016.