악성 문서 중심의 이메일 보안 솔루션 개발

(Malicious Document-Based Email Security Solution)

팀 명	ROAT
지도교수	유승재
팀 원	이 경 재 (팀장) 김 근 수 김 정 욱 전 유 병 송 지 현

2025. 11. 4

중부대학교 미래융합공학부 정보보호학전공

목 차

1. 서론

1.1 연구 배경

최근 이메일은 사이버 공격의 가장 보편적 진입점으로 자리잡았습니다. 공격자들은 첨부된 문서 파일을 통해 악성코드나 원격 명령을 실행시키는 방식으로 공격을 수행하며, 특히 한국 환경에서는 HWP/HWPX 문서가 자주 악용됩니다.
기존의 보안 솔루션은 주로 스팸 탐지(SpamAssassin) 또는 서명 기반 백신(ClamAV) 중심이라, 문서 내부의 구조적 위협까지 탐지하는 데 한계가 있었습니다.
이에 따라 본 연구에서는 HWP/HWPX 파일 내부를 정적으로 분석하고, 정적·동적 진단을 통합한 이메일 보안 파이프라인을 구축하고자 하였습니다.

1.2 연구 필요성

이메일을 통한 악성코드 유입은 여전히 가장 빈번한 공격 벡터로 꼽히며, 그 중 HWP/HWPX 문서는 국내 환경에서 특히 자주 악용되고 있습니다.
하지만 기존 백신 및 필터링 시스템은 문서 내부에 삽입된 EPS, WMF 등 임베디드 오브젝트 기반 공격을 제대로 탐지하지 못하는 한계가 있습니다.
이에 본 연구는 HWP/HWPX 파일의 구조적 특징을 기반으로 한 정적 탐지 모델을 구현하고, 이를 자동화된 분석 파이프라인에 통합하여 대응 효율을 높이고자 하였습니다.

1.3 연구 목적 및 주제선정 이유

본 연구의 목적은 이메일 첨부파일 중 HWP/HWPX 문서에 포함된 악성 행위를 정적으로 분석하고,
이를 SpamAssassin, ClamAV, Cuckoo 등 기존 보안 엔진과 연동하여 정확한 악성 판정과 시각화를 수행하는 것입니다.
또한 탐지 결과를 데이터베이스에 저장하여 웹 대시보드 형태로 가시화함으로써,
운영자가 메일 보안 현황을 직관적으로 확인할 수 있는 체계를 구축하고자 하였습니다.

2. 관련 연구

2.1 이메일 보안 기술 동향

이메일 보안은 과거 단순한 스팸 필터링 중심에서 벗어나, 최근에는 첨부파일·링크·행위 기반 탐지로 확장되고 있습니다.
SpamAssassin과 같은 오픈소스 엔진은 메일 헤더와 본문을 기반으로 스팸 여부를 판단하며, ClamAV는 첨부파일 내 바이러스 서명을 탐지합니다.
그러나 이러한 방식은 알려진 악성코드나 서명 기반 공격에는 강하지만, 문서 내부의 구조적 취약점을 악용하는 공격에는 한계가 있습니다.
이에 따라 최근에는 정적 분석과 동적 분석을 결합한 하이브리드 분석 기법이 주목받고 있으며, 본 연구 역시 이 접근을 채택하였습니다.

2.2 문서 포맷 악성코드 분석 사례

EPS(PostScript) 기반 공격은 exec, run, system 등의 연산자를 이용하여 코드 실행을 유도하는 방식이 주로 사용됩니다.
또한 WMF(GDI 기반 이미지 포맷)는 렌더링 중 메모리 손상 취약점을 이용할 수 있으며,
HWPX(OPC 기반 구조)는 .rels 파일의 외부 리소스 참조(TargetMode="External")를 통해 악성 파일을 불러올 가능성이 존재합니다.
이러한 공격 기법들은 기존의 백신 엔진으로는 탐지가 어렵기 때문에, 문서 포맷의 내부 구조를 직접 분석하는 접근이 필수적입니다.

3. 프로젝트 내용

3.1 시스템 구성

본 연구의 시스템은 다음과 같은 흐름으로 구성되었습니다.

1. 이메일 수신(MX) 후 /Maildir/new에 저장
2. header-auto.py로 헤더 추출 및 SpamAssassin 점수 분석
3. body-auto.py로 본문 추출, link-auto.py로 URL 수집
4. attachment-auto.py로 첨부파일 저장 및 압축 해제(zip, 7z, alz 등)
5. clamav.py로 파일 재귀 검사 수행
6. ole.py를 통해 OLE 구조 분석 및 매크로·임베디드 오브젝트 추출
7. hwp-analyzer.py로 HWP/HWPX 구조 분석 및 악성 지표 탐지
8. report.py로 각 결과 통합, final_db.py로 MySQL 적재
9. Flask 웹 UI에서 통계, 점수, 파일별 세부 분석 결과 시각화

3.2 분석기 및 파이프라인 구현

HWP 정적 분석기(hwp-analyzer.py)는 HWP 파일의 FileHeader, DocInfo, BinData 영역을 파싱하고,
BinData 내 임베디드 바이너리(EPS, WMF, PE, ZIP 등)를 식별하였습니다.
EPS 내부의 PostScript 연산자(exec, system, run, file, putinterval 등)를 탐지하고,
zlib으로 압축된 데이터는 해제하여 재검사를 수행하였습니다.
또한 HWPX 구조에서는 ZIP 내부의 /bindata/ 영역과 .rels 파일을 탐색하여 외부 참조를 탐지하였습니다.
정적 분석 결과는 지표별 가중합으로 계산되어 0~10점으로 표현되며,
0~3점은 정상(Benign), 4~6점은 의심(Suspicious), 7점 이상은 악성(Malicious)으로 분류하였습니다.
동적 분석은 Cuckoo Sandbox를 통해 파일을 실행하고, 생성된 JSON 보고서에서 점수 및 악성 행위를 추출하여 통합 리포트에 병합하였습니다.

3.3 정적 가중치 설계

본 절에서는 HWP(OLE)와 HWPX(OPC+ZIP) 문서에 대한 정적 분석 가중치 정책을 정의하였습니다.
가중치는 파일 포맷의 일관성, 임베디드 바이너리 매직, EPS(PostScript) 위험 연산자, 은닉·난독 신호, URL 특성, 파싱 에러, 확장자 위장 여부를 중심으로 설정하였으며,
최종 점수를 기준으로 문서의 악성 여부를 판정하도록 설계하였습니다.

3.3.1 HWP(OLE) 가중치

1. 포맷/컨테이너 일관성

- .hwp인데 OLE 구조가 아닌 경우: +3

→ 정상 HWP는 OLE(Compound File) 구조를 가져야 하며, 불일치는 확장자 위장 가능성이 높다고 판단하였습니다.

- FileHeader 암호화 비트(0x10) 설정 시: +3

→ 내용 은닉 또는 분석 방해 시도로 간주하였으나, 정상 문서에서도 나타날 수 있어 단독 신호로는 판단하지 않았습니다.

2. 임베디드 바이너리/매직 태그

- PE, ZIP, 7z, RAR, EPS, WMF, OLE 등의 매직 태그 탐지 시: +2

→ 문서 내 페이로드 흔적 가능성이 있어 중간 가중치를 부여하였습니다.

- WMF 존재 시 추가 가중치 +2

→ 과거 실전 악용 사례가 다수 존재하여 보수적으로 강화하였습니다.

3. EPS(PostScript) 위험 연산자

- Mild(file, filter, putinterval 등): +2
- Danger(exec, system, run): +3
- 압축 해제 후 EPS 연산자 재등장: +3

→ zlib 등으로 은닉 후 재등장할 경우 탐지 회피 정황으로 간주하였습니다.

4. 콘텐츠 난독·은닉(보조 지표)

- 고엔트로피(≥7.5): +1
- 대용량 블롭(≥200KB): +1

→ 정상 이미지나 템플릿에서도 나타날 수 있으므로 보조 지표로만 사용하였습니다.

5. URL/링크 특성

- IP 기반 URL: +2
- 단축 URL: +1
- 비표준 포트 사용: +1

→ 수집된 URL의 개수는 평가하지 않고 위험 특성만 반영하였습니다.

6. 파싱 에러

- Parse error 발생 시: +1

→ 문서 파손을 통한 분석 회피 가능성이 있다고 판단하였습니다.

7. 확장자 위장

- .hwp인데 실제 매직이 ZIP인 경우: +2 → HWPX 로직으로 전환하였습니다.
- .hwp인데 실제 매직이 MZ(PE)인 경우: +9 → 즉시 악성으로 판정하였습니다.
- .hwp인데 원본이 EPS인 경우: 0점에서 시작 후 EPS 연산자·URL 신호로 가중치를 추가하였습니다.

3.3.2 HWPX(OPC+ZIP) 가중치

1. ZIP 엔트리 및 바이너리 검사

- /bindata/, /embeddings/ 등에서 위험 포맷 매직 탐지 시: +2
- WMF 존재 시 추가 가중치: +2

- EPS 연산자 탐지 시(mild +2 / danger +3)
- 고엔트로피 +1, 대용량 +1
- 실행성 확장자(.exe, .dll, .js 등) 포함 시: +2

→ 실행성 징후가 누적될수록 점수가 상승하도록 설계하였습니다.

2. External 관계

- .rels 파일 내 TargetMode="External" 존재 시: +2

→ 외부 리소스 연결 시도를 탐지한 경우로 간주하였습니다.

3. URL/링크 특성

- IP 기반: +2, 단축: +1, 비표준 포트: +1

→ HWP와 동일한 정책을 적용하였습니다.

4. 파싱/ZIP 문제

- BadZip 또는 기타 오류 발생 시: +1

→ 분석 회피 또는 파일 손상 가능성으로 간주하였습니다.

3.3.3 공통 판정 기준

Malicious (악성): 총점 7점 이상
Suspicious (의심): 4~6점
Benign (정상): 0~3점
위 기준에 따라 각 파일의 위험도를 평가하였습니다.

3.3.4 가중치 설계 근거 요약

1. EPS/PostScript의 위험성

EPS는 단순한 이미지 포맷이 아닌, 명령 실행이 가능한 프로그래밍 언어 구조를 가지고 있습니다.
exec, run, system 등 명령 실행 연산자와 file, currentfile, filter 등의 파일 조작 연산자를 통해 외부 명령 실행이나 파일 접근이 가능하였습니다.
이에 EPS 관련 연산자에는 높은 가중치를 부여하였습니다.

2. OLE/Packager(Ole10Native)의 위험성

HWP 문서는 OLE 기반 구조로, 내부에 실행 가능한 개체를 임베딩할 수 있습니다.
특히 Ole10Native나 ObjectPool 등은 사용자의 클릭만으로 EXE나 스크립트가 실행될 수 있어 높은 보안 위험으로 평가하였습니다.

3. WMF/EMF의 위험성

Windows GDI 기반 포맷인 WMF는 과거 다수의 코드 실행 취약점이 보고된 바 있습니다.
이에 해당 매직을 탐지하면 추가 가중치를 부여하였습니다.

4. HWPX의 External 관계 위험성

HWPX 구조의 .rels 파일 내 외부 참조는 클릭 유도형 공격이나 C2 연결로 이어질 수 있어 중간 수준의 가중치를 부여하였습니다.
특히 IP 기반·단축 URL·비표준 포트를 사용하는 경우 추가로 가중하였습니다.

5. 엔트로피 및 대용량의 보조 신호화
고엔트로피·대용량은 은닉이나 압축 흔적을 의미할 수 있지만, 정상 이미지에서도 발생할 수 있어 보조 신호로만 취급하였습니다.

6. 압축 해제 후 재검출 강화
공격자가 EPS나 URL을 zlib 등으로 감춘 뒤 BinData에 은닉하는 경우가 많습니다.
따라서 압축 해제 후 동일한 지표가 다시 검출될 경우 추가 가중치(+3)를 부여하였습니다.

7. 확장자 위장 탐지
메일 게이트웨이나 사용자를 기만하기 위해 .hwp 확장자를 사용하면서 실제 헤더가 MZ(실행 파일)인 사례가 존재하였습니다.
이 경우 즉시 9점을 부여하여 악성으로 판정하였습니다.

3.3.5 운용 지침
초기 운용 단계에서는 EPS mild 및 보조 지표의 가중치를 낮게 유지하고,
EPS danger, External, PE, WMF 등의 강한 지표를 중심으로 탐지를 수행하였습니다.
운용 중 수집된 오탐 사례를 기반으로 주기적으로 가중치를 조정하며,
가중치와 룰은 외부 설정 파일로 분리하여 핫스왑이 가능하도록 관리하였습니다.

3.4 DB 및 웹 대시보드 연계
report.py에서 SpamAssassin, ClamAV, Hex, Cuckoo, hwp-analyzer의 결과를 통합한 후
final_db.py가 MySQL 데이터베이스에 적재하였습니다.
Flask 기반 웹 인터페이스는 파일명, 점수, 악성 지표, 기간별 통계 등을 시각적으로 표현하며, Cuckoo Sandbox의 동적분석 결과 보고서를 다운로드 받을 수 있습니다.

4. 결론

4.1 결론 및 기대효과
본 연구를 통해 이메일을 통한 HWP/HWPX 악성 문서의 정적 분석과 시각화를 통합적으로 수행할 수 있는 시스템을 구축하였습니다.
기존 백신과 스팸 필터가 탐지하지 못하는 문서 내부의 위협을 정확히 식별하였으며,
정적·동적 분석의 결합으로 탐지 신뢰성을 높이고 운영 효율성을 개선하였습니다.

본 시스템은 향후 공공기관·기업의 메일 보안 체계 강화에 기여할 수 있을 것으로 기대됩니다.

4.2 향후 계획

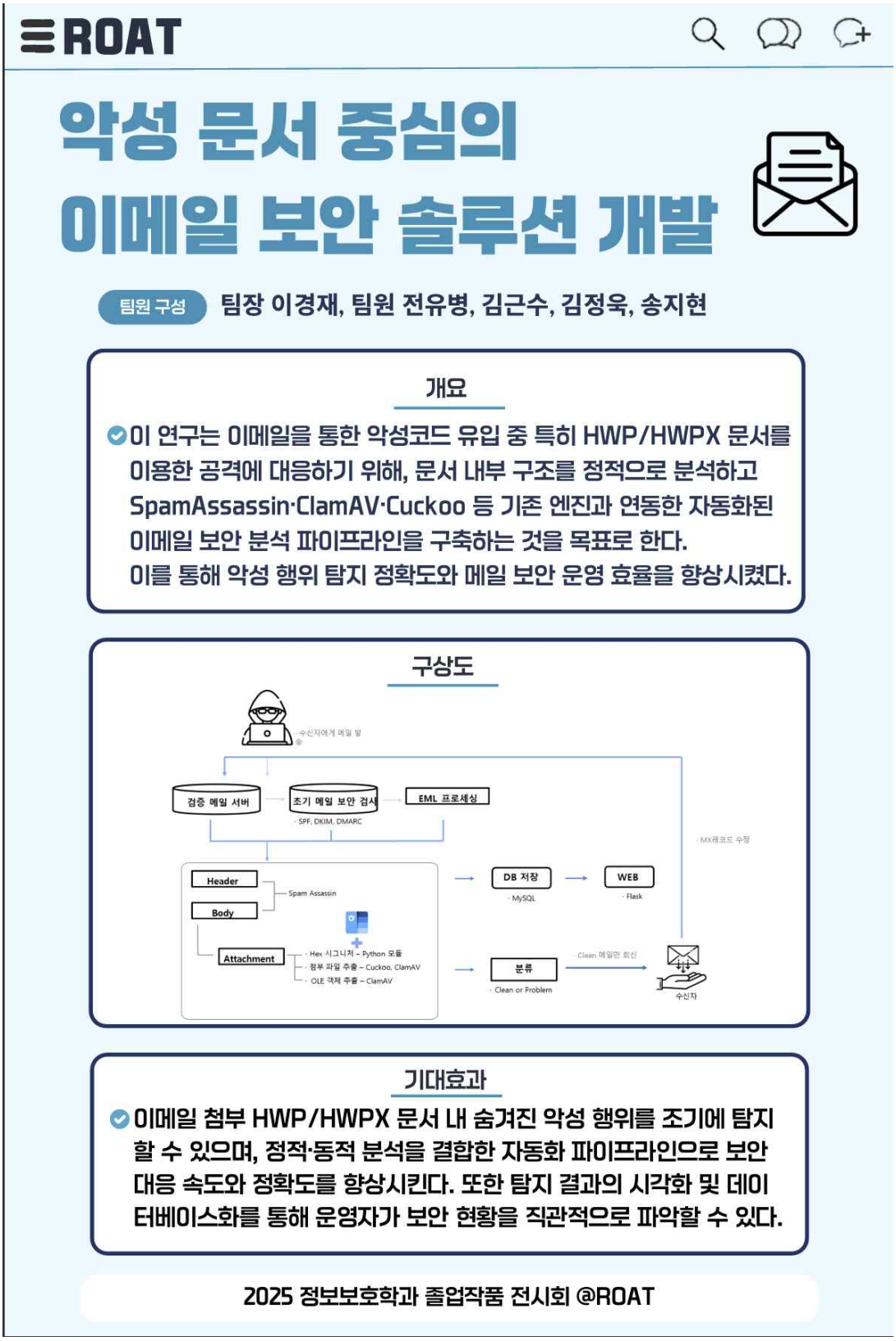
HWPX 악성 실 샘플을 확보하여 룰셋을 확장하고,
가중치 및 탐지 규칙을 외부 설정 파일로 분리하여 실시간 업데이트가 가능하도록 개선할 예정입니다..
또한 HWP 렌더링 엔진을 이용한 반(半)동적 분석 기능을 추가하여,
문서 실행 중 발생하는 행위를 자동으로 수집·기록할 수 있도록 시스템을 고도화할 계획입니다.

5. 참고 자료

(1) 한국인터넷진흥원(KISA), 「한글문서 포맷 취약점 분석 백서」, 2023
(2) Cuckoo Sandbox Documentation v2.0
(3) SpamAssassin Rule Guide
(4) ClamAV Engine Manual v1.2
(5) CERT/CC, EPS 기반 악성코드 사례 보고서, 2022

6. 별첨

6.1 소개 자료 (폼보드)



6.2 프로젝트 소개

프로젝트 시연 영상

https://drive.google.com/file/d/172T4bSyhIDqcm6BvxmhFQa82mAlecz56/view?usp=sharing

6.3 팀 소개