학사 학위논문

안전한 리눅스 사용을 위한 보안 기술에 관한 연구

A Study on Security Techniques for the Safe Use of Linux

이돈규(李敦圭)

정보보호학과

공과대학

중부대학교

2007

안전한 리눅스 사용을 위한 보안 기술에 관한 연구

A Study on Security Techniques for the Safe Use of Linux

A Study on Security Techniques for the Safe Use of Linux

Advisor : Professor Byoungcheon Lee

by

Donkyu Lee

Department of Information Security

School of Engineering

Joongbu University

A thesis submitted to the faculty of Joongbu University in partial fulfillment of the requirements for the degree of Bachelor of Science in the School of Engineering

Chubu, Chungnam, Korea

Nov. . 2006

Approved by

_____

Professor Byoungcheon Lee

Major Advisor

안전한 리눅스 사용을 위한 보안 기술에 관한 연구

이 돈 규

위 논문은 중부대학교 학사학위논문으로 지도교수위원회의 논문심사에 통과되었음을 인정합니다.

2006년 11월 일

지도교수위원회 위원장 이 병 천 (인)

위 원 양 정 모 (인)

위 원 유 승 재 (인)

D.K. 이돈규 Donkyu Lee

99912387

안전한 리눅스 사용을 위한 보안 기술에 관한 연구

A Study on Security Techniques for the Safe Use of Linux

School of Engineering, 2006, 49p.

Major Advisor : Prof. Byoungcheon Lee

논문초록

최근에 인터넷과 같은 개방형 네트워크상에서의 다양한 정보 제공을 위해 홈페이지 등을 만들 때 oracle등의 데이터를 저장 할 수 있는 연동 소프트웨어의 사용이 증대되고 있는 상황이지만 oracle등의 연동 소프트웨어는 자체적인 보안관리가 안되므로 oracle등의 연동 소프트웨어는 사용 중인 OS로부터 보안이 되어야 연동소프트웨어에 저장된 데이터를 안정적으로 관리할 수 있다. 이러한 사항 속에서 데이터를 안정적으로 관리하기 위해서는 기밀성, 통신 상대방의 인증 및 무결성 확보, 익명성 제공 등과 같은 보안 서비스 이용이 필수적이다. 본 논문에서는 데이터에 대한 안정적인 관리를 위해 연동소프트웨어가 주로 사용하는 OS인 LINUX에 대한 보안 기술에 대해 관련 법규, 보안기술, 해킹 방법, 보안 기술의 미래 등에 대해 연구하였다. 효율적으로 LINUX 보안하기 위해 기존문서 등에서 사용하기 어렵게 설명 한 것에 대해 보다 쉽게 기술하였고 이에 대해 system오류 등에 대해 대처 방안을 연구하였다.

현재 사용하고 있는 OS인 LINUX에 대한 문제점을 분석하여 이에 대

해 해결 방법에 대해 기술하였다. 그리고 네트워크 보안 아키텍처등과 같이 기술적인 부분에 대해서도 연구 및 그에 대해 기법도 기술하였다.

목차

그림목차

표목차

약어목록

APR : Alternate Path Retry

BOF : Buffer Overflower

CIPE : Cryptographic IP Encapsulation

CFS : Cyber Forum System

DoS : Denial of Service

DDoS : Distributed Denial-of-Service

FSB : Format String Bug

GGI : Generic Graphics Interface project

IPSEC : IP security protocol

IPS : Instruction Per Second

PPP : Point-to-Point Protocol

PAM : Pluggable Authentication Modules, 장착 방식 인증 모듈

SSL : Secure Sockets Layer

S-HTTP 기법 : multiple key management mechanisms, 다중 열쇠관리 기법

SMIME : Secure Multipurpose Internet Mail Extension

SSH : Secure SHell

SRP : Secure Remote Password Protocol

SNMP : Simple Network Management Protocol

SSL : Secure Sockets Layer

TCP/IP : Transmission Control Protocol/Internet Protocol

VPN : Virtual Private Network, 가설사설망

Ⅰ. 서　　론

1.1. 연구배경

인터넷과 같은 개방형 네트워크상에서의 다양한 정보 제공을 위해 홈페이지 등을 만들 때 oracle등의 데이터를 저장 할 수 있는 연동 소프트웨어의 사용이 증대되고 있는 상황이지만 oracle등의 연동 소프트웨어는 자체적인 보안관리가 안되므로 oracle등의 연동 소프트웨어는 사용 중인 OS로부터 보안이 되어야지 연동소프트웨어에 저장된 데이터를 안정적으로 관리를 할 수 있다. 이러한 사항 속에서 데이터를 안정적으로 관리하기 위해서는 기밀성, 통신 상대방의 인증 및 무 결성 확보, 익명성 제공 등과 같은 보안 서비스 이용이 필수적이다. 본 논문에서는 데이터에 대한 안정적인 관리를 위해 연동소프트웨어가 주로 사용하는 OS인 LINUX에 대한 보안 기술에 대해 관련 법규, 보안기술 해킹 방법, 보안 기술의 미래 등에 대해 연구가 필요하다.

1.2. 연구 범위

보다 안정적인 데이터 관리를 위한 LINUX사용의 보안에 대해 관련 법규에 대해 표로 구성하여 이에 대해 경각심을 알리며 보안의 필요성과 어떠한 것을 보안해야하며 그 관리에 대해 살펴보기로 한다.
호스트 보안등과 같은 보안의 종류와 정의, LINUX 상에 사용 할 수 있는 보안 방법에 대한 tip과 현재 사용하고 있는 보안 기술의 종류인 PGP와 공개 열쇠 암호 기법 (Public Key Cryptography)등의 정의, 보안기술의 성능과 문제점에 대해 살펴보기로 한다.
또한 해커나 크래커, 악의적인 목적을 가진 사람들이 주로 사용 하는 해킹 방법인 DoS, DDoS 등에 대해 정의 및 어떻게 해서 피해를 입게 되는지에 대해 tip과 함께 제시한다.

또한 현재 LINUX 보안 기술에 대한 문제점에 분석을 하고 그에 대한 대체 방안과 미래 LINUX 보안 기술 동향을 [그림 3] 보안 플랫폼의 발전 등을 분석하고 발전 방향을 제시한다.

1.3. 논문구성

본 논문의 구성은 다음과 같다. 먼저 II장 보안의 수칙과 보안의 방법에서는 보다 안정적으로 데이터를 보호하기 위해 우리가 모르고 지나갈 수 있는 보안 수칙과 그에 대한 법률 사항 및 보안 방법에 대해 에 살펴보고, III장에서는 LINUX 보안기술의 종류와 설명 그에 대한 간단한 tip을 제공하고 이에 대해 제시하였다. 또한 여러 가지 해킹 기법을 제시하였다.

IV, V장에서는 미래의 LINUX 보안 기술의 미래와 대체 방안 및 동향에 대해 기술하였고, VI장에서는 결론 및 앞으로의 연구 방향에 대해 서술하였다.

1.4. 왜 보안이 필요한가?

항시 변화하는 글로벌 데이터 커뮤니케이션의 세계에서, 그리고 값싼 인터넷 연결이 가능한 현재서, 또한 빠르게 움직이는 소프트웨어 개발에 있어서, 보안은 갈수록 중요한 문제로 떠오르고 있다.

컴퓨터라는 것이 개발 초기부터 보안을 염두를 두고 만든 것이 안이기에 보안은 근래에 와서야 기본적 필요조건으로 등장하게 되다. 나쁜 보기를 들자면, 인터넷 상에서 한 데이터가 A 지점에서 B 지점으로 흐르는 중간의 여러 다른 지점에서, 다른 사용자들이 데이터를 가로 채거나 심지어 변경해 버릴 수 있는 기회를 갖게 된다. 심지어 여분의 시스템에 있는 다른 사용자들이 여러분의 데이터를 (여러분이 의도하지 않은 다른 어떤 것으로) 악의로서 변경할 수도 있을 것이다.

"크래커"라고도 불리는 침입자들에 의해서 여러분의 시스템이 무단

사용될 수도 있으며, 이들은 뛰어난 지식을 악용함으로서 여러분인 척 신분을 위장하거나, 정보를 훔치거나, 또는 여러분이 여러분 시스템을 사용하고자 함을 거부할 수 있다.

1.5. 얼마나 안전한 것이 안전한 것인가?

우선 마음에 새겨 두어야 할 것은 어떠한 시스템도 "완벽하게 안전" 할 수 없다는 것이다. 여러분이 할 수 있는 최선의 방법은 여러분의 시스템에 침입하는 일을 가능한 어렵게 만드는 것뿐이다. 평균의 가정용 리눅스 사용자로서는 크래커에 대비하기 위해서 그리 많은 것이 필요하지 않다. 하지만 (은행, 통신 회사 등의) 위치가 노출된 잘 알려져 있는 리눅스의 사용자들 은 보다 많은 작업을 해야 한다.

계산에 두어야 할 사항은 시스템의 보안을 강화하면 할수록 시스템을 쓰기에 불편하게 된다는 것이다. 여러분은 시스템을 사용해야 하는 사용자의 관점에서도 시스템을 봄으로서 보안성과 편의성에 대한 균형을 잡아야 할 것이다. 예로서, (보안의 입장을 고려해서) 여러분의 시스템에 모뎀으로 접속해 들어오는 사용자 모두에게 call - back 모뎀을 쓰도록 할 수도 있을 것이다.

비록 이 방법을 쓰면 보안을 보다 강화할 수는 있겠지만, 사용자의 입장에서는 로그인을 하기에 불편하게 만드는 것이 된다. 또는, 네트워크이나 인터넷에 아예 연결되지 않게 리눅스 시스템을 만들 수도 있겠지만, 이것은 유용성을 제한하게 되는 것일 것이다.

만약 중간 규모 이상의 대형 사이트라면, 어떤 수준의 보안이 필요하고 이것을 점검하기 위해서는 어떤 절차 검사가 필요한 것인가를 밝히는 보안 수칙을 준비하는 것이 좋다.

1.6. 무엇을 보호할 것인가?

어떤 위험에 대비해야 할 것인가, 어떤 위험부담을 감수하거나 감수하지 않을 것인가, 그렇다면 결과적으로 시스템이 얼마나 취약하게 되는 것인가 등을 미리 생각해 놓는 것이 좋을 것이다. 무엇을 보호하는가, 왜 보호하는가, 이 보호 대상의 가치는 얼마나 되는가, 그리고 데이터와 자산에 대해서 누가 책임을 질 것인가를 분석하자.

위험 요소(risk)란 침입자가 시스템을 성공적으로 침입하는 경우를 암시한다. 여러분에게 중요한 파일을 침입자가 읽거나 쓰고, 심지어는 프로그램을 실행할 수 있는 가? 중요한 데이터를 지울 수 있는가? 여러분이나 여러분의 회사가 중요한 업무를 실행하는 것을 훼방 놓을 수 있는가? 또한 여러분의 계정이나 시스템에 접근을 할 수 있는 사람이 여러분을 사칭할 수도 있다는 것을 잊지 말아야 한다.

또한, 보안이 취약한 계정 한 개 때문에 전체 네트워크가 침입을 당하는 결과가 생길 수도 있다. 설사 단 한 명의 사용자라 하더라도 리모트 호스트 (r-host)의 사용을 허락해 주거나, 혹은 tftp 등의 보안이 불완전한 서비스의 사용을 허용함으로 침입자에게 "발 디딜 자리"를 주는 결과가 생기는 것이다. 침입자가 여러분 시스템이나 다른 시스템에 사용자 계정을 갖은 순간부터 이 계정은 다른 계정이나 다른 시스템으로의 접근을 얻는데 사용될 수 있다.

(1)위험 요소(threat)는 여러분 네트워크나 컴퓨터에 불법 접근(unauthorized access)을 얻고자 하는 생각이 있는 사람으로부터 생성된다. 누구를 신임해서 여러분의 시스템에 접근을 허락할 것인가, 어느 정도의 위험부담을 그 사람이 발생 시키는가 등을 잘 결정해야 한다. 침입자라는 집단은 여러 부류로 나누어지며, 보안 작업을 실행 할 때에는 그들 각 각의 특성을 염두에 두는 것이 좋다.

(2)궁금증이 많은 사람 (The Curious): 이 종류의 침입자는 기본적으

로 여러분이 어떤 시스템과 데이터를 가지고 있는 가 정도를 알고자 하는 것에 흥미를 둔다.

(3)악의가 있는 사람 (The Malicious): 이 종류의 침입자는 여러분의 시스템을 다운시키거나 여러분의 웹 페이지를 손상시키거나, 손해를 복구하게 하는 등으로 여러분의 시간과 돈을 낭비하게 만든다.

(4)명성을 얻으려는 사람 (The High-Profile Intruder): 이 종류의 침입자는 인기와 악명을 얻기 위해서 여러분의 시스템을 쓰려고 한다. 잘 알려진 시스템을 침투함으로서 자신의 능력을 선전하려고 하는 것이다.

(5)경쟁 자 (The Competition): 이 종류의 침입자는 여러분 시스템에 무슨 데이터가 있는가에 흥미를 둔다. 돈이 될 만한 무엇을 여러분이 가지고 있다고 생각하는 불특정인 일수도 있다.

(6)도용 자 (The Borrowers): 이 종류의 침입자는 그의 목적을 위해서 여러분 시스템을 무단 사용하면서 여러분의 시스템 자원을 훔치는 사람이다. 이들은 일반적으로 채팅이나 IRC 서버, 포르노 아카이브 사이트 등을 여러분의 컴퓨터에서 돌리고, 심지어는 DNS 서버를 돌리 기까지 한다.

(7)건너뛰기 도용 자 (The Leapfrogger): 이 종류의 침입자는 다른 시스템으로 들어가기 위한 도구로서 여러분의 시스템을 이용하려고 한다. 만약 여러분의 시스템이 많은 수의 호스트에 연결되어 있거나 게이트웨이로 사용되는 경우라면, 이런 분류의 침입자가 여러분 시스템을 깨고 들어오려는 노력을 하는 것을 이미 보았을 수도 있을 것이다.

- "취약성 (Vulnerability)" 이 있다 함은 여러분의 컴퓨터가 다른 네트워크로부터 보호가 되지 않거나, 누군가가 여러분 컴퓨터에 불법 접근을 얻을 가능성이 있는 경우를 뜻한다.

- 여러분의 시스템에 누군가가 침입했다면 무엇이 상관된 문제일까? 물론 다이내믹 PPP를 사용하는 개인 사용자의 관심은 인터넷이나 다른 큰 네트워크에 연결된 회사의 관심사와는 다르기는 하겠지만 말이다.

- 손상된 데이터의 횟수와 복구에 얼마나 시간이 걸릴 것인가? 초기의 조그만 시간 투자는 잃어버린 데이터를 회복하는데 낭비되는 시간의 십분의 일도 안 될 수 있다. 근래에 백업 전략을 점검하거나 백업된 데이터를 확인한 적이 있는지?

Ⅱ. 보안의 수칙과 보안의 방법

2.1. 보안 수칙

사용자들이 쉽게 이해하고 따를 수 있는 간단하고 일반적인 수칙을 만들도록 해야 한다. 수칙은 관리자 여러분이 수호하는 데이터를 보호하는 동시에, 사용자의 프라이버시도 지키도록 만들어져야 한다. 숙고해야 할 것들은 누가 시스템에 접근을 가질 것인가 (친구들이 내 계정을 써도 될 것인가?), 누가 시스템에 소프트웨어를 설치하도록 허락될 것인가, 누가 어떤 데이터를 소유할 것인가 등과, 재해 시의 복구 대책, 시스템의 적절한 사용 등이다.

일반적으로 이용되고 있는 보안 수칙은 다음의 문장으로 시작 된다: 허락되어 있지 않은 사항은 금지 사항으로 간주할 것. 이것은 시스템 관리자 여러분이 허락하지 않은 시스템 서비스를 일반 사용자가 사용을 하면 않 된다는 뜻이다. 이 수칙은 관리자 여러분의 일반 계정에조차도 적용이 되도록 해야 할 것이다. "도대체 이것의 허가권은 귀찮구먼. 그냥 루트로 들어가서 해 버리지 뭐" 하는 따위는, 너무도 당연히 알려져 있는 침입 법에 사용됨은 물론이고 아직 발견조차 되지 않은 침입 법까지 발전 사용될 보안 개구멍을 열어 놓는 것과 다름없는 것이다.

rfc1244는 네트워크 보안 수칙을 만드는 방법을 설명해 주고 있다.

rfc1281은 보안 수칙의 예제를 각 단 계의 자세한 설명과 함께 설명해 주고 있다.

마지막으로, ftp://coast.cs.purdue.edu/pub/doc/policy에 있는 COAST 아카이브를 가보면, 실제 사용되고 있는 보안 수칙이 어떻게 만들어져 있는 볼 수 있다. 그리고 현재 우리나라의 정보보호 보안 법률제도를 알아보면 아래의 표와 같다

표 1. 정보보호 보안 관련 법률(1)

형법 (법률 제5057호)		
구 분	**법 률 조 항**	**처 벌 형 량**
데이터 부정조작,변조	제227조의 제2항 (공전자기록등의 위작,변작)	10년 이하의 징역
	제228조(공전자기록등 부실기재)	5년 이하의 징역 또는 1천만원이하의 벌금
	제229조(위작공전자기록등행사)	10년이하의 징역
	제 232조의2(사전자기위작,변작)	5년 이하의 징역 또는 1천만원이하의 벌금
	제234조(위작사전자기록등 행사)	
업무방해	제314조 제2항 (컴퓨터등 장애 업무방해)	5년이하의 징역 또는 1.500만원이하의 벌금
컴퓨터 사기	제347조 제2항 (컴퓨터등 사용사기)	10년이하의 징역 또는 2천만원이하의 벌금
비밀침해	제140조 제3항 (공무상 비밀전자기록등 내용탐지)	5년이하의 징역 또는 700만원이하의 벌금
	제 316조 2항 (전자기록등 내용탐지)	3년이하의 징역이나 금고 또는 500만원이하의 벌금
전자기록 손괴 및 은닉	제366조(전자기록등 손괴)	3년이하의 징역이나 금고 또는 700만원이하의 벌금
	제 141조 1항 (공용 전자 기록등 손상)	7년이하의 징역 또는 7천만원이하의 벌금

공업및에너지기술기반조성에관한법률(법률 제5281호)		
구 분	**법 률 조 항**	**처 벌 형 량**
산업정보 위조. 변조	제22조 제1항(벌칙)	10년이하의 징역 또는 1억원이하의 벌금
산업정보 훼손. 비밀침해	제22조 제2항 제1호(벌칙)	5년이하의 징역 또는 5천만원이하의 벌금

신용정보의이용및보호에관한법률(법률 제4866호)		
구 분	**법 률 조 항**	**처 벌 형 량**
신용정보 변경. 검색.삭제	제32조 제11호(벌칙)	3년이하의 징역 또는 3천만원이하의 벌금

표 2. 정보보호 보안 관련 법률(2)

전산망보급확장과 이용촉진에관한법률(법률 제5219호)		
구 분	법 률 조 항	처 벌 형 량
전산망 보호조치 침해.훼손	제30조의 2(벌칙)	3년이하의 징역 또는 3천만원이하의 벌금
전자문서 위작. 변작.행사	제29조 제1항(벌칙)	10년이하의 징역 또는 1억원이하의 벌금
타인의 정보 훼손. 침해.도용	제30조(벌칙)	5년이하의 징역 또는 5천만원이하의 벌금

전기통신사업법(법률 제5220호)		
구 분	법 률 조 항	처 벌 형 량
통신비밀침해.누설	제70조 제4호(벌칙)	3년이하의 징역 또는 3천만원이하의 벌금

화물유통촉진법(법률 제5160호)		
구 분	법 률 조 항	처 벌 형 량
물류정보 위조.변조	제54조의 2(벌칙)	10년이하의 징역 또는 1억원이하의 벌금
물류정보 훼손. 비밀침해	제54조의 3(벌칙)	5년이하의 징역 또는 5천만원이하의 벌금
전산망 보호조치의 침해.훼손	제54조의 4(벌칙)	3년이하의 징역 또는 3천만원이하의 벌금

무역업무자동화촉진에관한법률(법률 제5211호)		
구 분	법 률 조 항	처 벌 형 량
무역정보 위조. 변조	제25조 제1항(벌칙)	1년이상 10년이하의 징역 또는 1억원이하의 벌금
무역정보 훼손. 비밀침해	제26조 제3호(벌칙)	5년이하의 징역 또는 5천만원이하의 벌금

2.2. 보안의 방법

2.2.1. 호스트 보안

일반적으로 대부분의 컴퓨터 보안 모델은 호스트 보안에 관계된 것이다. 보안 모델을 바탕으로 각 호스트에 대한 정보보호를 강화할 수 있으며, 호스트에 영향을 미치는 정보보호 위협 요소들을 막을 수 있다. 현재 전산 환경에서 호스트 정보보호에 가장 큰 방해 요인은 전산 환경의 다양성과 복잡성이다.

여러 컴퓨터 개발 업체에서 생산되는 컴퓨터 시스템들이 서로 다른 운영체제에서 작동하며, 정보 보호 상의 문제도 제품마다 상이하다. 동일한 업체에서 제공되는 시스템이라 할지라도 운영 체제의 버전이 바뀜에 따라 정보보호 문제점들이 서로 다르다. 제공하는 업체와 운영체제가 동일할지라도 내부 시스템 구성, 응용 프로그램, 서비스 등이 다를 경우에도 정보보호 문제들이 서로 다를 수 있다. 또한 업체에서 제공되는 소프트웨어내의 결점(버그)들도 호스트의 보안성을 약화시킬 수 있다. 이러한 이유 때문에 호스트의 보안성을 유지하는데 많은 노력과 어려움이 따르게 된다.

호스트 정보보호 모델은 정보보호 요구사항이 명확하거나 규모가 작은 네트워크에 더 적합하다. 실제로 모든 네트워크는 자체 정보보호 계획에 호스트 정보보호에 대한 사항을 포함하고 있다. 비록 네트워크 정보보호 모델을 채택하고 있다 하더라도 일부 시스템은 호스트 정보보호 모델이 적용 되어야 한다. 예를 들어 내부 네트워크에 대해 방화벽 시스템을 설치하였다 하더라도 외부 망에 노출되어 있는 시스템에 대해서는 호스트 정보보호 모델이 적용되어야 한다.

호스트 정보보호는 시스템에 대한 전문적인 지식을 갖고 있고 특별한 접근 권한이 부여된 사용자에 의해 관리되어야 한다. 시스템의 수가 증가할수록 특별 권한을 가진 사용자의 수도 자연히 증가하게 된다. 호스

트를 안전하게 보호하는 것이 호스트를 네트워크에 연결시키는 것보다 더 어려우며, 네트워크에 연결된 호스트 중 안전하지 않은 호스트에 의한 정보보호 침해가 전체 네트워크에 막대한 피해를 줄 수 있다.

2.2.2. 네트워크 보안

네트워크에서는 하나의 메시지가 한쪽 망에서 다른 쪽 망으로 전송된다. 메시지 전송과정에서 양쪽의 망은 메시지의 정확한 교환을 위해 상호 협조해야 한다. 즉, 양쪽 호스트 간에 메시지 전송 경로와 통신 프로토콜에 대한 협조가 이루어져야 한다. 정보보호 침해로부터 전송 메시지를 보호하기 위해서는 데이터 비밀성과 사용자 인증 등에 대한 적절한 정보보호 대책이 구현되어야 한다. 확실한 메시지 전송을 위해서는 신뢰할 수 있는 제3의 주체가 필요하다. 제3의 주체는 네트워크 침입자가 알지 못하도록 양쪽 망간에 정보를 책임지고 전송하는 역할을 한다.

일반적으로 네트워크 정보보호의 가장 큰 위협은 허가되지 않은 사용자가 실수 또는 고위로 시스템에 침입하여 시스템 자원을 손상시키는 것이다. 또 다른 유형의 위협은 시스템 취약성을 이용하여 프로그램 에디터나 컴파일러 같은 유틸리티 프로그램에 영향을 줄 수 있는 로직을 컴퓨터 시스템에 설치하는 것이다. 아래 두 종류의 위협이 프로그램에 의해 주어질 수 있다.

(1) 정보 접근 위협 : 접근이 허용되지 않는 사용자가 자료를 가로채거나 수정하는 것.

(2) 서비스 위협 : 컴퓨터 시스템의 서비스 결함을 이용하여 정당한 사용자의 사용을 방해하는 것.

허가되지 않은 접근을 막을 수 있는 가장 효과적인 정보보호 대책은 허가된 사용자 외에는 모든 접근을 차단하도록 설계된 방화벽 시스템을 양쪽 시스템 사이에 설치하는 것이다. 접근이 허가된 사용자에 대해

서도 접근이 이루어진 후에 사용자의 활동 상황을 감시하고 활동 상황을 로그 하는 여러 가지 내부 통제가 필요하다.

2.2.3. 물리적인 보안

2.2.3.1. Hard Ware를 이용

현재 우리가 사용하는 많은 컴퓨터에는 열쇠로 잠글 수 있는 기능을 가지고 있어서 이것을 이용하여 사용하는 방법이다. 또한 부팅을 제한할 수 있는 기능도 있고 시스템내의 BIOS를 이용하면은 시스템의 부팅 시 패스워드를 물어 볼 수 있게 설정을 할 수도 있을 것이다.

2.2.3.2. Lock(x lock & v lock)

만약에 리눅스를 관리하는 도중에 자리를 비울 일이 생길 것이다. 그러면 다른 사람이 이 컴퓨터를 건들지 못하도록 방지를 할 수가 있다. x lock은 엑스윈도우의 화면은 잠그는 것이고 v lock은 가상 터미널을 잠그는데 사용을 하는데 둘 다 그냥 터미널에서 실행을 함으로써 잠궈지는 것이다. 우선 vlock를 먼저 살펴보자 아래의 예처럼 실행을 하면 간단하게 잠글 수 있다.

```
{{{{[lee@linux:~$]vlock
*** This tty is not a VC (virtual console). ***
*** It may not be securely locked. ***
This TTY is now locked.
Please enter the password to unlock.
hwinnt\'s Password: }}}}
```

위의 두 줄은 네트워크상에서 텔넷으로 접속을 해서 이 터미널이 가상터미널이 아니라고 나온 것인데 직접 리눅스에서 명령어를 주게 되면 alt + function key로 터미널을 바꾸어서 사용 할 수 있다는 말이 나

올 것이다. 이렇게 vlock 이라는 명령어를 주게 되면 은 가상 콘솔을 잠궈 버린다. 다시 프롬프트로 돌아 가려하면은 패스워드를 알아야 할 것이다. vlock 의 기본적인 옵션은 다음과 같다.

-a,--all 모든 가상 터미널을 잠그고 싶을 때에 사용하는 옵션이다.

-c,--current 현제의 터미널을 닫힐 수 있는 옵션인데 이것은 기본 옵션이다.

-h,--help 도움말 출력을 할 때 사용 하는 옵션이다.

-v,--version vlock에 대한 버전을 출력 한다.

xlock은 터미널에서와는 달리 사운드 지정 배경 선정 등 많은 옵션을 가지고 있는데 이것은 맨 페이지를 참고 해야 할 것이다(옵션이 너무 많으므로). 또한 엑스 윈도우에서 화면 잠금 이라는 메뉴를 찾아보면 있을 것이다. 이것은 엑스 윈도우 관리자에 따라 모양이 많이 틀 릴 것이다.

2.2.4. System의 Log 파일 처리

리눅스에서 침입자의 가 있었는지 알아보는 것이 간단하게 로그 파일을 보는 것으로도 알 수 있다. 사용자가 log 파일에서 확인해야 할 사항은 다음과 같다.

(1) 짧거나 불완전한 기록.

(2) 이상한 시간 표시를 가진 기록.

(3) 잘못된 허가권이나 소유권을 가진 기록.

(4) 재부팅이나 서비스의 재시작에 대한 기록.

(5) 없어진 기록.

(6) su 사용기록과 이상한 곳으로부터의 접속 기록.

Ⅲ. 리눅스보안기술의 종류와 설명

3.1. 리눅스보안기술의 종류와 설명

3.1.1. PGP와 공개 열쇠 암호 기법 (Public Key Cryptography)

PGP (PGP Pretty Good Privacy) 등에 사용되고 있는, 공개 열쇠 암호 기법은 하나의 열쇠로 암호화하고 또 다른 열쇠로 복호 화하는 (두 개의 열쇠를 쓰는) 암호 기법을 쓴다. 전통적인 암호 기법은 동일한 하나의 열쇠로 암호화와 복호화를 둘 다 처리해 왔다. 이 (한 개뿐인) "비밀 열쇠"는 (암호화하는 쪽과 복호 화하는 쪽의) 양편이 모두 가지고 있어야 했고, 무슨 수로든 보안을 유지하면서 한 쪽에서 다른 상대방으로 전달되었어야 했다.

이렇게 보안을 유지하면서 열쇠를 전달해 주어야만 되는 어려운 수고를 덜어 주기 위해서, 공개 열쇠 암호 법은 두 개의 키를 사용한다. 각 개인의 공개 열쇠는 누구나 암호화에 쓸 수 있도록 배포되고 이에 상응하는 (복호화에 사용될) 개인의 비밀 열쇠는 개인이 보관한다.

공용 열쇠 암호 기법과 비밀 열쇠 암호 기법에는 각 장점이 있고, 차이점은 이 항목의 끝 부분에 적어 놓은 RSA FAQ를 읽어보기 바란다.

리눅스는 PGP를 잘 지원해 준다. PGP와 2.6.2와 5.0이 잘 작동된다고 알려져 있다. PGP에 대한 기본 안내문과 사용법을 알고 싶으면 PGP FAQ를 읽기 바란다. 국제판 PGPi: http://www.pgpi.org/doc/faq/.

미국 정부는 강력 암호 기법을 군용 무기로 취급하고 있고, 이에 따라서 PGP 등의 강력 암호 기법을 전자적 매체를 통해서 송출하는 것을 "수출 제한 조치"로 금하고 있으므로, 여러분 국가에 맞는 버전을 사용

하도록 하라.

http://mercury.chem.pitt.edu/~tiho/LinuxFocus/English/November1997/article7.html를 보면 리눅스에 피지피를 설치하는 자세한 설명서가 있다. 새로운 버전의 리눅스에는 패치를 구해서 붙여야 되는데, ftp://metalab.unc.edu/pub/Linux/apps/crypto에서 구할 수 있다.

또한 피지피를 오픈 소스 형태의 무료 판으로 재구성하는 계획이 진행되고 있다. 지엔유피지 (GnuPG)는 무료 판 PGP의 완성본이다. 이것은 IDEA나 RSA를 사용하지 않기 때문에 (수출 제한 조치에 걸리지 않고) 제한 없이 쓸 수 있다. 지엔유피지는 OpenPGP 규격에 거의 들어맞게 제작되어 있다. GNU 프라이버시 가드 웹 페이지에 가면 자세한 정보를 얻을 수 있다.

http://www.gnupg.orghttp://www.rsa.com/rsalabs/newfaq/에 있는 RSA FAQ에서 좀 더 정보를 얻을 수 있다. 여기에서 "디피-헬만 (Diffie-Hellamn)", "공용 열쇠 암호 기법 (public-key cryptography)", "전자 인증(DigitalCertificates)" 등의 용어에 대한 정보를 얻을 수 있을 것이다.

3.1.2. SSL, S-HTTP, HTTPS 그리고 S/MIME

SSL : SSL, 혹은 시큐어 소켓 레이어 (Secure Sockets Layer)는 인터넷 상에서의 보안을 위해서 넷스케이프 사에서 개발한 것이며 클라이언트/서버 인증용으로 쓰인다. SSL은 트랜스포트 Layer에서 작동되며 (많은 종류의 데이터들을 사용자가 인지하지 못하는 배경 투명 작업으로 암호화하는) 안전하며 암호화된 통신로(通信路: channel)를 만들어 준다. SSL의 예제는 넷스케이프 커뮤니케이터 (혹은 나비게이터)로 시큐어 사이트의 파일을 열어 볼 때 쉽게 볼 수 있으며 (넷스케이프 사의 데이터 인 크립션을 비롯한) 커뮤니케이터를 이용한 보안 통신 (secure communication)의 기초로 쓰인다.

http://www.consensus.com/faqs/ssl-talk-faq.html에서 추가 정보를 얻에서 추가 정보를 얻을 수 있다. 넷스케이프 회사의 다른 종류의 보안 기술은 http://home.netscape.com/security/index.html에 가면 있다.

S-HTTP : S-HTTP는 인터넷 상에서 보안을 담당하는 또 다른 종류의 프로토콜이다. 다중 열쇠 관리 기법 (multiple key management mechanisms)을 지원하며, 데이타를 주고받는 두 사람이 사용하는 암호 연산 기법 (cryptographic algorithm)의 일치를 옵션 교섭을 통해서 지원하는 동시에, 기밀성 (confidentiality), 인증 (authentication), 보전 성 (integrity: 데이터의 무 결성), 송신 사실 증명 기능 (non-repudiability) 등을 공급해 준다. S-HTTP는 사용 허가된 특별한 소프트웨어로 만 사용이 되도록 제한되어 있으며, 암호화될 대상 데이타를 부분 부분적으로 잘라서 (블록) 암호화해 준다.

S/MIME : S/MIME (Secure Multipurpose Internet Mail Extension)은 전자 우편과 인터넷 상의 메시지를 암호화하기 위한 인크립션 기준이다. RSA에서 개발한 공개 기준이니 만큼, 언젠가는 리눅스에서도 볼 수 있었으면 한다. S/MIME에 대한 추가 정보는 http://www.rsasecurity.com/standards/smime/에서 구할 수 있다.

3.1.3. 리눅스 IPSEC 기술법

CIPE를 포함한 여러 형식의 데이터 인크립션을 포함해서, 리눅스용의 IPSEC 사용 기술 법에는 여러 가지가 있다. IPSEC은 IETF가 IP 네트워크 레벨 상에서 암호 기법 상으로 안전한 통신을 하기 위한 목적으로 만들었으며, 인증 (authentication), 보전 성 (integrity), 접근 관리, 기밀성 등을 제공해 주는 제품이다. IPSEC에 대한 정보와 인터넷 드래프트 문서는 http://www.ietf.org/html.charters/ipsec-charter.html에서 구할 수 있다. 여기에서는 열쇠 관리 기법을 쓰는 다른 프로토콜에 대한 링크와 IPSEC 메일링 리스트, 그리고 메일링 리스트의 아카이브 등

을 찾을 수 있다. 여기서는 열쇠 관리 (Key Management)에 관한 여러 프로토콜에 대한 링크와 IPSEC의 메일링 리스트와 아카이브를 볼 수 있다.

애리조나 대학에서 개발하고 있는 "x-커널의 리눅스용 구성본" (x-kernel Linux implementation)은 x-커널이라는 네트워크 프로토콜을 쓰는 오브젝트-베이스 프레임워크이고, http://www.cs.arizona.edu/xkernel/hpcc-blue/linux.html에서 구할 수 있다. 가장 간단하게 설명을 하자면, x-커널은 커널 차원에서 메시지를 통과시키는 방법이다.

"Linux FreeS / WAN IPSEC"이라는 IPSEC 구성본의 무료 배포판도 있다. 제작자의 웹 페이지에 가보면 "이러한 서비스는 신뢰할 수 없는 네트워크들 (untrusted networks) 상에서, 신뢰할 수 있는 터널 통로(secure tunnel)를 만들도록 해준다. 신뢰할 수 없는 네트워크를 지나게 되는 모든 통신은 IPSEC 게이트웨이 머신 (gateway machine)을 사용해서 암호화되어서 송신되고, 끝 부분의 수신 지점에서 다시 복호화되게 된다. 결과적으로 버츄얼 프라이비트 네트워크가 (Virtual Private Network: VPN - 가상사설망) 만들어지는 것이다. 비록 이 네트워크가 공개적인 인터넷으로 연결된 여러 사이트를 포함한다 해도, 실질적으로는 통신 보안이 되는 네트워크가 구성되는 것이다." 라고 적혀 있다.

이 프로그램은 http://www.xs4all.nl/~freeswan/에서 다운로드 받을 수 있고, 이 문서가 만들어 질 당시에 이미 1.0 버전이 만들어져 나와 있다.

다른 분야의 암호화 기법은 (수출 제한 조치 때문에) 기본적으로 배포 본에 포함되지 않는다.

3.1.4. 시큐어 쉘 ssh와 스텔넷 (Stelnet)

ssh와 스텔넷 (Stelnet)은 원격 시스템으로 접속을 하고, 암호화된 커넥션을 유지하기 위한 프로그램 뭉치다.

openssh는 rlogin, rsh, 그리고 rcp의 대체용으로, 이 셋 보다 더 안정적인 풀그림들의 뭉치다. SSH는 두 호스트간의 통신 암호화와 사용자 인증을 위해서 공개 열쇠 암호 기법을 사용한다. 세션 하이젝킹 (Session Hijacking)과 DNS 스푸핑을 방지해 주면서, 원격 호스트에 로그인하거나, 호스트끼리 데이타를 복사하기 위해 사용될 수 있다. 송수신 시의 데이타 컴프레션을 실행하며, 호스트간의 X11 통신의 통신 보안을 실행해 준다.

이제는 ssh 구성본이 여러 가지 만들어져 있다. 데이터 펠로우스에서 만든 원래의 상업용 구성 본은 http://www.datafellows.com에서 구할 수 있다.

성능이 뛰어난 openssh는 데이터 펠로우스 사의 초기 구성 본에 기초를 두었으며 특허권이나 각 회사 전용의 소스를 전혀 사용하지 않도록 완전히 재구성되어 있다. 무료이며 BSD 사용권 (BSD License)에 기초를 두고 배포 사용된다. 이것은 http://www.openssh.com에서 구할 수 있다.

ssh를 기초부터 다시 오픈 소스로 구성한 "psst..."도 있다.

윈도우스 웍크스테이션 SSH에서 리눅스 SSH로 연결할 수도 있다. 윈도우스 클라이언트용으로 만든 무료 제품이 많은데, http://guardian.htu.tuwien.ac.at/therapy/ssh/ 등이고, 데이타펠로우스사(社)에서 만드는 유료 제품은 http://www.datafellows.com/에 있다.

SSLeay는 넷스케이프사의 SSL를 무료 판으로 구성한 것으로 시큐어 텔넷, 아파치 모듈, 여러 가지 데이터베이스, DES와 IDEA 그리고 불로우피쉬 (Blowfish) 등의 여러 종류 알고리듬을 포함한다.

SSLeasy는 에릭 영 (Eric Young)이 개발한 것으로, 넷스케이프사의 시큐어 소켓 레이어 프로토콜 (Secure Sockets Layer Protocol)의

작동을 무료 구성 판으로 만든 것이다. 이것에는 시큐어 텔넷, 아파치용 모듈, 여러 데이터베이스, 디이에스와 IDEA 불로우피쉬 등을 포함한 알고리듬 등의 포함되어 있다.

텔넷 연결 시에 암호화를 할 수 있는 시큐어 텔넷의 교체 품이 인라이브러리를 써서 만들어져 있다. 스텔넷(Stelnet)은 SSH와는 달리 넷스케이프가 만든 SSL (Secure Sockets Layer)를 사용한다. http://www.psy.uq.oz.au/~ftp/Crypto/에 있는 SSLeay FAQ를 읽어보면 시큐어 텔넷과 시큐어 FTP에 대한 것을 찾을 수 있다.

SRP는 (Secure Remote Password Protocol) 또 다른 텔넷/ftp 통신 보안용 구성의 하나이다. 제작자의 웹 페이지에 가보면 다음과 같은 설명을 하고 있다.

"SRP 프로젝트는 통신 보안 목적의 인터넷 프로그램을 무료로 전 세계에 배포하는 것이 목적으로 개발되고 있다. 완전한 통신 보안이 되는 텔넷과 FTP 디스트리 뷰션을 시작점으로 해 서, (현재의) 빈약한 네트워크 상의 인증 시스템을 사용자가 편하게 쓸 수 있는 강력한 것으로 교체하고자 한다. 보안은 선택적으로 제공되어서는 안 되며, 당연히 기본적으로 제공되어야만 하는 것이다."

자세한 정보는 http://srp.stanford.edu/srp에서 구할 수 있다.

3.1.5. PAM (팸) - 장착 방식 인증 모듈 (Pluggable Authentication Modules)

새로운 버전의 레드햇에는 "PAM"이라는 통일된 인증 방식이 들어 있다. PAM은 (사용자 여러분이 이진 파일을 다시 컴파일할 필요가 없이) 인증 법, 제한 사항, 지역 인증 법을 쉽게 인캡슐레이션 해 준다. PAM의 인캡슐레이션 처리 방법은 이 파일의 내용 밖의 문제이지만, PAM의 웹 사이트에 가서 꼭 보기를 권한다.

http://www.kernel.org/pub/linux/libs/pam/index.html

PAM으로 할 수 있는 일 가운데 몇 가지 만 들어보면 아래와 같다.

- 패스워드에 비 (非) DES 암호화 방법을 쓴다. (패스워드를 부루트 포스 공격을 써서 풀어내는 것이 어렵게 된다)
- 사용자들이 쓸 수 있는 (프로세스 수, 메모리의 양 등의) 자원을 제한하는 방법을 써서 서비스 거부식 공격 (Denial of Service: 이하 DoS)을 못하도록 한다.
- 패스워드를 쉐도우 패스워드로 감추는 것을 쉽게 할 수 있도록 한다.
- 특정한 사용자가 특정한 시간에 특정한 장소에서만 로그인할 수 있도록 제한 조정하는 것이 가능하다.

시스템을 설치하고 조정하기 시작한 지 몇 시간 안으로, 공격 시도 시점에서 막을 수 있다. 예를 들면, .rhosts 파일을 시스템 전체용으로 사용자 홈 디렉토리에서 사용하는 것을 막기 위해서 다음을 /etc/pam.d/rlogin에 PAM을 사용해서 넣을 수 있다.

```
#
# Disable rsh/rlogin/rexec for users
#
login auth required pam_rhosts_auth.so no_rhosts
```

3.1.6. 암호 기술이 적용된 IP 인캡슐레이션 (Cryptographic IP Encapsulation :CIPE)

이 소프트웨어의 일차적 목적은 -- 인터넷 등의 -- 개방형 패켓 네

트워크를 가로 질러가는 서브네트워크를 (가짜 메시지 주입, 트래픽 분석 등의 행위로부터) 보호하기 위한 방법을 제공하는 것이다.

CIPE는 데이타를 네트워크 수준에서 암호화한다. 네트워크의 호스트 사이에서 돌아다니는 패킷이 암호화된다. 암호화 엔진은 패킷들을 주고받는 드라이버 근처에 위치한다.

이것은 (소켓 수준에서 데이터를 연결함으로 암호화를 하는) SSH와는 다른 것이다. CIPE는 (가상사설망 구성하기 위해서) 터널링에 사용될 수 있다. 아래 수준 (Low-level)에서의 암호화는 (애플리케이션 소프트웨어를 수정할 필요가 없이) VPN에 연결되어 있는 두 네트워크 사이에서 투명하게 작동되도록 만들어 질 수 있는 이점이 있다.

IPSEC 기준은 (다른 일도 하지만) 암호화된 VPN을 만들기 위해서 사용될 수 있는 프로토콜의 집합을 정의한다. 반대적으로, 많은 옵션을 가지고 있는 IPSEC은 상대적으로는 헤비급이면서 복잡하며, 주어진 프로토콜 전부를 사용하는 경우는 아직은 드물면서도, (열쇠 관리 등의) 몇 문제는 아직 완벽히 해결되어 있지 않다.

CIPE는 좀 더 간단한 방식을 사용하는데, 초기 설정 시에 매개 변수 형식으로 (정말로 사용하고자 하는 인크립션 방식을 선택하는 등) 많은 조건에 대한 정해진 선택을 할 수 있다. 이것은 탄력적인 운영을 제한하기는 하지만, 간단한 (그리고, 이 이유로, 쉽게 디버그를 할 수 있는 등으로) 능률적인 설정을 가능하게 해 준다.

더 많은 정보를 아래 주소에서 더 얻을 수 있다.

http://www.inka.de/~bigred/devel/cipe.html

다른 크립토그라피의 경우와 마찬가지로 이것도, 수출 제한 조치 때문에, 커널과 함께 배포되지 않는다.

3.1.7. 커브로스 (Kerberos)

커브로스는 MIT의 아테나 프로젝트 아래에서 개발된 인증 방식이다. 사용자가 접속해 들어오면, 커브로스는 (패스워드를 사용해서) 사용자를 인증하고, 네트워크상에 흩어져 존재하는 서버와 호스트들에게 이 사용자의 신분을 증명해 주는 방법을 제공한다.

이 인증 법은 리모트 로그인 (rhost) 풀그림 등에 의해서 패스워드 없이 사용자가 다른 호스트로 (.rhost 파일을 대신해서) 접속을 할 수 있도록 해 준다. 이 인증 법은 또한, 보내는 사람 (발송인)이 가짜가 아닌 것을 보증하는 동시에, 메일이 정확한 사람 (수취인)에게 전달이 되도록 보증하기 위해서, 메일 시스템에 의해 사용될 수도 있다.

커브로스와 딸려 있는 많은 프로그램을 사용하는 궁극적인 효과는, 사용자가 시스템을 속여서 다른 사람인 척 "스푸핑"을 할 수 있는 능력을 거의 없애 버리는 데 있다.

커브로스를 호스트의 보안의 정도를 높이가 위한 첫 방법으로 쓰지는 말아야 한다. 이것은 매우 구성하기 힘들고, SSH처럼 광범위하게 사용되지는 않고 있다.

3.1.8. 쉐도우 패스워드

쉐도우 패스워드는 암호화되어 있는 패스워드 정보를 일반 사용자들로부터 비밀로 유지하기 위한 한 가지 방법이다. 최근에 나온 데비안은 쉐도우 패스워드를 기본적으로 사용하도록 되어 있으며, 다른 리눅스 구성 본은 암호화된 패스워드를 /etc/passwd 파일에 누구나 읽을 수 있을 수 있도록 저장한다.

누구라도 이 파일을 패스워드를 추측해 내는 풀그림에 돌려서 패스워드를 알아내려고 할 수 있다. 반면에 쉐도우 패스워드는 특별 권한이 있는 사용자들만 읽을 수 있도록 패스워드에 대한 정보를 /etc/shadow 파일에 저장한다. 쉐도우 패스워드를 사용하려면, 패스워드 정보를 읽어야 하는 모든 유틸리티들이 쉐도우 패스워드를 지원하

도록 제대로 리컴파일 되었는지 확인해야 한다.

반면에 (위의 설명한) PAM은 실행 프로그램들을 리컴파일할 필요 없이 단지 쉐도우 모듈을 장착시킴으로써 쉐도우 패스워드를 쓸 수 있도록 해준다.

3.1.9. 크랙(Crack)과 존 더 립퍼 (John the Ripper)

Passwd 프로그램을 실행할 때, "쉽게 추측할 수 없도록 만든다"는 패스워드 규칙을 어떤 이유가 있어서 실행하지 못하게 된 상황이라면, 여러분 스스로가 패스워드 격파 프로그램을 실행시켜서 실제의 사용자들이 안전한 패스워드를 쓰고 있는지 확인하는 것도 좋은 것이다.

패스워드 격파 프로그램은 간단한 방식으로 작동 한다. 사전에 있는 모든 단어와 그 변화형을 패스워드로 시도해 하고, 단어 하나하나를 암호화하면서 이미 암호화된 패스워드와 비교하는 것이다. 만약에 일치하는 단어를 찾게 되면, 암호를 알아낸 것이다.

원한다면 많은 패스워드 크랙 프로그램들을 구할 수 있을 것이다. 그 중에서 알아두면 좋은 두개가 바로 "크랙"과 "존 더 립퍼", http://www.openwall.com/john다.

CPU 자원을 엄청나게 소비할 것이지만, 이 풀그림을 여러분이 먼저 사용해 봄으로서 혹시나 공격자가 이런 풀그림을 사용해서 여러분의 시스템에 침입할 가능성이 있는지를 알아보는 동시에, 약한 패스워드를 쓰는 사용자들을 찾아내서 미리 알려줄 수가 있을 것이다.

공격자가 여러분의 passwd (유닉스에서는 /etc/passwd) 파일을 얻으려면 우선은 다른 개구멍을 이용해 먼저 들어와 있어야 하겠지만, 이런 개구멍 허점들이 여러분이 생각하는 것보다 훨씬 흔하다는 점 (즉, passwd가 저장된 파일을 빼내는 것이 어렵지 않다는 점)에 주의

해야 한다.

3.1.10. CFS와 TCFS (암호화 파일 시스템과 투명 암호화 파일 시스템)

CSF는 디렉토리 전체를 암호화하고, 사용자들이 문서를 암호화해서 저장할 수 있도록 하는 방법의 하나이다. 이것은 NFS 서버를 지역 컴퓨터에서 작동하는 방식으로 실행한다.

RPM을 http://www.zedz.net/redhat/에서 구할 수 있고, 작동 방식에 대한 정보는 ftp://ftp.research.att.com/dist/mab/에 더 있다.

TCSF는 CFS보다 좀 더 완성도를 높여서 (암호화/복호화 작업을 백그라운드에서 투명하게 실행함으로서) 암호화된 파일시스템을 쓰고 있는 사용자 입장에서는 암호화/복호화 작업이 눈에 보이지 않도록 한 것이다. http://edu-gw.dia.unisa.it/tcfs/에서 정보를 구할 수 있다.

파일시스템 전체에 사용하지 않을 수도 있다. 부분적으로 디렉토리 트리를 암호화하는 것에만 쓰일 수도 있는 것이다.

3.1.11. X11, SVGA와 디스플레이 보안

3.1.11.1. X11

디스플레이의 보안은 중요하다. 입력되는 패스워드를 공격자가 가로채거나, 여러분이 모니터로 읽고 있는 문서나 정보를 보거나, 심지어는 Super User의 권한을 얻기 위해 보안 개구멍을 이용하기까지 하는 일들을 막기 위해서다. 도청 자(sniffer)들이 여러분과 원격 시스템 사이의 상호작용을 모두 볼 수 있도록 허락하는 셈이라 할 수 있는, 네트워크상에서의 원격 X 응용 프로그램 수행도 위험천만한 일이다.

X는 많은 접근 통제 장치를 가지고 있다. 가장 간단한 것은 호스트에 기초를 두는 것이다. 여러분의 디스플레이어 접근할 수 있는 호스트들을 xhost를 사용해서 지정할 수 있다. 안전한 방법은 아니다. 누

군가가 여러분의 컴퓨터에 이미 근접할 수 있다면, 그는 xhost +그들의 컴퓨터라는 명령어를 사용해서 쉽게 숨어 들어올 수 있다. 아울러 신임되지 않은 기계 (untrusted machine)의 접속을 허락하면, 그쪽의 누구라도 여러분의 디스플레이를 침탈할 수 있다.

로그인을 위해서 xdm을 (xdm: X 디스플레이 매니저, x display manager) 사용하고 있다면, 더 나은 접근 방법인 MIT-MAGIC-COOKIE-1을 구해서 사용하는 것을 고려해 보도록 하라. 128 비트짜리 "쿠키(cookie)"가 만들어져서 .Xauthority 문서에 저장된다. 원격 컴퓨터에서 여러분의 디스플레이에 접근하는 것을 허용할 필요가 있다면, 그 컴퓨터로부터의 접근만을 제공하기 위해 xauth 명령과 여러분의 .Xauthority 파일에 들어 있는 정보를 적용할 수있다.

http://metalab.unc.edu/LDP/HOWTO/mini/Remote-X-Apps.html에 있는 Remote-X-Apps mini-howto를 보도록 하라.

보안 유지되는 X의 접속을 만들기 위해서 ssh를 쓸 수 있다 (위의 [ssh]를 참조할 것). 앤드 유저의 시점에서는 투명하게 작동되면서도, 암호화되지 않은 자료가 네트워크상에 떠다니지 않도록 하는 방법이 되는 장점이 있다.

X보안에 대해 더 많은 정보가 필요하면 Xsecurity의 맨 페이지(man)를 보기 바란다. 보다 안전한 방법은 xdm을 써서 콘솔에 로그인 하도록 하고, ssh를 써서 X 프로그램을 원격 수행하려는 원격 사이트들로 가는 것이다.

3.1.11.2 SVGA

SVGAlib 프로그램들은 리눅스 컴퓨터에 있는 모든 비디오 하드웨어에 접근할 수 있도록 이것의 SUID가 root로 정해져 있다.

이것은 매우 위험한 것이다. 만일 이 프로그램들이 깨지면, 쓸 수 있는 콘솔을 살리기 위해서 다시 부팅 시켜야 한다. 여러분이 실행시키고 있는 SVGA 프로그램들이 진품인지, 그리고 최소 수준이나마

믿을 수 있는 것들인지 확인하라. 더 나은 방법은 SVGA 프로그램들을 아예 수행시키지 않는 것이다.

3.1.11.3. GGI (Generic Graphics Interface project)

리눅스 GGI 계획은 여러 가지의 리눅스 비디오 인터페이스 문제들을 해결하고자 노력하고 있다.. GGI는 비디오 코드 일부분을 리눅스 커널 안으로 옮겨 실행하는 방식으로 비디오 시스템에 대한 엑세스를 관리할 것이다.

이것은 GGI가 (정해 놓은 양호 상태로) 언제라도 여러분의 콘솔을 복구해 줄 수 있다는 것을 의미한다. 또한 여러분 콘솔에서 트로이 목마식의 로그인 프로그램이 돌지 않도록 하기 위해서, 보안 경제 열쇠(관리)도 허락 될 것이다.

http://synergy.caltech.edu/~ggi

3.2. 여러 가지 해킹 기법

3.2.1. DoS, DDoS

DoS(Denial of Service)란, 서버의 존립목적인 서비스를 못하도록 하는 방법으로서 공격 장소에 따라 local과 remote로 나눈다.

3.2.1.1. local DoS

일반적으로 local DoS공격은 공격자가 시스템에 들어와서 시행하는 방법이다.

끝임 없이 프로세스를 만든다거나 잘못된 명령을 루프로 계속 돌린다거나 기타 시스템에 크게 무리를 주어, 시스템을 마비시키는 공격법이다. 예) Exploer 4.0에서 자기참조프레임

3.2.1.2 remote DoS

대부분의 네트워크상에서 이루어지는 DoS공격법으로서, 가장 대표적으로 얼마 전 일본 사이트들에 대한 국내 네트즌 들의 항의성 "리로딩"

공격도 이에 해당 한다.

좀 더 세밀하게 들어가면 ping공격, ICMP(Internet Control Message Protocol)공격, 메일폭탄 등등이 있다..

3.2.1.3. DDoS

DDoS(Distributed Denial-of-Service)는 여러 대의 장비를 이용해 엄청난 분량의 데이터를 한 곳의 서버에 집중적으로 전송함으로써, 특정 서버의 정상적인 기능을 방해하는 것을 말한다.

야후, 이베이, ZD넷, CNN 등 대형 인터넷업체들을 대상으로 발생한 cracking사고들이 이 수법에 의해 것이다.

이 방법은 공격자가 자신의 장비에서 보안이 취약한 다른 여러 곳의 서버에 침투해 해킹 마스터 프로그램을 몰래 설치한 뒤, 여기에 공격할 대상과 시간을 지정해 두면, 이들 서버가 한 개의 목표 서버에 수많은 패킷을 전송함으로써 일시에 서버기능을 무력화시키는 방법이다.

3.2.2 System오류

시스템상의 오류를 통하여 공략하는 방법을 의미한다.

3.2.2.1 환경변수의 취약성

다중 사용자를 지원하는 서버용 시스템은 각 사용자에 맞는 환경변수를 지정하여서 지정한 환경 하에서 사용자가 좀 더 편리하게 작업을 할 수 있도록 도와주고 있다. 그러데, 이러한 환경변수 가운데 시스템의 취약성을 나타내어주는 일부의 환경변수들이 존재한다.

예1) 상대경로

어떠한 suid프로그램에서 "/bin/ls"를 프로그램중간에 call한다고 가정하자. 일반적으로 "/bin" 디렉토리는 환경변수 상에서 PATH의 기본 값으로 지정해주고 있다. 그래서 프로그래머가 system("ls") 라는 코드로 ls 를 실행하게 된다면 공격자는 PATH의 일반경로를 변경하여 /bin/ls

가 아닌 임의의 다른 ls라는 이름의 프로그램을 실행하게 할 수 있다

예2) IFS

IFS(internal file seperator)는 프로그램 아큐먼트 간을 구분해주는 공백을 의미한다.

이 역시 환경 변수 하에서 적용되는 것인데, 이번에는 프로그래머가 절대경로를 잘 사용하여 system ("/bin/ls")라고 코드를 넣었다고 가정하자, 그러나 환경변수하의IFS를 "/" 으로 변경한다면 어떻게 될까..흐름은 "/bin/ls" 가 아닌 "bin ls" 라고 인식을 할 것이다. 그렇다면 원하는 ls가 실행되는 것이 아니라 bin이란 이름의 실행 파일을 실행시키려고 시도 할 것이다 이때, 공격자는 bin이란 이름의 악의적 프로그램을 suid상에서 실행토록 할 수 있다. -> sysinfo2.0.6이상 버전에서 패치

예3) 그 밖의 환경변수들

export명령을 통해서 일반적인 자신 계정의 환경 변수 값을 볼 수 있다. ssh(Secure Shell)의 환경변수를 이용해서 '-r' 퍼미션의 내용(예 /etc/shadow)도 읽을 수 있는 버그가 발표 되었다.

```
$export RESOLV_HOST_CONF=/etc/shadow
$ssh aa 2> shadow
$more shadow
```

3.2.2.2. race condition

race condition이란 공격법은 최초 8lgm이란 해커그룹의 공격에서 그 유래를 들 수 있다.

이들은 mail을 통해서 해당계정의 spool파일을 덮어 쓴다는 것에 착안하여 SunOS 4.x버전의 /bin/mail의 경우 사용자들의 메일 이 저장되는 /var/spool/mail/user_id 에 해당되는 파일 중 deamon과 같은 시스템계정의 파일을 root의 .rhost 파일로 링크를 건 뒤, daemon 사용자에게 메일을 보냄으로서 자원 간에 경쟁을 통해 .rhost를 변경함으로써

공격이 이루어진다.

```
ln -s ~root/.rhosts daemon
% echo "+ +" | mail daemon
% rlogin localhost -l root
```

즉 /bin/mail 프로그램이 /var/spool/mail/user_id 에 해당 하는 파일이 심볼릭 링크인지 아닌지를 확인하지 않은 채 root 의 권한으로 /var/spool/mail/user_id에 해당되는 파일에 수신된 E-mail의 내용을 덮어쓰기 때문에 발생한 것이다.

3.2.2.3 ptrace

최근 들어서 시스템오류로는 프로세스가 시그널 스트럭쳐를 공유할 때 발생되는 공격법이 나왔고. Linux Kernal 2.2.x 이하에서 /usr/bin/passwd를 이용한 ptrace공격법과, FreeBSD 의 모든 suid프로그램에 적용되는 공격법등이 선보이고 있다.

3.2.3. 프로그램의 오류

suid 프로그램이나 daemon프로그램 등에서 발생되는 오류 등을 찾아서 공격하는 방법을 말한다.

3.2.3.1. BOF(Buffer Overflower)

프로그램 오류의 가장 대표적인 방법이 BOF이다. BOF는 변수 간 경계 값을 체크하지 않았을 경우 발생하는 것으로서 스택상의 입력되는 변수의 값을 설정보다 많이 넣어서 값이 흘러 넘치도록 하는 것인데, 문제는 이렇게 흘러넘친 값을 함수의 리턴 어드레스까지 변경하게 하여 실행흐름을 변조할 수 있다는 것이다.

3.2.3.2 FSB(Format String Bug)

FSB는 비교적 최근에 발표된 버그로서 프로그램 상에 string을 지정해주지 않고 변수이름으로 입출력 할 때에 발생되는 버그로서, 프로그

램이 실행되고 변수를 입력 받을 때 공격자는 임의의 string을 지정하여 스택상의 메모리 값을 임의로 변조하는 방법을 말한다.

%hn , %n -> 스트링 앞까지의 문장 열 개수 출력을 이용해서 함수의 리턴어드레스나, 스택상의 중요 포인터 값들을 변조 할 수 있다.

3.2.4. 네트워크의 취약성

기본적으로 Hacking의 전제는 네트워크가 활성화 되어있을 때 가능하다. 네트워크 설계상의 취약점들을 노려서 공격하는 방법은 공격기법 면에서 상당한 기술적 능력이 요구되는 방법 중에 하나다.

3.2.4.1 Spoofing

바로 자기 자신의 식별 정보를 속여 다른 대상 시스템을 공격하는 기법이다. 네트워크상의 공격자는 TCP/IP 프로토콜 상의 취약성을 기반으로 해킹 시도 시 자신의 시스템 정보(IP 주소, DNS 이름, Mac 주소 등)를 위장하여 감춤으로써 역추적이 어렵게 만든다. 이러한 스프핑 공격은 패킷 스니퍼링이나 서비스 거부 공격, 세션 하이재킹(Session Hijacking) 등의 다른 여러 가지 공격을 수행 가능하게 한다.

스프핑 공격의 종류를 알아보면 다음과 같다. 어떤 정보를 속이느냐에 따라 세분화될 수 있다.

(1) IP 스프핑

IP 스프핑은 말 그대로 IP 정보를 속여서 다른 시스템을 공격하는 것이다. IP 스프핑을 통해 서비스 거부 공격(TCP Syn flooding, UDP flooding, ICMP flooding)을 수행할 수도 있으며, 공격대상 컴퓨터와 서버 사이의 연결된 세션에 대해서 세션 끊기도 가능하다.

TCP/IP 상의 프로토콜 취약성은 1985년 Robert T. Morris의 논문 “A Weakness in the 4.2 BSD Unix TCP/IP Software”에 업급이 되었으며, 특히 희대의 해커 케빈미트닉은 실제 해킹에서 IP 스프핑 공격을 통해 모토롤러, 선마이크로시스템즈, NEC, 노벨 등의 컴퓨터 전산망에

침투, 소프트웨어 및 각종 자료 등을 훔친 혐의로 1995년 체포되었다.

케빈 미트닉은 정확하게는 TCP Syn flooding + TCP 순서 번호 예측(TCP Sequence Nubmer Guessing)

+ IP 스프핑을 사용하였다. 이는 클라이언트와 서버와의 통신 사이에 해커 컴퓨터가 끼어들어 클라이언트를 TCP Syn flooding 서비스 거부 공격으로 전혀 반응하지 못하게 한 후, 해커가 이 클라이언트인 것으로 가장하여 서버와 통신하는 기법이다. 이러한 공격은 TCP/IP 프로토콜의 문제점인 TCP 순서 번호 생성이 매 초당 일정하게 증가한다는 것과 호스트에 대한 인증 시 IP의 소스 주소만을 사용한다는 것으로 인하여 가능하였다.

순서 번호는 연결 지향형 프로토콜인 TCP 프로토콜에서 두 호스트 간의 패킷 전달이 손실 없이 이루어졌는지 체크하기 위한 일종의 패킷 번호표이다. 해커는 일단 클라이언트를 TCP Syn flooding 공격으로 봉쇄한다. 이후 서버의 순서번호를 예측하여 IP 스프핑 된 위조 패킷을 발송함으로써 서버를 속여 침투하는 것이다. IP 기반의 인증만을 제공하는 Unix의 rlogin, rsh 등의 r 계열 서비스들은 이러한 IP 스프핑 공격에 취약할 수 밖에 없다.

(2) APR 스프핑

ARP 프로토콜은 32bit IP 주소를 48bit의 네트워크 카드 주소(Mac Address)로 대응시켜 주는 프로토콜이다. 우리가 실제로 IP 주소를 통해 네트워크 연결을 시도하면 TCP/IP에서는 해당 IP에 해당하는 네트워크 카드 주소를 찾아 연결하게 된다.

이러한 IP 주소와 네트워크 카드 주소의 대응 테이블은 스위치나 기타 네트워크 장비 및 사용자 컴퓨터에서 arp cache 테이블이라는 곳에 위치하게 된다.

해커가 이 테이블 상의 정보를 위조하게 되면 공격 대상 컴퓨터와 서

버 사이의 트래픽을 해커 자신의 컴퓨터로 우회시킬 수 있다. 우회된 트래픽으로부터 해커는 패스워드 정보 등 유용한 정보를 마음껏 획득할 수 있다.

(3) 이메일 스프핑

이메일 발송시 송신자의 주소를 위조하는 것이다. 간단한 방법으로는 이메일 송신자 From 필드에 별칭(alias) 필드를 사용할 수 있다. 이메일 발송 시 별칭을 설정한 경우에는 별칭 주소로 이메일이 발송된다. 이러한 경우 메일을 받아보는 사람은 실제 이메일 송신자가 아닌 별칭 필드만을 확인하는 경우에는 이메일의 송신자가 별칭 필드에서 온 것으로 알게 된다.

요즘 들어서 극성인 대량의 스팸 메일과 바이러스 감염 메일은 송신자의 주소가 아예 존재하지 않는 이메일 주소를 사용한다. 또한 이메일을 발송한 메일 서버 또한 직접적인 메일 발송 서버가 아닌 중계 서버이므로 메일을 발송한 자를 추적하기란 쉽지 않다.

(4) DNS 스프핑

DNS 프로토콜은 인터넷 연결 시 도메인 주소를 실제 IP 주소로 대응시켜 주는 프로토콜이다. 인터넷 연결 시 사용하는 DNS 서버가 IP 주소를 찾아달라는 요청을 받았을 때, 자기 자신의 도메인이 아닌 주소에 대해서는 보다 상위 단의 DNS 서버로부터 재귀적(recursive)인 방식으로 IP 주소를 찾아 알려준다.

만약 해커가 어떤 도메인의 DNS 컴퓨터를 장악하여 통제하고 있다면 최종적으로 얻어진 IP 주소는 원래 사용자가 찾아가고자 하였던 홈페이지가 아닌 다른 홈페이지로 연결되게 된다. 이는 요청을 발송했던 DNS와 응답을 주는 DNS 사이의 트래픽을 해커가 스니퍼링 함으로써 Query ID라는 값을 통해 해커의 사이트 IP를 최종 응답으로 넘겨주도록 하는 것이다.

사용자가 쇼핑몰을 이용하고자 하였다면 해커에 의해 조작된 홈페이

지 내에서 자신의 아이디와 필드, 신용 카드 정보를 기입함으로써 개인 정보를 탈취당할 수 있다. 위와 같은 스프핑 공격들은 실제로 인터넷 상의 툴로써 공개가 되어 있으며 여러 가지 다른 복합적인 공격과 같이 사용될 수 있다.

그러나 각각의 공격 방법에 있어서 제약 및 전제 사항이 있으므로 모두 완벽하게 성공되지는 않는다. 스프핑 공격은 패킷 필터링 접근 제어와 IP 인증 기반 접근제어, 취약점 서비스 사용의 제거, 암호화 프로토콜의 사용을 통해서 방어가 가능하다. 인터넷 상에 떠돌아다니는 정보는 항상 안전한 것이 아니므로 스프핑 공격과 같은 것을 통해 언제라도 위조되었을 가능성도 있을 수 있음을 간과해서는 안되겠다.

3.2.4.2. Sniffing

사전적인 의미로 스니핑(Sniffing)이란 '코를 킁킁거리다', '냄새를 맡다' 등의 뜻이 있다. 사전적인 의미와 같이 해킹 기법으로서 스니핑은 네트워크상에서 자신이 아닌 다른 상대방들의 패킷 교환을 엿듣는 것을 의미한다. 간단히 말하여 네트워크 트래픽을 도청(eavesdropping)하는 과정을 스니핑이라고 할 수 있다. 이런 스니핑을 할 수 있도록 하는 도구를 스니퍼(Sniffer)라고 하며 스니퍼를 설치하는 과정은 전화기 도청 장치를 설치하는 과정에 비유될 수 있다.

TCP/IP 프로토콜은 학술적인 용도로 인터넷이 시작되기 이전부터 설계된 프로토콜이기 때문에 보안은 크게 고려하지 않고 시작되었다. 대표적으로 패킷에 대한 암호화, 인증 등을 고려하지 않았기 때문에 데이터 통신의 보안의 기본 요소 중 기밀성, 무결성 등을 보장할 수 없었다. 특히 스니핑은 보안의 기본 요소 중 기밀성을 해치는 공격 방법이다.

인터넷은 말 그대로 광범위한 네트워크이며 공개된 네트워크이다. 패킷이 송수신될 때, 패킷은 여러 개의 라우터를 거쳐서 지나가게 되며 중간 ISP 라우터에 접근 권한을 가지는 사람이라면 해당 패킷을 쉽게 잡아낼 수 있다. 그런데 문제는 이렇게 쉽게 얻어낼 수 있는 많은 패킷

의 내용은 암호화 되지 않는다는 것이다.

물론 xDSL, 케이블 모뎀 등을 사용하는 일반 가정 사용자가 이러한 패킷을 아주 쉽게 볼 수 있는 것만은 아니다. 그러기 위해서는 패킷이 흘러가는 네트워크의 중간 경로를 얻어내야 한다. 전화에 도청기를 설치하는 과정을 연상하면 이해하기 쉬울 것이다. 직접 도청하고자 하는 전화기에 도청기를 설치하는 방법, 그리고 중계 회선에 도청기를 설치하는 방법이 있을 것이다.

두 가지의 차이는 직접 도청하고자 하는 전화기에 설치한다면 그 전화기의 내용만 도청할 수 있다는 것, 중계 회선에 설치한다면 그 중계 회선을 통해 연결된 모든 전화기의 내용을 도청할 수 있다는 것이 될텐데 스니핑 역시 마찬가지이다.

대표적인 시나리오는 다음과 같다.

(1) 다양한 공격 기법을 통해 실제 공격 대상 시스템에 관리자 권한을 얻어낸 후 스니핑 도구를 설치하여 스니핑

(2) 공격 대상 기업의 다른 호스트에 대한 접근 권한을 얻어내서 그 호스트를 이용하여 스니핑

(3) ISP 장비에 대한 시스템 권한을 얻어내어 스니핑 도구를 설치하여 스니핑

다음은 이러한 스니핑 기법을 통하여 FTP 접속을 스니핑함으로 사용자 ID와 패스워드를 얻어낸 화면이다. 이 경우, Ethereal이라는 툴을 이용한 것으로 쉽게 해당 서버에 hackme라는 peace! 패스워드를 사용하는 사용자가 존재한다는 사실을 알았다.

물론 스니핑은 이와 같은 공격만을 위해 사용되는 것은 아니다. 네트워크 트래픽 분석이나 트러블슈팅 등에 사용되며 그 외에도 네트워크상에 이루어지는 공격을 탐지하는 침입탐지시스템에 쓰일 수 있다. .

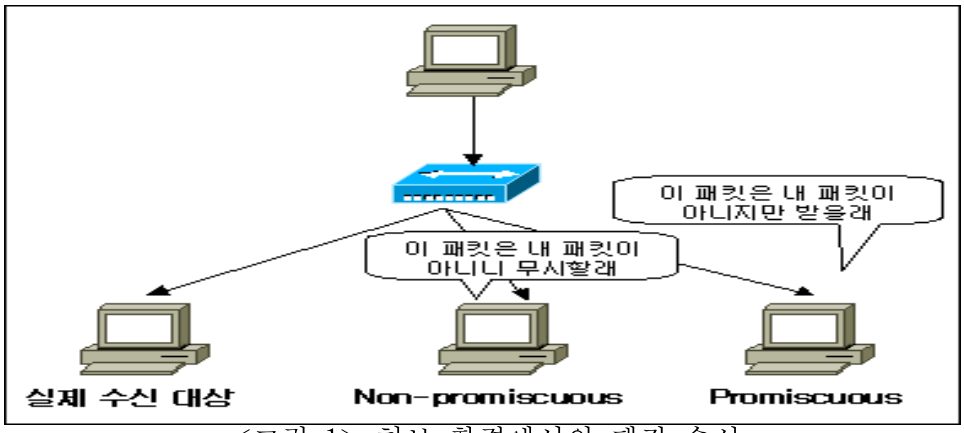
실질적인 스니핑 공격의 예는 아래와 같다.

(1) 허브 환경에서의 스니핑

허브(Hub)는 기본적으로 들어온 패킷에 대해 패킷이 들어온 포트를 제외한 모든 포트에 대해 패킷을 보내는 리피터(Repeater) 장비이다. 사실 여러분의 기업에서 허브를 사용하고 있고 여러분의 시스템이 그 허브에 연결되어 있다면 여러분은 원하던 원치 않던 간에 계속하여 다른 사람의 패킷들을 받아보고 있었던 것이다.

물론 네트워크 드라이버, OS 커널 등의 수준에서 MAC 주소를 보아 자신이 아닌 다른 이들의 패킷은 버려지기 때문에 그것을 쉽게 느낄 수는 없었을 것이다. 하지만 여러분 시스템의 NIC를 promiscuous 모드로 동작하게 한다면 다른 이들의 패킷 또한 버리지 않고 받아볼 수 있다. 이제 스니핑 도구를 통해 해당 패킷을 저장하고 분석하기만 하면 된다.

다음 그림은 이 내용을 설명한 것이다. 모든 패킷은 실제 수신 대상이 아닌 호스트에게도 전달되며 Promiscuous 모드로 동작하는 호스트는 다른 수신 대상의 패킷 또한 볼 수 있다.

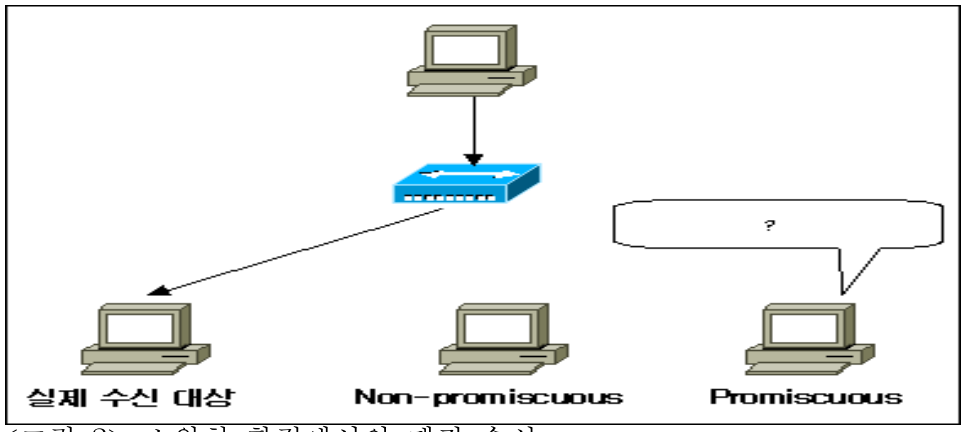


<그림 1> 허브 환경에서의 패킷 송신

(2) 스위치 환경에서의 스니핑

스위치는 기본적으로 Layer 2 헤더 정보인 MAC 주소 정보를 이용하여 패킷이 어떤 목적지로 보내질지를 결정한다. 따라서 허브 환경에서와 달리 패킷은 실제 수신 대상에게만 보내지게 되며 공격 대상이 아무리 인터페이스를 Promiscuous 모드로 세팅하였다 하더라도 그 내용

을 훔쳐 볼 수는 없다.



<그림 2> 스위칭 환경에서의 패킷 송신

(3)스니핑에 취약한 프로토콜

앞서 말한 바와 같이 일단 패킷을 악의적인 사용자가 가로채어 보는 것은 그리 어려운 일이 아니다. 이런 시도를 탐지하는 방법도 알려져 있기는 하지만 일단 이런 시도 자체를 완전 봉쇄하는 것은 불가능하다고 볼 수 있다.

하지만 이렇게 얻어낸 패킷이 모두 유용한 것은 아니며, 암호화되지 않거나 암호화되더라도 너무도 간단한 방법으로 되어 쉽게 복호화 해낼 수 있는 그런 프로토콜에서 사용되는 패킷이 공격자에 의해 사용될 수 있다. 그런 종류의 프로토콜을 스니핑에 취약한 프로토콜이라 할 수 있는데 스니핑에 취약한 프로토콜을 몇 가지 예로 든다.

1) Telnet, Rlogin

Telnet, Rlogin의 사용자 ID, 패스워드를 비롯한 모든 통신의 내용은 암호화되지 않아 모든 통신 내용을 쉽게 볼 수 있다.

2) HTTP

HTTP의 사용자 인증으로 많이 사용되는 Basic Authentication 방법은 아주 기본적인 방법으로 encode되기 때문에 쉽게 사용자 ID, 패스워드 정보를 얻어낼 수 있다.

3) SNMP

SNMP 프로토콜은 단순 네트워크 관리 프로토콜(Simple Network Management Protocol)이라는 이름과 같이 보안을 거의 고려하지 않았다. SNMP 프로토콜은 SNMPv1, SNMPv2, SNMPv3로 나뉘어 뒤로 갈수록 보안은 강화되었지만 아직도 가장 많이 사용되는 것은 SNMPv1 프로토콜이다. SNMP의 패스워드와 같은 역할을 하는 커뮤니티 이름을 비롯한 모든 통신 내용이 암호화 되지 않는다.

4) 기타

NNTP, POP, FTP, IMAP, SMTP 등

(4)스니핑의 방어

스위치에 브로드캐스트 도메인, MAC 주소 수동 설정 등을 함으로 패킷을 가로채는 시도를 줄일 수는 있으나 앞서 말한 바와 같이 다른 사용자가 패킷을 가로채는 시도를 원천 봉쇄하는 것은 불가능하다. 따라서 패킷을 가로채더라도 그것의 내용을 가지고 어떠한 행동조차 할 수 없도록 암호화 기법을 이용하는 것이 가장 일반적이고 중요한 스니핑의 방어 기법이라고 할 수 있다.

1) SSL 적용

HTTP, IMAP, POP, SMTP, Telnet 등은 SSL을 적용하여 HTTPS, IMAPS, POPS, SMTPS, Telnets 등으로 할 수 있다. SSL은 물론 HTTP에 가장 많이 활용되며 이를 적용하여 사용자 이름, 패스워드 및 전자 상거래 결재 정보 등 웹 서핑의 내용을 암호화 할 수 있다.

2) PGP, S/MIME

SMTP 상으로 보내지는 메일은 기본적으로 암호화 되지 않기 때문에 스니핑하여 그 내용을 쉽게 얻어낼 수 있다. PGP, S/MIME 등을 이용하여 메일에 대한 암호화 기능을 제공할 수 있다.

3) SSH

암호화 통신을 제공하여 Telnet, FTP, RCP, Rlogin 등을 대치할 수 있다.

4) 사설망 혹은 가상사설망(VPN)

스니핑이 우려되는 네트워크상에 전용선(leased line)으로 직접 연결함으로 중간에 도청되는 것을 막는 것이 사설망이다. 하지만 이는 거리가 멀어질수록 인터넷을 이용하는 것에 비해 비용이 매우 비싸질 수 밖에 없다. 인터넷 회선을 이용하며 사설망의 효과를 줄 수 있는 것이 VPN입니다. VPN 장비 간의 암호화를 통해 도청을 막을 수 있다.

스니핑은 다양한 형태로 네트워크상에서 이루어질 수 있으며 다음과 같은 두 가지 단계로 볼 수 있다.

- 패킷 가로채기
- 가로챈 패킷 디코딩을 통해 주요 정보 획득

패킷을 가로채는 시도는 차단하기 매우 어려우며 디코딩을 통해 주요 정보를 얻어내는 것을 막기 위해 SSL, SSH, VPN, PGP 등 다양한 기법이 이용될 수 있다.

Ⅳ. 현재 리눅스보안기술의 보안 대책방안

4.1. 리눅스보안 기술의 단점

첫 번째로 리눅스보안기술의 단점은 리눅스는 기술 인력이 없고, 유닉스·윈도처럼 기술지원 서비스의 주체가 없다는 점이 최고의 단점이다. 이에 따라 보안성과 시스템 안정성 문제에 대한 우려가 있게 마련이다. 요즈음 들어서부터 HP 등 하드웨어 기업들이 리눅스에 대한 기술 지원을 하겠다고 하면서 인식이 개선되었지만. 아직도 부족한 것이 기술 인력이다.

두 번째로 리눅스는 거의 대부분의 프로그램들이 소스코스를 공개하고 있다는 것에 있다. 소스코드가 공개되어있기 때문에 그 소스의 치명적인 약점을 공격하여 리소스를 얻어갈 확률이 커 보안위협에 노출되어 있는 것이다.

하지만 소스에 관한 것은 개개인 사용자가 자신의 시스템의 맞게 변경시킬 수 있기 때문에 오히려 안전한 것이 될 수도 있다. 리눅스 자체는 보안에 취약한 것이 아니나, 보안을 사용하는 프로그램에 문제가 생기는 것이기 때문이다.

4.2. 현재 리눅스보안기술의 보안 대책방안

- 만약 finger 서비스를 꼭 해야 한다면 수정된 버전을 설치하라. 사용자의 홈 디렉토리, 로그인 한 호스트이름 등을 알릴 필요는 없는 것이다.

- 절대적으로 필요하지 않다면 NIS 를 사용하지 말 것이며, NFS 도 가능한 사용하지 않는다.

- 절대로 NFS 파일시스템을 완전히(Worldwide) 공개하지 말 것이며, 가능한 읽기 전용으로 만들어라.

- 서버를 방어하고 보호한다. 단지 관리할 수 있는 계정만을 허용한다.

- inetd 나 portmapper 등에서 불필요한 서비스가 없는지 점검한다. 만약 접속하는 시스템의 로그를 기록하고 싶다면 wrapper 를 사용하는 것이 좋은데, 표준 유닉스보다 우수한 로그와 특히 네트워크 공격에 대한 기록이 우수하다. 그리고 가능한 보안 관련 정보를 입수하기 위해 syslog 의 loghost 메카니즘을 이용하라.

- 완벽하게 믿을 수 있는 시스템이 없다면 신뢰하는 호스트를 없앤다.

- Shadow 패스워드와 잘못된 패스워드를 가려내는 패스워드 명령을 사용한다. 사용하지 않는 계정과 시스템을 없앤다.

- 최근의 자료(참고문헌)와 도구들을 수집하며, 보안사고 문제나 보안관련 문제를 지속적으로 남들과 대화한다. 최소한 CERT mailing list 와 보안잡지, Firewall mailing list, 보안 관련 뉴스그룹 등에 가입한다. 무관심도 보안 문제에 가장 큰 적이다.

- 기관의 모든 호스트에 대해 가능한 최신 패치를 설치한다.

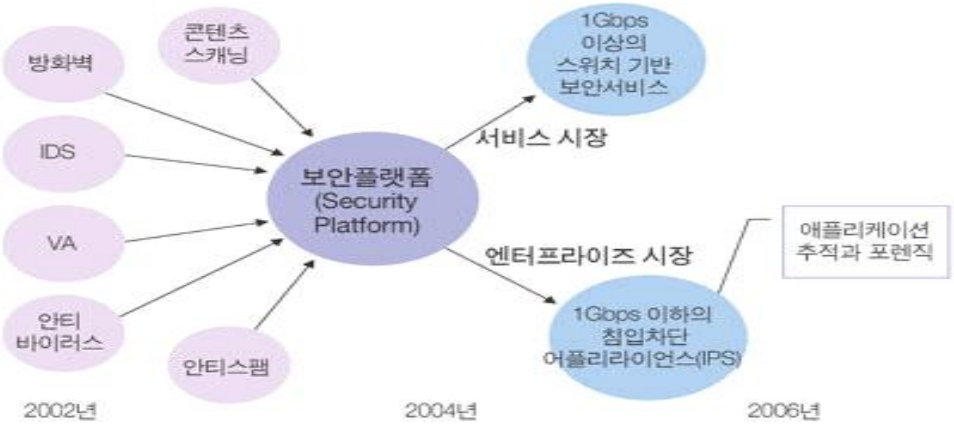
네트워크 방화벽시스템, Kerberos, 일회용 패스워드시스템 등이 보안 문제를 해결하는데 도움이 되지만 위에서 여기에서 알려진 이러한 공격에 모두 안전한 것은 아니다. 즉 위 3, 4 가지 우수한 보안 기법을 사용하는 것이 좋다고 권고하지만 모두 해결되는 것은 아니라는 말이다.

V. 미래 리눅스 보안기술 동향

5.1. 네트워크 보안 플랫폼과 IPS

전문기관들이 예측하는 <그림 3> 보안 플랫폼의 발전에서 느낄 수 있듯이 보안 플랫폼의 발전 자체가 앞으로의 보안 시장에서 가장 중요한 요소중 하나가 될 것이다. 전형적인 발전 방향의 하나는 리눅스 플랫폼에 IDS, 방화벽, VA(Vulnerability Assessment) 도구, 안티바이러스 게이트웨이 등이 통합되는 것이다.

그리고 이 플랫폼은 다른 네트워크 보안 기능을 지속적으로 추가해 나갈 것으로 예상하는데 요즈음 모든 사람들을 성가시게 하는 이메일 문제와 관련해서 이메일 콘텐츠 스캐닝이나 안티스팸 기능 등이 추가적으로 고려되어질 것이다. 또한 엔터프라이즈 보안을 위해서 이들은 포렌직 정보를 제공해 준법감시(Compliance) 요구를 충족시키는 도구가 될 것이다.



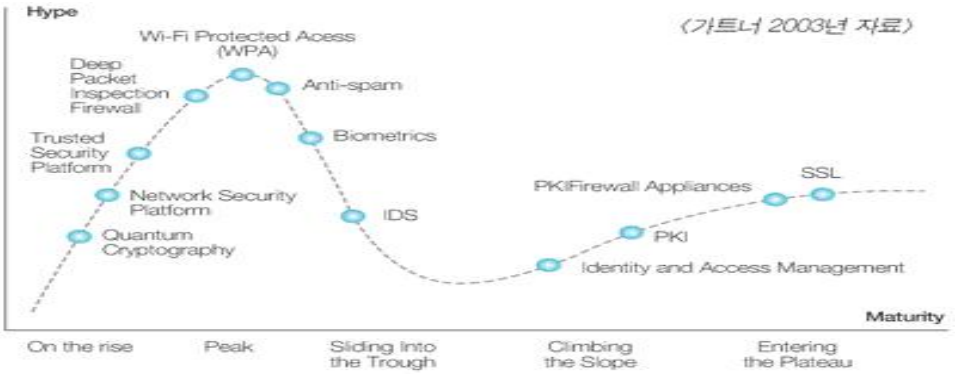
<그림 3>보안 플랫폼의 발전

향후 예상하는 엔터프라이즈를 위한 보안 플랫폼의 하드웨어는 어플라이언스 형태의 침입차단 장치이기 때문에 현재 IDS기반의 IPS 어플라이언스가 기능을 추가해 나가는 과정에서 이상적인 보안플랫폼으로 이전하는 상당한 기간 동안 IPS의 이름을 그대로 이어받아 사용할 것으로 보인다. IPS 측면으로만 생각해 본다면 현재의 IPS는 계속 새로운

기능을 추가하여 침입을 차단하는 통합보안솔루션 박스의 모습으로 거듭 날 것이라는 예상을 할 수도 있다.

보안 플랫폼 시장은 대역폭에 따라서 크게 두 가지로 구분이 될 수 있으며 1Gbps 이상의 서비스 시장과 1Gbps 미만의 엔터프라이즈 시장으로 나눌 수 있다. 현재도 그런 경향을 볼 수가 있는데, 대역폭 제공이 중요한 서비스 시장에서는 스위치 기반의 IPS가 시장 점유율이 높고, 엔터프라이즈 시장에서는 관리가 좀더 편한 어플라이언스 형태의 IPS가 시장을 주도하고 있는 것으로 보인다. 앞으로 당분간은 하드웨어적인 태생이 다른 두 가지 IPS가 시장을 구분할 것으로 보인다. 그러나 장기적으로 하드웨어 기술이 발전하면 할수록 이들의 구분 또한 모호해 질 수도 있을 것이다.

가트너에서는 네트워크 보안 플랫폼(Network Security Platform)의 현재 위치를 <그림 4> 보안의 하이프(Hype) 사이클에서와 같이 기술의 초기진입 단계로서 기대가 부풀려진 모습으로 보여 주고 있다. 기대만큼 기술이 성숙하려면 아직 해야 할 일이 많다는 것을 알려주고 있는 것이다.



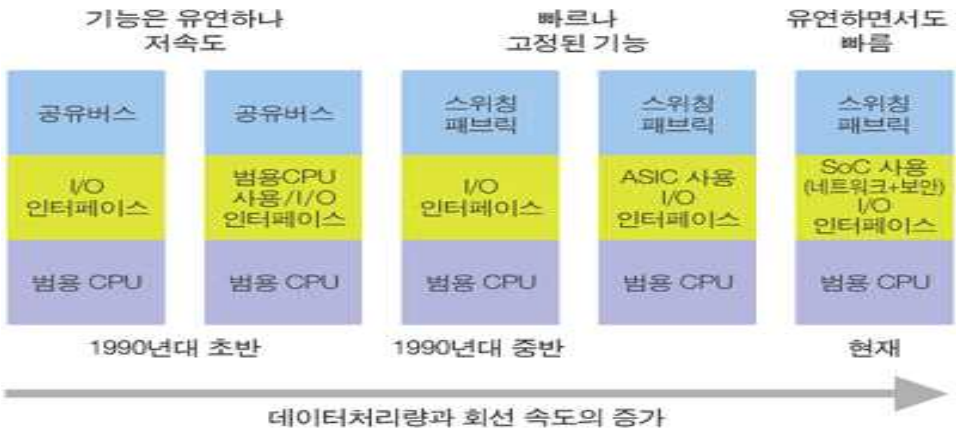
<그림 4> 보안의 하이프 사이클

그러나 실제적인 시장에서 IPS의 적용 효과는 현재의 기능만으로도 상당한 효과를 얻을 수 있는 것을 볼 수 있다. 이는 하이프 사이클과 같이 기술적으로 이상적인 목표를 향해서 발전하고 있는 IPS를 현재의

보유 기능으로 비교해서 평가하는 것이 아니다.

최근 웜 바이러스 등의 확산이 수만 노드도 10분여 만에 감염시키는 빠른 속도를 보이지만 기존의 IDS나 방화벽 등이 신속히 보안의 위협을 차단하지 못하는데서 IPS의 자동 차단효과가 돋보이고 있는 것이다. 그래서 특히 IT규모는 크면서 상대적으로 관리 인원이 적은 캠퍼스 네트워크 등에서 IPS의 도입 및 검토가 활발하게 이뤄지고 있다.

리눅스 기반의 네트워크 보안 어플라이언스가 하나의 추세이듯이 스위치 기반의 IPS도 하나의 추세가 될 수 있다. <그림 5> 네트워크 보안을 위한 시스템 아키텍처 발전에서 보듯이 처리속도는 빠르나 보안기능의 추가가 어려운 ASIC과 처리속도는 느리나 기능의 추가가 쉬운 CPU의 장점들만을 취할 수 있는 SoC(System-on-Chip)가 최근에 도입되는 것을 볼 수 있다. 몇 해전부터 네트워크 벤더들이 보안 회사들을 인수하면서 네트워크 장비에 보안기술을 내장하기 시작했고 빠른 보안기술 발전을 수용하기 위해 SoC를 채택한 네트워크 하드웨어 시스템 아키텍처가 일반화되고 있다.



<그림 5> 네트워크 보안을 위한 시스템 아키텍처 발전

5.2. 네트워크 보안 아키텍처와 IPS

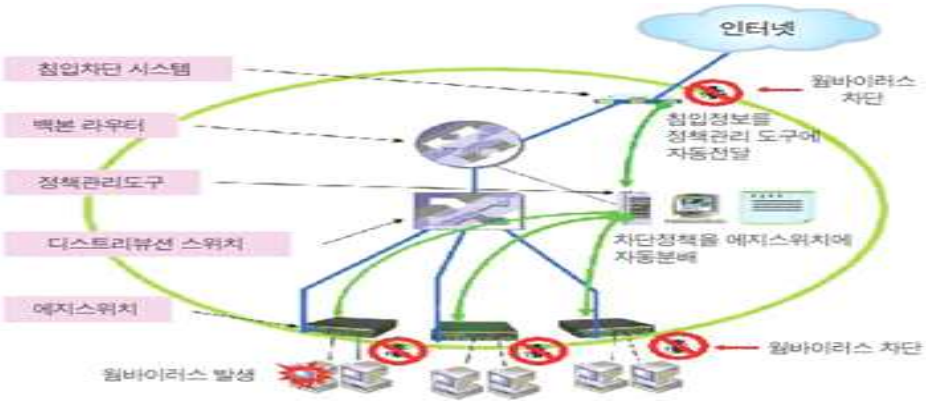
지금까지 IPS 등과 같은 보안 장비가 보안의 위협을 해결해 왔고 앞으로도 보안 장비의 역할은 계속 중요할 것이다. 그러나 실제로는 네트워크 보안을 설계시에 보안 제품만을 적용해서 웜바이러스 등의 피해를 줄이는 것이 한계가 있다. 최근에는 보안의 위협들이 모든 네트워크 포트를 통해서 빠르게 확산하고 있기 때문이다.

에지 스위치를 포함한 모든 네트워크 포트로부터 보안의 위협이 상존하는 현실에서 IPS만으로 더 이상 안전한 네트워크를 유지하는데 어려움을 많이 겪게 된다는 것이다. 그리고 이제 보안 사고의 피해를 본 후에 반응적으로 보안장비를 추가하는 모습으로는 보안의 위협으로부터 자유로울 수가 없다. 이 때문에 이제는 안전한 네트워크를 위해서 아키텍처적인 접근이 필요하고 IPS도 이에 대한 요소로서 고려해야한다.

네트워크 보안 아키텍처 기반으로 준비하지 못한 네트워크 상에서는 유비 쿼터스 환경으로 진행하면 할수록 사용자의 관리가 어려워지고 보안의 위협은 더욱 증가하게 된다. 특히 에지로부터의 보안 위협이 상존하는 환경이 되어 IPS 등과 같은 보안제품으로만 주요 네트워크 관문에서 자동으로 웜 바이러스 등을 차단하여 보안의 위협을 제거하는 것이 어렵다. 그렇다고 모든 네트워크 장비의 포트에 IPS를 설치할 수는 없다.

보안의 시스템 아키텍처적인 측면에서 IPS는 자동으로 침입차단을 하는 것뿐만 아니라 발생하는 보안 이벤트를 모든 에지 스위치 포트에도 연동하는 것이다. IPS는 주요 관문에서 유해한 데이터를 차단 하는 것 뿐만 아니라 에지 스위치의 포트를 포함해 모든 포트에서도 동시에 동일한 침입을 자동으로 차단하는 연동기능을 보유해야 한다는 것이다.

<그림 6> 에지와 인터넷 관문의 자동 차단 시스템 구성에서는 안전한 네트워크(Secure Network) 개념을 도입한 구성 사례와 이를 통해 네트워크 내에 모든 포트 차단의 자동화를 이루어 안전한 엔터프라이즈 네트워크를 유지하는 모습을 보여주고 있다.



<그림 6> 에지와 인터넷 관문의 자동 차단 시스템 구성

IPv6까지 고려하는 장기적인 관점으로 보면 에지 스위치와 연동하는 IPS의 모습에서 미래의 네트워크 보안 아키텍처의 모습도 일부를 볼 수 있다. 현재는 VPN에서만 사용자의 데이터 부분이 암호화가 되지만 IPv6에서는 모든 사용자 데이터 부분이 기본적으로 암호화 기능을 가진다. 기존의 콘텐츠를 분석하는 방화벽이나 IDS 등의 보안장비들이 능력을 발휘하지 못하는 환경이 되는 것이다.

이렇게 사용자의 데이터가 암호화되는 환경에서는 IP주소나 TCP주소 등의 컨텍스트(Context) 기반으로 다양한 플로우 그룹을 생성하여 차단/QoS/오차 한계(Rate Limiting) 등의 보안관련 기능을 중앙 집중적으로 관리해 이를 스위치에서 구현해야 한다. 여기서 IPS의 기능은 사용자의 데이터가 점차 암호화되는 과도기적 환경에서 컨텍스트 기반의 보안 정책을 컨 텍스트 기반의 보안 정책으로 변환하는 도구의 역할을 하는 것이다.

5.3. 사용자 입장에서의 요구 파악 '시급'

한 국내업체에서 수년 전 처음으로 IPS를 개발했다고 했었지만 안정성 문제로 주목을 받지 못했다. 그러나 그 후 IPS가 외국 기업들이 K인증이라는 환경의 장벽을 피할 수 있어서 보안솔루션 판매의 돌파구가 되었고, 이러한 측면이 IPS의 좋은 효과를 부각시키지 못하고 단점으로 비춰지기도 했다.

그러나 실제로 IPS의 적용 후 많은 효과를 볼 수 있었기 때문에 입소문을 타고 급격히 시장이 확대되고 있다. 이제는 보안을 하는 모든 기업에서 IPS를 이야기하고 있고 이러한 추세는 거스를 수 없어 보인다. 이제 국내 기업들에서는 IPS도 K인증을 제도화하려고 한다. 그러나 국내 기업들이 항상 잊지 않아야 할 것은 K인증에 안주하지 않고 사용자 입장에서 무엇이 필요한지 능동적으로 기능을 개선해 나가야 한다는 것이다.

네트워크 이슈의 시대적 변화는 Y2K를 지나 지금은 안전한 네트워크가 이슈가 되고 있고 아키텍처로서의 보안이 일반화되는 것이다. IPS도 네트워크 보안 아키텍처 속에서 필요한 기능 등을 끊임없이 추가해 사용자들에게 안전한 네트워크를 제공해야 할 것이다.

Ⅵ. 결　　론

네트워크는 날이 갈수록 복잡해지고 있고 그 반작용으로 네트워크 침입기술이 더욱 고도화, 지능화 되고 있다. 하지만 이 기술들은 웹과 그 밖의 인터넷 컨텐츠의 발달로 소위 스크립트 키디(Script kiddy)라 불리는 비전문적인 침입자들에게 쉽게 그 길에 들어 설 수 있게 해주고 있기 때문에 이제는 정확한 목적이 없는 대량 공격(Massive Attack) 시대가 되었고 네트워크 연결된 호스트나 그 밖 장비들은 이제는 더 이상 그런 위협에서 안전하다고는 말할 수 없다.

그런 위협지대에서 우리가 안전하게 방어하는 방법은 현재 보안의 최신동향을 빠르게 알아보고 그 동향에 맞추어 소스의 수정, 패치, 업그레이드를 통하여 위협점을 제거해 나가는 방법뿐이다.

그리고 보안에 대해서 단지 CEO, Admin에게만 국한하지 말고 상대적으로 보안인식 및 보안대책이 미비한 일반 사용자들 대상하는 방식으로 지속적으로 발전되어 나가는 기술을 일반 사용자들도 주기적으로 자동 갱신되는 백신의 사용과 더불어 보안정보에 지속적인 관심을 갖고, 주기적인 OS패치 및 취약점 점검 등의 대응을 하여야만 안전하게 인터넷 문화를 향유할 수 있을 것이다.

참 고 문 헌

[정부동향]

- [CERTCC] WORM_NETSKY.B 예보

. http://www.certcc.or.kr/cvirc/Alert/warning/2004/netsky_b.html

- [국정원] 국정원, '국가사이버안전센터' 설립

http://www.nis.go.kr/servlet/remote.FlowView/3718017486710910491 67877/bodo040219.htm

.

http://www.dt.co.kr/dtspelst_sec_view.html?gisaid=200402200201025 1699002&setitle=보안

[해킹취약점]

- Windows XP 사용자 모드에서 커널 모드 권한 획득 가능

. http://www.securiteam.com/windowsntfocus/5TP0B2KC0K.html

[웹사이트]

- 리눅스 포탈 http://www.linux.co.kr
- 리눅스 한글문서 프로젝트 http://www.kldp.org
- 해커스쿨 http://hackerschool.org
- 해커즈랩 http://www.hackerslab.org

[기술자료]

- [A3] Windows 시스템의 lsass.exe 서비스 거부공격 발생에 대한 보안권고안

감사의 글

먼저 오늘이 있기까지 많은 어려움 속에서도 헌신적인 사랑으로 키워 주시고 보살펴 주신 부모님께 깊은 감사를 드립니다.

본 논문이 완성되기까지 아낌없는 지도와 격려를 해 주신 이병천 교수님께 깊은 감사를 드리며, 바쁘신 와중에도 논문을 심사해 주신 양정모 교수님과 유승재 교수님께 감사를 드립니다. 그리고 학사과정 동안 많은 가르침을 주신 이용호 교수님, 이완범 교수님, 노창배 교수님, 이완복 교수님께도 감사를 드립니다.

논문작성에 도움을 준 동료 학우들께도 감사를 드립니다.

또한, 마음의 후원자인 동생, 친지 그리고 친구들에게도 진심으로 고마움을 전합니다.