1주차 목/금요일

UNIX System Security Management

이 장에서는 UNIX 시스템의 보안 위협 요소 (Security Threats) 및 취약점(Vulnerabilities) 과 보안 대책(Countermeasures)등을 알아본다. 또한, 알려진 보안 취약점 공격 도구와 보안 강화 도구 등을 실습한다.

목 차

참고 자료

“Essential System Administration”, *Aeleen Frisch, O’Reilly & Associates, Inc*

“UNIX System Administration Handbook”, *Evi Nemeth, Pentice-Hall*

“UNIX System Security Tools”, *Seth T. Ross, Mc Graw Hill*

“Practical UNIX & Internet Security”, *Simson Garfinkel and Gene Spafford, O’Reilly & Associates, Inc.*

“TCP/IP Network Administration”, *Craig Hunt, O’Reilly & Associates, Inc*

“The New Hacker’s Dictionary 3rd Ed.”, *Eric S. Raymond*

“유닉스 시스템 보안”, 정재훈, 영진출판사

“Linux Security HOWTO”, *Kevin Fenzi*

http://www.ibiblio.org/mdw/HOWTO/Security-HOWTO.html

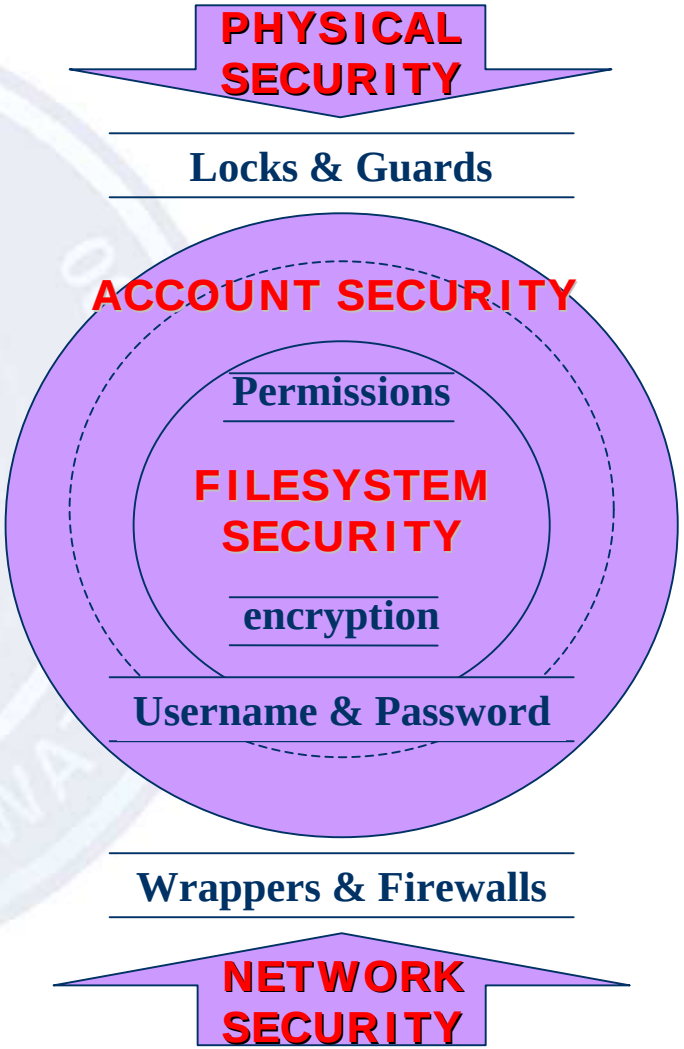
1. UNIX 보안 모델

UNIX 보안 모델

- 물리적 보안
- 계정 보안
- 파일시스템 보안
- 네트워크 보안

UNIX 보안 위협

- 물리적 보안 위협
- *Social Engineering*
- 운영체제나 프로그램 자체의 결함
- 관리자의 운영상의 오류



2. 물리적 보안

물리적 보안 위협요소

- 전산실 출입통제 부재로 인한 도난 및 손괴
  - 시스템 혹은 장비의 도난
  - 네트워크 장비의 훼손
- 시스템/콘솔의 접근통제 부재로 인한 불법적인 접근
  - 불법적인 시스템 부팅
  - 로그인된 콘솔에 접근

보안대책

- 전산실 출입통제 및 콘솔 접근통제
  - **Smart Card(ID card), Bio-metric Access control**
  - 출입일지 기록(출입시간, 사용목적 등)
  - **CD-ROM or Floppy Booting** 통제
  - **Screen Lock**

3. 계정 보안 (1)

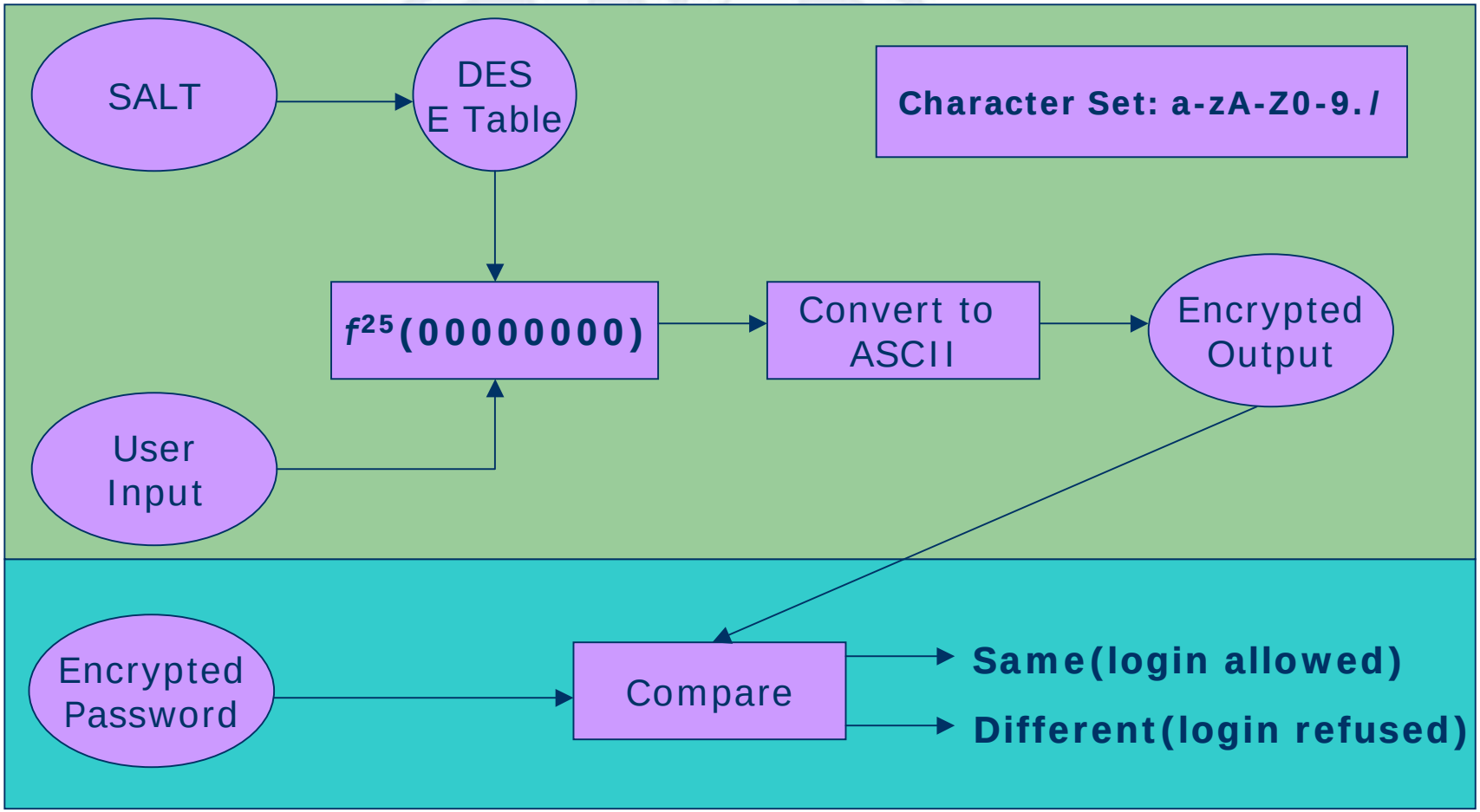
계정 보안 위협요소 - **impersonation**

- Accounts without password
- Inactive account
- 패스워드 추측 및 사전 공격
  - **nutcrack, John the Ripper**
- 네트워크 도청
  - **tcpdump, sniffit**
- Social Engineering
- Super User 권한의 오남용

관련용어: 암호(cryptography), 접근 통제(access control), 인증(authentication), 크랙(crack), DES, shadow,MD5, RSA, Accountability(책임추궁성), 수동적 공격(passive attack), IPSec/IPv6

참고 : UNIX Password Encryption (1)

Modified DES Unix Standard Password Encrypt



참고 : UNIX Password Encryption (2)

Modified DES Unix Standard Password Encrypt

```
/*  char  *crypt(const char  *key, const char  *salt) in <unistd.h>
    Jin-Yul, Kim (jykim@hackerslab.com)
    Unix Password Cracker - Basic
*/
#include <unistd.h>
int main(int argc, char *argv[])
{
        if(argc != 3) {
                fprintf(stderr, “Two arguments required: key and salt values\n”);
                exit(0);
        }
        printf(“Encrypted password of the key and salt you typed : %s\n”,
                crypt(argv[1], argv[2]));
}
```

Compilation Notes:
gcc -o crack crack.c -lcrypt

실습 : 무차별 사전 공격

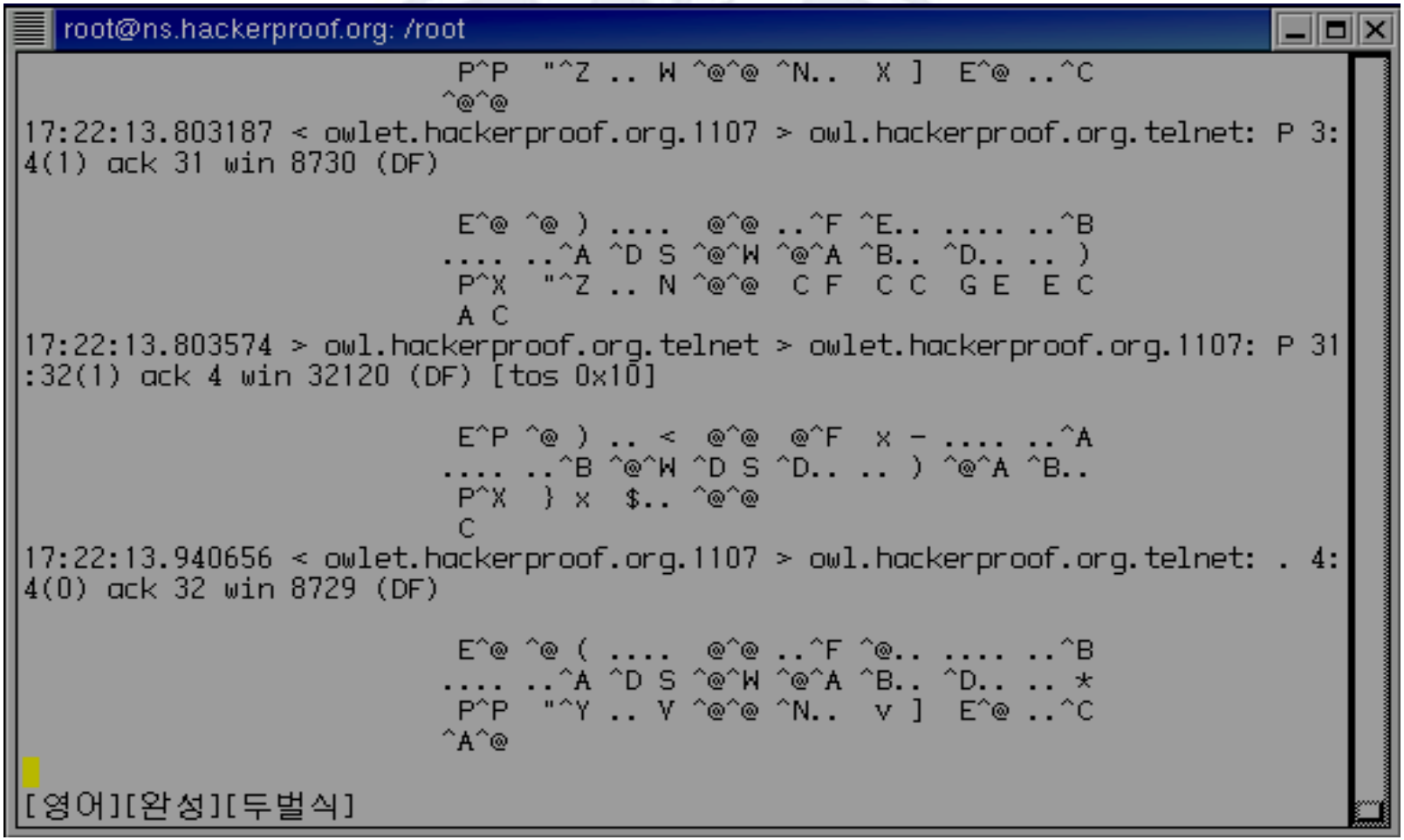
nutcrack

root@ns.hackerproof.org: /home/jykim/unixsec/nutcrack.1.0

```
[root@ns nutcrack.1.0]# ls
nutcrack  password  readme  words
[root@ns nutcrack.1.0]# ./nutcrack password words

Nutcracker version 1.0
Copyright 2000 by Ryan T. Rhea

got dict file: words

got passwd file: password

cracking...

user name         status           password
---------------   --------------   ----------------
root              CRACKED          frog
bob               unable to crack  X
tom               NO PASSWORD      NONE
ben               disabled         -
bitch             disabled         -
jeff              unable to crack  X
joe               disabled         -
jack              CRACKED          12345678

[영어][완성][두벌식]
```

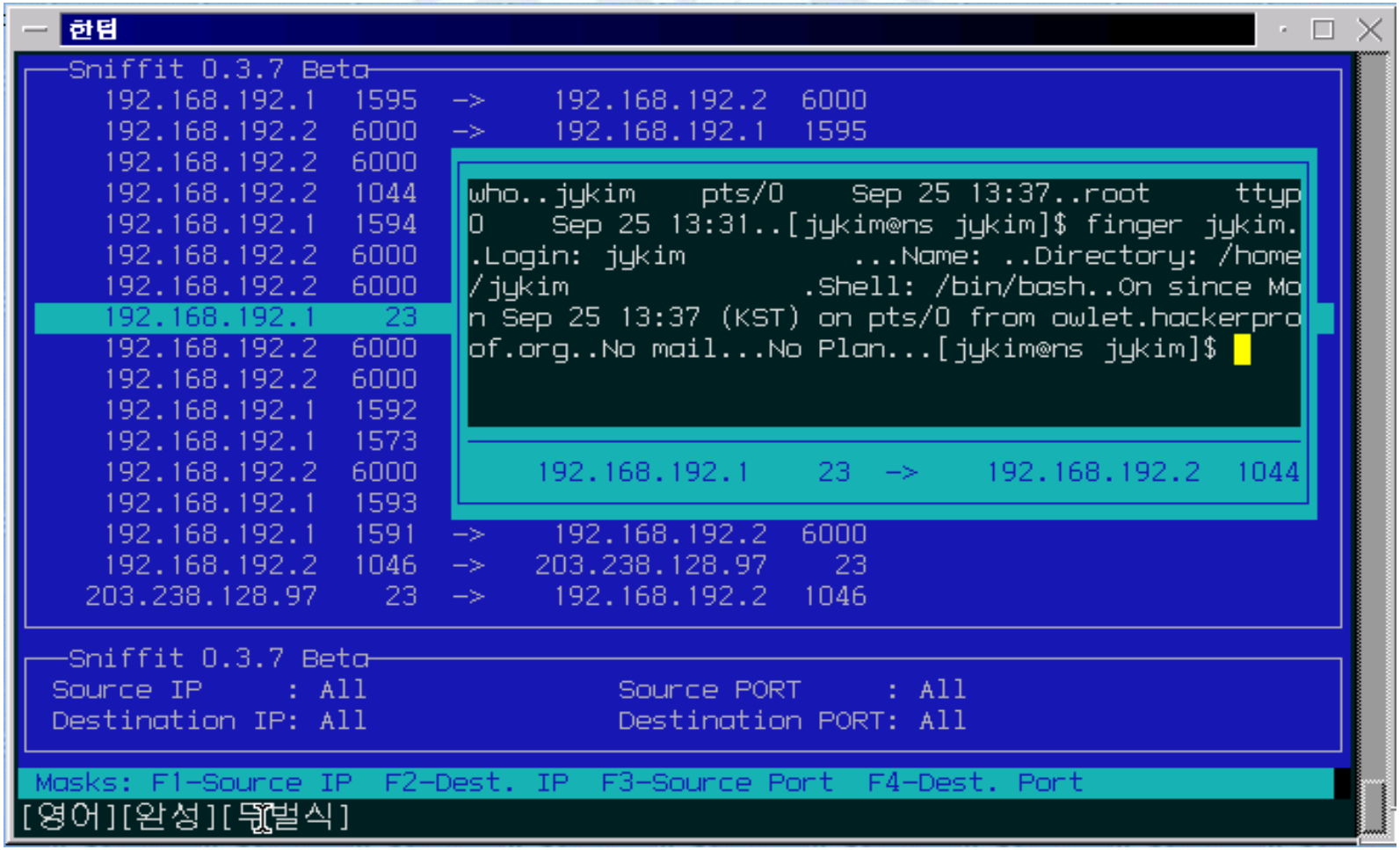
실습 : 네트워크 도청 (1)

tcpdump



실습 : 네트워크 도청 (2)

sniffit



3. 계정 보안 (2)

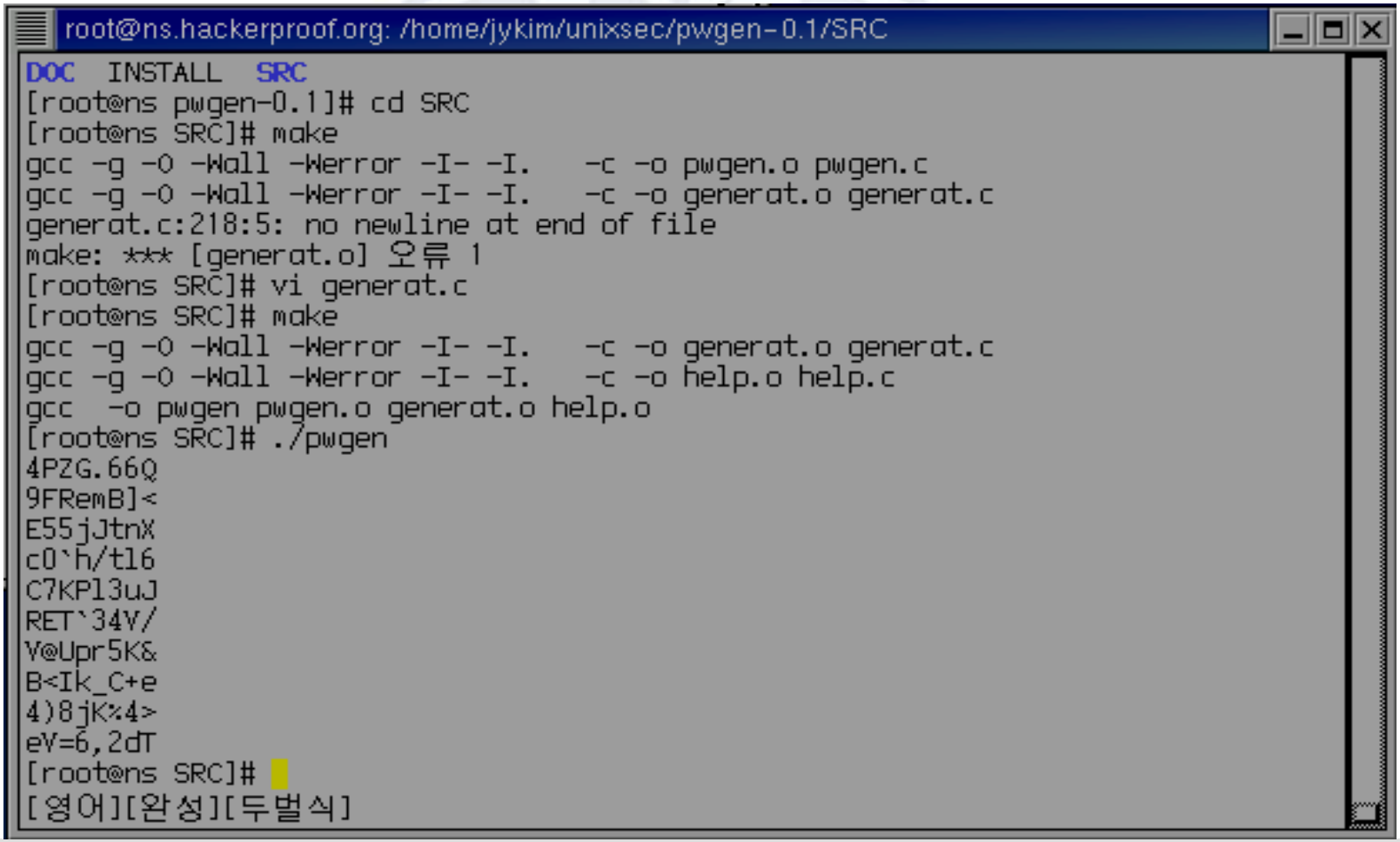
계정 보안 강화 대책

- 안전한 패스워드, 쉐도우 패스워드, 사전/사후 패스워드 검사
  - **In SVR4, at least 6 characters long and at least two letters and one digit or punctuation character.**
- One-Time Password(OTP) 및 Secure 통신 도구 사용
  - **Secure Shell(SSH)**
  - **Stelnet**
  - **SRP(Secure Remote Password Protocol)**
- PAM(Pluggable Authentication Modules)
- Restricted Environment(chroot)
- Kerberos 인증 시스템

관련용어: 암호(cryptography), 접근 통제(access control), 인증(authentication), 크랙(crack), DES, shadow,MD5, RSA, Accountability(책임추궁성), 수동적 공격(passive attack), IPSec/IPv6

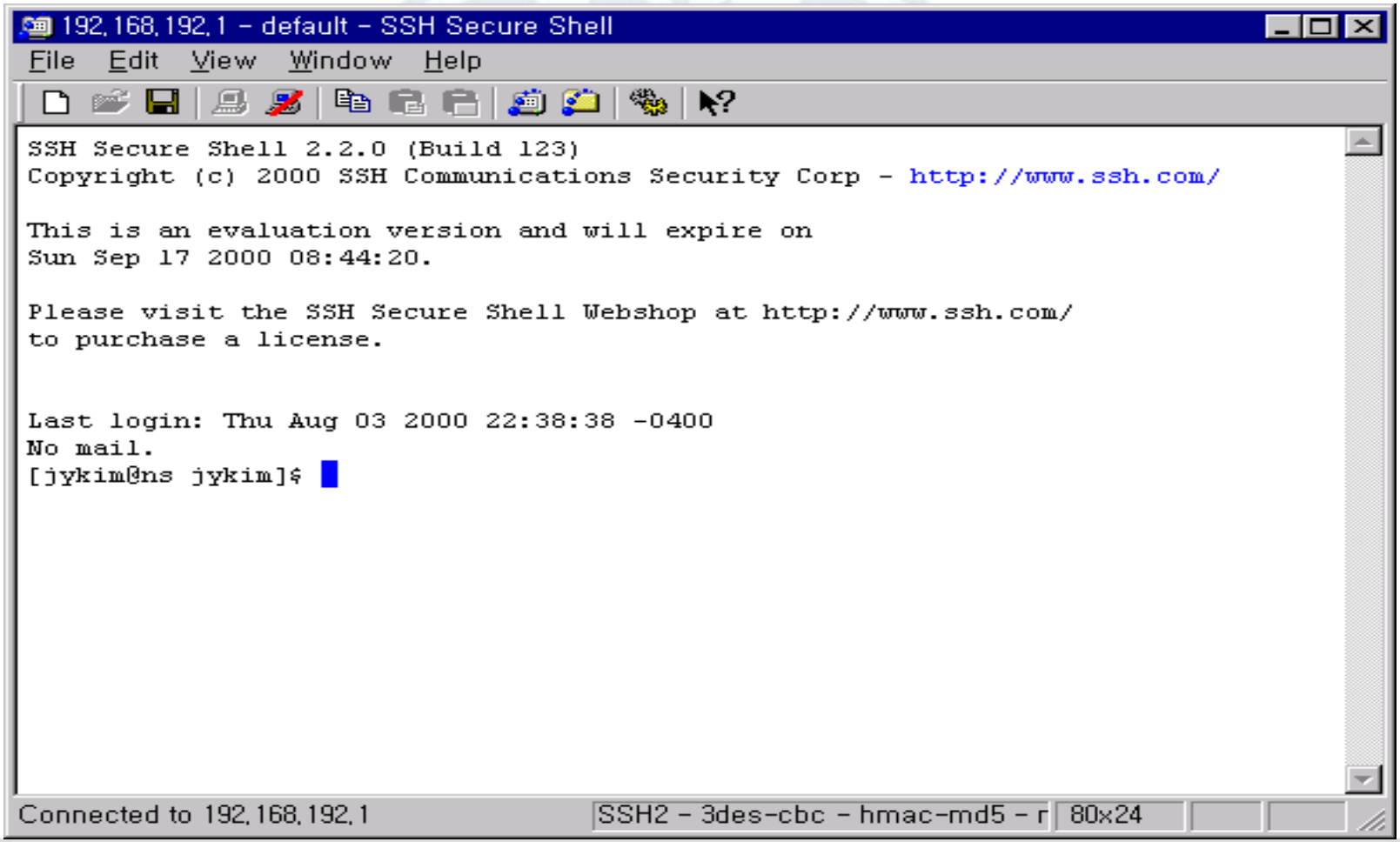
실습 : 패스워드 보안 강화 도구

Random Password Generator



실습 : 패스워드 보안 강화 도구

Secure Shell



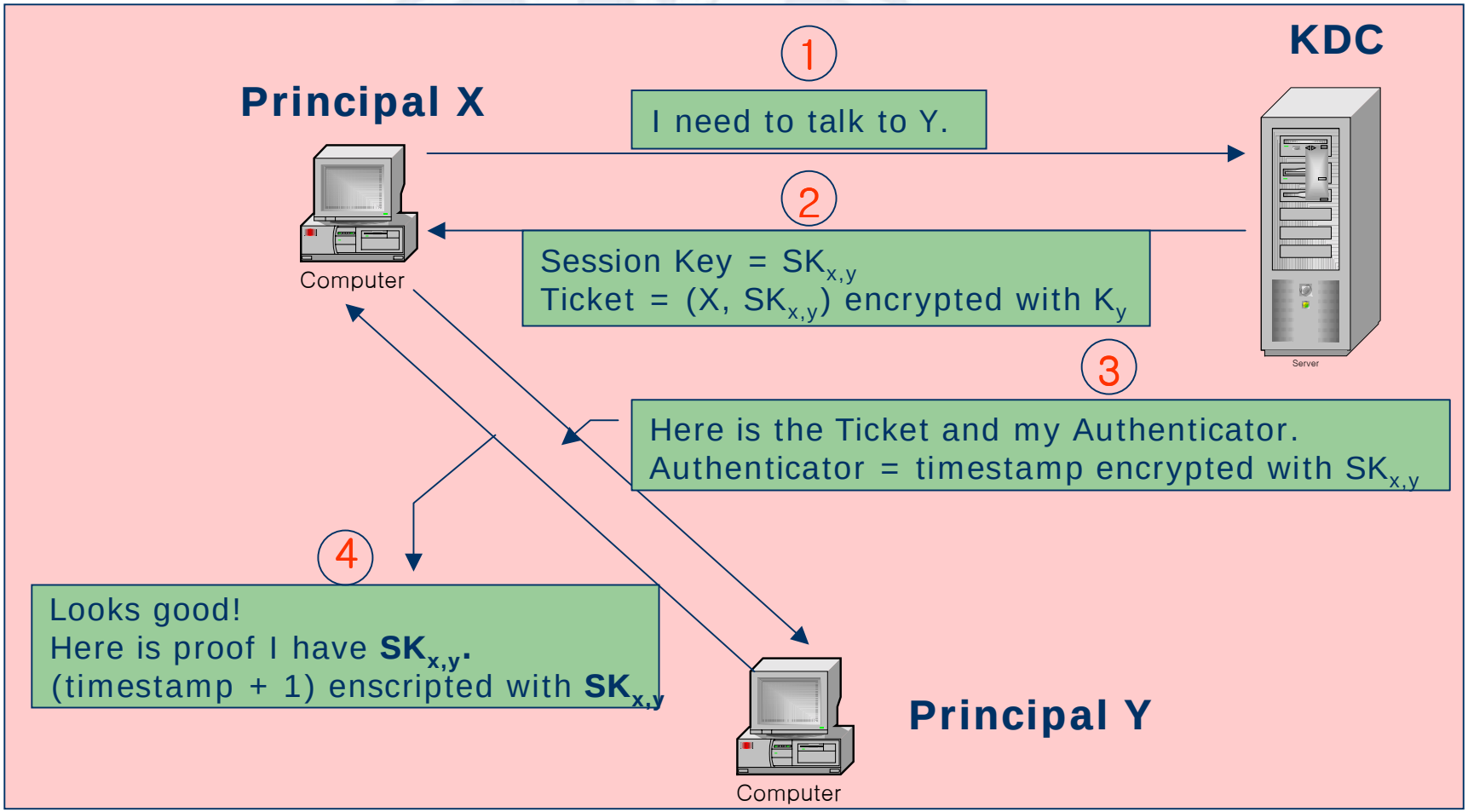
실습 : PAM

PAM(Pluggable Authentication Modules)

- 모듈타입
  - **auth, account, password, session**
    - auth : login과 같은 password check 서비스에 사용
    - account : account의 유효성 검사
      - 계정이 **expired** 되지 않았는지 검사
      - 로그인에 대한 통제(로그인 가능 시간 등)
    - password : password 설정시 password의 유효성 검사
      - **proactive password checking**
      - **dictionary-based crack**
    - session : login session 동안에 필요한 권한/자원 사용에 대한 로깅
- Config flag(인증방법)
  - **required, requisite, sufficient, optional**
- module
  - **pam_securetty.so, pam_unix.so, pam_nologin, pam_cracklib.so, etc**

실습 : 인증 시스템

Kerberos 인증 시스템



4. 파일시스템 보안 (1)

파일시스템 보안 위협 요소 - integrity

- 파일 및 디렉토리의 소유권 및 허가권 설정 오류
- 백도어 및 트로이 목마 프로그램
- 소프트웨어 설치 및 구성 설정 오류
  - **Setuid 프로그램 및 네트워크 관련 프로그램의 설정 오류**
- 프로그래밍 오류로 인한 취약점
  - **Race Condition**
  - **Buffer / Heap Overflow, Format String Bugs**
  - **DoS - System Resource Exhaustion Schemes**
  - 기타

관련용어: 임의적 접근통제(DAC; Discretionary Access Control), MAC(Mandatory Access Control), 무결성(Integrity), Race Condition, Input Validation Error, 서비스 거부(Denial of Service)

실습 : 소유권과 허가권

파일 및 디렉토리의 소유권 및 허가권

- File Permissions & Ownership
  - Owner : User(u), Group(g), Other(o) and All(a)
  - Access Modes : Read(r), Write(w), eXecute(x) and sticky(t), setuid/gid(s)

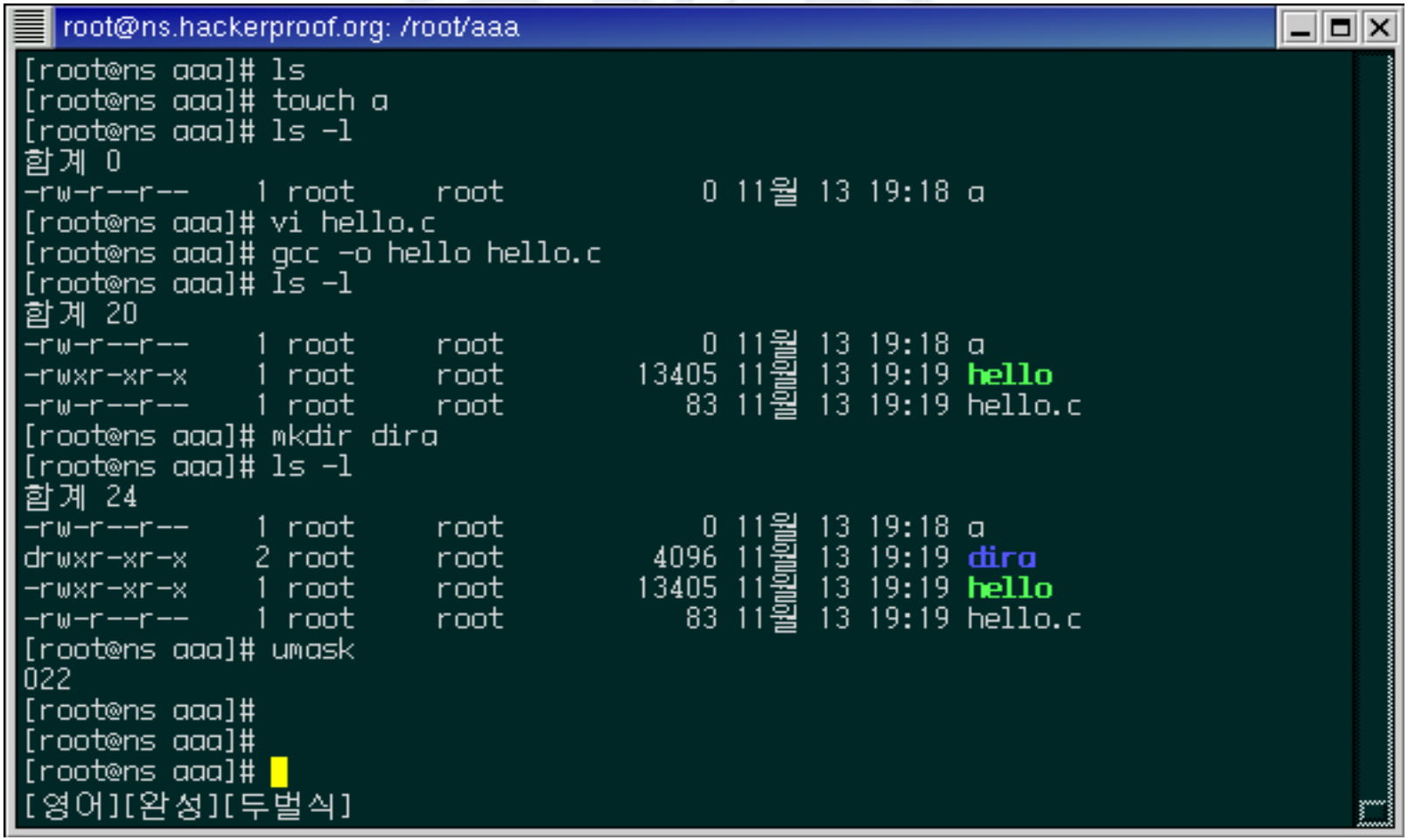
Access	Meaning on File	Meaning on Directory
r	View file contents	Search directory contents(e.g., use *ls*)
w	Alter file contents	Alter directory contents(e.g. delete files in it)
x	Run executable file	Make it your current directory (*cd* to it)

- Special purpose access modes

Code	Name	Meaning
t/T	Sticky bit	Keep executable in memory after exit
s/S	SUID or SGID bit	Set process user ID or group ID on execution

실습 : umask

umask(User file-creation mode mask)



실습 : 백도어와 트로이 목마

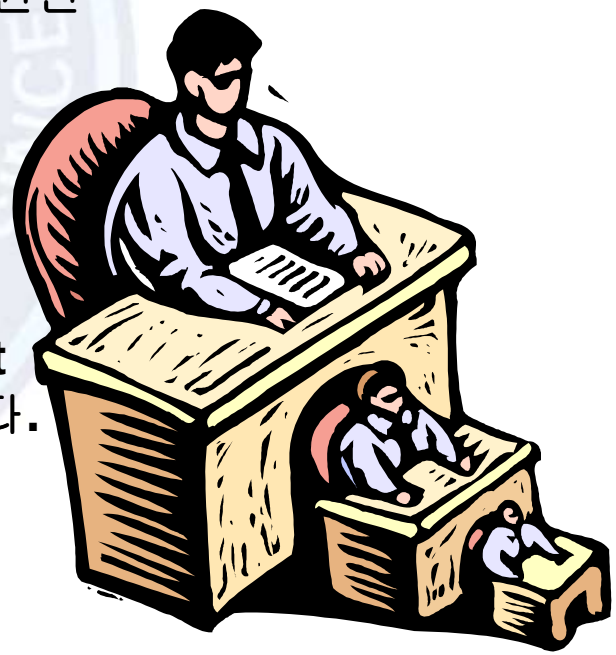
백도어 및 트로이 목마 프로그램

- 백도어(Backdoor)
  - 주로 **setuid** 비트가 켜진 프로그램으로 단지 실행하는 것만으로 혹은 간단한 조작만으로 쉽게 **root** 권한을 획득할 수 있도록 작성
  - **Example)**
    - cp /bin/sh /tmp/sh
    - chmod u+s /tmp/sh
- 트로이 목마(Trojan Horse)
  - 마치 정상적으로 동작하는 것으로 보이나 실제로는 엉뚱한 동작을 수행하도록 의도된 프로그램으로 다른 사용자의 실행을 유도하거나, 네트워크 데몬으로 동작
  - **Example : rootkit(lrk5.tar.gz)**
    - login.c
    - tcp.c (tcpwrapper의 가짜 버전)

실습 : Setuid 프로그램

Setuid 프로그램

- 주로 네트워크 서버 데몬이나 시스템 관리도구
  - 일반 사용자에게 필요에 의해 잠시 특권을 부여할때 사용된다.
  - **setuid bit**가 켜진 프로그램을 실행할 경우, 이 프로그램을 실행하는 동안 **"파일 소유자의 권한 (euid)"**을 가지게 된다.
    - 프로세스 생성 및 실행시
    - 파일 및 디렉토리 생성 및 접근시
    - 즉, 새로운 프로세스 생성 및 파일 생성시 effective user ID를 따른다.
  - **Backdoor, Buffer Overflow** 및 **Input Validation** 오류를 이용한 해킹에 사용된다.



참고 : 소프트웨어 설치 보안 가이드

Minimizing the Risks from Freely-Available Software

- 특권을 갖지 않은 일반 사용자 권한으로 테스트할 것!
- 미리 컴파일된 바이너리 파일보다 소스 코드로부터 직접 실행파일을 생성 및 실행할 것!
- 압축된(archived) 파일을 풀기전에 미리 테스트 및 안전한 영역에 풀고, 의심스런 바이너리 파일이 없는지 파일의 종류를 확인할 것!
- 의심스런 동작을 수행하지 않는지 소스 코드 및 Makefile 그리고 configuration 파일의 내용을 살펴볼 것!
- strings 등의 명령으로 오브젝트 파일을 살펴볼 것!
- SUID 및 SGID 프로그램 실행시 최소한의 권한으로 수행하는지 살펴볼 것!

실습 : 프로그래밍 오류

프로그래밍 오류로 인한 취약점

- Race Condition
  - 주로 임시 파일을 생성하는 **setuid** 프로그램에 대한 공격
  - **8lgm(http://www.8lgm.org)**
- Buffer Overflow
  - 주로 **root** 소유의 **setuid** 프로그램에 대한 공격으로 버퍼의 경계검사를 하지않아 발생
  - **Stack-based Overflow & Heap-based Overflow**
- 기타
  - **strings, gdb**
    - 실행 프로그램에 포함된 함수명이나 문자열 등의 정보를 출력

5. 파일시스템 보안 (2)

파일시스템 보안 강화 대책

- 주기적인 파일시스템 무결성 검사
  - **find, tripwire, md5, pgp**
- 시스템 보안 취약점 검사
  - **COPS, Tiger**
- 파일 시스템 암호화
  - **CFS(Crypto File System), TCFS**
- 파일시스템 백업 및 복구
  - **tar, dd, restore**
- Mandatory Access Control
  - **LIDS(Linux Intrusion Detection System)**

관련용어: 무결성(Integrity), 취약점(Vulnerability), Security Scanner, MD5(Message Digest V.5) Backup(on-site, off-site),

실습 : 무결성 검사

find, diff, grep, strings

- root 소유의 setuid 파일 찾기
- 최근 3일 내에 변경된 파일 목록
- 기존의 파일과 신규 파일의 변화된 내용
- 파일내에서 특정 문자열 검색

tripwire

- 모든 파일에 대하여 MD5 메시지 다이제스트 생성
- 파일 속성의 변화에 대하여 보고

실습 : 보안 취약점 검사

보안 취약점 검사

- COPS
  - **Compilation: ./reconfig 후 make**
- Tiger
  - 각종 인증 파일 점검
    - /etc/group, /etc/passwd, .rhosts, .netrc
  - 환경 설정 점검
    - PATH, ftp, mail aliases, cron, inetd, NFS export
  - 시스템 파일의 소유권/허가권 점검
  - 최근에 추가된 실행파일들 및 변경된 파일 조사

실습 : 백업 및 복구

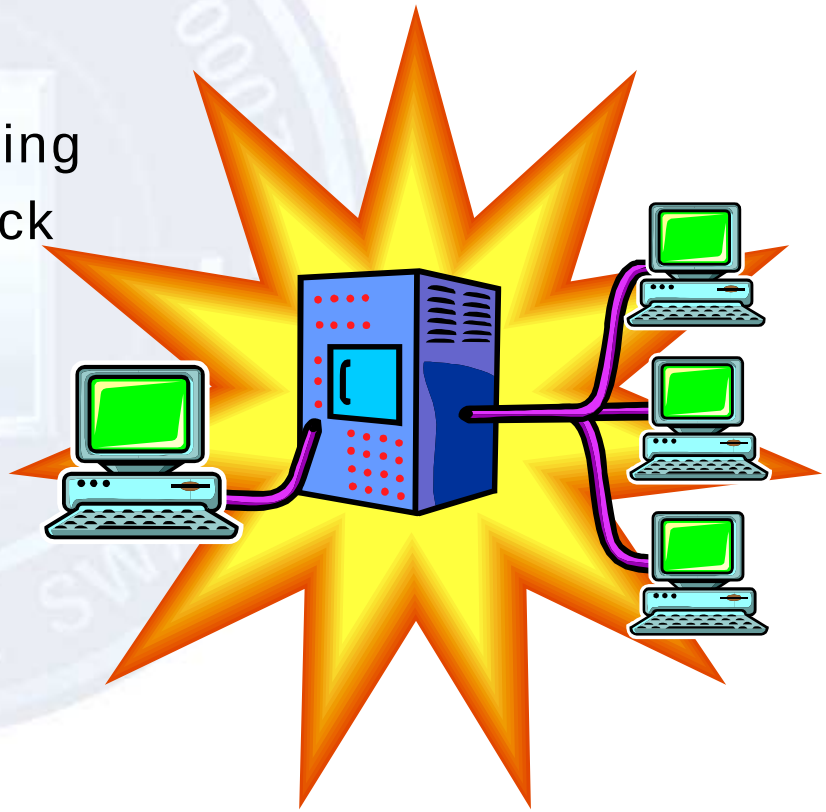
파일시스템 백업 및 복구

- tar(Tape Archive)
- dd(Disk Dump)
- cpio(Copy into or out)
- dump / restore
- mtio(magtape in/out)
- rmt(remote magtape)
- rdump / rrestore

5. 네트워크 보안 (1)

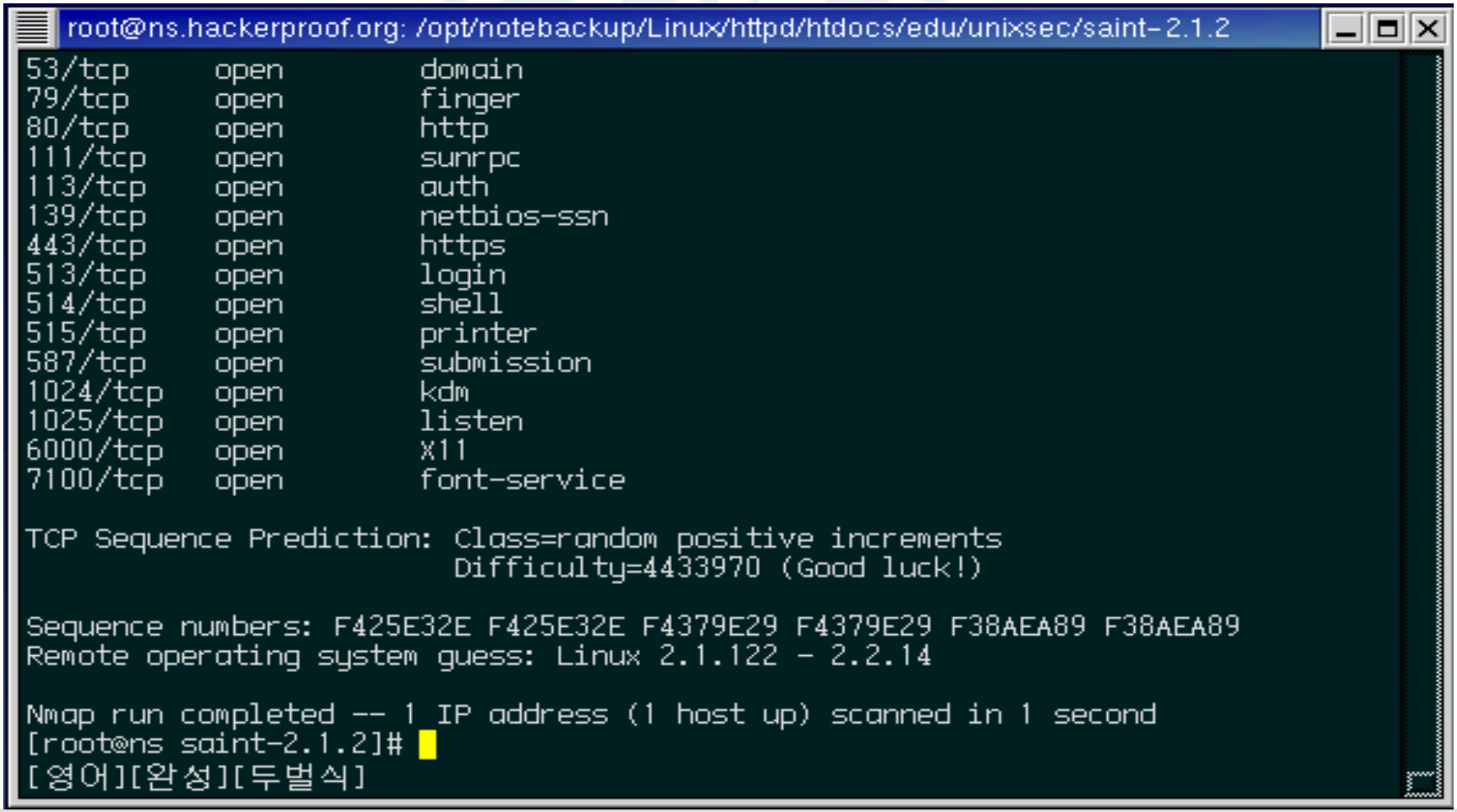
네트워크 보안 위협 요소 - **availability**

- Port Scanning & OS Detection
- Vulnerability Scanning
- NFS 및 NIS 정보수집 및 공격
- IP Spoofing & Session Hijacking
- Reverse Telnet / Bounce Attack
- DoS & DDoS



실습 : Port Scanning

nmap(Network Mapper)



실습 : Vulnerability Scanning

Vulnerability Scanning

- satan, satan, iss
- mscan, sscan
- cgiscan
- showmount –e hostname

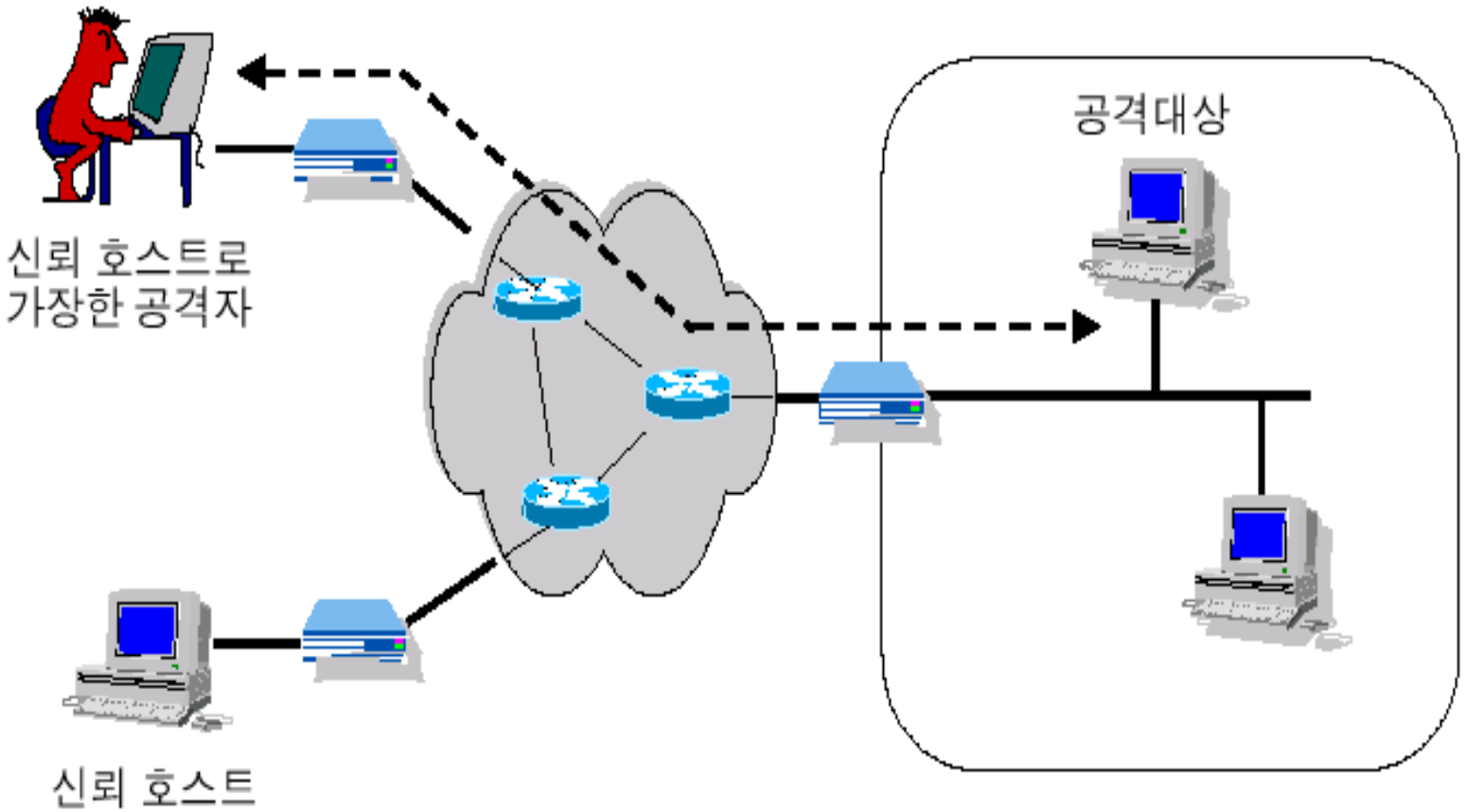
실습 : 리모트 공격

NFS/NIS Attacks

- Gathering Information
  - **rpcinfo –p victim_host, rusers, showmount –e victim_host, pscan –n**
- NFS Daemon Bugs
  - **nfsd, rpc.mountd, rcp.statd**
- R*-command Authentication Threats
  - **/etc/hosts.equiv**
  - **$HOME/.rhosts**

실습 : 리모트 공격

IP Spoofing & Session Hijacking



실습 : 리모트 공격

DoS/DDoS(Distributed Denial of Service)

- 여러 대의 컴퓨터가 한대의 서버를 공격하여 서비스를 하지 못하도록 방해하는 공격방법.

Victims in mid-February 2000

- Yahoo
- CNN Interactive
- Amazon.Com
- eBay
- Datek Online
- E*Trade
- ZDNet
- Buy.com

5. 네트워크 보안 (2)

네트워크 보안 강화 대책

- 인가되지 않은 접근차단 및 침입차단
  - **tcpwrapper, firewalls(ipchains,ipfilter)**
    - 호스트 수준 혹은 네트워크 수준의 필터링
- 보안 툴을 이용한 보안 점검
  - **SATAN, SAINT, ISS**
  - **mscan, sscan**
  - **nessus**
- 네트워크 감시 및 침입탐지
  - **netstat, tcpdump, snoop, sniffer**
  - **IDS - snort**

실습 : 인가되지 않은 접근 차단

TCP Wrapper – Restricting the Unauthorized Hosts

- /etc/inetd.conf : *inetd(8), inetd.conf(8)*

```
# <service_name> <sock_type> <proto> <flags> <user> <server_path> <args>
ftp         stream    tcp    nowait   root     /usr/sbin/tcpd    in.ftpd -l -a
telnet      stream    tcp    nowait   root     /usr/sbin/tcpd    in.telnetd
pop-3       stream    tcp    nowait   root     /usr/sbin/tcpd    ipop3d
imap        stream    tcp    nowait   root     /usr/sbin/tcpd    imapd
finger      stream    tcp    nowait   nobody   /usr/sbin/tcpd    in.fingerd
tftp        dgram     udp    wait     root     /usr/sbin/tcpd    in.tftpd
bootps      dgram     udp    wait     root     /usr/sbin/tcpd    bootpd
```

- /etc/hosts.{allow,deny} : *hosts_access(5),hosts_options(5)*

```
#<daemon_list> : <client_list> [ : shell-command ]
in.fingerd, in.rshd, in.rexecd : 192.168.192.
in.rlogind, in.telnetd, in.ftpd : 203.235.

ALL : UNKNOWN
ALL : PARANOID
in.fingerd, in.rlogind, in.telnetd, in.ftpd : ALL : spawn(/usr/sbin/safe_finger –l @h \
| /bin/mail –s %d-%h root) &
```

실습 : 침입 차단 (1)

ipchains(for Linux) – Packet Filtering Firewall

- 규칙의 목표(Target)

ACCEPT	패킷을 통과시킨다.
DENY	패킷을 무시한다.
REJECT	DENY와 같이 패킷을 무시하지만, 패킷을 보낸이에게 거절을 알림.
MASQ	IP Masquerading에 사용된다.(only for '**forward**' chains)
REDIRECT	패킷을 로컬 호스트의 특정 포트로 리다이렉션한다. (for 'Transparent Proxy')
RETURN	이전 사슬(chain)으로 되돌아가서 규칙을 찾는다.
User-Defined Chains	사용자 정의 사슬을 찾는다.

- 사슬(Chain)의 종류
  - **input – incoming packet** 필터링
  - **forward – forwarding packet**에 대한 필터링
  - **output – outgoing packet**에 대한 필터링
- Examples: Protection of Ping Attack
  - **ipchains -I input icmp –j deny**
- gfcc(GTK+ Firewall Control Center): http://icarus.autostock.co.kr

실습 : 침입 차단 (2)

Blocking spoofed or private addresses

- # rules for standard unroutables
  - **ipchains –A input –i eth0 –s 255.255.255.255/32 –b –j DENY**
  - **ipchains –A input –i eht0 –s 127.0.0.0/8 –b –j DENY**
- # rules for private(RFC1918) addresses
  - **ipchains –A input –i eth0 –s 10.0.0.0/8 –b –j DENY**
  - **ipchains –A input –i eth0 –s 172.16.0.0/12 –b –j DENY**
  - **ipcahins –A input –i eth0 –s 192.168.0.0/16 –b –j DENY**
- # rules for reserved addresses(multicast)
  - **ipchains –A input –i eth0 –s 240.0.0.0/5 –b –j DENY**
- # rules for protection internal network from spoofing
  - **ipchains –A input –i eth0 –s (internal network here) –j –l DENY**

실습 : 침입 차단 (3)

Blocking spoofed or private addresses (cont'd)

- Disable the source routing
  - **echo 0 > /proc/sys/net/ipv4/conf/all/accept_source_route**
  - **echo 0 > /proc/net/ipv4/conf/default/accept_source_route**
  - *First line sets a general policy, second sets a default that will be applied to all interface.*
- Testing your configuration
  - **ipchains –C input –i eth0 –p tcp –s 192.168.0.1/32 –d 0.0.0.0/0 25**
- Here,
  - **-A input : input chain**에 **rule**을 추가
  - **-i eth0 : rule**을 적용할 인터페이스
  - **-s :** 필터링을 위해 사용될 소스 어드레스
  - **-b : bi-directional**
  - **-l : syslog** 기능을 이용해 패킷을 기록**(log)**
  - **-j DENY :** 패킷을 거절**(sliently drop)**

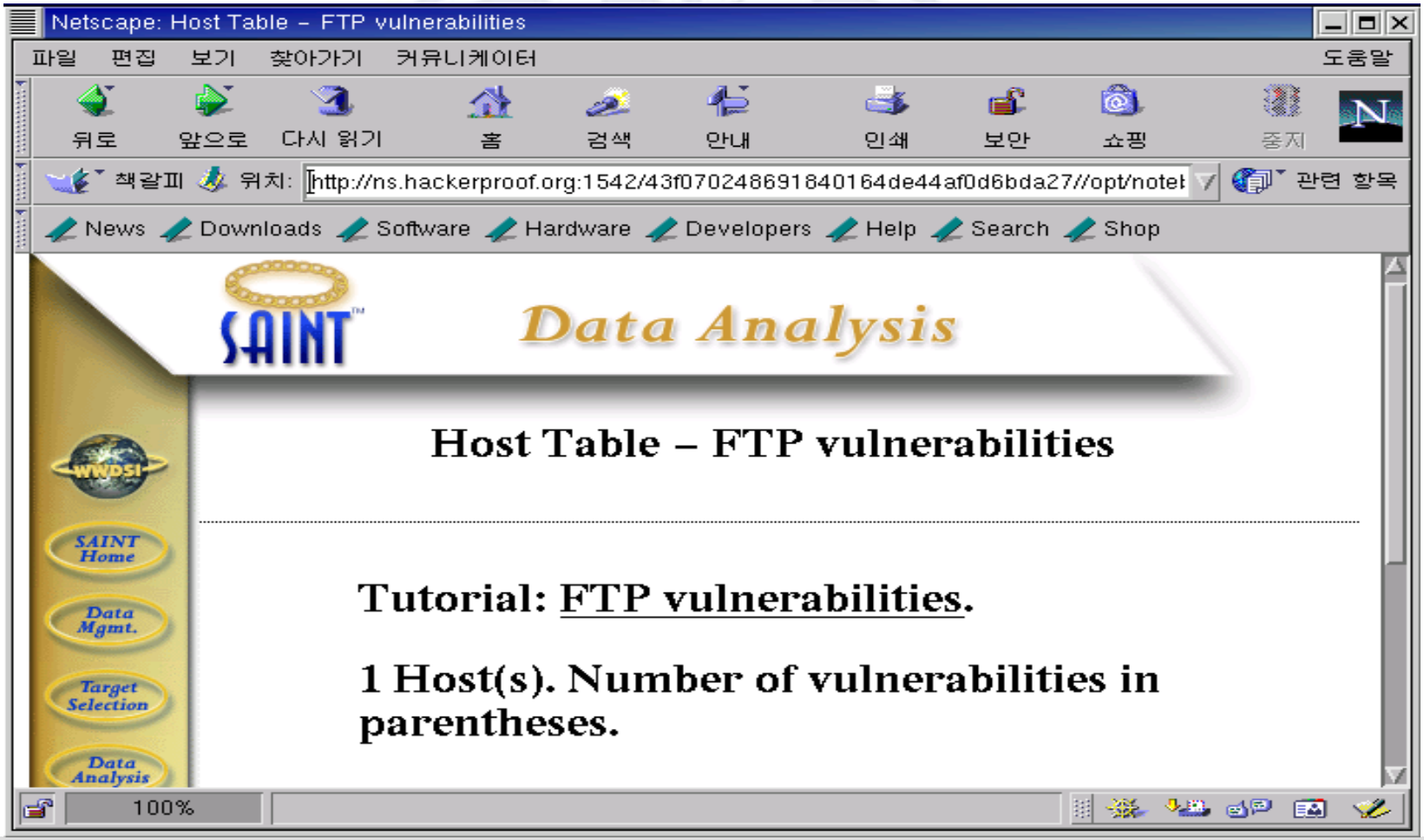
실습 : 침입 차단 (4)

Blocking Specific ICMP traffic

- rule to block incoming ICMP echo requests
  - **ipchains –A input –i eth0 –p icmp –s 0.0.0.0/0 –d 0.0.0.0/0 8 –l –j DENY**
- rule to block outgoing ICMP echo replies
  - **ipchains –A output –i eth0 –p icmp –s 0.0.0.0/0 –d 0.0.0.0/0 –l –j DENY**
- rule to block outgoing time exceeded and unreachable message
  - **ipchains –A output –i eth0 –p icmp –s (local network) –d 0.0.0.0/0 11 –l –j DENY**
  - **ipchains –A output –i eth0 –p icmp –s (local network) –d 0.0.0.0/0 3 –l –j DENY**

실습 : 보안 툴을 이용한 보안점검

SAINT



실습 : 네트워크 감시 및 침입 탐지

netstat

- 네트워크 연결, 라우팅 테이블, 인터페이스 통계, masquerade 연결 등을 보여줌
  - **netstat, netstat -nr**
  - **netstat -t, netstat -u, netstat -w**
  - **netstat -a, netstat --listening**
  - **netstat -c**

tcpdump

- “네트워크 도청” 참조

실습 : 침입 탐지 (1)

Snort

root@ns.hackerproof.org: /var/log/snort

```
-*> Snort! <*-
Version 1.6.3
By Martin Roesch (roesch@clark.net, www.snort.org)
USAGE: snort [-options] <filter options>
Options:
        -A          Set alert mode: fast, full, or none  (alert file alerts only)
                   "unsock" enables UNIX socket logging (experimental).
        -a          Display ARP packets
        -b          Log packets in tcpdump format (much faster!)
        -c <rules> Use Rules File <rules>
        -C          Print out payloads with character data only (no hex)
        -d          Dump the Application Layer
        -D          Run Snort in background (daemon) mode
        -e          Display the second layer header info
        -F <bpf>    Read BPF filters from file <bpf>
        -g <gname> Run snort gid as `gname' user or uid after initialization
        -h <hn>     Home network = <hn>
        -i <if>     Listen on interface <if>
        -l <ld>     Log to directory <ld>
        -M <wrkst> Sends SMB message to workstations in file <wrkst>
                   (Requires smbclient to be in PATH)
        -n <cnt>    Exit after receiving <cnt> packets
        -N          Turn off logging (alerts still work)
        -o          Change the rule testing order to Pass|Alert|Log
[영어][완성][두벌식]
```

실습 : 침입 탐지 (2)

cont'd

- Preprocessors
  - **preprocessor <name>:<options>**

- Available preprocessor modules
  - **mindfrag**
    - preprocessor minfrag: 128
  - **http_decode**
    - preprocessor http_decode: 80 8080
  - **portscan**
    - preprocessor portscan: 192.168.1.0/24 5 7 /var/log/portscan.log
  - **portscan-ignorehosts**
    - preprocessor portscan-ignorehosts: 192.168.192.5/32 192.168.3.0/24

실습 : 침입 탐지 (3)

cont'd

- Output modules
  - **output <name>: <options>**

- Available output modules
  - **alert_syslog**
    - output alert_syslog: <facility> <priority> <optios>
  - **log_tcpdump**
    - output log_tcpdump: snort.log
  - **log_postgresql**
    - output log_postgresql: <database name>, <parameter list>

실습 : 보안 감사 (1)

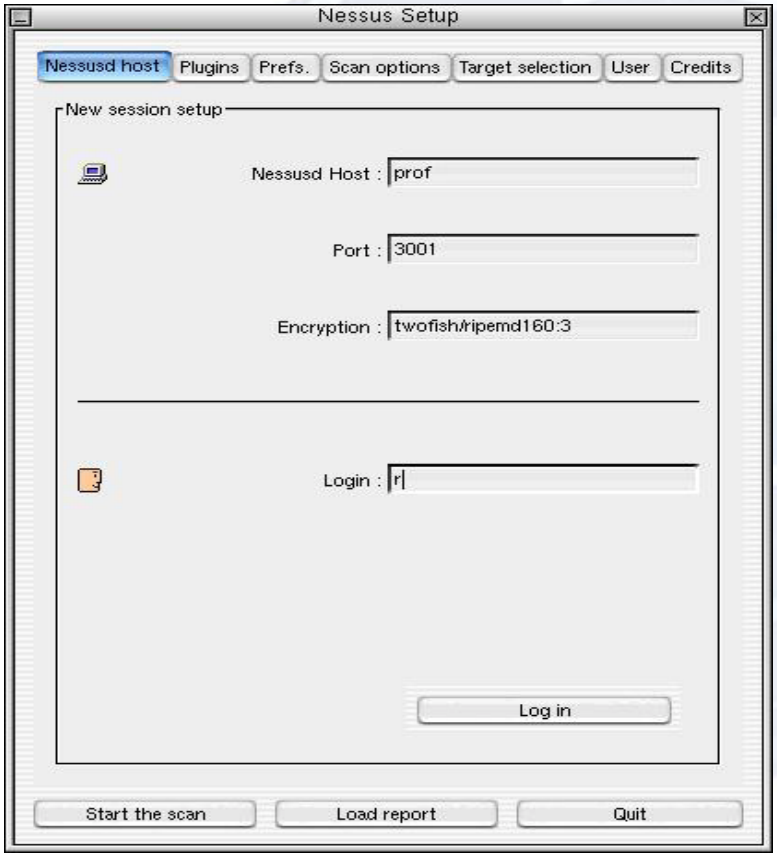
nessus

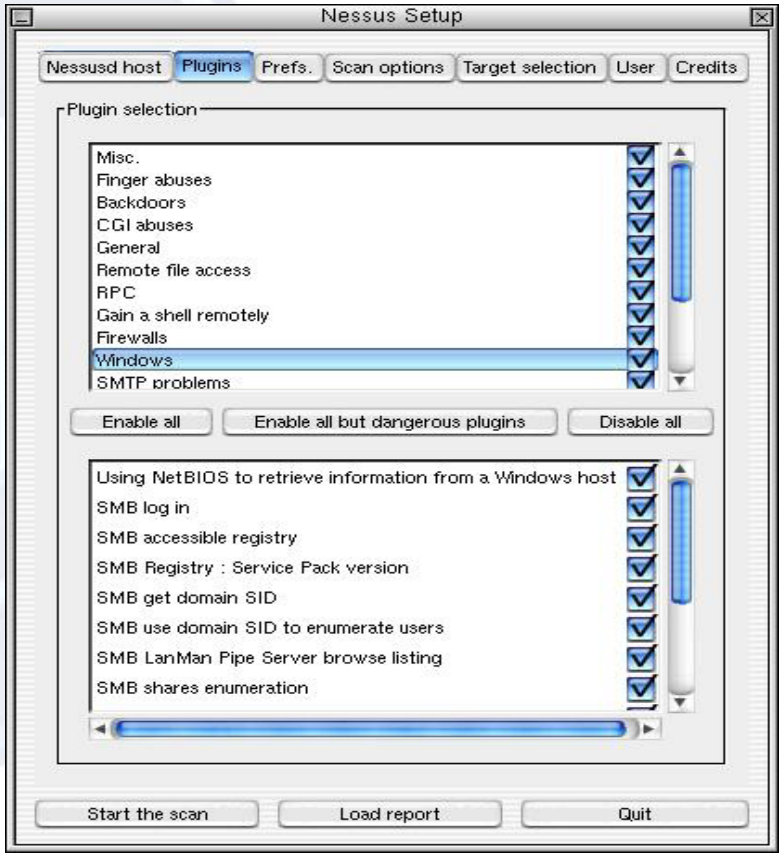
- Remote network security auditor
- Install nessus
  - **Install packages**
    - rpm –Uvh nessus-client-1.0.5-1.i386.rpm
    - rpm –Uvh nessus-devel-1.0.5-1.i386.rpm
    - rpm –Uvh nessus-plugins-1.0.5-1.i386.rpm
    - rpm –Uvh nessus-1.0.5-1.i386.rpm
  - **Create a nessusd account**
    - nessus-adduser
  - **Configure your nessus daemon**
    - Edit /usr/local/etc/nessus/nessusd.conf
  - **Start nessusd**
    - nessusd -D

실습 : 보안 감사 (2)

cont'd

- Client configuration

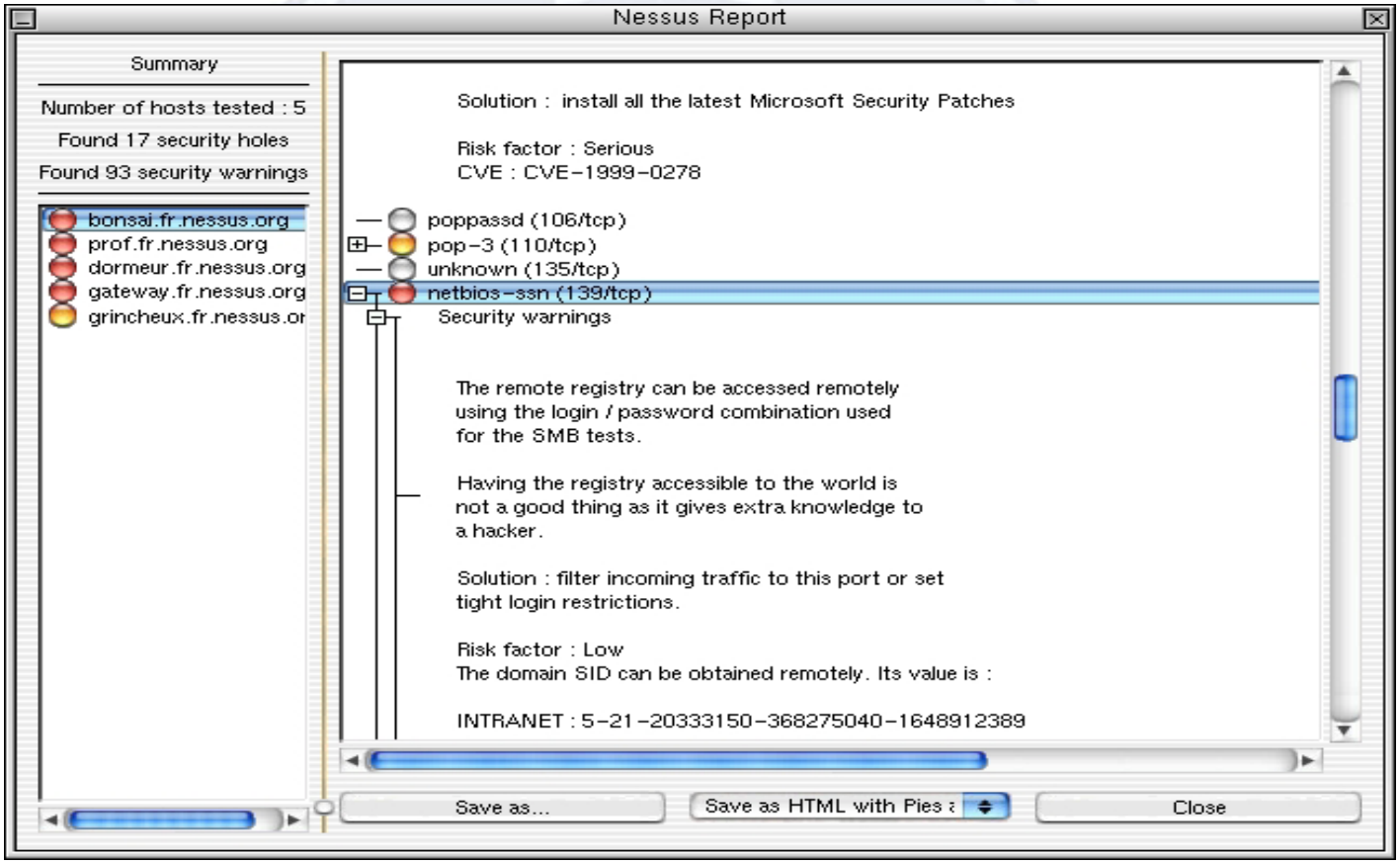




실습 : 보안 감사 (3)

cont'd

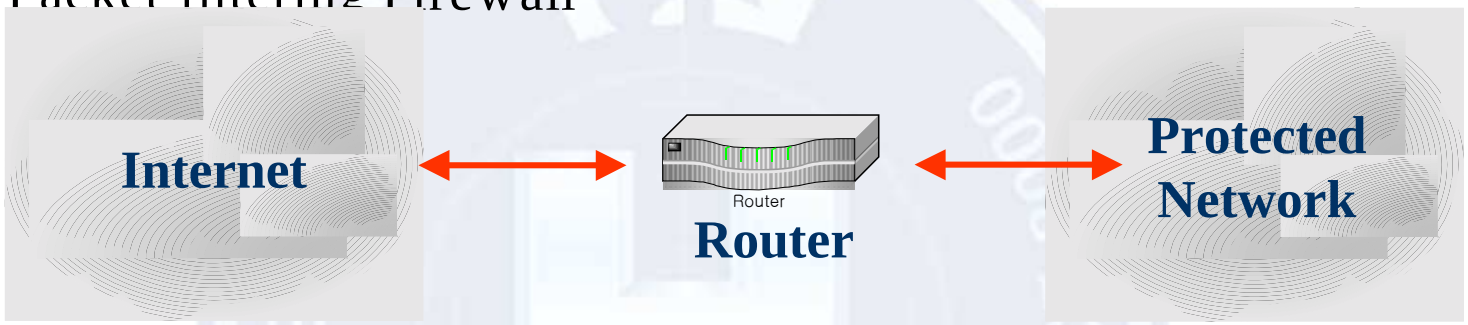
- Report interpretation



5. 네트워크 보안 (3)

Network-level Firewalls

- Packet-filtering Firewall



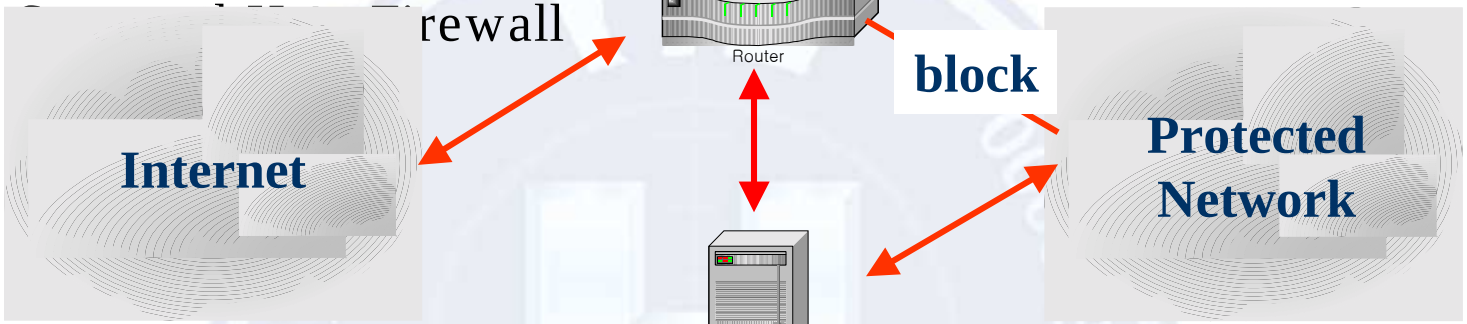
Application Gateways

- Dual-homed Host Firewall

5. 네트워크 보안 (4)

Screened Host

- rewall



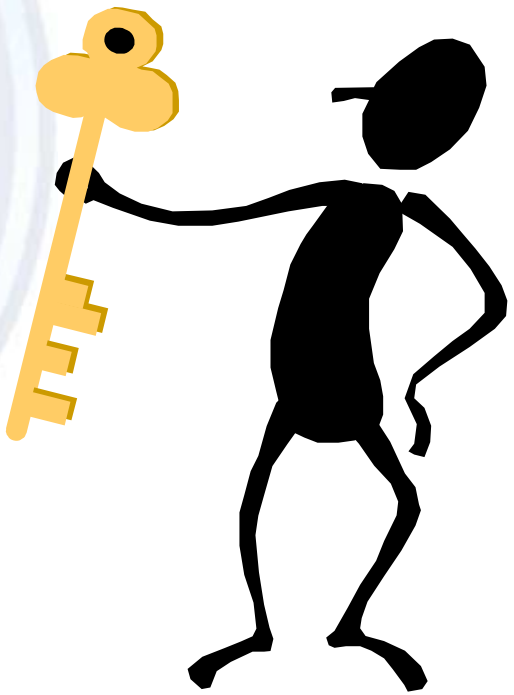
Screened Subnet

- Screened Subnet Firewall

6. UNIX 보안 관리 (1)

일반 사용자 관점의 보안 관리

- 좋은 패스워드 선택 및 주기적인 변경
- 안전한 통신을 제공하는 도구 사용
- 파일 및 디렉토리 허가권
  - **chmod, umask**
- 파일 암호화
  - **crypt, pgp, etc**
- 환경 설정 파일 관리
  - **.profile, .rhosts, .netrc, .exrc, etc**



6. UNIX 보안 관리 (2)

프로그래머 관점의 보안 관리 (1)

- 환경변수
  - 외부 환경 변수 무시(LD_LIBRARY_PATH, LD_PRELOAD, etc)
  - 필요한 환경 변수는 직접 설정(IFS, PATH, etc)
  - 절대 경로를 이용한 외부 프로그램 실행

- 버퍼 오버플로우
  - 입력 문자열의 길이 검사
  - 경계 검사를 지원하는 함수 및 컴파일러 사용
  - Overflow_wrapper 이용

6. UNIX 보안 관리 (2)

프로그래머 관점의 보안 관리 (2)

- Race Condition / 임시파일
  - 새로운 파일 생성시 access("file", W_OK);open("file",O_CREATE) 대신에 open("file", O_CREATE |O_EXCL); 함수 이용
  - 임시 파일 생성시 tmpnam(), tempnam(), mktemp() 대신에 mkstemp(), tmpfile() 함수 이용
  - 링크 파일 점검시 lstat() 혹은 open() 후 fstat() 함수 이용
- 자원 관리
  - Resource Exhaustion 방지
    - 메모리
    - 디스크
    - 기타 자원

6. UNIX 보안 관리 (3)

관리자 관점의 보안 관리

- 사용자 관리
  - 불필요한 계정 사용금지
  - 사전 및 사후 패스워드 검사
- 시스템 설정
  - **Network** 관련 서버 설정
    - inetd, tcp_wrapper, httpd, ftp, sendmail
- 보안 패치
- 로그 감사

7. 결 론

"완전 무결한 시스템은 존재하지 않는다."

- 지속적인 보안 관리 필요
  - 신속한 보안 패치
  - 보안 도구 이용
    - 취약성 검사 도구
    - 침입차단 및 침입탐지 시스템

위험 분석 → 보안 정책 → 보안 대책 → 보안 관리

- 위험 분석
  - 자산의 가치와 위협 및 취약점 분석

"보안 관리의 목표는 완전무결이 아니라
보안 위협으로 인한 피해를 최소화하고,
최대한의 가용성**(availability)**을
제공하는 것이다."