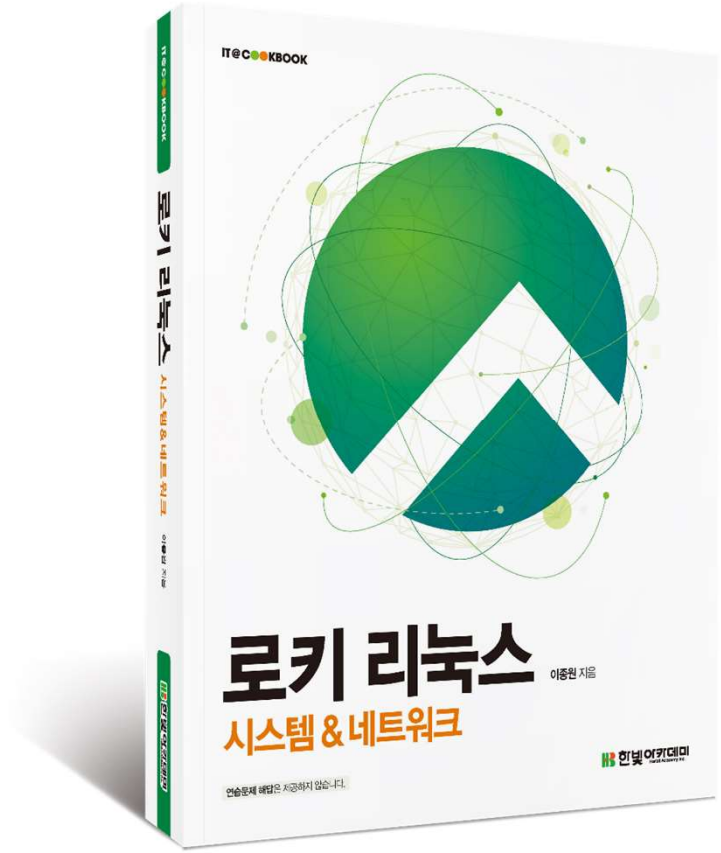

Chapter 15 리눅스 보안의 기초

목차

01 정보 보안의 기초

02 시스템 로그

03 방화벽 관리

04 보안 관리 도구

학습목표

- 정보 보안의 중요성을 이해하고 보안의 3요소를 설명할 수 있다.
- 로그가 무엇인지 이해하고 로그 파일을 관리할 수 있다.
- 방화벽을 설정하여 서비스나 포트를 허용하거나 차단할 수 있다.
- 보안 관리 도구를 이해하고 사용할 수 있다.

00 Preview

00 Preview

15장의 내용 구성

- 보안에 대한 기본적인 내용을 이해하고 있어야 함
- 리눅스는 보안 관리를 위해 로그 기능을 제공.
  로그는 시스템의 여러 서비스가 동작하면서 남기는 메시지로, 이 메시지를 잘 살펴서 이상이 없는지 확인해야 함
- 방화벽은 외부의 침입을 차단하는 가장 기본적인 도구
- 리눅스의 보안을 강화하기 위해 인증 기능을 제공하는 PAM과 더욱 강력한 접근 권한을 제공하는 SELinux에 대해 알고 있어야 함

01 정보 보안의 기초

01 정보 보안의 기초

■ 정보 보안의 정의

- 정보 보안: 정보 자산을 여러 가지 위협으로부터 보호하여 기밀성, 무결성, 가용성을 유지하는 것
- 이 세 가지를 정보 보안의 3요소라고 하며, 영어의 머리글자를 따서 'CIA 삼각형'이라고도 함

그림 15-1 정보 보안의 3요소

01 정보 보안의 기초

기밀성

- 기밀성: 허가 받은 사용자만 해당 정보에 접근할 수 있도록 하는 것. 정보가 불법적으로 공개되거나 노출되지 않도록 하는 것
- 이와 관련된 기능으로 각종 서비스를 사용할 때 사용자 인증하기, 읽기·쓰기 등에 접근 제어 설정하기, 데이터 암호화하기 등이 있음

무결성

- 무결성: 정보가 무단으로 변조되지 않았다는 것. 해당 정보가 완전하고 정확하다는 것을 보장. 믿을 수 있는 정보라는 뜻
- 무결성을 유지하기 위해 정보가 원본과 동일하다는 것을 보증하는 여러 가지 전자 서명 기법을 활용

01 정보 보안의 기초

■ 가용성

- 가용성: 인가를 받은 사용자가 필요할 때 정보나 서비스에 접근할 수 있는 것
- 정보가 있어도 필요한 순간에 접근할 수 없다면 의미가 없음
- 가용성은 외부의 공격을 받아서 문제가 발생하기도 하지만 전력 불안정, 하드웨어와 소프트웨어의 결함에 의해 문제가 발생할 수도 있으므로 시스템 관리자는 주기적으로 시스템 상태를 점검해야 함

01 정보 보안의 기초

■ 불필요한 서비스 통제하기

- 보안 위협은 네트워크를 통해 발생. 꼭 필요하지 않은 서비스 포트는 모두 막아 두어야 하는데, 일반적으로 모든 포트를 막고 서비스를 제공하려는 포트만 열어주는 것이 좋음
- 서비스를 통제하는 방법으로 불필요한 서비스 자체를 제거하는 방법과 방화벽에서 패킷을 필터링하는 방법을 함께 사용하는 것이 바람직함

■ 소프트웨어 패치 실시하기

- 주요 리눅스 배포판은 이미 제공한 소프트웨어에 대한 패치를 지속적으로 제공하고 있음
- 시스템 관리자는 패치 관련 발표에 주의를 기울이고 있다가 패치가 나오면 즉시 설치하는 것이 좋음
- 원래 패치는 드러난 어떤 문제를 고치는 것
- 패치를 설치한 이후에 시스템이 정상적으로 동작하는지 상태를 재확인할 필요가 있음

01 정보 보안의 기초

주기적으로 점검하기

- 시스템에 언제 어떤 문제가 발생할지 알 수 없음. 그러므로 프로세스의 목록과 사용자의 상태, 서비스의 동작 상태, 네트워크 연결 상태, 디스크의 남은 용량 등을 주기적으로 확인해야 함
- 점검할 항목을 체크리스트로 만들고 매일, 매주 등 주기를 정해 점검하는 습관을 길러야 함
- 시스템과 네트워크의 점검을 도와주는 도구를 적극적으로 활용하는 것이 좋음

백업하기

- 여러 가지 보안 대책을 설정하고 주기적으로 점검한다고 해도 문제는 어디선가 발생
- 이 때를 대비하여 주요 시스템 설정과 소프트웨어, 사용자 데이터 등을 주기적으로 백업해야 함
- 문제가 발생했을 때 빠르게 복구하는 방법도 연습해 두어야 함

01 정보 보안의 기초

■ 공부하는 시스템 관리자

- 해킹 기법이 날로 교묘해져 이를 방어하기 위해 다양한 보안 기술이 등장하고 있음
- 시스템이 계속 업그레이드되고 네트워크가 더욱 복잡해지며 사용자의 요구 수준은 점점 높아지고 있음
- 예전에 배운 지식과 기술에만 의지한다면 도태되고 말 것
- 시스템 관리자는 늘 공부하는 자세로 신기술의 검토와 적용을 고민하고, 서비스를 안정적으로 운영하기 위해 꾸준히 노력해야 함

02 시스템 로그

02 시스템 로그

■ 로그 파일

- 로그는 커널과 리눅스 시스템이 제공하는 여러 서비스 및 응용 프로그램이 발생시키는 메시지를 뜻함
- 로그를 저장한 파일을 로그 파일이라고 하는데, 리눅스 시스템에는 다양한 로그 파일이 있고 각 로그 파일에 저장되는 정보의 종류도 다양함
- 로그 파일을 통해 시스템 상태를 확인할 수 있으므로, 보안 사고에 대비하며 사고가 발생했을 때는 원인을 규명하고 침입 경로를 추적하기 위해 가장 기본적으로 로그 파일을 확인하게 됨

02 시스템 로그

■ 주요 로그 파일

- 리눅스 시스템에는 기본 로그 파일을 비롯해 다양한 로그 파일이 있으며 대부분의 로그 파일은 /var/log 디렉터리에 있음
- 시스템에 따라 /var/log 디렉터리의 파일 종류나 개수가 다르게 출력될 것

```
[user1@localhost ~]$ ls /var/log
README@              hawkey.log-20230924  speech-dispatcher/
anaconda/            hawkey.log-20231001  spooler
audit/               hawkey.log-20231008  spooler-20230921
boot.log             httpd/               spooler-20230924
boot.log-20230711    kdump.log            spooler-20231001
boot.log-20230829    lastlog              spooler-20231008
boot.log-20230901    mail/                sssd/
boot.log-20230922    maillog              tallylog
boot.log-20230924    maillog-20230921     tuned/
(생략)
```

02 시스템 로그

■ 주요 로그 파일

- /var/log 디렉터리의 내용을 보면 같은 파일명에 날짜가 붙은 파일이 여러 개 있음. 이는 로그를 계속 한 파일에 저장하면 파일 크기가 너무 커져서 파일내용을 보거나 관리할 때 불편하기 때문에 나눈 것
- 시스템 관리자는 날짜가 오래된 로그 파일을 백업한 후 삭제하는 것이 좋음
- 로그 파일의 소유자는 대부분 root 계정이고 접근 권한이 600으로 설정됨. root 계정만 읽거나 쓸 수 있게 설정됨
- 보안 측면에서도 일반 사용자 계정에서 로그 파일의 내용을 함부로 볼 수 없게 하는 것이 바람직함

02 시스템 로그

■ 주요 로그 파일

```
[user1@localhost ~]$ ls -l /var/log
합계 5516
lrwxrwxrwx. 1 root root    39 7월  9 10:39 README -> ../../usr/share/doc/systemd/README.
logs
drwxr-xr-x. 2 root root  4096 7월  9 10:46 anaconda/
drwx------. 2 root root    23 7월  9 10:47 audit/
-rw-------. 1 root root 56412 10월 9 10:41 boot.log
-rw-------. 1 root root 37392 7월 11 21:08 boot.log-20230711
-rw-------. 1 root root 36134 8월 29 20:31 boot.log-20230829
-rw-------. 1 root root 54174 9월  1 20:56 boot.log-20230901
-rw-------. 1 root root 17947 9월 22 07:26 boot.log-20230922
-rw-------. 1 root root 72049 9월 24 00:00 boot.log-20230924
-rw-------. 1 root root 17974 9월 25 20:02 boot.log-20230925
-rw-------. 1 root root 37619 10월 2 09:25 boot.log-20231002
-rw-rw----. 1 root utmp  2304 10월 2 18:59 btmp
(생략)
```

02 시스템 로그

■ 주요 로그 파일

표 15-1 리눅스의 주요 로그 파일

로그 파일	설명
/var/log/anaconda/*	리눅스 설치 과정에서 발생한 메시지를 기록한다.
/var/log/audit/*	auditd 데몬이 생성한 정보를 기록한다.
/var/log/boot.log	부팅 시 서비스 데몬의 실행 상태를 기록한다.
/var/log/btmp	실패한 로그인 기록으로, 바이너리 파일이어서 last -f /var/log/btmp 또는 lastb 명령으로 확인할 수 있다.
/var/log/cron	cron 실행 결과와 오류 메시지를 기록한다.
/var/log/cups/*	cupsd 데몬이 생성하는 로그를 기록한다. cupsd 데몬은 인터넷 프린팅 프로토콜을 지원하는 데몬이다.
/var/log/dnf.log	dnf 명령으로 설치된 패키지와 관련된 메시지를 기록한다.
/var/log/httpd/*	아파치 웹 서버에 의해 생성된 메시지를 기록한다.
/var/log/journal/*	journald 데몬이 관리하는 메시지를 기록한다.
/var/log/firewalld	firewalld 데몬이 생성하는 메시지를 기록한다.
/var/log/lastlog	각 계정의 가장 최근 로그인 정보를 기록하며 lastlog 명령으로 확인할 수 있다.
/var/log/mail/*	메일 송수신과 관련된 메시지를 기록한다.
/var/log/mariadb/*	MariaDB에 의해 생성된 메시지를 기록한다.
/var/log/messages	시스템 공통 로그로 시스템의 부팅과 커널, 대다수 서비스가 생성하는 메시지를 저장한다. 단, rsyslog가 설치되어 있어야 한다.
/var/log/samba/*	삼바에 의해 생성된 메시지를 기록한다.
/var/log/wtmp	로그인 정보를 기록하며 last 명령으로 확인할 수 있다.
/var/log/xferlog	ftp 서버의 데이터 전송 내역을 기록한다.
/var/secure	사용자들의 원격 접속 정보를 기록한다.

02 시스템 로그

■ 로그 파일 관리하기

- /var/log 디렉터리에 있는 README 파일의 내용을 확인해 보면 다음과 같음

```
[user1@localhost ~]$ cat /var/log/README
You are looking for the traditional text log files in /var/log, and they are
gone?

Here's an explanation on what's going on:

You are running a systemd-based OS where traditional syslog has been replaced
with the Journal. The journal stores the same (and more) information as classic
syslog. To make use of the journal and access the collected log data simply
invoke "journalctl", which will output the logs in the identical text-based
format the syslog files in /var/log used to be. For further details, please
refer to journalctl(1).
(생략)
```

- README 파일의 내용을 간단히 요약하면, 전통적인 로그 관리 방법인 syslog가 journal 기능으로 대체되었다는 것

02 시스템 로그

■ 로그 파일 관리하기

- journal은 기존 syslog 형식에 따라 로그를 저장하고, 저장된 로그에 접근하기 위해 journalctl 명령을 사용. 이 명령은 기존 /var/log 디렉터리에 있는 파일과 같은 텍스트 기반 형식으로 로그 내용을 출력
- 리눅스 시스템의 기존 로그 파일 중 대표라고 할 수 있는 /var/log/messages 파일은 rsyslog가 동작해야 생성되는데, journal에서 messages 파일을 대체하는 것이 바로 journalctl 명령

```
[user1@localhost ~]$ journalctl
10월 09 10:41:10 localhost kernel: Linux version 5.14.0-284.11.1.el9_2.x86_64 (mockbuild@i>
10월 09 10:41:10 localhost kernel: The list of certified hardware and cloud instances for >
10월 09 10:41:10 localhost kernel: Command line: BOOT_IMAGE=(hd0,msdos1)/vmlinuz-5.14.0-28>
10월 09 10:41:10 localhost kernel: Disabled fast string operations
(생략)
```

02 시스템 로그

rsyslog 데몬

- 리눅스 시스템의 로그 파일 중 일부 파일은 rsyslog라는 로그 관리 데몬에 의해 통제
- 로키 리눅스에는 rsyslog가 기본으로 설치되어 있음

```
[user1@localhost ~]$ rpm -qa | grep rsyslog
rsyslog-logrotate-8.2102.0-113.el9_2.x86_64
rsyslog-8.2102.0-113.el9_2.x86_64
rsyslog-relp-8.2102.0-113.el9_2.x86_64
rsyslog-gssapi-8.2102.0-113.el9_2.x86_64
rsyslog-gnutls-8.2102.0-113.el9_2.x86_64
[user1@localhost ~]$ systemctl status rsyslog
● rsyslog.service - System Logging Service
     Loaded: loaded (/usr/lib/systemd/system/rsyslog.service; enabled; preset: enabled)
     Active: active (running) since Mon 2023-10-09 10:41:19 KST; 18min ago
       Docs: man:rsyslogd(8)
             https://www.rsyslog.com/doc/
   Main PID: 985 (rsyslogd)
(생략)
```

02 시스템 로그

rsyslog 데몬

- rsyslog가 동작하고 있으므로 /var/log 디렉터리에 기본 로그 파일인 messages 파일이 생성되어 있음

```
[user1@localhost ~]$ ls /var/log/mess*
/var/log/messages           /var/log/messages-20230924  /var/log/messages-20231008
/var/log/messages-20230921  /var/log/messages-20231001
```

- ournal과 rsyslog는 시스템에 공존할 수 있으며 journal 데몬이 기본 도구 역할을 함

02 시스템 로그

rsyslog 서비스 설정 파일

- rsyslog 서비스를 설정하는 파일은 /etc/rsyslog.conf
  이 파일에는 어떤 로그를 어떻게 처리할 것인지를 규칙으로 정의해 놓았음
- 텍스트 파일이므로 관리자가 vi로 수정할 수 있음
- 규칙은 로그 메시지 중 어떤 메시지를 선택할 것인지를 정의한 필터(선택자라고도 함) 부분과 선택된 메시지를 어떻게 처리할 것인지를 정의한 동작 부분으로 구성
- 칙은 한 행에 필터와 동작으로 작성하고 공백문자나 탭으로 구분

02 시스템 로그

rsyslog 서비스 설정 파일

- /etc/rsyslog.conf 파일의 설정 내용 중 규칙(Rules) 부분을 보면 다음과 같음

```
[user1@localhost ~]$ cat /etc/rsyslog.conf
# rsyslog configuration file

(생략)
#### RULES ####

# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                                 /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none                /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                              /var/log/secure
(생략)
```

02 시스템 로그

■ 필터

- rsyslog의 필터는 기능명과 우선순위로 구성되며 다음과 같은 형식으로 작성

```
기능명.우선순위
```

02 시스템 로그

■ 기능명

- 기능명은 로그 메시지를 생성하는 프로그램을 지정한 것
- 기능명은 [표 15-2]의 키워드 중에서 하나를 사용함

표 15-2 rsyslog 필터의 기능명

기능명	코드	관련 프로그램
*	–	모든 기능
mark	–	rsyslog 내부용
kern	0	시스템 커널
user	1	사용자 프로세스
mail	2	sendmail과 기타 메일 관련 프로그램
daemon	3	일반적인 시스템 데몬
auth	4	인증 관련 명령
syslog	5	rsyslog 데몬 내부 메시지
lpr	6	인쇄 시스템
news	7	유즈넷 뉴스 시스템
uucp	8	uucp 통신(현재는 사용하지 않음)
cron	9	cron 데몬
authpriv	10	좀 더 민감한 보안 메시지
ftp	11	ftp 데몬
local0~7	16~23	여덟 가지 로컬 메시지

02 시스템 로그

■ 우선순위

- 메시지의 심각도를 나타내는 우선순위는 [표 15-3]의 키워드를 사용
- 심각도는 emerg부터 debug까지 8단계로 구분하지만 각 단계를 명확하게 구분하기 어려울 수도 있음

표 15-3 rsyslog 메시지의 우선순위

우선순위	의미	우선순위	의미
emerg	매우 긴급한 비상 상태	warning	경고 메시지
alert	긴급한 상태	notice	단순 메시지
crit	중대한 상태	info	정보성 메시지
err	오류 상태	debug	디버깅용 메시지

02 시스템 로그

■ 필터 구성

- 기능명과 우선순위를 결합하는 방법을 [표 15-4]의 예를 통해 살펴보자

표 15-4 rsyslog 필터 구성의 예

필터 구성	의미
kern.*	우선순위에 상관없이 커널의 모든 로그 메시지를 선택한다.
mail.crit	메일에서 crit 이상 우선순위(crit, alert, emerg)의 모든 로그 메시지를 선택한다.
cron.!info,!debug	cron에서 info와 debug를 제외한 모든 로그 메시지를 선택한다.
mail.=info	메일에서 심각도가 info인 경우만 로그 메시지를 선택한다.

02 시스템 로그

■ 동작

- 동작은 필터가 선택한 메시지를 어떻게 처리할지를 정의한 것
- 동작에는 메시지를 파일로 저장하기, 메일로 전송하기, 화면으로 출력하기 등이 있음

표 15-5 rsyslog 동작의 종류

필터와 동작	의미
. @192.168.1.50	메시지를 192.168.1.50의 rsyslog 데몬으로 보낸다.
. @@abc.com:18	메시지를 abc.com의 18번 포트로 TCP를 통해 보낸다.
. 파일명	메시지를 지정한 파일에 저장한다.
. user1, user2	메시지를 user1, user2 사용자의 화면에 출력한다.
. *	메시지를 현재 로그인한 모든 사용자에게 보낸다.
cron.* ~	cron이 발생시킨 모든 메시지를 무시한다.
kern.* ^exe;form	커널이 발생시킨 메시지를 form에 따라 형식을 조정하여 exe 프로그램에 전달하고 exe 프로그램을 실행한다.

02 시스템 로그

■ 동작

- 예를 들어 커널이 발생시킨 메시지 중 우선순위가 crit 이상인 메시지를 /var/log/kern.log 파일에 저장하려면 다음과 같이 설정함

```
kern.crit /var/log/kern.log
```

- 위의 메시지를 파일에도 저장하고 user1 사용자의 화면에도 출력하려면 다음과 같이 각각 두 줄로 작성해야 함

```
kern.crit /var/log/kern.log
kern.crit user1
```

02 시스템 로그

journal 기능

- journal은 systemd 데몬의 구성 요소로 로그 파일의 관리를 담당하며, 전통적으로 로그를 관리해 온 rsyslog 데몬과 병행하여 사용할 수 있음
- 로깅 데이터는 journald 데몬이 수집· 가공하여 journals라고 불리는 바이너리 파일로 저장.
  이 파일에는 커널이나 사용자 프로세스의 메시지, 시스템 서비스의 표준 출력과 표준 오류 등이 저장되며 사용자가 편집할 수 없음

```
[user1@localhost ~]$ systemctl status systemd-journald
● systemd-journald.service - Journal Service
     Loaded: loaded (/usr/lib/systemd/system/systemd-journald.service; static)
     Active: active (running) since Mon 2023-10-09 10:41:16 KST; 34min ago
TriggeredBy: ● systemd-journald.socket
             ● systemd-journald-dev-log.socket
       Docs: man:systemd-journald.service(8)
             man:journald.conf(5)
   Main PID: 785 (systemd-journal)
```

- journald 데몬의 실행 파일 이름은 systemd-journald

02 시스템 로그

journal 기능

- journal이 저장한 로그를 보려면 journalctl 명령을 사용

journalctl

- 기능 journal 로그를 관리한다.
- 형식 journalctl [옵션]
- 옵션 -n 행수: 가장 최근에 기록된 로그 중 행수만큼 출력한다.
  -r: 가장 최근 로그가 먼저 출력된다.
  -o short¦verbose: 지정한 형식으로 출력한다.
    short: syslog 형식으로 출력한다.
    verbose: 로그의 상세한 내용까지 출력한다.
  -f: 최근 로그를 자동으로 출력한다.
  -p 우선순위: 우선순위로 필터링하여 출력한다.
  -b 시간: 현재 부팅 이후의 로그만 출력한다.
  --since=시간 --until=시간: 시간을 필터링하여 출력한다.
  필드명=값: 필드명으로 필터링하여 출력한다.
- 사용 예 journalctl
    journalctl o verbose

02 시스템 로그

journal 기능

- root 권한이 없는 일반 사용자는 journalctl 명령으로 해당 계정이 생성한 로그 내용만 볼 수 있음
- 만약 일반 사용자가 adm 그룹에 속해 있다면 전체 로그를 볼 수 있음
- 현재 user1 계정은 adm 그룹에 속해 있음

02 시스템 로그

■ 로그 파일의 전체 내용 보기

- 옵션 없이 journalctl 명령을 실행하면 로그 파일의 전체 내용이 출력

```
[user1@localhost ~]$ journalctl
10월 09 10:41:10 localhost kernel: Linux version 5.14.0-284.11.1.el9_2.x86_64 (mockbuild@i>
10월 09 10:41:10 localhost kernel: The list of certified hardware and cloud instances for >
10월 09 10:41:10 localhost kernel: Command line: BOOT_IMAGE=(hd0,msdos1)/vmlinuz-5.14.0-28>
10월 09 10:41:10 localhost kernel: Disabled fast string operations
10월 09 10:41:10 localhost kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating >
10월 09 10:41:10 localhost kernel: x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
10월 09 10:41:10 localhost kernel: x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers '
(생략)
```

02 시스템 로그

■ 가장 최근의 로그 내용 보기: -n 옵션

- 로그 파일에 기록된 내용 중 가장 최근의 로그를 확인하려면 -n 옵션으로 보려는 행수를 지정
- 예를 들어 가장 최근의 로그를 세 행만 보려면 다음과 같이 지정

```
[user1@localhost ~]$ journalctl -n 3
10월 09 13:15:42 localhost.localdomain systemd[1]: Finished dnf makecache.
10월 09 13:28:46 localhost.localdomain NetworkManager[1161]: <info>  [1696825726.2806]
dhc>
10월 09 13:36:43 localhost.localdomain cupsd[1168]: REQUEST localhost - - "POST /
HTTP/1.1>
lines 1-3/3 (END)
```

02 시스템 로그

■ 로그의 상세한 내용 보기: -o 옵션

- 로그의 전체 항목을 자세하게 보려면 -o 옵션을 사용
- 예를 들어 가장 마지막 로그를 자세히 보려면 다음과 같이 지정
  로그 항목을 보면 UID, GID, 호스트 이름 등 상세한 정보가 모두 기록되어 있음

```
[user1@localhost ~]$ journalctl -n 1 -o verbose
Mon 2023-10-09 13:36:43.939942 KST [s=90e91ddaa1074f879b0a92a7b5773482;i=b85;b=fff7
6a58130]
    _BOOT_ID=fff76a58130e4f85a8b606488eb2714c
    _MACHINE_ID=b79f0cabfac84e2ab3527f85365b740d
    PRIORITY=6
    _UID=0
    _GID=0
    _SYSTEMD_SLICE=system.slice
    _TRANSPORT=journal
    _CAP_EFFECTIVE=1fffffffff
    _HOSTNAME=localhost.localdomain
(생략)
```

02 시스템 로그

■ 로그를 자동으로 출력하기: -f 옵션

- -f 옵션은 로그에 새로운 기록이 저장될 때마다 자동으로 출력. 이 옵션을 사용하면 로그를 출력한 후 프롬프트를 출력하지 않고 다음 로그가 저장되면 출력하기 위해 기다리고 있음
- 이 기능은 관리자가 로그 메시지를 실시간으로 확인하기 위해 주로 사용.
  출력을 종료하려면 Ctrl +c를 입력

```
[user1@localhost ~]$ journalctl -f
10월 09 13:13:46 localhost.localdomain NetworkManager[1161]: <info>  [1696824826.2821]
dhcp4 (ens160): state changed new lease, address=192.168.147.129
10월 09 13:15:38 localhost.localdomain systemd[1]: Starting dnf makecache...
10월 09 13:15:40 localhost.localdomain dnf[3717]: Rocky Linux 9 - BaseOS    4.11 kB/s
¦ 4.1 kB     00:00
10월 09 13:15:41 localhost.localdomain dnf[3717]: Rocky Linux 9 - AppStream 6.1 kB/s
¦ 4.5 kB     00:00
(생략)
      → 새로운 로그를 기다리고 있다.
```

02 시스템 로그

■ 우선순위로 필터링하여 출력하기: -p 옵션

- [표 15-3]의 우선순위로 필터링하여 로그를 출력하려면 -p 옵션을 사용
- 예를 들어 오류 상태(err) 로그만 출력하려면 다음과 같이 작성
  실제 터미널에서 보면 오류 메시지가 빨간색으로 출력됨

```
[user1@localhost ~]$ journalctl -p err
10월 09 10:41:17 localhost.localdomain kernel: piix4_smbus 0000:00:07.3: SMBus Host
Contro>
10월 09 10:41:18 localhost.localdomain kernel: Bluetooth: hci0: unexpected cc 0x0c12
lengt>
10월 09 10:41:18 localhost.localdomain kernel: Bluetooth: hci0: Opcode 0x c12 failed:
-38
10월 09 10:41:19 localhost.localdomain systemd[1]: Failed to start Enable File System
Quot>
10월 09 10:41:19 localhost.localdomain alsactl[1026]: alsa-lib main.c:1559:(snd_use_
case_m>
(생략)
lines 1-8/8 (END)
```

02 시스템 로그

■ 시간으로 필터링하여 출력하기: -b, --since --until

- 현재 부팅 이후의 로그만 출력하려면 -b 옵션을 사용하고, 특정 시간 이후나 시간의 범위를 정하려면 --since=시간 [--until=시간] 옵션을 사용
- 예를 들어 2023년 10월 09일 12:00 이후의 로그만 출력하려면 다음과 같이 작성

```
[user1@localhost ~]$ journalctl --since="2023-10-09 12:00"
10월 09 12:01:01 localhost.localdomain CROND[3658]: (root) CMD (run-parts /etc/cron.
hourly)
10월 09 12:01:01 localhost.localdomain run-parts[3661]: (/etc/cron.hourly) starting
0anacr>
10월 09 12:01:01 localhost.localdomain run-parts[3667]: (/etc/cron.hourly) finished
0anacr>
10월 09 12:01:01 localhost.localdomain CROND[3657]: (root) CMDEND (run-parts /etc/cron.
hou>
10월 09 12:13:46 localhost.localdomain NetworkManager[1161]: <info>  [1696821226.2788]
dhc>
10월 09 12:18:02 localhost.localdomain anacron[3568]: Job `cron.monthly' started
(생략)
```

02 시스템 로그

■ 시간으로 필터링하여 출력하기: -b, --since --until

- 시간의 범위를 지정하려면 --until 옵션을 같이 사용

```
[user1@localhost ~]$ journalctl --since="2023-10-09 12:00" --until="2023-10-09 12:05"
10월 09 12:01:01 localhost.localdomain CROND[3658]: (root) CMD (run-parts /etc/cron.
hourly)
10월 09 12:01:01 localhost.localdomain run-parts[3661]: (/etc/cron.hourly) starting
0anacr>
10월 09 12:01:01 localhost.localdomain run-parts[3667]: (/etc/cron.hourly) finished
0anacr>
10월 09 12:01:01 localhost.localdomain CROND[3657]: (root) CMDEND (run-parts /etc/cron.
hou>
lines 1-4/4 (END)
```

02 시스템 로그

■ 필드명으로 필터링하여 출력하기: -F

- 로그 기록 중 특정 필드의 값만 확인하고 싶을 때는 -F 옵션을 사용
- 필드명은 앞의 예에서 journalctl -n 1 -o verbose로 자세하게 출력한 로그 기록을 참조
- 예를 들어 로그 메시지를 생성한 systemd 서비스가 무엇인지 확인하려면 다음과 같이 입력

```
[user1@localhost ~]$ journalctl -F _SYSTEMD_UNIT
dnf-makecache.service
session-4.scope
geoclue.service
fwupd.service
(생략)
```

02 시스템 로그

■ 필드명으로 필터링하여 출력하기: -F

- 필드명의 특정 값을 지정하면 좀 더 구체적으로 로그 기록을 필터링할 수 있음
- 예를 들어 UID가 0(root)인 사용자가 crond.service에서 발생한 로그 기록을 필터링하여 확인하려면 다음과 같이 입력

```
[user1@localhost ~]$ journalctl _UID=0 _SYSTEMD_UNIT=crond.service
10월 09 10:41:20 localhost.localdomain crond[1216]: (CRON) STARTUP (1.5.7)
10월 09 10:41:20 localhost.localdomain crond[1216]: (CRON) INFO (RANDOM_DELAY will be
scal>
10월 09 10:41:20 localhost.localdomain crond[1216]: (CRON) INFO (running with inotify
supp>
10월 09 11:01:01 localhost.localdomain CROND[3555]: (root) CMD (run-parts /etc/cron.
hourly)
10월 09 11:01:02 localhost.localdomain anacron[3568]: Anacron started on 2023-10-09
(생략)
```

02 시스템 로그

GUI 기반 로그 관리 도구

- 그놈은 로그를 관리할 수 있는 GUI 도구인 로그뷰어를 제공
- 그놈 로그뷰어의 패키지는 pkgs.org 사이트에서 검색하여 설치하면 됨
- 리눅스 콘솔에서는 처음 로그인한 계정에서 gnome- system- log를 실행
- 만약 root 계정으로 로그인하지 않았다면 로그뷰어를 실행하기 위해 사용자의 암호를 물어봄

03 방화벽 관리

03 방화벽 관리

로그

- 로그는 시스템의 상태를 알려주는 메시지
- 로그를 보고 시스템에 이상이 있는지 확인할 수 있지만, 이미 어떤 상황이 발생한 이후에 남은 기록을 보는 것
- 즉, 로그로는 사전에 외부의 공격을 차단할 수 없음
- 네트워크를 통한 외부의 접속을 차단하려면 방화벽을 사용해야 함

03 방화벽 관리

■ 방화벽 동작 확인

- 방화벽이 동작 중인지 확인해 보자. 방화벽의 서비스 이름은 firewalld.service
- 다음 예를 보면 방화벽 서비스가 동작(active) 중임을 알 수 있음
- 만약 방화벽이 설치되어 있지 않다면 설치해야 하고, 설치는 되어 있지만 동작하지 않는다면 동작시켜야 함

```
[user1@localhost ~]$ systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
     Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset:
enabled)
     Active: active (running) since Mon 2023-10-09 10:41:20 KST; 3h 21min ago
       Docs: man:firewalld(1)
   Main PID: 1036 (firewalld)
(생략)
```

03 방화벽 관리

방화벽 동작 확인

- 방화벽을 시작하거나 종료하려면 다음 명령을 사용

```
systemctl stop firewalld
```

```
systemctl start firewalld
```

03 방화벽 관리

GUI 도구로 방화벽 설정

- firewall 데몬은 동적으로 방화벽을 관리할 수 있게 하며 IPv4와 IPv6를 모두 지원
- 방화벽을 동적으로 관리한다는 것은 언제든지 방화벽의 설정을 변경할 수 있고, 변경된 설정이 즉시 적용된다는 의미
- 방화벽의 변경 내용을 실행하기 위해 별도로 변경 내용을 저장하고 적용하는 과정이 필요 없음
- 이는 방화벽을 다시 실행하기 위해 기존 네트워크 연결을 끊을 필요가 없다는 뜻

03 방화벽 관리

GUI 도구로 방화벽 설정

- 방화벽을 관리하는 GUI 도구인 firewall-config를 dnf 명령으로 설치

```
[root@localhost ~]# dnf install firewall-config
(생략)설치되었습니다:
  dbus-x11-1:1.12.20-7.el9_2.1.x86_64          firewall-config-1.2.1-1.el9.noarch

완료되었습니다!
```

- 리눅스 콘솔에 root로 로그인한 것이 아니므로 로그인한 계정에서 firewall-config를 실행 하면 root 계정의 암호를 물어봄
- 암호를 입력하면 [그림 15-2]와 같은 창이 뜸

```
[user1@localhost ~]$ firewall-config &
```

03 방화벽 관리

GUI 도구로 방화벽 설정

그림 15-2 방화벽 설정 창

- firewall-config의 사용법을 간단하게 살펴보자. 방화벽에서 서비스를 열거나 닫으면 현재 시스템을 사용 중인 다른 사용자들에게도 영향을 미치므로 주의해야 함

03 방화벽 관리

설정

- 현재 설정하는 내용을 바로 적용할 것인지를 결정
- 설정값이 '런타임'이면 서비스나 포트 등의 항목을 변경했을 때 즉시 적용되지만 시스템을 다시 시작하면 초기화됨
- 예를 들어 서비스의 telnet 체크박스에 체크 표시를 하면 바로 적용되지만 시스템을 다시 시작하면 telnet 서비스가 허용되지 않는 것
- 설정값이 '영구적'이면 설정한 내용이 시스템을 다시 시작해도 그대로 적용됨

03 방화벽 관리

■ 영역

- 방화벽은 신뢰도 수준에 따라 네트워크를 여러 개의 영역으로 구분하여 사용할 수 있음
  영역은 [표 15-6]과 같이 구분되며 이 영역 중 하나를 기본 영역으로 지정할 수 있음.
  현재의 기본 영역은 public. 영역은 시스템 관리자가 추가·삭제·편집할 수 있음

표 15-6 영역의 종류

영역	내용
block	모든 네트워크 접속 요청이 거부된다.
dmz	dmz로 구분된 영역에 있는 컴퓨터만 공개적으로 접근할 수 있다.
drop	모든 접속 요청이 거부되고 응답도 하지 않는다. 내부에서 외부로 접속하는 것만 가능하다.
external	외부 네트워크를 위해 사용하는 영역으로 선택된 서비스만 접속이 허용된다.
home	홈 영역으로, 대체로 이 영역에 있는 다른 컴퓨터를 믿을 수 있다. 선택된 서비스만 접속이 허용된다.
internal	내부 네트워크를 위한 것으로, 대체로 이 영역에 있는 다른 컴퓨터를 믿을 수 있다. 선택된 서비스만 접속이 허용된다.
nm-shared	이 영역은 연결을 공유할 때 NetworkManager가 사용하는 부분이다.
public	공개 영역으로 선택된 서비스만 접속이 허용된다.
trusted	이 영역에 있는 컴퓨터의 모든 네트워크 연결을 허용한다.
work	작업 영역으로, 대체로 이 영역에 있는 다른 컴퓨터를 믿을 수 있다. 선택된 서비스만 접속이 허용된다.

03 방화벽 관리

■ 방화벽 설정하기

- 방화벽에서는 서비스, 포트, 프로토콜, 마스커레이딩, 포트 포워딩, ICMP 필터를 설정할 수 있음
  - **서비스**: http, ssh, ftp 같은 이미 잘 알려진 서비스 중 신뢰할 수 있는 서비스를 열거나 닫는 것. 신뢰할 수 있는 서비스의 체크박스에 체크 표시를 하면 됨
  - **포트, 소스 포트:** 서비스 목록에 없는 네트워크 서비스를 열어주기 위해 해당 서비스의 포트 번호를 직접 지정할 때 사용. 포트를 추가할 때는 [그림 15-3]과 같이 포트 번호와 프로토콜(tcp, udp)을 선택하면 됨

그림 15-3 포트 추가 창

03 방화벽 관리

■ 방화벽 설정하기

– **프로토콜**: 서비스 목록에 없는 네트워크 서비스를 열어주기 위해 해당 프로토콜을 직접 지정할 때 사용. 프로토콜을 추가할 때는 [그림 15-4]와 같이 기존 프로토콜 이름을 선택하거나 직접 입력

그림 15-4 프로토콜 추가 창

03 방화벽 관리

■ 방화벽 설정하기

- **마스커레이딩**: 내부의 IP 주소를 하나의 외부용 주소로 변환하는 것으로, 리눅스 시스템을 라우터 기능으로 사용할 때 적용
- **포트 포워딩**: 외부에서 들어오는 패킷을 내부 주소의 특정 포트로 전달하는 것으로, 마스커레이딩과 함께 사용
- **ICMP 필터**: ICMP 프로토콜로 보내는 메시지에서 거부할 유형을 표시하는 것
  예를 들어 'echo reply'를 거부한다면 해당 시스템은 ping 명령에 응답하지 않게 됨

03 방화벽 관리

■ 방화벽 관리 명령

- 방화벽은 명령으로도 관리할 수 있음

firewall-cmd

- 기능 firewalld를 설정한다.
- 형식 firewall-cmd [옵션]
- 옵션 -v: firewalld의 버전을 출력한다.
  -h: 도움말을 출력한다.
  --reload: 방화벽의 규칙을 다시 읽어온다.
  --get-default-zone: 기본 영역을 출력한다.
  --set-default-zone=영역 이름: 기본 영역을 설정한다.
  --add-ACTION: ACTION을 추가한다.
  --remove-ACTION: ACTION을 제거한다.
  ACTION: service=서비스명, port=포트 번호/프로토콜(tcp/udp)
  --list-services: 허용된 서비스 목록을 출력한다.
  --list-ports: 허용된 포트 목록을 출력한다.
  --permanent: 방화벽에 저장한다. reload나 재시작을 해야 적용된다.
- 사용 예 firewall-cmd --add-service=telnet
  firewall-cmd --remove-service=telnet
  firewall-cmd --add-port=5500/tcp
  firewall-cmd --list-services

03 방화벽 관리

■ 허용하는 서비스 목록 보기

- 현재 방화벽에서 허용하는 서비스 목록을 확인하는 옵션은 --list-services
- 다음 예에서는 허용하는 서비스가 cockpit, dhcpv6-client, ssh인데 이는 시스템에 따라 다를 수 있음

```
[root@localhost ~]# firewall-cmd --list-services
cockpit dhcpv6-client ssh
```

■ 서비스 추가하기

- telent 서비스를 허용하도록 방화벽에 추가해 보자. 서비스를 추가한 뒤 다시 허용 목록을 확인.
  다음 예를 보면 telnet이 추가되었음을 알 수 있음

```
[root@localhost ~]# firewall-cmd --add-service=telnet
success
[root@localhost ~]# firewall-cmd --list-services
cockpit dhcpv6-client ssh telnet
```

03 방화벽 관리

■ 서비스 삭제하기

- 방화벽에서 telnet 서비스를 삭제해 보자
- 서비스를 삭제한 뒤 역시 허용 목록을 확인해 보면 telnet이 삭제되었음
- 서비스를 삭제할 때는 그 서비스를 사용하는 사용자가 있는지 확인한 뒤에 삭제해야 함
- 갑자기 서비스 연결이 끊어지면 사용자가 작업하던 내용을 잃어버릴 수도 있으므로 주의해야 함

```
[root@localhost ~]# firewall-cmd --remove-service=telnet
success
[root@localhost ~]# firewall-cmd --list-services
cockpit dhcpv6-client ssh
```

03 방화벽 관리

■ 포트 추가하기

- 방화벽에 포트를 추가해 보자
- 포트를 추가할 때는 tcp나 udp 프로토콜을 지정해야 함
- 어떤 프로토콜인지 확실하게 모른다면 둘 다 추가하며, 포트를 추가한 뒤에 목록에서 확인
- 다음 예에서는 임의로 5000번 포트를 추가했음

```
[root@localhost ~]# firewall-cmd --add-port=5000/tcp
success
[root@localhost ~]# firewall-cmd --add-port=5000/udp
success
[root@localhost ~]# firewall-cmd --list-ports
5000/tcp 5000/udp
```

03 방화벽 관리

■ 포트 삭제하기

- 방화벽에 추가한 포트를 삭제해 보자. 다음 예에서는 앞서 추가한 5000번 포트를 삭제

```
[root@localhost ~]# firewall-cmd --remove-port=5000/tcp
success
[root@localhost ~]# firewall-cmd --remove-port=5000/udp
success
[root@localhost ~]# firewall-cmd --list-ports
```

03 방화벽 관리

■ 설정 내용 저장하기

- --permanent 옵션은 바로 적용되지 않지만, 방화벽에 설정 내용을 저장
- 이 옵션으로 지정한 항목을 적용하려면 reload하거나 firewalld를 다시 시작해야 함
- 바로 적용하기와 방화벽에 저장하기를 동시에 하려면 런타임 적용과 permanent 옵션 적용을 모두 실행해야 함. 이때 --permanent 옵션은 제일 먼저 나와야 함
- 예를 들어 방화벽에서 telnet 서비스 허용을 런타임으로 바로 적용하기와 저장하기를 함께 실행하려면 다음과 같이 입력

```
[root@localhost ~]# firewall-cmd --add-service=telnet
success
[root@localhost ~]# firewall-cmd --permanent --add-service=telnet
success
```

04 보안 관리 도구

04 보안 관리 도구

NMap

- NMap은 내 서버나 원격의 서버가 사용 중인 포트, 운영체제 등을 스캔하여 출력
- NMap은 네트워크 관리용으로 사용되고, 취약한 포트의 사용 여부를 확인할 수 있으므로 보안용으로도 사용
- 그러나 스캔하는 것만으로도 보안 침입을 위한 준비 과정으로 간주하므로 원격 서버를 마구 스캔하면 안 됨

04 보안 관리 도구

NMap 설치하기

- NMap은 리눅스에 기본적으로 설치된 명령이 아니므로 관리자가 추가로 설치해야 사용할 수 있음
- dnf 명령으로 NMap을 설치해 보자. 설치할 패키지는 nmap

```
[root@localhost ~]# dnf install nmap
(생략)
설치되었습니다:
  nmap-3:7.91-12.el9.x86_64

완료되었습니다!
```

04 보안 관리 도구

■ NMap 설치하기

nmap

- 기능 네트워크를 탐색하고 보안을 점검한다.
- 형식 nmap [옵션] [목적지 주소]
- 옵션 -sS: TCP SYN을 스캔한다.
  -sT: TCP 연결을 스캔한다.
  -sP: ping을 스캔한다.
  -sU: UDP를 스캔한다.
  -sO: IP 프로토콜을 스캔한다.
  -O: 운영체제를 확인한다.
  -v: 스캔 결과를 상세하게 출력한다.
  -p 포트 번호: 지정한 포트만 스캔한다.
  (예 -p22; -p1-65535; -p U:53,111,T:21-25,80)
  -F: 빠른 모드(Fast mode)로 기본 스캔보다 적은 수의 포트만 스캔한다.
- 사용 예 nmap 192.168.0.1
  nmap -O 192.168.0.1
  nmap -sT -O -v 192.168.0.1

04 보안 관리 도구

■ 옵션 없이 nmap 실행하기

- 아무 옵션 없이 nmap을 실행하면 지정한 호스트에서 현재 열려 있는 포트를 요약하여 출력
- localhost를 스캔한 예. 22, 111, 631번 포트가 열려 있음을 알 수 있음

```
[root@localhost ~]# nmap localhost
Starting Nmap 7.91 ( https://nmap.org ) at 2023-10-09 14:23 KST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000070s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 997 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
111/tcp open  rpcbind
631/tcp open  ipp

Nmap done: 1 IP address (1 host up) scanned in 0.56 seconds
```

04 보안 관리 도구

■ 특정 서버 스캔하기

- IP 주소를 사용하여 특정 서버를 지정하고 -O 옵션도 지정해 보자
- -O 옵션은 해당 시스템의 운영체제 정보를 알려줌
- 윈도가 설치된 PC를 스캔하면 다음과 같이 출력됨

```
[root@localhost ~]# nmap -O 192.168.0.7
Starting Nmap 7.91 ( https://nmap.org ) at 2023-10-09 14:26 KST
Nmap scan report for 192.168.0.7
Host is up (0.00090s latency).
Not shown: 994 filtered ports
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
912/tcp  open  apex-mesh
1025/tcp open  NFS-or-IIS
5357/tcp open  wsdapi
Warning: OSScan results may be unreliable because we could not find at least 1 open and
1 closed port
Device type: general purpose
Running: Microsoft Windows XP|7|2012
OS CPE: cpe:/o:microsoft:windows_xp::sp3 cpe:/o:microsoft:windows_7 cpe:/o:microsoft:windows_
server_2012
OS details: Microsoft Windows XP SP3, Microsoft Windows XP SP3 or Windows 7 or Windows
Server 2012

OS detection performed. Please report any incorrect results at https://nmap.org/submit/
.
Nmap done: 1 IP address (1 host up) scanned in 11.03 seconds
```

04 보안 관리 도구

■ 특정 서버 스캔하기

- 이 스캔 결과로 알 수 있는 정보는 다음과 같음
  - 이 시스템에서 열려 있는 포트는 135/tcp 등 총 여섯 개다.
  - 이 시스템의 운영체제는 마이크로소프트 윈도다.
  - 이 시스템을 스캔하는 데 11.03초가 걸렸다.

04 보안 관리 도구

UDP 포트 스캔하기

- UDP 포트가 열려 있는지 확인하려면 -sU 옵션을 사용
- 다음 예에서는 localhost의 UDP 포트를 스캔함

```
[root@localhost ~]# nmap -sU -v localhost
Starting Nmap 7.91 ( https://nmap.org ) at 2023-10-09 14:28 KST
Initiating UDP Scan at 14:28
(생략)
PORT     STATE          SERVICE
111/udp  open|filtered  rpcbind
5353/udp open|filtered  zeroconf
(생략)
```

- 앞의 실행 결과를 보면 localhost에는 UDP 포트 111, 5353번이 열려 있음을 알 수 있음

04 보안 관리 도구

■ 특정 네트워크를 대상으로 포트 스캔하기

- 네트워크 주소를 지정하면 특정 네트워크 전체를 스캔할 수 있음
- 이 경우 네트워크에 연결된 전체 시스템의 포트 상태를 일괄적으로 점검할 수 있음
- 다음은 192.168.0.0 네트워크를 스캔한 예
- /24는 24비트로 192.168.0.까지가 네트워크 주소
- 네트워크 전체를 스캔하므로 실행하는 데 시간이 걸림

```
[root@localhost ~]# nmap -sT -O  -v 192.168.0.0/24
[root@localhost ~]# nmap -sT -O -v 192.168.0.0/24
Starting Nmap 7.91 ( https://nmap.org ) at 2023-10-09 14:29 KST
Initiating Ping Scan at 14:29
Scanning 256 hosts [4 ports/host]
Completed Ping Scan at 14:30, 2.64s elapsed (256 total hosts)
Initiating Parallel DNS resolution of 256 hosts. at 14:30
Completed Parallel DNS resolution of 256 hosts. at 14:30, 11.17s elapsed
Initiating Connect Scan at 14:30
Scanning 64 hosts [1000 ports/host]
Discovered open port 135/tcp on 192.168.0.7
Discovered open port 445/tcp on 192.168.0.7
Discovered open port 80/tcp on 192.168.0.1
Discovered open port 139/tcp on 192.168.0.7
Discovered open port 1025/tcp on 192.168.0.7

Connect Scan Timing: About 9.38% done; ETC: 14:35 (0:04:59 remaining)
Connect Scan Timing: About 12.45% done; ETC: 14:38 (0:07:09 remaining)
Connect Scan Timing: About 15.57% done; ETC: 14:39 (0:08:14 remaining)
Discovered open port 5357/tcp on 192.168.0.7
Connect Scan Timing: About 22.98% done; ETC: 14:38 (0:06:46 remaining)
Increasing send delay for 192.168.0.1 from 0 to 5 due to 11 out of 34 dropped probes
since last increase.
Increasing send delay for 192.168.0.7 from 0 to 5 due to 13 out of 42 dropped probes
since last increase.
(생략)
```

04 보안 관리 도구

PAM

- 사용자가 시스템의 서비스를 사용하려면 사용자의 신원을 확인하는 인증을 받아야 함.
  로키 리눅스에서는 많은 프로그램이 중앙 집중화 된 인증 기법을 사용하는데 이 기법이 바로 PAM
- PAM은 'pluggable authentication modules'의 약자로 우리말로는 삽입형 인증 모듈 이라고 번역할 수 있음
- PAM은 모듈 방식으로 구성되어 있어 시스템 관리자가 필요에 따라 인증 모듈을 추가·삭제·편집할 수 있음

04 보안 관리 도구

PAM 설정 파일

- PAM 모듈의 설정 파일은 /etc/pam.d 디렉터리에 있음. 설정 파일은 서비스별로 존재함
- 현재 /etc/pam.d 디렉터리의 내용은 다음과 같음.
  sshd, samba, crond, vsftpd 등 그동안 다룬 서비스에 대한 설정 파일도 있음을 알 수 있음
- 패키지가 설치되면서 서비스 이름과 같은 이름으로 PAM 설정 파일을 /etc/pam.d 디렉터리에 생성한 것. 각 설정 파일에는 해당 서비스를 이용하기 위해 어떻게 인증할 것인지를 설정해 놓았음

```
[root@localhost ~]# ls /etc/pam.d
atd                 gdm-autologin           passwd          smartcard-auth   sudo-i
chfn                gdm-fingerprint         password-auth   smtp             system-auth
chsh                gdm-launch-environment  polkit-1        smtp.sendmail    systemd-user
cockpit             gdm-password            postlogin       sshd             vlock
config-util         gdm-pin                 remote          sssd-shadowutils vmtoolsd
crond               gdm-smartcard           runuser         su               vsftpd
cups                login                   runuser-l       su-l             xserver
fingerprint-auth    other                   samba           sudo
```

04 보안 관리 도구

PAM 설정 파일 형식

- PAM 설정 파일은 다음과 같은 형식으로 구성됨

```
<모듈 인터페이스> <제어 플래그> <모듈 이름> <모듈 인자>
```

04 보안 관리 도구

■ 모듈 인터페이스

- 현재 모듈 인터페이스로 다음 네 가지를 쓸 수 있음. 각각은 인증 과정의 각 단계에 대응함
  - **auth**: 이 모듈은 사용자를 인증하는 데 사용된다.
    예를 들어 암호가 정확한지를 확인 한다. 또한 이 모듈은 그룹 지정에도 적용된다.
  - **account**: 이 모듈은 접근이 허용되는지를 확인한다.
    예를 들어 사용자 계정이 유효 한지 또는 사용자가 해당 날짜에 로그인할 수 있는지를 확인한다.
  - **password**: 이 모듈은 사용자 계정의 암호를 바꾸는 데 사용된다.
  - **session**: 이 모듈은 사용자의 세션을 설정하고 관리한다. 또한 사용자의 홈 디렉터리를 마운트하는 것과 같이 접근을 허용하는 데 필요한 부가적인 작업을 수행한다.
- 각 모듈은 이와 같은 인터페이스를 일부 또는 전부 제공함.
  예를 들어 sshd 모듈은 앞의 네 가지 모듈 인터페이스를 모두 제공함

04 보안 관리 도구

■ 제어 플래그

- 모든 PAM 모듈은 수행 후에 성공이나 실패를 생성. 제어 플래그는 특정 모듈의 성공과 실패를 어떻게 처리할 것인지를 알려줌. 제어 플래그는 다음 여섯 가지 중 하나를 설정
  - **required**: 해당 모듈은 인증을 계속하기 위해 반드시 성공해야 한다.
    만약 실패하면 사용자는 다른 모든 모듈의 테스트가 끝날 때까지 결과를 받지 못한다.
  - **requisite**: 해당 모듈은 인증을 계속하기 위해 반드시 성공해야 한다.
    만약 이 지점 에서 실패하면 사용자는 실패에 대한 메시지를 즉시 받는다.
  - **sufficient**: 실패하면 이 모듈의 결과가 무시된다.
    만약 이 모듈이 성공하고 앞선 required 모듈 중 실패가 없으면 인증 성공을 리턴한다.
  - **optional**: 이 모듈의 결과는 무시된다.
    이 모듈은 해당 인터페이스에 다른 모듈이 없는 경우에만 인증에 성공하는 데 필요하다.

04 보안 관리 도구

■ 제어 플래그

- **include**: 모듈 인자로 지정한 설정 파일에서 모든 행을 읽어온다.
- **substack**: 모듈 인자로 지정한 설정 파일에서 모든 행을 읽어오는데, 종료 동작에 대한 평가 때문에 모듈 스택의 나머지 동작을 무시하지 않는다.

04 보안 관리 도구

■ 모듈 이름

- 모듈 이름은 삽입 가능한 것을 지정함
- 예전 버전에서는 모듈의 절대 경로를 사용했으나 현재는 PAM 모듈이 /lib64/security 디렉터리에 위치하여 디렉터리 이름은 생략하고 바로 모듈 이름만 지정함

```
[root@localhost ~]# ls /lib64/security
pam_access.so         pam_gdm.so             pam_postgresok.so      pam_systemd.so
pam_cap.so            pam_gnome_keyring.so   pam_pwhistory.so       pam_time.so
pam_chroot.so         pam_group.so           pam_pwquality.so       pam_timestamp.so
pam_cockpit_cert.so   pam_issue.so           pam_rhosts.so          pam_tty_audit.so
pam_console.so        pam_keyinit.so         pam_rootok.so          pam_umask.so
pam_debug.so          pam_lastlog.so         pam_securetty.so       pam_unix.so
(생략)
```

04 보안 관리 도구

■ 모듈 인자

- 모듈 인자는 인증 과정에서 정보가 필요한 일부 모듈에 정보를 전달함
- 예를 들면 pam_ userdb.so 모듈은 사용자 정보가 저장된 버클리 DB에 대한 경로 정보가 필요함

04 보안 관리 도구

PAM 파일의 예

- PAM 파일에는 같은 모듈 인터페이스를 가진 모듈이 여러 개 설정될 수 있음.
  이 모듈들은 같은 목적을 위해 함께 사용되는 것
- PAM 파일의 예를 통해 그 구성을 살펴보자.
- 다음은 /etc/pam.d/vsftpd 파일의 내용임. 맨 앞의 숫자는 설명을 위해 붙인 행 번호

```
1 #%PAM-1.0
2 session    optional     pam_keyinit.so    force revoke
3 auth       required     pam_listfile.so item=user sense=deny file=/etc/vsftpd/
ftpusers onerr=succeed
4 auth       required     pam_shells.so
5 auth       include      password-auth
6 account    include      password-auth
7 session    required     pam_loginuid.so
8 session    include      password-auth
```

04 보안 관리 도구

■ PAM 파일의 예

```
1 #%PAM-1.0
2 session    optional     pam_keyinit.so    force revoke
3 auth       required     pam_listfile.so item=user sense=deny file=/etc/vsftpd/
ftpusers onerr=succeed
4 auth       required     pam_shells.so
5 auth       include      password-auth
6 account    include      password-auth
7 session    required     pam_loginuid.so
8 session    include      password-auth
```

① 1행은 주석. 주석은 #로 시작함

② 2행은 프로세스를 호출할 때 사용자의 기본 암호가 아닌 세션 암호를 사용

③ 3~5행은 로그인 인증을 위한 모듈

- **pam_listfile.so**: 이 모듈은 /etc/vsftpd/ftpusers 파일에 기반하여 서비스 이용을 허용하거나 거부. 이 파일에 사용자 이름이 있으면 ftp 서비스 이용을 거부함

04 보안 관리 도구

PAM 파일의 예

```
1 #%PAM-1.0
2 session    optional     pam_keyinit.so    force revoke
3 auth       required     pam_listfile.so item=user sense=deny file=/etc/vsftpd/
ftpusers onerr=succeed
4 auth       required     pam_shells.so
5 auth       include      password-auth
6 account    include      password-auth
7 session    required     pam_loginuid.so
8 session    include      password-auth
```

- **pam_shells.so**: 이 모듈은 사용자의 셸이 /etc/shells 파일에 있는 정당한 셸인지 확인
- **password-auth**: password-auth 파일에 설정된 내용을 읽어 옴.
  이 파일에는 암호와 관련하여 자세하게 설정되어 있음
- pam_listfile.so와 pam_shells.so 모듈은 required이므로 인증을 계속하기 위해 반드시 성공해야 함

04 보안 관리 도구

■ PAM 파일의 예

```
1 #%PAM-1.0
2 session    optional     pam_keyinit.so    force revoke
3 auth       required     pam_listfile.so item=user sense=deny file=/etc/vsftpd/
ftpusers onerr=succeed
4 auth       required     pam_shells.so
5 auth       include      password-auth
6 account    include      password-auth
7 session    required     pam_loginuid.so
8 session    include      password-auth
```

④ 6행은 계정의 유효성을 확인하는 것으로 password-auth 파일의 설정에 따라 확인

⑤ 7~8행의 session은 로그 기록을 남기고, password-auth 파일의 설정에 따라 확인

04 보안 관리 도구

SELinux

- SELinux는 'Security-Enhanced Linux'의 약자로 미국의 NSANational Security Agency가 개발
- SELinux는 침입자가 특정 프로그램을 해킹하여 root 권한을 가졌더라도 다른 프로 그램에서는 root 권한을 행사하지 못하도록 막음
- 이를 위해 SELinux는 기존의 접근 제어 기능과 다른 좀 더 강력한 접근 제어 기능을 제공

04 보안 관리 도구

SELinux의 기초

- SELinux는 리눅스 커널에 강제 접근 제어(MAC)를 구현한 것으로, 기존의 임의적 접근 제어(DAC) 외에 추가적인 접근 제어를 사용
- SELinux를 사용할 때 파일, 디렉터리, 장치 등은 객체로 정의되고 사용자, 프로세스, 프로그램은 주체로 정의
- 대부분의 운영체제는 주체가 객체를 사용할 때 DAC를 적용
- DAC를 사용하는 운영체제에서는 사용자가 자신이 소유한 파일(객체)을 통제할 수 있음
- 예를 들어 리눅스에서는 사용자가 자신의 홈 디렉터리에 중요한 정보가 저장된 파일을 생성하고 이를 누구나 읽을 수 있도록 설정할 수 있음
- 따라서 DAC에만 의존하는 방법은 더 강력한 보안을 구현하는 데 적합하지 않음
- DAC는 오직 사용자의 계정과 소유권에만 기반하여 접근 제어를 하기 때문

04 보안 관리 도구

SELinux의 기초

- DAC는 사용자의 역할이나 프로그램의 기능과 신뢰도, 데이터의 민감성과 무결성 등을 고려하지 않음
- SELinux는 MAC를 리눅스 커널에 추가함. CentOS에서는 MAC가 기본으로 동작함
- MAC는 보안 정책을 시스템의 파일과 프로세스에 전반적으로 강제 적용할 수 있도록 함
- 이때 파일의 유형이나 사용자의 역할에 따라 접근 여부를 세부적으로 정의하여 통제함
- 한 가지 알아 둘 점은 SELinux가 바이러스 체크 프로그램 같은 것이 아니며, 방화벽이나 암호 같은 다른 보안 시스템을 모두 대체하는 것도 아니고 종합 보안 솔루션도 아니라는 점

04 보안 관리 도구

SELinux 설정 파일

- SELinux를 설정하는 파일은 /etc/selinux/config로, 이 파일의 내용은 다음과 같음

```
[root@localhost ~]# cat /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
(생략)
#
SELINUX=enforcing
# SELINUXTYPE= can take one of these three values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

04 보안 관리 도구

SELinux 설정 파일

- 이 설정 파일은 SELinux의 세부 정책을 설정하는 파일이 아님.
  이 설정 파일은 SELinux의 적용 여부를 결정하고 적용할 정책을 지정함
  - **SELINUX**: SELinux를 사용할지를 결정. 이 항목의 값이 Enforcing이면 SELinux를 적용.
    Permissive이면 SELinux 정책이 강제로 적용되지 않고 경고 메시지만 출력됨.
    Disable이면 SELinux가 적용되지 않음
  - **SELINUXTYPE**: SELinux의 정책 유형을 지정.
    타깃으로 정해진 프로세스만 보호할지, 선택된 프로세스만 보호할지를 지정함
- 이 파일을 수정하면 시스템을 재시작해야 적용됨

04 보안 관리 도구

SELinux 설정 명령

- SELinux의 적용 여부를 바꾸기 위해 매번 /etc/selinux/config 파일을 수정하고 재시작할 수는 없음
- SELinux의 적용을 즉시 바꾸려면 명령을 사용해야 함
- getenforce 명령으로 현재의 SELinux 상태를 알 수 있고, setenforce 명령으로는 SELinux의 상태를 바꿀 수 있음

04 보안 관리 도구

■ SELinux 설정 명령

① 현재의 SELinux 상태를 알아보자

```
[root@localhost ~]# getenforce
Enforcing
```

② SELinux의 적용을 중지하려면 setenforce 0 명령을 사용.

적용을 중지한 후 geten force 명령으로 상태를 확인해 봄

```
[root@localhost ~]# setenforce 0
[root@localhost ~]# getenforce
Permissive
```

SELinux의 상태가 Enforcing에서 Permissive로 바뀌었음. SELinux의 적용이 중지된 것

③ 다시 SELinux를 적용하려면 setenforce 1 명령을 실행

```
[root@localhost ~]# setenforce 1
[root@localhost ~]# getenforce
Enforcing
```

Thank you!

한빛아카데미 Hanbit Academy, Inc.