인하공업전문대학
INHA TECHNICAL COLLEGE

# 서버구축실습

2023년 2학기

# 서버 구축 실습 STACK

리눅스 서버

리눅스 네트워크
서비스 관리

- 네트워크 설정
- 텔넷 SSH
- FTP
- DB
- Apache(웹서버)
- NFS & SAMBA

리눅스
시스템 관리

- 디스크관리
- 패키지 관리
- 사용자 관리

리눅스
기본 명령

- 디렉터리와 파일
- 문서 편집
- 셸사용
- 접근 권한 설정
- 프로세스 관리

리눅스
개요 및 설치

- 리눅스 개요
- AWS 리눅스 인스턴스 생성
- 명령 사용 환경

리눅스 보안

# 01 정보 보안의 기초

# 01. 정보 보안의 기초

## ■ 정보 보안의 3요소

- 기밀성
  - 기밀성은 허가받은 사용자만 해당 정보에 접근할 수 있도록 하는 것
  - 사용자를 인증, 읽기·쓰기 등의 접근 제어, 데이터의 암호화 등
- 무결성
  - 정보가 무단으로 변조되지 않았다는 것, 즉 해당 정보가 완전하고 정확하다는 것을 보장
  - 전자 서명 기법을 활용
- 가용성
  - 인가를 받은 사용자가 필요할 때 정보나 서비스에 접근할 수 있는 것

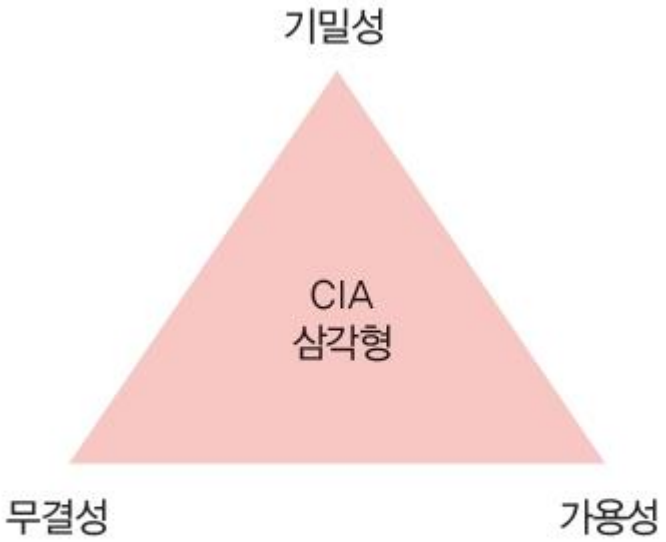



그림 15-1 정보 보안의 3요소

# 01. 정보 보안의 기초

## 보안 기본 조치

- 불필요한 서비스 통제하기
- 소프트웨어 패치 설치하기
- 주기적으로 점검하기
- 백업하기
- 공부하는 시스템 관리자

# 02 시스템 로그

# 02. 시스템 로그

## 로그

- 커널과 리눅스 시스템이 제공하는 여러 서비스 및 응용 프로그램이 발생시키는 메시지

## 주요 로그 파일: /var/log 디렉터리

```
user1@myubuntu:/$ ls /var/log
alternatives.log        dmesg.3.gz              sssd
alternatives.log.1      dmesg.4.gz              syslog
alternatives.log.2.gz   dpkg.log               syslog.1
apache2                 dpkg.log.1              syslog.2.gz
apport.log              dpkg.log.2.gz           syslog.3.gz
apport.log.1            faillog                 syslog.4.gz
apt                     fontconfig.log          ubuntu-advantage-timer.log
aptitude                gdm3                    ubuntu-advantage.log
auth.log                gpu-manager-switch.log  ubuntu-advantage.log.1
auth.log.1              gpu-manager.log         ubuntu-advantage.log.2.gz
auth.log.2.gz           hp                      unattended-upgrades
auth.log.3.gz           installer               vmware-network.1.log
auth.log.4.gz           journal                 vmware-network.2.log
(생략)
```

같은 파일명에 번호가 붙은 파일이 여러 개 있는 것은 한 파일에 저장하면 파일 크기가 너무 커져서 파일 내용을 보거나 관리할 때 불편하기 때문에 나눈 것

# 02. 시스템 로그

## ■ 주요 로그 파일

데몬 : 백그라운드에서 실행되는 프로그램

표 15-1 리눅스의 주요 로그 파일

| 로그 파일 | 설명 |
|---|---|
| /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/cups/* | cupsd 데몬이 생성하는 로그를 기록한다. cupsd 데몬은 인터넷 프린팅 프로토콜을 지원하는 데몬이다. |
| /var/log/apache2/* | 아파치 웹 서버에 의해 생성된 메시지를 기록한다. |
| /var/log/journal/* | journald 데몬이 관리하는 메시지를 기록한다. |
| /var/log/ufw | 방화벽 데몬이 생성하는 메시지를 기록한다. 인바운드 규칙 |
| /var/log/lastlog | 각 계정의 가장 최근 로그인 정보를 기록하며 lastlog 명령으로 확인할 수 있다. |
| /var/log/mail.log | 메일 송수신과 관련된 메시지를 기록한다. |
| /var/log/mysql/* | MariaDB에 의해 생성된 메시지를 기록한다. |
| /var/log/syslog | syslog가 생성하는 공통 로그를 기록한다. |
| /var/log/samba/* | 삼바에 의해 생성된 메시지를 기록한다. |
| /var/log/wtmp | 로그인 정보를 기록하며 last 명령으로 확인할 수 있다. |

# 02. 시스템 로그

## ■ 로그 파일 예: /var/log/syslog

```
user1@myubuntu:~$ cat /var/log/syslog
(생략)
Jan  9 15:22:44 myubuntu systemd[1]: Stopping Samba SMB Daemon...
Jan  9 15:22:44 myubuntu systemd[1]: smbd.service: Deactivated successfully.
Jan  9 15:22:44 myubuntu systemd[1]: Stopped Samba SMB Daemon.
Jan  9 15:22:44 myubuntu systemd[1]: Starting Samba SMB Daemon...
Jan  9 15:22:44 myubuntu systemd[1]: Started Samba SMB Daemon.
Jan  9 15:30:01 myubuntu CRON[73635]: (root) CMD ([ -x /etc/init.d/anacron ] && if [ !
-d /run/systemd/system ]; then /usr/sbin/invoke-rc.d anacron start >/dev/null; fi)
Jan  9 15:30:23 myubuntu systemd[1]: Started Run anacron jobs.
Jan  9 15:30:23 myubuntu anacron[73636]: Anacron 2.3 started on 2022-01-09
Jan  9 15:30:23 myubuntu anacron[73636]: Normal exit (0 jobs run)
Jan  9 15:30:23 myubuntu systemd[1]: anacron.service: Deactivated successfully.
```

날짜와 시간, 로그를 발생시킨 호스트 이름, 데몬 이름, 메시지 내용 등으로 구성되며 한 행씩 기록

# 02. 시스템 로그

## ■ 전통적인 로그 관리 방법은 이제 journal 기능으로 대체

- journal은 기존 syslog 형식에 따라 로그를 저장하고, 저장된 로그에 접근하기 위해 journalctl 명령을 사용

```
user1@myubuntu:/$ journalctl
-- Journal begins at Sun 2021-11-21 12:05:33 KST, ends at Sun 2022-01-09 15:36:>
11월 21 12:05:33 ubuntu kernel: Linux version 5.13.0-21-generic (buildd@lgw01-a>
11월 21 12:05:33 ubuntu kernel: Command line: BOOT_IMAGE=/boot/vmlinuz-5.13.0-2>
11월 21 12:05:33 ubuntu kernel: KERNEL supported cpus:
11월 21 12:05:33 ubuntu kernel:   Intel GenuineIntel
11월 21 12:05:33 ubuntu kernel:   AMD AuthenticAMD
11월 21 12:05:33 ubuntu kernel:   Hygon HygonGenuine
11월 21 12:05:33 ubuntu kernel:   Centaur CentaurHauls
11월 21 12:05:33 ubuntu kernel:   zhaoxin   Shanghai
11월 21 12:05:33 ubuntu kernel: Disabled fast string operations
(생략)
```

# 02. 시스템 로그

## rsyslog 데몬(로그 관리 데몬)

- rsyslog 서비스를 설정하는 파일은 /etc/rsyslog.d 디렉터리에 있는 *.conf 파일
- 설정 파일들에는 어떤 로그를 어떻게 처리할 것인지를 규칙으로 정의
- 규칙 = 필터 + 동작

## 필터

- 로그 메시지 중 어떤 메시지를 선택할 것인지를 정의
- 기능명과 우선순위로 구성

```
기능명.우선순위
```

- 기능명
  - 로그 메시지를 생성하는 프로그램을 지정
- 우선순위
  - 메시지의 심각도 표시

# 02. 시스템 로그

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 | 보다 민감한 보안 메시지 |
| ftp | 11 | ftp 데몬 |
| local0~7 | 16~23 | 여덟 가지 로컬 메시지 |

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 예

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

- 필터가 선택한 메시지를 어떻게 처리할지를 정의한 것
- 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 예
  - 커널이 발생시킨 메시지 중 우선순위가 crit 이상인 메시지를 /var/log/kern.log 파일에 저장하기

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

  - 메시지를 파일에도 저장하고 user1 사용자에게도 메시지를 출력

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

  - /etc/rsyslog.d/50-default.conf 파일에 설정된 기본 규칙 사례

```
auth,authpriv.*                 /var/log/auth.log
*.*;auth,authpriv.none          -/var/log/syslog
#cron.*                         /var/log/cron.log
#daemon.*                       -/var/log/daemon.log
kern.*                          -/var/log/kern.log
```

# 02. 시스템 로그

## journal 기능

- journal은 systemd 데몬의 구성 요소로 로그 파일의 관리를 담당
- 전통적으로 로그를 관리해온 rsyslog 데몬과 병행하여 사용 가능
- 로깅 데이터는 journald 데몬이 수집·가공하여 journals라고 불리는 바이너리 파일로 저장
- journals 파일에는 커널이나 사용자 프로세스의 메시지, 시스템 서비스의 표준 출력과 표준 오류 등이 저장되며 사용자가 편집할 수 없음
- journald 데몬의 실행 파일 이름은 systemd-journald

# 02. 시스템 로그

## ■ 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

- 로그 파일의 전체 내용 보기

```
user1@myubuntu:/$ journalctl
-- Journal begins at Sun 2021-11-21 12:05:33 KST, ends at Sun 2022-01-09 15:39:>
11월 21 12:05:33 ubuntu kernel: Linux version 5.13.0-21-generic (buildd@lgw01-a>
11월 21 12:05:33 ubuntu kernel: Command line: BOOT_IMAGE=/boot/vmlinuz-5.13.0-2>
11월 21 12:05:33 ubuntu kernel: KERNEL supported cpus:
11월 21 12:05:33 ubuntu kernel:   Intel GenuineIntel
11월 21 12:05:33 ubuntu kernel:   AMD AuthenticAMD
11월 21 12:05:33 ubuntu kernel:   Hygon HygonGenuine
11월 21 12:05:33 ubuntu kernel:   Centaur CentaurHauls
11월 21 12:05:33 ubuntu kernel:   zhaoxin   Shanghai
11월 21 12:05:33 ubuntu kernel: Disabled fast string operations
(생략)
```

# 02. 시스템 로그

- 가장 최근의 로그 내용 보기: -n 옵션

```
user1@myubuntu:/$ journalctl -n 5
-- Journal begins at Sun 2021-11-21 12:05:33 KST, ends at Sun 2022-01-09 15:39:>
 1월 09 15:39:01 myubuntu systemd[1]: phpsessionclean.service: Deactivated succ>
 1월 09 15:39:01 myubuntu systemd[1]: Finished Clean php session files.
 1월 09 15:39:01 myubuntu CRON[73725]: pam_unix(cron:session): session opened f>
 1월 09 15:39:01 myubuntu CRON[73726]: (root) CMD (   [ -x /usr/lib/php/sessionc>
 1월 09 15:39:01 myubuntu CRON[73725]: pam_unix(cron:session): session closed f>
lines 1-6/6 (END)
```

# 02. 시스템 로그

- 로그의 상세한 내용 보기: -o 옵션

```
user1@myubuntu:~$ journalctl -n 1 -o verbose
-- Journal begins at Sun 2021-11-21 12:05:33 KST, ends at Sun 2022-01-09 15:43:>
Sun 2022-01-09 15:43:33.272389 KST [s=974e6c4a1cc44270a502aed7c7bd700f;i=11043;>
    PRIORITY=6
    _TRANSPORT=journal
    _SELINUX_CONTEXT=unconfined
    _BOOT_ID=594f7e1b507842f7841ab26ce6af2286
    _MACHINE_ID=33edce09bc144b6baa199b3bca4a4a71
    _HOSTNAME=myubuntu
    SYSLOG_FACILITY=3
    _UID=0
    _GID=0
    _CAP_EFFECTIVE=1fffffffff
    TID=1
    CODE_FILE=src/core/unit.c
    CODE_LINE=5306
    CODE_FUNC=unit_log_success
    SYSLOG_IDENTIFIER=systemd
```

# 02. 시스템 로그

- 로그를 자동으로 출력하기: -f 옵션

```
user1@myubuntu:~$ journalctl -f
-- Journal begins at Sun 2021-11-21 12:05:33 KST. --
 1월 09 15:43:22 myubuntu whoopsie[1154]: [15:43:22] offline
 1월 09 15:43:22 myubuntu systemd[1]: Starting Network Manager Script Dispatcher
Service...
 1월 09 15:43:22 myubuntu dbus-daemon[928]: [system] Successfully activated service
'org.freedesktop.nm_dispatcher'
 1월 09 15:43:22 myubuntu systemd[1]: Started Network Manager Script Dispatcher
Service.
```

```
 1월 09 15:43:24 myubuntu whoopsie[1154]: [15:43:24] online
 1월 09 15:43:33 myubuntu systemd[1]: NetworkManager-dispatcher.service: Deactivated
successfully.
```

➜ 새로운 로그를 기다리고 있다.

# 02. 시스템 로그

- 우선순위로 필터링하여 출력하기: -p 옵션

```
user1@myubuntu:~$ journalctl -p err
-- Journal begins at Sun 2021-11-21 12:05:33 KST, ends at Sun 2022-01-09 15:43:>
11월 21 12:05:33 ubuntu kernel: piix4_smbus 0000:00:07.3: SMBus Host Controller>
11월 21 12:05:33 ubuntu kernel: sd 32:0:0:0: [sda] Assuming drive cache: write >
11월 21 12:05:50 ubuntu kernel: hub 2-2:1.0: hub_ext_port_status failed (err = >
11월 21 12:05:57 ubuntu gdm-password][1466]: gkr-pam: unable to locate daemon c>
11월 21 12:05:58 ubuntu pulseaudio[1488]: ALSA가 장치에 새 데이터를 쓰도록 재촉>
11월 21 12:05:58 ubuntu pulseaudio[1488]: 이는 대부분 ALSA 드라이버 'snd_ens137>
11월 21 12:05:58 ubuntu pulseaudio[1488]: POLLOUT 세트로 불러 오려했지만 결과적>
11월 21 12:06:02 ubuntu gdm-launch-environment][918]: GLib-GObject: g_object_un>
```

# 02. 시스템 로그

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

```
user1@myubuntu:~$ journalctl --since="2021-08-12 12:00"
-- Journal begins at Mon 2021-06-07 20:44:02 KST, ends at Thu 2021-08-12 17:17:01
KST. --
 8월 12 16:24:58 myubuntu systemd[1]: Started Run anacron jobs.
 8월 12 16:24:59 myubuntu anacron[39498]: Anacron 2.3 started on 2021-08-12
 8월 12 16:24:59 myubuntu systemd[1]: Starting Daily apt download activities...
```

```
user1@myubuntu:~$ journalctl --since="2022-01-01 00:00" --until="2022-01-05 12:00"
-- Journal begins at Sun 2021-11-21 12:05:33 KST, ends at Sun 2022-01-09 15:48:>
 1월 01 09:28:16 myubuntu systemd-resolved[709]: Clock change detected. Flushin>
 1월 01 09:28:16 myubuntu systemd[1]: Started Run anacron jobs.
 1월 01 09:28:16 myubuntu anacron[12701]: Anacron 2.3 started on 2022-01-01
 1월 01 09:28:16 myubuntu anacron[12701]: Will run job `cron.daily' in 5 min.
```

# 02. 시스템 로그

- 필드명으로 필터링하여 출력하기: -F

```
user1@myubuntu:~$ journalctl -F _SYSTEMD_UNIT
user@1000.service
session-57.scope
session-2.scope
session-75.scope
colord.service
packagekit.service
whoopsie.service
```

# 02. 시스템 로그

- 필드명의 특정 값을 지정하면 좀 더 구체적으로 로그 기록을 필터링 가능
  - 예: UID가 0(root)인 사용자가 cron.service에서 발생한 로그 기록을 필터링

```
user1@myubuntu:~$ journalctl _UID=0 _SYSTEMD_UNIT=cron.service
-- Journal begins at Sun 2021-11-21 12:05:33 KST, ends at Sun 2022-01-09 15:48:>
11월 21 12:05:41 ubuntu cron[778]: (CRON) INFO (pidfile fd = 3)
11월 21 12:05:41 ubuntu cron[778]: (CRON) INFO (Running @reboot jobs)
11월 21 12:17:01 ubuntu CRON[4287]: pam_unix(cron:session): session opened for >
11월 21 12:17:01 ubuntu CRON[4288]: (root) CMD (   cd / && run-parts --report />
11월 21 12:17:01 ubuntu CRON[4287]: pam_unix(cron:session): session closed for >
11월 21 12:30:01 ubuntu CRON[4323]: pam_unix(cron:session): session opened for >
11월 21 12:30:01 ubuntu CRON[4324]: (root) CMD ([ -x /etc/init.d/anacron ] && i>
```

# 03 방화벽 관리

# 03. 방화벽 관리

## 방화벽

- 네트워크를 통한 외부의 접속을 차단
- 우분투는 자체적으로 방화벽을 관리하는 도구인 ufw를 제공

## 방화벽 동작 확인하기

```
user1@myubuntu:~$ dpkg -l | grep ufw
ii  gufw                                        21.04.0-0ubuntu1
all          graphical user interface for ufw
ii  ufw                                         0.36.1-1ubuntu1
all          program for managing a Netfilter firewall
```

```
user1@myubuntu:~$ sudo ufw status
상태: 비활성
```

```
sudo ufw enable
```

```
sudo ufw disable
```

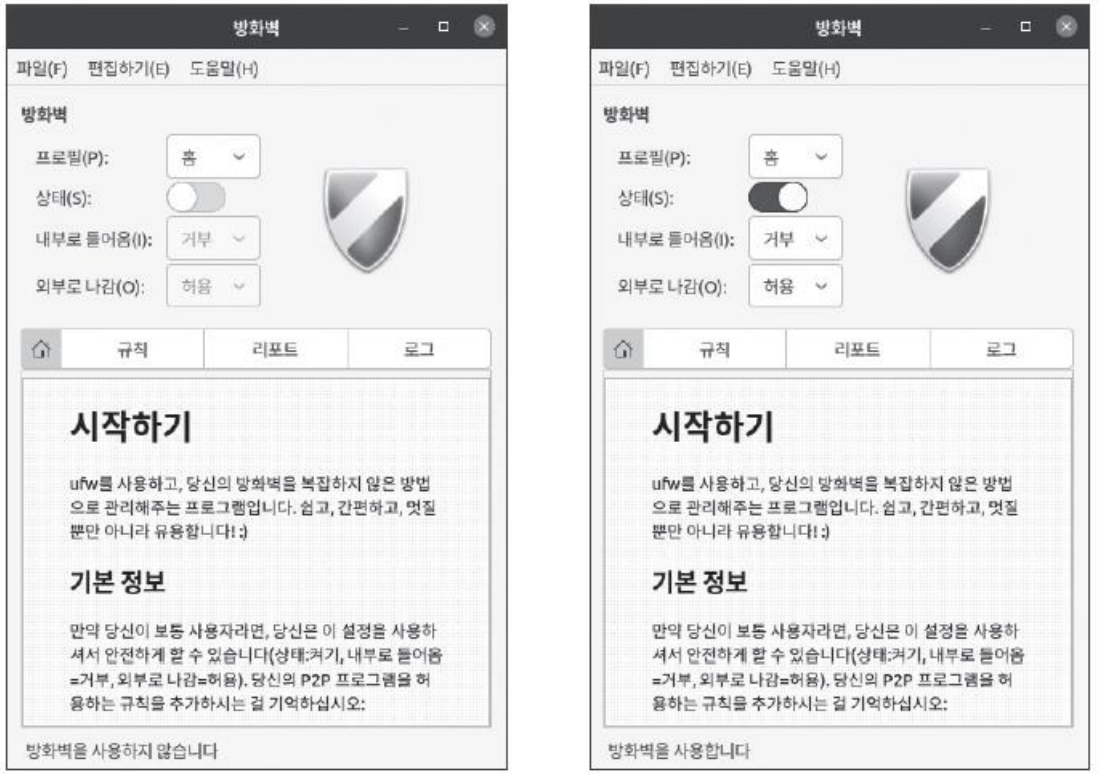
# 03. 방화벽 관리

## GUI 도구로 방화벽 설정하기

- 인증을 위해 암호 입력

```
user1@myubuntu:~$ sudo apt install gufw
```


그림 15-2 gufw 접속 인증 창




(a) 방화벽 비활성 상태 (b) 방화벽 활성 상태

그림 15-3 gufw 동작 화면

# 03. 방화벽 관리

## gufw의 사용법

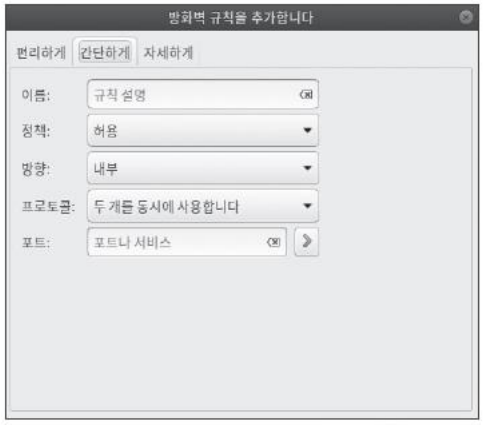
- 프로필(P)
  - 현재 설정하는 내용을 적용할 환경을 설정
  - 설정할 수 있는 값은 홈, 사무실, 공용이며 각 환경에 따라 방화벽을 다르게 설정 가능
- 상태(S)
  - 방화벽 전체를 켜거나 끌 수 있음
- 내부로 들어옴(I)/외부로 나감(O)
  - 시스템으로 들어오는 트래픽과 시스템에서 밖으로 나가는 트래픽에 대한 기본값을 설정
  - 일반적으로 시스템으로 들어오는 트래픽은 모두 거부하고 밖으로 나가는 트래픽은 허용하는 것이 기본값
  - 허용과 거부 외에 거절 reject 이 있음: 거절은 접속을 거부하고 거절된 이유를 알려줌

# 03. 방화벽 관리

- 규칙: 현재 규칙에 +를 선택하여 규칙을 추가하거나, -를 선택하여 규칙을 삭제


그림 15-4 gufw 규칙 화면


(a) 편리하게 모드




(b) 간단하게 모드


(c) 자세하게 모드

그림 15-5 gufw 규칙을 정하는 세 가지 모드 화면

**자세하게 모드**: 규칙의 이름, 번호, 정책, 방향, 인터페이스 선택, 로그 기록 여부, TCP/UDP 선택뿐만 아니라 출발지와 목적지의 주소, 포트 번호 등을 자세하게 설정

**편리하게 모드**: 방화벽을 적용할 응용 분야를 게임, 오디오/비디오, 시스템, 오피스 등으로 구분하고 다시 세부 카테고리를 정하여 방화벽 정책을 정할 수 있음

**간단하게 모드**: 규칙의 이름을 사용자가 정할 수 있으며, TCP/UDP 선택과 포트 번호나 서비스명을 사용자가 직접 지정하고 정책을 적용

# 03. 방화벽 관리

- 리포트: 현재 열려 있는 포트를 보고
- 로그: 방화벽과 관련된 로그 기록 확인


그림 15-6 gufw 리포트 화면


그림 15-7 gufw 로그 화면

# 03. 방화벽 관리

## ■ 방화벽 관리 명령: ufw

**ufw**

- **기능** 방화벽을 설정한다.
- **형식** ufw 서브 명령
- **옵션** enable: 방화벽을 활성화한다.
  disable: 방화벽을 비활성화한다.
  default allow|deny|reject [incoming|outgoing]: 방화벽의 기본 동작을 설정한다.
  status [verbose]: 방화벽의 상태를 출력한다.
  allow 서비스|포트/프로토콜: 지정한 서비스나 포트를 허용한다.
  deny 서비스|포트/프토로콜: 지정한 서비스나 포트를 거부한다.
  delete 명령: 명령으로 설정한 규칙을 삭제한다.
- **사용 예** ufw deny telnet  ufw allow 23/tcp  ufw status

# 03. 방화벽 관리

## 방화벽의 상태 보기

```
user1@myubuntu:~$ sudo ufw status
상태: 활성
```

## 규칙 추가하기

- 예: http 서비스를 허용하도록 방화벽에 규칙을 추가

```
user1@myubuntu:~$ sudo ufw allow http
규칙이 추가되었습니다
규칙이 추가되었습니다 (v6)
user1@myubuntu:~$ sudo ufw status
상태: 활성
목적                         동작         출발
--                           --           --
22/tcp                       ALLOW        Anywhere
80/tcp                       ALLOW        Anywhere
22/tcp (v6)                  ALLOW        Anywhere (v6)
80/tcp (v6)                  ALLOW        Anywhere (v6)
```

# 03. 방화벽 관리

- 서비스 거부하기
  - 예: telnet 서비스 거부

```
user1@myubuntu:~$ sudo ufw deny telnet
규칙이 추가되었습니다
규칙이 추가되었습니다 (v6)
user1@myubuntu:~$ sudo ufw status
상태: 활성
목적                         동작          출발
--                         --          --
22/tcp                     ALLOW       Anywhere
80/tcp                     ALLOW       Anywhere
23/tcp                     DENY        Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
80/tcp (v6)                ALLOW       Anywhere (v6)
23/tcp (v6)                DENY        Anywhere (v6)
```

# 03. 방화벽 관리

- 규칙 삭제하기
  - 예: telnet규칙 삭제

```
user1@myubuntu:~$ sudo ufw delete deny telnet
규칙이 삭제되었습니다
규칙이 삭제되었습니다 (v6)
user1@myubuntu:~$ sudo ufw status
상태: 활성

목적                         동작         출발
--                           --           --
22/tcp                       ALLOW        Anywhere
80/tcp                       ALLOW        Anywhere
22/tcp (v6)                  ALLOW        Anywhere (v6)
80/tcp (v6)                  ALLOW        Anywhere (v6)
```

# 03. 방화벽 관리

- 포트 추가하기
  - 예: 5000번 포트를 추가

```
user1@myubuntu:~$ sudo ufw allow 5000/tcp
규칙이 추가되었습니다
규칙이 추가되었습니다 (v6)
user1@myubuntu:~$ sudo ufw status
상태: 활성

목적                        동작         출발
--                          --           --
22/tcp                      ALLOW        Anywhere
80/tcp                      ALLOW        Anywhere
5000/tcp                    ALLOW        Anywhere
22/tcp (v6)                 ALLOW        Anywhere (v6)
80/tcp (v6)                 ALLOW        Anywhere (v6)
5000/tcp (v6)               ALLOW        Anywhere (v6)
```

# 03. 방화벽 관리

- 특정 IP 주소의 접속 설정하기
  - 예: 192.168.1.90에서 요청하는 ftp 서비스를 허용

```
user1@myubuntu:~$ sudo ufw allow from 192.168.1.90 to any port ftp
규칙이 추가되었습니다
user1@myubuntu:~$ sudo ufw status
상태: 활성

목적                        동작         출발
--                          --           --
22/tcp                      ALLOW        Anywhere
80/tcp                      ALLOW        Anywhere
5000/tcp                    ALLOW        Anywhere
21/tcp                      ALLOW        192.168.1.90
```

# 04 보안 관리 도구

# 04. 보안 관리 도구

## NMap

- 내 서버나 원격의 서버가 사용 중인 포트, 운영체제 등을 스캔하여 출력
- 스캔하는 것만으로도 보안 침입을 위한 준비 과정으로 간주하므로 원격 서버를 마구 스캔하면 안됨

## NMap 설치하기

```
user1@myubuntu:~$ sudo apt install nmap
```

# 04. 보안 관리 도구

## 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
  - 지정한 호스트에서 현재 열려 있는 포트를 요약하여 출력

```
user1@myubuntu:~$ nmap localhost
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-09 16:05 KST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000077s latency).
Not shown: 989 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
631/tcp  open  ipp
3306/tcp open  mysql
5902/tcp open  vnc-2

Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
```

# 04. 보안 관리 도구

- 특정 서버 스캔하기
  - IP 주소를 사용하여 특정 서버를 지정
  - -O 옵션은 해당 시스템의 운영체제 정보를 알려줌

```
user1@myubuntu:~$ sudo nmap -O 192.168.0.7
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-09 16:05 KST
Stats: 0:23:42 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 99.99% done; ETC: 16:29 (0:00:00 remaining)
Nmap scan report for 192.168.0.7
Host is up (0.0037s latency).
Not shown: 992 closed ports
PORT     STATE    SERVICE
135/tcp  open     msrpc
139/tcp  open     netbios-ssn
445/tcp  open     microsoft-ds
514/tcp  filtered shell
902/tcp  open     iss-realsecure
912/tcp  open     apex-mesh
1025/tcp open     NFS-or-IIS
5357/tcp open     wsdapi
Aggressive OS guesses: Microsoft Windows XP SP3 (97%), Microsoft Windows XP SP3 or
Windows 7 or Windows Server 2012 (96%), Actiontec MI424WR-GEN3I WAP (95%), DD-WRT
v24-sp2 (Linux 2.4.37) (94%), Linux 3.2 (94%), VMware Player virtual NAT device (93%),
Linux 4.4 (92%)
```

# 04. 보안 관리 도구

- UDP 포트 스캔하기
  - -sU 옵션을 사용

```
user1@myubuntu:~$ sudo nmap -sU -v localhost
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-09 16:18 KST
Initiating UDP Scan at 16:18
Scanning localhost (127.0.0.1) [1000 ports]
Discovered open port 111/udp on 127.0.0.1
Discovered open port 137/udp on 127.0.0.1
Discovered open port 5353/udp on 127.0.0.1
Completed UDP Scan at 16:18, 1.23s elapsed (1000 total ports)
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000050s latency).
Not shown: 995 closed ports
PORT     STATE         SERVICE
111/udp  open          rpcbind
137/udp  open          netbios-ns
138/udp  open|filtered netbios-dgm
631/udp  open|filtered ipp
5353/udp open          zeroconf

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.29 seconds
           Raw packets sent: 1002 (28.983KB) | Rcvd: 2003 (86.398KB)
```

# 04. 보안 관리 도구

- 특정 네트워크를 대상으로 포트 스캔하기

```
user1@myubuntu:~$ sudo nmap -sT -O -v 192.168.147.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-09 16:19 KST
Initiating ARP Ping Scan at 16:19
Scanning 255 hosts [1 port/host]
Completed ARP Ping Scan at 16:20, 1.94s elapsed (255 total hosts)
Initiating Parallel DNS resolution of 255 hosts. at 16:20
Completed Parallel DNS resolution of 255 hosts. at 16:20, 0.19s
Nmap scan report for 192.168.147.0 [host down]
Nmap scan report for 192.168.147.3 [host down]
```

```
445/tcp  open  microsoft-ds
5902/tcp open  vnc-2
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Uptime guess: 2.612 days (since Fri Jan  7 01:38:27 2022)
Network Distance: 0 hops
TCP Sequence Prediction: Difficulty=260 (Good luck!)
IP ID Sequence Generation: All zeros
```

# 04. 보안 관리 도구

## PAM

- 사용자의 신원을 확인하는 인증하기 위해 사용하는 중앙 집중화된 인증 기법
- PAM은 ‘pluggable authentication modules’의 약자: 삽입형 인증모듈
- PAM은 모듈 방식으로 구성되어 있어 시스템 관리자가 필요에 따라 인증 모듈을 추가·삭제·편집 가능

# 04. 보안 관리 도구

## PAM 설정 파일

- PAM 모듈은 /etc/pam.d 디렉터리에 설정 파일을 가지고 있음

```
user1@myubuntu:~$ ls /etc/pam.d
atd                             lightdm-greeter
chfn                            login
chpasswd                        mate-screensaver
chsh                            newusers
common-account                  other
common-auth                     passwd
common-password                 polkit-1
```

# 04. 보안 관리 도구

## PAM 설정 파일 형식

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

- 모듈 인터페이스
  - auth: 이 모듈은 사용자를 인증하는 데 사용된다. 예를 들어 암호가 정확한지를 확인. 또한 이 모듈은 그룹 지정에도 적용
  - account: 이 모듈은 접근이 허용되는지를 확인. 예를 들어 사용자 계정이 유효한지, 사용자가 해당 날짜에 로그인할 수 있는지를 확인
  - password: 이 모듈은 사용자 계정의 암호를 바꾸는 데 사용
  - session: 이 모듈은 사용자의 세션을 설정하고 관리. 또한 사용자의 홈 디렉터리를 마운트하는 것과 같이 접근을 허용하는 데 필요한 부가적인 작업을 수행

# 04. 보안 관리 도구

- 제어 플래그 : 제어 플래그는 특정 모듈의 성공과 실패를 어떻게 처리할 것인지를 알려줌
  - 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. 그러나 만약 이 지점에서 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.
  - 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
  - 현재는 PAM 모듈이 /lib/x86_64-linux-gnu/security 디렉터리에 위치

```
user1@myubuntu:~$ ls /lib/x86_64-linux-gnu/security
pam_access.so       pam_gdm.so              pam_nologin.so      pam_systemd.so
pam_cap.so          pam_gnome_keyring.so    pam_permit.so       pam_tally.so
pam_cifscreds.so    pam_group.so            pam_pwhistory.so    pam_tally2.so
pam_debug.so        pam_issue.so            pam_pwquality.so    pam_time.so
pam_deny.so         pam_keyinit.so          pam_rhosts.so       pam_timestamp.so
pam_echo.so         pam_lastlog.so          pam_rootok.so       pam_tty_audit.so
pam_env.so          pam_limits.so           pam_securetty.so    pam_umask.so
pam_exec.so         pam_listfile.so
(생략)
```

- 모듈 인자: 인증 과정에서 정보가 필요한 일부 모듈에 정보를 전달

# 04. 보안 관리 도구

## PAM 파일의 예

- /etc/pam.d/vsftpd 파일

```
 1 # Standard behaviour for ftpd(8).
 2 auth    required        pam_listfile.so item=user sense=deny file=/etc/ftpusers
onerr=succeed
 3
 4 # Note: vsftpd handles anonymous logins on its own. Do not enable pam_ftp.so.
 5
 6 # Standard pam includes
 7 @include common-account
 8 @include common-session
 9 @include common-auth
10 auth    required        pam_shells.so
```

# 04. 보안 관리 도구

- common-auth 파일

```
user1@myubuntu:~$ vcat /etc/pam.d/common-auth
(생략)
# here are the per-package modules (the "Primary" block)
auth    [success=2 default=ignore]      pam_unix.so nullok_secure
auth    [success=1 default=ignore]      pam_sss.so use_first_pass
# here's the fallback if no module succeeds
auth    requisite                       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
auth    required                        pam_permit.so
# and here are more per-package modules (the "Additional" block)
auth    optional                        pam_cap.so
# end of pam-auth-update config
```

- pam_unix.so nullok: pam_unix.so는 계정의 정상 여부를 확인하기 위해 필요한 모듈이다. pam_unix.so 모듈은 계정이 만료되지는 않았는지, 정해진 기간에 암호를 변경했는지 등을 확인한다. nullok 인자는 pam_unix.so 모듈이 빈 암호도 허용한다는 것을 의미한다.
- pam_deny.so, pam_permit.so: 이 모듈들은 접근을 거부하거나 허용