Linux 보안(firewalld,SELinux)

- RHEL, CentOS 기반의 리눅스 배포판은 커널 레벨에 두 가지 중요한 보안 기능이 탑재되어 있으며 하나는 firewalld 의 기반이 되는 netfilter 로 네트워크 패킷 필터링을 수행하며 허가되지 않은 곳으로부터 네트워크 연결을 차단하여 시스템을 보호한다.
- 다른 하나는 보안강화 리눅스(SELinux; Security Enhanced Linux) 라는 커널 기반의 보안 모듈이다.

방화벽

- 방화벽은 외부에서 원치 않는 트래픽으로부터 시스템을 보호할 수 있는 방법이다.
- 사용자는 방화벽 세트를 정의하여 호스트 시스템에서 들어오는 네트워크 트래픽을 제어할 수 있다.
- 이러한 규칙은 들어오는 트래픽을 정렬하고 차단하거나 통과하도록 허용하는데 사용된다.
- 미리 설정된 보안 규칙에 따라 수신 및 발신 네트워크 트래픽을 모니터링하고 제어하는 네트워크 보안 시스템이다.
- 방화벽은 일반적으로 신뢰할 수 있는 보안 내부 네트워크와 다른 외부 네트워크 간의 장벽을 설정한다.
- Red Hat Enterprise Linux 7 에서 방화벽은 Red Hat Enterprise Linux 설치 중에 자동으로 활성화되는 firewalld 서비스 에 의해 제공된다.
- 그러나 킥 스타트 구성에서 서비스를 명시 적으로 비활성화 한 경우, 다시 활성화 할 수 있다.

리눅스 보안 1) firewalld 서비스

filewalld 란

- Linux 커널 2.2 까지는 ipchains 이라는 패킷 필터링/방화벽 프레임워크가 구현되어 있었고 2.4 부터는 더 유연하고 다양한 기능을 가진 netfilter 로 프레임워크가 교체 되었다.
- iptables 은 netfilter 프레임워크의 최상단에 위치하는 사용자 레벨의 프로그램으로 시스템 관리자는 iptables 명령어로 리눅스 서버로 들어오고 나가는 패킷을 필터링하거나 포트 포워딩을 설정할 수 있으며 방화벽으로도 사용할 수 있다.
- iptables 는 숙련된 관리자가 아니면 사용이 어려운 단점이 있었는데 이런 문제를 해결하고자 RHEL/CentOS 7 부터는 방화벽을 firewalld 라는 데몬으로 교체하였고 이에 따라 사용자 레벨의 프로그램은 iptables 명령어 대신 명령행에서는 firewall-cmd , GUI 환경에서는 firewall-config 를 사용하게 되었다.

firewalld 서비스

- 일반 사용자로도 firewalld 의 현재 상태를 표시 할 수 있다 .

  $ systemctl status firewlld
- enable 상태나 running 상태가 아닌 경우, root 사용자로 다시 활성화 시킬 수 있다.

  # systemctl start firewalld

  # systemctl enable firewalld

영구적 규칙과 정책 재구동

- 기본적으로 firewall-cmd 로 방화벽 정책을 변경했을 경우 현재 구동되고 있는 firewalld 에 즉시 적용되지만 정책은 지속성이 없이 임시로 적용되며 정책을 재구동하는 명령어인 firewall-cmd --reload 를 실행하거나 시스템을 재부팅하면 예전 정책으로 다시 초기화 되며 이로 인해 서비스의 장애가 발생할 수 있다.
- 이 때문에 방화벽 정책을 영구적으로 유지하기 위해서는 --permanent 옵션을 추가해서 실행하면 되지만 이는 즉시 적용되지 않고 firewall-cmd --reload 명령어로 방화벽 정책을 재구동하거나 재부팅을 하기 전에는 변경한 방화벽 설정이 적용되지 않는다.
- firewalld 를 처음에 사용할 때 이 때문에 혼란을 겪고 정책 설정을 잘못한 걸로 오해하는 사용자가 많으므로 아래 표를 보고 방화벽 정책의 즉시 적용 여부와 지속성 여부를 꼭 익혀 두어야 한다.

- 방화벽 정책 적용 여부

	firewall-cmd	firewall-cmd --permanent
즉시 적용(firewall-cmd –reload 불필요)	예	아니오
재부팅 시 정책의 지속 여부	아니오	예

- 예로 ***fireall-cmd*** 로 정책을 추가했을 경우 즉시 반영되지만 만약 잘못된 설정이었을 경우에 f***irewall-cmd --reload*** 명령어를 실행하면 예전 정책으로 복구된다.
- 하지만 ***firewall-cmd --permanent*** 로 정책을 추가했을 경우 **firewall-cmd *--reload*** 명령어를 실행해야 변경한 정책이 적용되며 예전 정책으로 복구하려면 새로 추가한 정책을 삭제하고 다시 방화벽을 구동해야 한다.
- 만약 방화벽 정책 변경 시 즉시 반영하고 재부팅시에도 유지되도록 하고 싶으면 firewall-cmd 를 두 번 호출하면 되며 한 번은 --permanet 옵션을 주고 한 번은 옵션을 빼면 된다.

Network Zone

- firewalld 는 사용자가 해당 네트워크 내의 interface 와 트래픽에 적용하기도 결정한 신뢰 수준에 따라 네트워크를 여러 영역으로 분리한다.
- 사전 정의된 영역(zone)은 /usr/lib/firewalld/zones/ 디렉토리에 저장되며 사용 가능한 모든 네트워크 인터페이스에 즉시 적용될 수 있다.
- 이러한 파일은 수정된 후에 /etc/firewalld/zones/ 디렉토리에 복사된다.
- 다음은 미리 정의 된 영역의 기본 설정에 대한 설명이다.

zone	설명
block	incoming 네트워크에 대한 패킷이 응답메세지 출력 후 거부된다. (출력메세지 : icmp-host-prohibited(IPv4), icmp6-adm-prohibited(IPv6))
dmz	개방된 네트워크와 연결되어 있지만 제한적으로 내부 네트워크에 접속이 가능한 호스트를 위한 영역이다.
drop	incoming 네트워크 패킷은 통지없이 삭제되며 outgoing 네트워크 연결만 허용된다.
external	masquerading 이 가능한 외부 네트워크에서 사용 (특히 라우터의 경우). 네트워크의 다른 컴퓨터가 사용자 컴퓨터에 해를 끼치지 않는다고 믿지 않을 때 사용. 선택한 수신 연결 만 허용된다.
home	네트워크의 다른 컴퓨터를 주로 신뢰하는 경우 가정에서 사용한다. 선택한

	수신 연결 만 허용된다.
internal	주로 네트워크의 다른 컴퓨터를 신뢰할 때 내부 네트워크에서 사용한다. 선택한 수신 연결 만 허용된다.
public	네트워크상의 다른 컴퓨터를 신뢰하지 않는 공공 장소에서 사용. 선택한 수신 연결 만 허용된다.
trusted	모든 네트워크 연결이 허용된다.
work	네트워크상의 다른 컴퓨터를 주로 신뢰하는 직장에서 사용. 선택한 수신 연결 만 허용된다.

- 기본 설정된 zone 목록 확인

```
# firewall-cmd --get-zones
work drop internal external trusted home dmz public block
```

- 사전 정의된 zone 에 대한 자세한 정보 출력(어떤 zone 이 어떤 정책을 가지고 있는 확인)

```
# firewall-cmd --list-all-zones
work
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  sourceports:
  icmp-blocks:
  rich rules:

drop
  target: DROP
  icmp-block-inversion: no
  interfaces:
  sources:
  services:
  ports:
```

```
  protocols:
  masquerade: no
  forward-ports:
  sourceports:
  icmp-blocks:
  rich rules:
....
```

→ 각 zone 별 services 항목을 보면 해당 zone 에서 기본적으로 허용하는 서비스 포트를 확인할 수 있으며 dmz 는 ssh 를, internal 은 dhcpv6-client,mdns 등을 허용하는 것을 알 수 있다.

- 현재 활성화된 zone 출력

```
# firewall-cmd --get-active-zone
public
    interfaces: ens33
```

→ 현재 활성화된 zone 은 public 인 것을 알 수 있다.

- 현재 활성화된 zone 정보 출력

```
# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens33
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  sourceports:
  icmp-blocks:
  rich rules:
```

- 특정 존의 정보 출력

```
# firewall-cmd --list-all --zone work
work
  target: default
  icmp-block-inversion: no
```

```
  interfaces:
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  sourceports:
  icmp-blocks:
  rich rules:
```

- zone 바꾸기

```
# firewall-cmd --set-default-zone=dmz
success
# firewall-cmd --get-active-zone
dmz
  interfaces: ens33
# firewall-cmd --set-default-zone=public
success
# firewall-cmd --get-active-zone
public
  interfaces: ens33
```

zone 만들기

- 만약 firewalld 에 내장된 zone 설정이 현재 서비스에 딱 들어맞지 않는다면 --new-zone 옵션을 사용하여 새로 zone 을 만들 수 있다.
- 다음은 webserver 라는 이름의 새로운 zone 을 만든다.

```
# firewall-cmd --permanent --new-zone=webserver
success
```

- 새로 생성한 존은 바로 반영되지 않으므로 방화벽 정책을 다음과 같이 다시 읽어 들여야 한다.

```
#firewall-cmd --reload
success
# firewall-cmd --get-zones        → 확인
work drop webserver internal external trusted home dmz public block
```

사전 정의된 서비스

- firewalld 에는 방화벽 정책 변경 시 용이하도록 많이 사용하는 서비스를 사전에 정의해 놓고 있으며 --get-services 옵션으로 전체 목록을 확인할 수 있다.

```
# firewall-cmd --get-services
RH-Satellite-6 amanda-client amanda-k5-client bacula bacula-client ceph ceph-mon
dhcp dhcpv6 dhcpv6-client dns docker-registry dropbox-lansync freeipa-ldap
freeipa-ldaps freeipa-replication ftp high-availability http https imap imaps ipp
....
```

서비스 추가

- 웹 서버는 http(80), https(443) 으로 접근을 허용해야 하므로 --add-service=SERVICENAME 구문으로 서비스를 추가해준다.
- --add-service 는 여러 개의 서비스를 동시에 기술할 수 없으므로 http 와 https 를 따로 기술해준다.

```
1) webserver 존 상세 내용 확인
# firewall-cmd --list-all --zone webserver
webserver
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services:                    -- 목록이 비어있음
  ports:
  protocols:
  masquerade: no
  forward-ports:
  sourceports:
  icmp-blocks:
  rich rules:
2) webserver zone 에 http, https 서비스 추가
# firewall-cmd --permanent --zone=webserver --add-service=http
success
# firewall-cmd --permanent --zone=webserver --add-service=https
success
```

3) 확인

 # firewall-cmd --list-all --zone webserverwebserver

target: default
icmp-block-inversion: no
interfaces:
sources:
services: -- --permanent 옵션 사용으로 즉시 적용되지 않음
ports:
protocols:
masquerade: no
forward-ports:
sourceports:
icmp-blocks:
rich rules:

4) 적용(reload)

 # firewall-cmd –reload

success

 # firewall-cmd --list-all --zone webserver

webserver
target: default
icmp-block-inversion: no
interfaces:
sources:
services: http https
ports:
protocols:
masquerade: no
forward-ports:
sourceports:
icmp-blocks:
rich rules:

서비스 삭제

- --remove-service=SERVICENAME 구문을 사용하여 http, https 서비스 삭제

```
# firewall-cmd --zone=webserver --remove-service=http
success
# firewall-cmd --zone=webserver --remove-service=https
success
# firewall-cmd --list-all --zone webserver
webserver
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services:          --permanent 옵션을 사용하지 않아 즉시적용됨
  ports:
  protocols:
  masquerade: no
  forward-ports:
  sourceports:
  icmp-blocks:
  rich rules:
```

포트 추가

- 만약 사전 정의된 서비스가 아닌 다른 포트를 사용하는 경우
  –add-port=포트번호/프로토콜 형식으로 등록할 수 있다.
- 포트 번호가 범위일 경우-를 구분자로 등록할 수 있다.
- 다음은 9090 에서 9100 까지 TCP 포트를 허용하여 방화벽을 설정하는 예제이다.

```
# firewall-cmd --permanent --zone=webserver --add-port=9090-9100/tcp
success
# firewall-cmd --reload
success
# firewall-cmd --list-all --zone webserver
webserver
  target: default
  icmp-block-inversion: no
```

```
  interfaces:
  sources:
  services: http https
  ports: 9090-9100/tcp
  protocols:
  masquerade: no
  forward-ports:
  sourceports:
  icmp-blocks:
  rich rules:
```

포트 삭제

- --remove-port=포트번호/프로토콜형식으로 사용하면 되며 add 와 마찬가지로 범위일 경우 –를 구분자로 사용한다.

```
# firewall-cmd --permanent --zone=webserver --remove-port=9090-9100/tcp
success
# firewall-cmd --reload
success
# firewall-cmd --list-all --zone webserver
webserver
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: http https
  ports:
  protocols:
  masquerade: no
  forward-ports:
  sourceports:
  icmp-blocks:
  rich rules:
```

➢ 방화벽에 대한 더 많은 내용 : https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/security_guide/sec-using_firewalls

리눅스 보안 2)SELinux

- SELinux 는 미국 국가안보국(NSA - National Security Agency)이 개발한 Flask 라는 보안 커널을 리눅스에 이식한 강제 접근 통제 커널 보안 모듈이다.
- NSA 는 리눅스에 강제 접근 통제를 구현하기 위해 리눅스 커널의 많은 부분을 수정했으며 그 결과물을 리눅스 커뮤니티에 기증하여 2003 년부터 2.6 버전의 커널에 공식 포함되었다.
- RHEL 기반의 배포판에는 버전 4 부터 공식적으로 포함되었으며 원활한 사용을 위해 ls, cp, mv, ps 등의 시스템의 많은 유틸리티가 SELinux 를 지원하게 수정이 되었고 아파치 웹 서버, ftp 서버, 삼바 서버등 주요 서비스 데몬 프로세스를 위한 SELinux 정책이 기본 탑재되었다.
- 현재는 많은 제품들이 SELinux 를 지원하므로 기본적인 개념과 설정 방법을 익힌다면 큰 어려움 없이 사용할 수 있을 정도이며 이를 사용한다면 다음과 같은 장점이 있다.

  사전 정의된 접근 통제 정책 탑재

  사용자, 역할, 타입, 레벨 등의 다양한 정보를 조합하여 어떤 프로세스가 어떤 파일, 디렉터리, 포트 등에 접근 가능한지에 대해 잘 정의된 접근 통제가 제공되므로 강제 접근 통제 적용을 위해 시스템 관리자가 할 일이 대폭 줄었다.

  "Deny All, Permit Some" 정책으로 잘못된 설정 최소화

  서두에 말한 "모든 걸 차단하고 필요한 것만 허용"하는 정책은 단순하면서 강력한, 정보 보호를 위한 최선의 정책으로 SELinux 의 보안 정책도 이 방식으로 사전에 설정되어 있으므로 잘못된 기본 설정으로 인한 보안 취약점이 최소화 된다.

  최소 권한 정책에 따른 취약점 감소

  SELinux 는 setuid 비트가 켜져 있거나 루트로 실행되는 데몬 프로세스 프로그램들은 샌드박스안에서 별도의 도메인으로 격리되어 낮은 등급으로 실행되므로 루트 권한을 탈취해도 해당 도메인에만 영향을 미치고 전체 시스템에 미치는 영향을 최소화 한다.

  예로 아파치 httpd 서버의 보안 취약점을 통해 권한을 획득했어도 낮은 등급의 권한을 부여 받으므로 공격자는 정해진 타입의 파일만 읽을 수 있으므로 /etc/passwd 파일을 가져갈 수 없으며 mysql 데이타 파일에도 접근할 수 없다.

또 80, 443 같이 웹 서비스에 필요한 포트에만 접근을 허용하고 있으므로 웹 서버 권한을 획득했어도 ssh 로 다른 서버에 접근할 수 없으므로 이차 피해를 최소화할 수 있다.

잘못된 설정과 버그로부터 시스템 보호

데이터와 기밀성과 무결성을 적용할 수 있으며 신뢰할 수 없는 입력에서 프로세스를 보호할 수 있다. 예로 버퍼의 입력 길이 등을 제대로 체크하지 않아서 발생하는 버퍼 오버 플로우 공격(buffer overflow attack)의 경우 SELinux 는 어플리케이션이 메모리에 있는 코드를 실행할 수 없게 통제하므로 데몬 프로그램에 버퍼 오버 플로우 버그가 있어도 쉘을 쉽게 얻을 수 없다.

SELinux 의 한계

SELinux 의 주요 목표는 잘못된 설정이나 프로그램의 보안 버그로 인해 시스템이 공격 당해도 시스템과 데이터를 보호하고 2 차 피해를 막는 것으로 SELinux 로 모든 보안 요건이 충족되지는 않는다.
중요한 서비스라면 SELinux 외에 추가 보안 수단을 사용하여 시스템과 서비스를 보호해야 하며 방화벽이나 백신, 침입 탐지 시스템(Intrusion Detection System) 등과 혼용해서 사용해야 더욱 견고한 시스템을 구성할 수 있다.

SELinux 의 사용

SELinux 의 동작 모드

- Disabled
- Enforcing

  설치 후 기본 모드

  SELinux 의 정책과 룰에 어긋나는 동작 모두 차단
- Permssive

  정책에 어긋나는 동작은 감사 로그를 남기고 허용

현재 SELinux 모드 표시 및 설정 변경

- 현재의 **SELinux** 모드 표시

```
# sestatus
```

```
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Max kernel policy version:      28
# getenforce

Enforcing
```

- SELinux 모드 전환

```
# setenforce 0          (동일 명령어 : #setenforce Permissive)

# getenforce

Permissive

# sestatus

SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   permissive
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Max kernel policy version:      28
```

- permissive 로 전환한 후에 sestatus 의 결과를 보면 Current mode 는 permissive 이지만 Mode from config file 항목은 enforcing 으로 두 개가 다르다.
- setenforce 명령어로 설정한 모드는 영구적이지 않으며 재부팅하면 설정이 초기화 된다.
- 부팅 시 기본으로 설정할 모드는 /etc/sysconfig/selinux 을 열어서 SELINUX=enforcing 항목을 수정하면 되며 재부팅해야 반영된다.

#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.
SELINUX=enforcing
# SELINUXTYPE= can take one of three two values:
#       targeted - Targeted processes are protected,
#       minimum - Modification of targeted policy. Only selected processes are protected.
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

- 부팅시에 SELinux 를 해제하려면 위의 설정파일에서 SELINUX=disabled 로 변경하면된다.

SELinux boolean 이란?

- SElinux 는 보안 정책 설정 편의를 위해 사전에 정의된 규칙 집합들을 갖고 있으며 이 규칙들을 SELinux 불린(Boolean) 이라고 한다.
- 예로 아파치 httpd 가 cgi 를 실행할 수 있게 하려면 여러 가지 보안 컨텍스트 설정을 해야 하지만 이런 번거로움을 없애기 위해 이런 과정을 묶어서 httpd_enable_cgi 라는 불린을 제공하고 있으며 true 로 설정하면 SELinux 는 아파치 웹 서버의 cgi 실행을 허용한다.
- 보안을 위해 대부분의 불린 설정은 기본 값이 off 이며 현재 설정을 확인하려면 getsebool 명령어를 사용하면 되며 확인하려는 불린명을 주거나 -a 옵션을 줄 경우 모든 불린의 설정 현황을 표시한다.
- 아래 명령은 아파치 httpd 가 SMTP 메일 서버에 연결할 수 있는지를 나타내는 불린인 httpd_can_sendmail 의 설정을 확인한다.

```
# getsebool httpd_can_sendmail
httpd_can_sendmail --> off
```

- 또는 아래와 같이 모든 불린을 출력한 후에 egrep 과 정규식을 활용하여 찾는 방법도 있다.

```
# getsebool -a | egrep "httpd(.*)sendmail"
httpd_can_sendmail --> off
```

SELinux boolean 설정

- httpd 서버가 메일 서버에 접속할 수 있도록 httpd_can_sendmail 불린을 on 으로 설정

```
# setsebool httpd_can_sendmail true
# getsebool httpd_can_sendmail
httpd_can_sendmail --> on
```