1.1. SELinux의 이해

Q: SELinux란?
A: 휘도라 코어(Fedora Core)의 SELinux(*Security-Enhanced Linux*)란 리눅스 보안 모듈 구조체(Linux Security Modules(LSM) framework)를 이용하여 리눅스 커널에 의무 접근 제어(Mandatory Access Control - MAC)를 구현하는 것이다. 표준 리눅스 보안(Standard Linux Security)은 자유재량 접근 제어(*Discretionary Access Control* – DAC) 모델이다. DAC 모델에서, 파일과 자원에 대한 결정권은 오직 해당 객체(objects)의 사용자(user id)에게 있고 소유권(ownership)에 따라 이뤄진다. 각 사용자와 그 사용자에 의해 실행된 프로그램은 자기에게 할당된 객체에 대해 전적으로 자유재량권을 갖는다. 이러한 상황에서는, 악의 있는 일반 혹은 루트 사용자(예로, setuid와 setgid)가 실행시킨 결함이 있는 소프트웨어를 통해 주어진 객체로 원하는 어떠한 일을 해도 막아낼 방법이 없으며 보안 정책을 시스템 전체에 걸쳐 시행되도록 할 방법이 없다.

MAC 시스템은 위와 같은 빠져있는 요소들을 제공한다. 첫 째, 보안 정책을 모든 프로세스나 객체에 대하여 관리차원으로 규정 지을 수 있다. 둘 째, 커널에 SELinux를 구현하면, 모든 프로세스와 객체를 제어할 수 있다. 셋 째, 결정은 단지 인증된 사용자(user identity)에 의해서가 아니라 이용 가능한(available) 모든 보안 관련 정보에 근거하여 이뤄진다.

SELinux하에서 MAC는 모든 주체(subjects – 사용자, 프로그램, 프로세스)와 객체(파일, 디바이스)에 대해서 국부적으로 허가(granular permissions)해 줄 수 있다. 응용프로그램에서 불필요한 부분은 제외하고 오직 필요한 기능에 대해서만 사용 권한을 안전하게 부여한다.

SELinux 구현은 역할과 유형 시행(Type Enforcement - TE)에 기초하여 추상적 사용자 수준 제어(abstracted user-level control)를 제공하는 역할 기반 접근 제어(role-based access control – RBAC)를 사용한다. TE는 접근 제어를 처리하기 위해서 테이블(매트릭스)을 이용한다. 주체는 영역(domain)을 갖고 객체는 유형을 갖으므로, 매트릭스에서 교차조회하여 이들의 상호작용을 규정한다. 이는 리눅스 시스템에 있는 모든 동작자(actor)에 대하여 극단적으로 국부 제어를 가능케 한다.

Q: SELinux 정책이란?
A: SELinux 정책은 사용자, 프로그램, 프로세스 그리고 이들의 동작 대상인 파일과 디바이스를 포함한 시스템 전체, 즉, 모든 주체와 객체에 대한 접근 허가(access permissions)를 기술한다. 휘도라 코아 정책은 관련 소스 패키지와 함께 패키지로 공급된다. 현재 공급되는 정책 패키지는 다음과 같다.

- selinux-policy-strict-*<version-arch>*.rpm and selinux-policy-strict-sources-*<version-arch>*.rpm
- selinux-policy-targeted-*<version-arch>*.rpm and selinux-policy-targeted-sources-*<version-arch>*.rpm

설치시 정책 소스는 /etc/selinux/*policyname*/src/policy에, 바이너리 정책 파일은 /etc/selinux/*policyname*/policy에 놓인다. 정책 소스는 최소 설치(ultra-minimal installations)에 포함되지 않는다. 유형과 영역에 대한 정책은 주체와 객체에 대한 보안 문맥과 구분해서 별도로 설정된다.

Q: SELinux 목표 정책(SELinux targeted policy)이란?
A: 처음 SELinux가 휘도라 코아에 포함됐을 때, NSA의 엄격한 정책이 시행됐었다. 시험 목적으로, 이는 그 엄격한 정책을 통해서 수백가지 문제점을 찾아낼 수 있었다. 뿐만 아니라, 휘도라 사용자들의 다양한 환경에 단 하나의 엄격한 정책을 적용한다는 것은 실효성이 없다는 것이 명백히 드러났다. 기본 설치 이외의 것에 대해서 단일 엄격한 정책을 적용하는 것은 현지 일반 사용자에게 전문 지식(기술)을 요구하는 것이었다.

이 시점에서, SELinux 개발자들은 사용자들이 선택한 내용을 검토하고, 다른 전략을 시도하기로 결정했다. 특정 데몬들, 특히, 깨지거나 손상(오염)되면 시스템을 황폐화시키거나 공격받기 쉬운 것들을 옥죄는(lock down) 데 초점을 맞추는 정책을 만들기로 했다. 시스템의 나머지 부분은 SELinux가 활성(enabled)이든 아니든 관계없이 똑같이 가동되도록, 즉, 마치 표준 리눅스 보안하에 운영되고 있는 것처럼 동작하도록 허용한다.

목표 정책 하에서, 대부분의 프로세스는 unconfined_t domain(무제한 영역)에서 가동된다. 그 이름이 의미하는 바와 같이, 이 프로세스들은 거의 SELinux 정책에 의한 제한을 받지 안는다. 그러나, 그 프로세스들은 여전히 표준 리눅스/유닉스 보안에 의해 통제된다.

특정 네트워크 데몬들은 전용 정책을 갖고 있어, 응용프로그램이 시작될 때,

무제한 정책(unconfined_t policy)이 전용 정책으로 전이된다. 예를 들면, 시스템 부팅시, init는 무제한 정책하에서 가동하지만, named가 시작할 때, named 영역으로 전이되고 적시에 적절한 정책에 의해 옥죄어진다.

각 구체적인 데몬에 대한 목표 정책을 활성화 혹은 비활성화 시키는 것에 관하여 더 알고 싶으면, system-config-securitylevel의 사용법을 참조하기 바란다.

Q: 어떤 데몬이 목표 정책에 의해 보호받는가?
A: 현재, 데몬 일람표에는 dhcpd, httpd(apache.te), named, nscd, ntpd, portmap, snmpd, squid 그리고 syslogd가 있다. 이 데몬들에 대한 정책 파일은 /etc/selinux/targeted/src/policy/domains/program에서 찾을 수 있다.

향후, 더 많은 데몬들이 목표 정책 보호(targeted policy protection)에 추가될 것이다.

Q: 어느 데몬을 목표 정책에 추가할 것인가? Sendmail, Postfix, MySQL, 혹은 PostgreSQL?
A: SELinux 개발자는 긍극적으로 ftp와 메일 에이젼트를 목표 정책에 추가하고자 한다. 예를 들면, vsftpd는 로그인과 비슷하게 동작한다. 즉, 로그인 후 새로운 프로세스가 사용자 문맥(실사용자 혹은 무명씨 문맥)하에 실행된다.

메일 에이젼트가 안고 있는 문제중 하나는 메일 에이젼트는 사용자 홈 디렉토리에 있는 파일을 조작해야 하는 일이 자주 생기는 것이다. 목표 정책의 목적중 하나는 사용자 홈 디렉토리에서 문제들을 분류하지 않는 것이다. 이 생각은 아직도 변함이 없다.

Q: 엄격한 정책은 어떤가? 이 정책 조차도 유효한가?(Does it even work?)
A: 엄격한 정책은 휘도라 코아에서 분명히 적용되고 있다. 이는 다른 사용자의 독특한 환경에 의해 문제를 야기시킬 수 있다. 엄격한 정책이 적절하게 운영되려면, 정책과 시스템 둘 다 미세조정(tweak)해야 할 지도 모른다.

엄격한 정책을 사용하는 데 좀더 용이하도록 하려고, SELinux 개발자들은 정책 변경을 더 쉽게 할 수 있도록 노력해왔다. 예를 들면, **system-config-securitylevel**은 재명명하기(relabel)를 시동 스크립트안에 갖고 있다(build).

Q: 파일 문맥이란?
A: 파일 문맥은 파일이나 디레토리의 보안 문맥을 기술하는 항구적 라벨(꼬리표)(persistent labels)을 만들기 위해 setfiles 명령에 의해 사용된다.

휘도라 코아는 검사, 복구, 재명명(check, restrore, and relabel) 등, 이상 세가지 옵션을 지원하는 명령 스크립트 fixfiles를 갖고 있다. 이는 사용자가 selinux-policy-targeted-sources 패키지를 설치하지 않고도 파일 시스템을 재명명(relabel)할 수 있도록 해준다. 명령 라인 사용법은 표준 setfiles 명령보다 더 친숙하다.

Q: 파일, 사용자, 프로세스의 보안 문맥을 확인하는 방법은?
A: 새 옵션 –Z는 주체나 객체의 문맥을 표시하기 위한 손쉬운 방법이다:

```
ls -alZ file.foo
id -Z
ps -eZ
```

Q: 영역과 유형은 어떻게 다른가?
A: 비록 영역이 프로세스의 유형으로 자주 언급될 지라도, 영역과 유형 사이에는 차이점이 없다. 이렇게 볼 때, 영역의 사용은 영역과 유형을 구분하는 전통적 TE 모델에 기인한다.

1.2. SELinux 제어하기

Q: SELinux 설치 및 설치 안하는 방법은?
A: 방화벽 설정 화면에서 선택한 사항에 기초하여 설치자(installer)가 처리한다. 기본 가동 정책(default running policy)은 목표 정책이며, 기본으로(by default) 가동된다.

Q: 사용중인 정책을 교체하는 방법은?
A:

정책 교체는 가볍게 취할 사안이 아니다.
연구 목적으로 시험 장비(test machine)에서 새 정책을 시도하는 이외, 생산 시스템(production system)에서는 다른 정책으로 교체하기 전에 현황을 심각하게 고려해야 한다.

교체 작업은 간단하다. 이는 매우 안전한 방법이지만, 우선 시험 시스템에서 일차 시도해 보는 것이 바람직하다.

한 가지 방법은 system-config-securitylevel을 사용하여 정책을 바꾸고 재명명(relabel)하도록 파일 시스템을 설정하는 것이다.

수작업 절차는 다음과 같다:

1. /etc/selinux/config을 편집하고 SELINUXTYPE=*policyname*으로 정책 유형을 바꾼다.
2. 재부팅하여 돌아올 수 있는 지 확인하기위해, SELINUX=permissive모드로 설정한다. 이렇게 하면, SELinux는 정확한 정책하에서 가동될 것이지만, 만일 부정확한 파일 문맥 명명(labeling)과 같은 문제가 있으면 로그인하도록 할 것이다.
3. sysadm_r 역할을 갖춘 root로 파일 시스템을 재명명한다(relabel):

   ```
   id -Z
   root:sysadm_r:sysadm_t
   fixfiles relabel
   ```

   옵션 -l */path/to/logfile*을 사용하여 표준 출력으로 로그를 볼 수 있고, 옵션 -o */path/to/file*을 사용하여 검토(checked)되거나 재명명(relabeled)된 모든 파일 리스트를 저장할 수 있다.

4. 시스템을 재부팅한다. 새 정책하에서의 재시작은 모든 시스템 프로세스가 적절한 문맥에서 시작되고 정책 변경으로 인한 모든 문제가 드러나게 한다.
5. sestatus –v 명령으로 발효된 변경사항을 확인한다. Permissive 모드로 가동된 새 시스템에서, avc: denied 메시지를 /var/log/messages 에서 확인한다. 이들은 새 정책하에 문제없이 시스템이 가동되도록 해결해 야 할 문제들을 표시해 준다.

6. 새 정책하에서 시스템이 만족스럽게 돌아갈 때, SELINUX=enforcing 으로 바꿔 실행 권한을 부여한다. 실시간에 enforcing 을 활성화 시키기 위해 재부팅하거나 **setenforce 1** 을 실행한다.

Q: 목표 정책하에서 특정 데몬에 대하여 SELinux protection 을 활성화/비활성화 방법은?

A: 특정 데몬의 부울 값을 제어하려면 **system-config-securitylevel** 을 사용한다. 예를 들어, 만일 apache 가 시스템상에서 올바르게 동작하도록 하기위해서는 SELinux 를 비활성화시켜야 한다면, **system-config-securitylevel** 으로 해당 값을 비활성화시키면 된다. 이는 apache.te 에 정의된 정책으로 전이되는 것을 막고 httpd 가 일반 리눅스 보안하에 있도록 한다.

Q: 부팅시 SELinux 를 비활성화시키는 방법은?

A: 커널 명령 라인에 selinux=0 를 추가한다.

SELinux 를 비활성화시킬 때 주의

이 옵션을 사용할 때는 매우 주의해야 한다. 만일 selinux=0 로 부팅하면, SELinux 가 비활성인 동안 만든 모든 파일은 SELinux 문맥 정보를 갖고 있지 않을 것이다. 적어도 파일 시스템을 재명명(relabel)해야 할지도 모르고, 복구를 위해 단일 사용자 모드로 부팅할 때 필요한 selinux=1 로 부팅이 안될 수도 있다.

Selinux=0 의 대안으로 /etc/selinux/config 의 SELINUX=disabled 을 사용하다.

Q: 부팅시 enforcing 을 활성화/비활성화하는 방법은?

A: /etc/sysconfig/selinux 설정 파일을 이용하여 SELinux 모드를 지정할 수 있다.

```
# 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= type of policy in use. Possible values are:
```

```
#       targeted - Only targeted network daemons are protected.
#       strict - Full SELinux protection.
SELINUXTYPE=targeted
```

enforcing 에 값을 지정하는 것은 enforcing 을 활성화하기 위해 커널을 부팅할 때 명령 라인에 enforcing=1 을 추가하는 것과 같다. 반면에, permissive 에 값을 지정하는 것은 enforcing 을 비활성화하기 위해 enforcing=0 를 추가하는 것과 같다. 명령 라인 커널 인자는 설정파일에 우선한다는 것을 명심하자(Note that the **command line kernel parameter overrides the configuration file**.).

그러나, disabled 에 값을 지정하는 것은 커널 부트 인자 selinux=0 을 사용하는 것과 다르다. 커널에서 SELinux 를 완전하게 비활성화시키는 것과 달리 대신 disabled 를 설정하는 것은 enforcing 을 비활성화하고 정책 적재를 하지않고 건너뛰는 것이다.

Q: 재부팅하지 않고 enforcing 모드를 임시로 비활성화시키는 방법은?
A: 이 상황은 보통 정책에 의해 금지된 동작를 수행할 수 없을 때 발생한다. 실시간으로 enforcing 모드를 비활성화시키기 위해서 sentenforce 0 명령을 실행한다. 작업이 끝나서 enforcing 모드로 돌아오기 위해서는 sentenforce 1 을 실행하면 된다.

sysadm_r 권한(역할) 필수
sentenforce 명령은 sysadm_r 권한을 갖고 수행해야 한다; 그러기 위해, **newrole** 명령을 사용하거나, 아니면, **su -**를 사용하여 root 로 사용자 전환을 하면, 자동으로 sysadm_r 권한을 얻을 수 있다.

Q: 부팅시 시스템콜 auditing 키거나 끄는 방법은?
A: 시스템콜 auditing 을 키기위해서는 audit=1 을 커널 명령 라인에 추가한다. 끄기위해서는 audit=0 을 추가한다.

시스템콜 auditing 은 기본적으로 꺼져있다. 켜지면, SELinux 가 denied(거부) 메시지가 발생했을 때, 실행중이던 시스템콜에관한 정보를 제공한다. 이는 정책을 디버깅할 때 유용하다.

Q: 재부팅을 하지않고 시스템콜 auditing 을 잠정적으로 끄는 방법은?
A: 현재는 지원되지 않고 있다. 향후, auditing 을 조정할 수 있는 유틸리티(utility)가 제공될 것이다.

Q: SELinux 설치에 관한 상태 정보(status info)를 얻는 방법은?
A: root 사용자 권한으로 **/usr/sbin/setstatus -v** 명령을 실행하라. 더 많은 정보가 필요하면, 매뉴얼 페이지 setstatus(8)을 참조하라.

1.3. 문제 해결하기

Q: 응용프로그램이 기대했던 데로 동작하지않고 avc: denied 란 메시지가 나타나면, 어떻게 이 문제를 해결하는 가?
A: 이 메시지는 현재 실행된 SELinux 정책이 그 응용프로그램의 동작을 허락하지 않기 때문이다. 이러한 일에는 여러 가지 사유가 존재한다.

첫째, 응용프로그램이 접근하려는 파일중 하나가 잘못 명명되어있을 수 있다. 만일 AVC 메시지가 특정 파일을 참조한다면, **ls -alZ** ***/path/to/file*** 을 수행하여 현재 참조하는 파일명(current label)을 조사해 보라. 만일 그것이 잘못되어 보이면, **restorecon -v** ***/path/to/file*** 을 시도해보라. 만일 파일과 관련된 매우 많은 거부(denials) 상황이 존재하면, **fixfiles relabel** 을 수행하거나, 반복적으로 디렉토리 경로를 재명명하기 위해서 -R 옵션과 함께 **restorecon** 을 수행하고 싶을 수 있다.

다른 때에는, 거부(denials) 현상은 정책에 의해 거부되도록 프로그램에 설정을 바꿔서 발생될 수 있다. 예를 들면, 만일 Apache 를 8800 포트로 바꾸면, 보안 정책, apache.te,도 관련하여 바꿔야 할 필요가 생긴다. 정책 작성에 관한 상세한 정보가 필요하면, 외부연결 리스트(External Link List)를 보라.

만일 작동중인 Apache 와 같이 특정 응용프로그램을 구하는 데 어려움이 있으면, 해당 응용프로그램에 enforcement 를 비활성화(disable)시키는 방법에 관하여 How to use system-config-securitylevel 를 보라.

Q: 기존 /home 파티션이 있는 시스템에 휘도라 코아를 설치하고 로그인을 할 수 없을 때 해결방법은?
A: /home 파티션이 정확하게 명명(label)되지 않았다. 두 가지 방법으로 이 문제를 쉽게 해결할 수 있다.

만일 /home 을 반복적으로 재명명(relabel)하고자 한다면:

```
/sbin/restorecon -v -R /home
```

만일 잘못 명명된 기타 파일이 없다고 확인하고자 한다면, 전체 파일 시스템을 재명명할 수 있다.

```
/sbin/fixfiles relabel
```

fixfiles 를 사용하여 policycoreutils 패키지를 설치할 필요가 있을 것이다.

Q: setfiles 나 fixfiles 를 사용하여 /home 을 재명명한 후, 여전히 SELinux 비활성 시스템(non-SELinux-enabled system)으로 /home 을 읽을 수 있을까?
A: 비 SELinux 배포판(non-SELinux distribution)의 파일이나 SELinux 가 비활성화(disabled)된 파일을 읽을 수 있다. 그러나, SELinux 비활성화 시스템에 의해 생성된 파일 또는 삭제나 재생성된 어느 파일도 보안 문맥을 갖고 있지 않을 것이다. 이는 ~/.bashrc 와 같은 파일에서 문제가 생길 수 있다. 휘도라 코아로 다시 돌아갈 때 /home 을 재명명해야 할 지도 모른다.

Q: 휘도라 코아와 비 SELinux 시스템간에 NFS 를 사용하여 디렉토리를 공유하는 방법은?
A: NFS 가 많은 파일 시스템을 명확하게 지원하기 때문에, SELinux 와 비 SELinux 시스템간에 디렉토리를 공유하는데 NFS 를 사용될 수 있다.

NFS 를 통해서 비 SELinux 파일 시스템을 마운트할 때, 기본적으로 SELinux 는 공유영역에 있는 모든 파일을 nfs_t 의 문맥을 갖고 있는 것으로 간주한다. Context=옵션 을 사용하여 수작업으로 그 값을 설정하여 기본문맥(default context)을 덮어쓸 수 있다. 예를 들면, NFS 로 마운트된 디렉토리의 파일들을 SELinux 에 대한 system_u:object_r:tmp_t 의 문맥을

갖고 있는 것처럼 보이게 할 것이다(appear to have a context of system_u:object_r:tmp_t to SELinux).

```
mount -t nfs -o context=system_u:object_r:tmp_t server:/shared/foo /mnt/foo
```

SELinux 가 파일 시스템을 NFS 를 경유하여 엑스포트(export)할 때, 생성된 파일은 자기가 만들어진 디레토리의 문맥을 갖을 것이다. 다시 말해서, 원격으로 마운트된 시스템(remote mounting system)의 SELinux 의 출현(presence)은 현지 보안 문맥(local security contexts)에 영향을 끼치지 않는다.

Q: 적절한 문맥을 갖는 사용자 홈 디렉토리와 함께 새 리눅스 사용자 계정 을 만드는 방법은?

A: 표준 useradd 명령을 사용하여 사용자 계정을 새로 만들 수 있지만, 우선 sysadm_r 의 문맥을 갖는 루트 사용자가 되어야 한다. 이 문맥 교환은 **su** 명령에 통합되어 있어 자동으로 일어난다.

```
su - root
id -Z
root:sysadm_r:sysadm_t
useradd auser
ls -Z /home
drwx------  auser  auser  root:object_r:user_home_dir_t /home/auser
```

새 사용자 디렉토리에 대한 초기 문맥은 루트의 정체성(identity)을 갖는다. 이후 파일시스템의 재명명은 그 정체성을 system_u 로 바꾼다. 이들은 역할(권한)과 유형이 동일하기 때문에 기능적으로 서로 같다 (object_r:user_home_dir_t).

Q: 다른 모든 SELinux 문서는 su 이 단지 리눅스 정체성만을 바꾸고 보안 역할(security role)은 아니라고 말한다.

A: 휘도라 개발팀은 기존 SELinux 관례(existing SELinux practice)와는 조금 다른 방향을 취했다. 보안 문맥 전이는 이제 pam_selinux 에 의하여 **su** 에 통합되었다. 이는 시스템 사용을 크게 단순화시켰다. 실제로, 이는 전통전인 **su** 와 SELinux **newrole** 을 두 단계대신 한 단계로 조합한 것과 같다.

setuid(2)와 같이 다른 형식의 Linux/UNIXR 정체성 변화는 SELinux 정체성 변화를 일으키지 않는다.

Q: 특정 프로그램에 대한 로그를 채우는 avc 오류에 골치를 앓고 있다. 어떻게 하면 그것에 대한 접근을 감시하지(audit) 않도록 선택할 수 있는가?
A: 예를 들어, 만일 dmesg 를 감시하지 않길 원했다면, /etc/selinux/targeted/src/policy/dmesg.te 파일에 다음사항을 넣으면 된다.

```
dontaudit dmesg_t userdomain:fd { use };
```

이는 모든 유저영역(user, staff 과 sysadm)에 대하여 터미널로 출력된 오류를 제거한다.

Q: 비록 허가 모드에서 운행될 지라도(even running in permissive mode), 많은 수의 avc denied 메시지가 나타난다.
A: 비 enforcing 모드에서는 실제 enforcing 모드에 있을 때보다 더 많은 메시지를 받는다. 이는 이전에 enforcing 모드에 있었던 것처럼 커널이 매 접근 거부를 기록하기 때문에 일어난다. 제한을 받지 않기 때문에, 더 많은 실행을 수행할 수 있고, 이는 더 많은 거부가 기록되는 결과를 초래한다.

Q: 오직 SELinux 가 enforcing 모드에 있을 때만 특정 허가 거부를 받지만, /var/log/messages 에는 어떠한 감시 메시지를 볼 수가 없다. 이들 침묵의 거부(silent denials)의 원인을 밝혀낼(identify) 수 있는 방법은?
A: 침묵의 거부에 대한 가장 공통적인 이유는 정책에 감시 메시지를 억제하도록 명시적 dontaudit 규칙을 내포하고 있기 때문이다. dontaudit 규칙은 은근한 거부가 감시 기록을 채울 때 이러한 방법으로 종종 사용된다.

특정 거부를 찾으려면, 모든 dontaudit 규칙의 감시(auditing)를 활성화시킬 필요가 있다.

```
cd /etc/selinux/targeted/src/policy make enableaudit
make load
```

활성화된 dontaudit 출력은 장황하다
모든 dontaudit 규칙의 감시(auditing)를 활성화하는 것은 다량의 감시 정보를 생산해낼 가능성이 높고, 그 대부분은 찾고자하는 거부와 무관하다.

은근하게 발생하는 거부에 대한 감시 메시지를 구체적으로 찾고자 할 때만 이 기술을 사용하라. 십중팔구(아마) 가능한한 빨리 dontaudit 규칙을 재활성화시키길 원할 것이다.

dontaudit 규칙을 재활성화하려면, 다음을 수행하라.

```
cd /etc/selinux/targeted/src/policy
make clean
make load
```

Q: 정책 패키지를 업그레이드할 때(예를 들어, yum 을 사용하여), 기존 정책 에 어떤 일이 발생하는가; 자동으로 업데이트되는가?
A: 패키지가 업데이트될 때 정책은 원래데로 재 적재된다. 이는 수작업 **make load** 를 대체한다.

어떤 상황에서, 파일 시스템을 재명명할 필요가 있을 수 있다. 이는 파일 문맥이 실효성이 없게 된 곳에서 SELinux 버그 수정 과정의 일 부분으로 혹은 정책 업데이트가 file_contexts 으로 변경될 때 발생될 수 있다.

파일 시스템을 재명명한 후, 재부팅이 꼭 필요한 것은 아니지만 모든 프로세스와 프로그램이 적절한 영역에서 수행되고 있는 지를 확인하는 데 필요하다. 이는 업데이트된 정책의 변경사항에 크게 영향을 받는다.

만일 정책 소스 패키지(예, selinux-policy-strict)를 설치했다면, 파일 시스템을 재명명하기 위해 다음 명령을 수행한다.

```
cd /etc/selinux/targeted/src/policy
make
make relabel
reboot
```

만일 정책 소스를 사용하지 않는다면, 다른 접근법은 **fixfiles** 명령을 사용하는 것이다.

fixfiles relabel
reboot

Q: 만일 응용프로그램 패키지와 함께 보급되는 정책이 재명명이 필요한 방법으로 변경된다면, RPM 은 패키지가 소유한 파일들을 재명명을 처리할 것인가?
A: 그렇다. 해당 패키지가 소유한 파일에 대한 보안 문맥은 그 패키지의 헤더 데이터(header data)에 저장되있다. 그 파일 문맥은 패키지 파일이 디스크에 놓여 있으므로 **cpio** 복사가 된 후 즉시(directly) 설정된다.

Q: 정책과 정책 소스 패키지사이에 어떤 관계가 있는가?
A: selinux-policy-targeted 와 같은 정책 패키지는 즉시 적용되는(working) SELinux 설치에 필요 조건인 반면 selinux-policy-targeted-sources 와 같은 정책 소스 패키지는 기본 정책을 커스터마이즈(customize)하고자 한다면 필요하다.

정책 패키지는 SELinux 보안 정책을 정의하는 데 필요한 최소한의 파일을 갖고 있다. 최소한의 설치 청사진을 지원하기 위해 크기를 줄였다.

정책 소스 패키지는 /etc/selinux/*policyname*/contexts/*과 /etc/selinux/*policyname*/policy/policy 를 만든 데 필요한 /etc/selinux/*policyname*/src/policy 에 소스 정의를 포함하고 있다. *Version* 은 그 정책의 버전 번호이다.

설치할 패키지 선택은 설치 유형에 달려있다. 만일 휘도라 코아 개발자가 정의한 기본 보안 정책 만을 사용하려면, 정책 패키지를 설치만 하면 된다. 어떤 방법이든 보안 정책을 커스터마이즈하거나 아니면 **setools** 를 실행하려면, 정책 소스를 설치해야 한다.

정책 패키지를 설치하거나 업데이트하면 파일을 설치한 후 새 정책을 적재한다. 마찬가지로, 정책 소스 패키지를 설치하거나 업데이트하는 것은 policy.*version* 파일 뿐만 아니라 file_contexts 파일을 재 구축하고 나서 현재 실효성 있는 정책으로 그 파일을 적재하는 것이다.

Q: 왜 /etc/selinux/*policyname*/policy/policy.<*version*>과 /etc/selinux/*policyname*/src/policy/policy.<*version*> 파일은 다른 (크기, md5sums, 날짜)를 갖는가?
A: 정책 패키지를 설치할 때, 미리 컴파일된 바이너리 정책 파일이 /etc/selinux 에 직접 들어간다. 정책 소스 패키지가 설치되거나 업데이트될 때, 바이너리 정책 파일이 /etc/selinux/*policyname*/src/policy 에 구축되고, 다시 /etc/selinux/*policyname*/policy/로 이동한다. 다른 구축 환경은 다른 크기, md5sums 와 날짜를 갖는 목표 파일을 만들게 된다.

Q: 새로운 정책 패키지가 시스템을 비활성화시킬(disable) 것인가?
A: 정책 패키지나 응용프로그램 패키지와 함께 배포되는 정책의 변경은 오류, 더 많은 거부 혹은 기타 원인 모를 동작을 유발시킬 가능성이 있다. 문제를 해결하기 위해, 한번에 하나씩 정책과 응용프로그램 패키지를 되돌려가면서 어느 패키지가 문제를 일으키는지 찾아낼 수 있다. 이전 패키지로 되돌리지 않으려면, 구버전 설정파일에 .rpmsave 확장자(꼬리표)를 붙 여 저장될 것이다. 문제 해결의 도움을 받으려면, 메일링 리스트, 버그질라, 그리고 IRC 를 활용하도록 한다. 능력이 있으면, 스스로 정책을 작성하거나 고쳐 문제를 해결할 수 있다.

Q: 정책 작성에 도움을 줄 수 있는 방법은?
A: 도움을 기꺼이 받아들일 것이다. SELinux 메일링 리스트, fedora-selinux-list@redhat.com 에 가입하는 것으로 시작할 수 있다; http://www.redhat.com/mailman/listinfo/fedora-selinux-list 에 등록하고 관련 문서를 읽을 수 있다. 비공식 FAQ 는 HOWTO 정보를 작성하는 일반 정책 (http://sourceforge.net/docman/display_doc.php?docid=14882&group_id=21266#BSP.1) 을 갖고 있다. 또 다른 새 자원은 기록하는 SELinux 정책 HOWTO(Writing SE Linux policy HOWTO)이다 (https://sourceforge.net/docman/display_doc.php?docid=21959&group_id=21266).

/etc/selinux/*policyname*/src/policy/에 있는 정책 파일을 보고 시험해보는 것이 상책이다. 실마리를 찾기 위해 /var/log/messages 에 있는 avc denied 메시지를 관찰하라.

정책 작성자에게 유용한 도구는 **/usr/bin/audit2allow** 인데 이것은 /var/log/messages 의 avc 메시지를 SELinux 에 의해 사용될 수 있는 규칙으로 번역해준다. 이 규칙들은 십중팔구 정리되어야 할 것이다.

audit2allow 명령은 세가지 방법으로 입력을 받을 수 있다. 기본은 표준입력(stdin)이다. -i 옵션을 사용하면 /var/log/messages 로부터 입력을 읽을 수 있고 -d 옵션을 사용하면 **dmesg** 출력으로부터 입력을 읽을 수 있다.

Q: 콘솔에 메시지가 넘쳐나고 있다. 어떻게 이것을 막아낼 수 있는가?
A: 유용한 제어를 되찾기 위해서는 **dmesg -n 1** 를 사용하여 콘솔로 나가는 커널 메시지를 끈다.

Q: 정책 소스를 설치하지 않고 기본 정책을 시험할 수 있는가?
A: 단지 selinux-policy-*policyname* 과 policycoreutils 패키지를 설치하면 SELinux 기본 정책을 시험할 수 있다. 정책 소스를 설치하지 않았으면, **fixfiles** 명령은 자동으로 파일 시스템 재명명을 처리해준다.

fixfiles relabel 은 **make relabel** 와 동일하다. 재명명 동안, /tmp 에 있는 모든 파일을 삭제하여, 구 파일 문맥 라벨을 갖고 있는 파일들을 정리한다.

기타 명령으로는 잘못 명명된 파일을 검사하는 **fixfiles check** 이 있고 잘못 명명된 파일을 고치지만 /tmp 에 있는 파일을 삭제하지 않는 **fixfiles restore** 이 있다. **fixfiles** 는 전 파일시스템을 재명명하는 것이 목적이므로 인자로서 디렉토리의 리스트를 취하지 않는다. 특정 디렉토리 경로를 재명명하려면, **restorecon** 을 사용하라.

Q: KDE 응용프로그램을 SELinux 하에서 작동하는 데 문제가 있다.
A: KDE 실행파일은 항상 **kdeinit** 으로 나타난다. 그런데 이것은 SELinux 정책으로 실행될 수 있는 것을 제한한다. 이는 모든 KDE 응용프로그램이 **kdeinit** 의 영역에서 실행되기 때문이다.

/tmp 와 /var/tmp 를 재명명할 수 없기 때문에 SELinux 를 설치할 때 문제가 종종 발생한다. 어느 파일이 어느 문맥을 갖어야 하는지 결정하는 좋은 방법은 없다.

해결책은 KDE 를 완전히 로그아웃하고(log out) 모든 KDE 임시 파일을 삭제하는 것이다.

rm -rf /var/tmp/kdecache- ***<username>***
rm –rf /var/tmp/ ***<other_kde_files>***

다음 로그인에서 문제가 고쳐져야 한다.

Q: SELinux=disabled 가 작용하지 않는 이유는?
A: /etc/sysconfig/selinux
파일에 있는 화잍 스페이스를 주의하라. 코드는 비록 끝에 붙은 스페이스일지라도 화잍 스페이스에 매우 민감하다.

1.4. SELinux 배치하기

Q: SELinux 를 사용하기 위해 어떤 파일 시스템을 선택해야 하는가?
A: 파일 시스템이 올바른 *security.** 이름공간(namespace)에 xattr 라벨를 지원해야한다. ext2/ext3 뿐 아니라 필요한 라벨을 지원하는 XFS 가 최근 추가되었다.

Q: SELinux 는 시스템 성능에 어떠한 영향을 미치는가?
A: 이것은 측정하기 힘든 변수이고 SELinux 가 실행되고 있는 시스템의 규모에 크게 의존한다. 성능이 마지막으로 측정됐을 때, 그 영향은 완벽하게 조정이 안된 코드(for completely untuned code)의 약 7%였다. 그 이후의 변경은 네크워크와 같이 어떤 경우에 더 악화시킬 가능성이 있다. SELinux 성능 튜닝은 개발팀이 최우선 과제이다.

Q: SELinux 를 효과를 제고하기위해(to leverage SELinx in) 첫째로 보아야 할 것은 어떤 유형의 배치/응용프로그램/시스템 등 인가(What types of deployments/applications/systems, etc.)?
A: 초기에는, SELinux 는 거의 특별한 기능을 수행하지 않지만 극도로 엄격한 보안을 유지하는 것이 중대한 인터넷에 접한 서버에서 동작할 것이다. 그러한 시스템은 전형적으로 모든 특별한 소프트웨어와 서비스를 제외시키고 웹 서버나 이메일 서버와 같은 매우 집중화된 서비스나 서비스 집합을 운영한다.

이러한 에지 서버(edge servers)에는 정책을 매우 엄격하게 적용할 수 있다(lock down). 이것은 다른 성분(components)과의 더 작은 수의 상호작용으

로 더 쉽게 이뤄진다. 마찬가지로, 특별한 제삼 응용프로그램을 운영하는 전용 박스가 좋은 후보이다.

향후, SELinux 는 모든 환경을 목표를 둘 것이다. 그곳에 도달하기 위해서, 공동체와 ISVs(independent software vendors)은 필요한 정책을 생산하기 위해 SELinux 와 협력할 필요가 있다. 지금까지, 특정 공격받기 쉬운 데몬에 초점을 맞춘 목표 정책뿐 아니라, 매우 제한적인 엄격한 정책이 쓰여졌다.

Q: 어떻게 SELinux 가 제삼 응용프로그램에 영향을 끼치는가?
A: 휘도라 코아에서 목표 SELinux 정책을 구현하는 한가지 목적은 제삼 응용프로그램이 수정없이 동작할 수 있도록 하는 것이다. 이것은 목표 정책이 그 응용프로그램에 대해 투명하므로(transparent) 제어를 하려하지 않고 실질적으로 표준 리눅스 보안에 의지한다(This works because the targeted policy is transparent to those applications it is not trying to control that it essentially falls back on standard Linux security.). 이 응용프로그램들은 특별한 보안 방법(extra-secure manner)으로 실행되진 않을 것이다. 정책은 이 응용프로그램이 MAC 에 의해 보호되도록 작성될 필요가 있다.

아주 많은 변형의 제삼 응용프로그램이 존재하고 이들이 SELinux, 심지어 목표 정책을 기동할 지라도 서로 어떻게 작용하는가(behave with)를 예측하는 것은 불가능하다. 발생된 문제(issues that arise)은 때때로 정책에서 해결될 수 있다. 응용 프로그램을 SELinux 하에서 동작하도록 수정될 필요가 있는, 미처 알지 못했던 응용 프로그램과 관련된 보안 문제를 SELinux 가 보여줄 수 있다.

휘도라 코아 시험자와 사용자가 공동체에 가져오는 한가지 중요한 가치는 제삼 응용프로그램의 광대한 시험이다. 그러한 마음가짐으로, 토론을 하기 위해 적절한 메일링 리스트(fedora-selinux-list@redhat.com)에 여러분의 경험을 보내주실 것을 바란다.