



13 SELinux-Anwendung

In diesem Kapitel werden wir die ersten Schritte der Installation und Inbetriebnahme von SELinux unter verschiedenen Distributionen betrachten. Anschließend stelle ich Ihnen die wichtigsten Befehle vor, die Sie als Administrator benötigen, um Ihr SELinux-System zu verwalten und zu benutzen. Im nächsten Kapitel besprechen wir dann erste Anpassungen des Systems über boolesche Variablen, während Sie im übernächsten Kapitel kleine Anpassungen an der Richtlinie selbst vornehmen werden.

13.1 Distributionen

SELinux ist keine eigene Distribution. Daher benötigen Sie zunächst immer eine Linux-Distribution, die Sie anschließend um SELinux erweitern. Am besten werden aktuell die *Fedora Core*-Distributionen unterstützt. Die in diesem Buch besprochene SELinux-Referenz-Richtlinie kommt seit Fedora Core 5 zum Einsatz. Alternativ können Sie aber auch die *Debian*-Distribution *Etch*, *Ubuntu*, *Gentoo* oder *Slackware* einsetzen. Falls Sie eine andere Distribution wünschen, müssen Sie die notwendigen Anpassungen selbst vornehmen. Unterschätzen Sie den Aufwand nicht. Eine Vielzahl von Programmen müssen modifiziert werden. Es genügt nicht, einige wenige Bibliotheken und Befehle zusätzlich zu installieren.



Achtung

Dies trifft auch und besonders auf die *SUSE*-Distribution ab Version 10.1 zu, bei der die Unterstützung für SELinux wieder aus dem Kernel und aus den Applikationen entfernt wurde.

13.1.1 Fedora Core

Fedora Core kommt direkt mit SELinux-Unterstützung. Ab Fedora Core 5 ist die *SELinux Reference-Policy* im Einsatz (siehe Abschnitt 13.2).

13.1.2 Debian und Ubuntu

Debian hat für die aktuelle Version Etch (4.0) die Unterstützung für SELinux aufgenommen. Dies beinhaltet dann auch die Unterstützung für die SELinux *Reference-Policy* (siehe Abschnitt 13.2). Die Unterstützung für *Ubuntu* wird auf <https://wiki.ubuntu.com/SELinux> vorangetrieben und dokumentiert.

13.1.3 Gentoo

Das *Gentoo*-Projekt hat unter <http://www.gentoo.org/proj/en/hardened/selinux/selinux-handbook.xml> eine hervorragende Dokumentation darüber, wie eine *Gentoo*-Installation um SELinux erweitert wird. Eine weitere Erläuterung hier ist daher unnötig. Die Integration der SELinux *Reference-Policy* wird gerade vorbereitet und ist bei Veröffentlichung des Buches möglicherweise schon abgeschlossen.

13.1.4 Slackware

Die Unterstützung für *Slackware* ist nicht weit fortgeschritten. Es existieren jedoch einige Pakete mit den für *Slackware* notwendigen angepassten Dateien:

- <http://projects.dimensionalistorm.net/selinux/>
- <ftp://ftp.diyab.net/selinux/>

Diese Pakete sind aber durchaus schon älter, und ihr Betrieb ist daher problematisch.

13.2 Welche SELinux-Policy?

Dieses Buch betrachtet in erster Linie die SELinux-*Reference-Policy*, die auf <http://seref-policy.sf.net> gepflegt wurde und nun unter <http://oss.tresys.com> zu finden ist. Ältere Distributionen (zum Beispiel Fedora Core 3, 4 und Red Hat Enterprise Linux 4) verwenden häufig auch noch die SELinux *Example-Policy*, die auf <http://selinux.sf.net> gepflegt wird. Sollten Sie eine derartige Distribution einsetzen wollen, sollten Sie auch den Teil V lesen. Hier wird auf die Unterschiede eingegangen.

Die wesentlichen Ziele, die zur Entwicklung der neuen *Reference-Policy* geführt haben, waren:

- einen Quelltext für die Erzeugung aller Policy-Varianten: Targeted, Strict und MLS
- integrierte Dokumentation innerhalb der Policy
- Modularität und Kapselung der einzelnen Bestandteile
- vereinfachte Verwaltung und Übersetzung
- integrierte Unterstützung von *MLS*

Obwohl die Modularität die tägliche Arbeit mit der Policy vereinfacht, kann dieses Ziel erst richtig verstanden werden, wenn auch der SELinux Policy Management Ser-

ver (siehe Kapitel 32) eingesetzt wird. Dieses Projekt befindet sich noch in der Entwicklung und wird die zentrale Administration der Policy durch unterschiedliche Benutzer erlauben.

Sowohl beim Einsatz der Example-Policy als auch bei der Reference-Policy bieten viele Distributionen sowohl eine Targeted als auch eine Strict-Policy als Variante an. Die meisten einführenden Beispiele basieren auf der Targeted-Variante. Diese Variante ist bei den meisten Distributionen der Default und erlaubt die verständlichere Darstellung der Beispiele. Die Eigenschaften der Targeted-Policy werden in Kapitel 17.1 besprochen. Viele Befehle sind jedoch nur bei Einsatz der Strict-Variante sinnvoll (z.B. `newrole`). Deren Unterschiede werden in Kapitel 17.2 besprochen.

13.3 Erste Schritte und SELinux-Befehle

Sobald Sie ein funktionstüchtiges Linux-System mit aktiviertem SELinux besitzen, sollten Sie sich als `root` anmelden¹.



Tipp

Wenn Sie nicht über ein entsprechendes System verfügen, befindet sich auf der CD ein Fedora Core 6-System mit installierter SELinux-Reference-Policy als VMWare-Image. Dieses können Sie in einer VMWare-Workstation oder mit dem beigelegten VMWare-Player betreiben. Die Anmeldung ist als Benutzer `root` mit dem Kennwort *kennwort* möglich.

Zunächst sieht direkt nach der Anmeldung alles normal aus. Möglicherweise erhalten Sie aber auch direkt Protokollmeldungen von SELinux auf der Konsole (siehe Abbildung 13.1).

Ihre erste Frage sollte nun sein: Wer bin ich? Dies beantwortet der Befehl `id`. Mit der Option `-Z` erhalten Sie nur die SELinux-Identität:

```
[root@supergrobi ~]# id
uid=0(root) gid=0(root) Gruppen=0(root),1(bin),2(daemon),
3(sys),4(adm),6(disk),10(wheel) context=root:staff_r:
staff_t:SystemLow-SystemHigh
```

Ihr Benutzer verfügt nun über die Rolle `staff_r`. Diese Rolle ist Personen vorbehalten, die administrative Tätigkeiten wahrnehmen. In dieser Rolle sind diese administrativen Tätigkeiten diesen Personen aber nicht gestattet. Das können Sie sogar recht leicht erkennen.

¹ Möglicherweise funktioniert die grafische Anmeldung nicht. Wechseln Sie dann mit `(STRG)+(ALT)+(F1-6)` auf eine Konsole, und melden Sie sich dort an.

```

Fedora Core release 5 (Bordeaux)
Kernel 2.6.15-1.2054_FC5 on an i686

localhost login: audit(1146705086.465:79): avc: denied { sendto } for pid=199
2 comm="udev" path=002F6F72672F667265656465736B746F702F68616C2F756465765F657665
6E74 scontext=system_u:system_r:udev_t:s0-s0:c0.c255 tcontext=system_u:system_r:
hald_t:s0 tclass=unix_dgram_socket
root
audit(1146705339.665:80): avc: denied { search } for pid=1976 comm="login" na
me="nscd" dev=dm-0 ino=65319 scontext=system_u:system_r:local_login_t:s0-s0:c0.c
255 tcontext=system_u:object_r:nscd_var_run_t:s0 tclass=dir
Password:
Last login: Wed May 3 18:58:32 on :0
audit(1146705341.353:81): avc: denied { read } for pid=2050 comm="bash" name=
".bash_profile" dev=dm-0 ino=162406 scontext=root:staff_r:staff_t:s0-s0:c0.c255
tcontext=root:object_r:sysadm_home_t:s0 tclass=file
[root@localhost ~]# _

```

Abbildung 13.1: Bei der Anmeldung protokolliert SELinux möglicherweise direkt Verletzungen der Richtlinie.

Hierfür müssen Sie sich aber zunächst weitere Informationen über die SELinux-Konfiguration beschaffen. Der Befehl `getenforce` teilt Ihnen den aktuellen Zustand des SELinux-Systems mit:

```
[root@supergrobi ~]# getenforce
Permissive
```

Mögliche Rückgabewerte des Befehls sind:

- Disabled: SELinux ist abgeschaltet.
- Permissive: SELinux ist angeschaltet und wertet die Policy aus. Verletzungen werden aber nur protokolliert und nicht verhindert. Jede Aktion ist erlaubt, als ob SELinux abgeschaltet wäre.
- Enforcing: SELinux ist angeschaltet und erzwingt die Einhaltung der Policy.

Der Befehl `sestatus` liefert Ihnen die gleichen Informationen ein wenig ausführlicher:

```
[root@supergrobi ~]# sestatus
SELinux status:                enabled
SELinuxfs mount:              /selinux
Current mode:                  permissive
Mode from config file:        permissive
Policy version:                20
Policy from config file:      strict
```

Hier erkennen Sie, dass SELinux angeschaltet ist und wo das SELinux-Dateisystem gemountet wird. Wir werden das SELinux-Dateisystem später noch kennenlernen. Die hier vorgestellten Befehle kommunizieren mit SELinux im Kernel über dieses Dateisystem. Es handelt sich um ein virtuelles Dateisystem ähnlich `/proc` und `/sys`. Sie erkennen den aktuellen Modus (*Permissive*) und den in der Konfigurationsdatei hinterlegten Modus. Die angegebene Version bezeichnet die SELinux-Version im Kernel und ist ein Anhaltspunkt für die verfügbaren Sprachelemente. Schließlich erkennen Sie noch, dass auf meinem Rechner die *Strict-Policy* eingesetzt wird.

Die hier angesprochene Konfigurationsdatei ist `/etc/selinux/config`. Diese Datei definiert den SELinux-Modus:

```
# 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 - SELinux is fully disabled.
SELINUX=permissive
# SELINUXTYPE= type of policy in use. Possible values are:
#     targeted - Only targeted network daemons are protected.
#     strict - Full SELinux protection.
SELINUXTYPE=strict

# SETLOCALDEFS= Check local definition changes
SETLOCALDEFS=0
```

Diese Datei wird bei dem Boot des Systems ausgewertet. Nach einer Änderung in der Datei genügt daher ein Reboot. Wenn Sie jedoch die Policy ändern möchten (*strict* oder *targeted*), genügt kein einfacher Reboot, da diese Policies unterschiedliche Security-Contexts verwenden. Zusätzlich ist ein *Relabeling* des Dateisystems erforderlich². Wir werden diesen Punkt noch ansprechen.

Kommen wir zurück zu unserem Ausgangspunkt. Der Benutzer *root* verfügt über die Rolle *staff_r*, die keine administrativen Arbeiten erlaubt. Testen Sie das zum Beispiel mit dem Befehl `ps -ef`. Dieser Befehl zeigt sämtliche Prozesse auf dem System an. Auf dem hier vorgestellten System funktioniert das zunächst, da es sich im *Permissive-Mode* befindet:

```
[root@supergrobi ~]# ps -ef
UID          PID  PPID  C  STIME TTY          TIME CMD
root           1     0  0  12:37 ?           00:00:01 init [5]
root           2     1  0  12:37 ?           00:00:00 [migration/0]
root           3     1  0  12:37 ?           00:00:00 [ksoftirqd/0]
root           4     1  0  12:37 ?           00:00:00 [watchdog/0]
root           5     1  0  12:37 ?           00:00:00 [migration/1]
```

² Üblicherweise genügt es, eine Datei `/.autorelabel` vor dem Reboot anzulegen.

```

root          6      1  0 12:37 ?          00:00:00 [ksoftirqd/1]
...
root         4122  2239  0 18:23 tty1        00:00:00 -bash
root         4297  4092  0 18:57 pts/2       00:00:00 ps -ef

```

Wechseln Sie nun auf dem System in den Enforcing-Mode. Hierzu müssen Sie nicht rebooten. Es genügt, den Befehl `setenforce` zu verwenden:

```

[root@supergrobi ~]# setenforce 1
[root@supergrobi ~]# getenforce
Enforcing
[root@supergrobi ~]# ps -ef
UID          PID  PPID  C  STIME TTY          TIME CMD
root         4092  4089  0 18:21 pts/2       00:00:00 -bash
root         4122  2239  0 18:23 tty1        00:00:00 -bash
root         4305  4092  0 18:58 pts/2       00:00:00 ps -ef

```

Nun sehen Sie nur noch eine kleine Auswahl der laufenden Prozesse. SELinux verhindert den Zugriff auf alle Prozesse, die nicht von Ihnen gestartet wurden! Dies gilt auch für andere Prozesse, die den Benutzer `root` verwenden. Versuchen Sie, auf die Heimatverzeichnisse anderer Benutzer oder auf das `proc`-Dateisystem zuzugreifen:

```

[root@supergrobi ~]# ls -l /home/
insgesamt 8
?----- ? ?      ?          ? /home/lost+found
?----- ? ?      ?          ? /home/student
[root@supergrobi ~]# ls -l /proc/kcore
ls: /proc/kcore: Keine Berechtigung

```

Nebenbei: In den Permissive-Modus wechselt SELinux ebenfalls mit dem Befehl `setenforce`:

```

[root@supergrobi ~]# setenforce 0
setenforce: setenforce() failed

```

Leider darf dies nur ein Benutzer mit der Rolle `sysadm_r`.

Sie sollten bis jetzt bereits einen Eindruck davon gewonnen haben, wie SELinux die Sicherheit eines Linux-System gewährleisten kann. Natürlich ist dafür eine fehlerfreie und umfangreiche Policy erforderlich. Diese wird jedoch bei den meisten Distributionen bereits mitgeliefert. Auch setzt dieses Beispiel die *Strict-Policy* voraus. Wir werden in diesem Buch uns in erster Linie mit der *Targeted-Policy* beschäftigen, da diese häufiger zum Einsatz kommt.

Wie wird nun das System administriert? Der Benutzer *root* hat auch Zugriff auf die Rolle *sysadm_r*, die über die notwendigen Privilegien verfügt. Um in die Rolle zu wechseln, verwendet er den Befehl `newrole`:

```
[root@supergrobi ~]# newrole -r sysadm_r
Authentifiziere root.
Passwort:
[root@supergrobi ~]# ls -l /proc/kcore
-r----- 1 root root 527896576 23. Aug 19:06 /proc/kcore
```

Anschließend kann er auch wieder auf alle Dateien zugreifen.

Weitere wichtige Befehle auf der Kommandozeile sind:

- `chcon`: Hiermit können Sie den Kontext einer Datei ändern (siehe Abschnitt 18.6).
- `restorecon`: Dieser Befehl stellt den Security-Context einer Datei entsprechend der Policy wieder her (siehe Abschnitt 18.16).
- `selinuxenabled`: Dieser Befehl kann in Scripts eingesetzt werden, um den Status von SELinux zu prüfen (siehe Abschnitt 18.24).
- `getsebool/setsebool`: Hiermit können Sie boolesche Variablen lesen und setzen (siehe Abschnitt 15 und Abschnitt 18.34).
- `audit2allow/audit2why`: Diese Befehle erlauben die Analyse der SELinux-Protokollmeldungen und die Anpassung der Richtlinie (siehe Kapitel 16 und Abschnitt 18.3).

Dies soll als erste Einführung in SELinux genügen. Zunächst werden Sie das System genauso verwenden können wie bisher. Jedoch werden einige Funktionen nicht mehr zur Verfügung stehen. Dann müssen Sie die Protokollmeldungen analysieren (siehe Kapitel 14) und können die Richtlinie über boolesche Variablen (siehe Kapitel 15) oder über die Erweiterung der Richtlinien anpassen (siehe Kapitel 16). Eine typische Zusammenfassung der täglichen Aufgaben und ihre Lösung ist in Kapitel 19 zu finden. Wenn Sie aber tiefer einsteigen möchten und vielleicht sogar für komplett neue Dienste die Richtlinien entwickeln möchten, müssen Sie sich mehr mit der Sprache und ihren Bestandteilen beschäftigen. Dann sollten Sie Teil IV durcharbeiten.

