



19 Typische SELinux-Administrationsaufgaben

Viele Administratoren schalten als Erstes bei dem Einsatz der entsprechenden Distributionen SELinux ab. Das ist schade und meist unnötig, wenn grundsätzliche Kenntnisse des Systems existieren.

Dieses Kapitel versucht die typischen Probleme beim Einsatz von SELinux zu erklären und Lösungen aufzuzeigen.

19.1 Abschalten von SELinux

Vielleicht wollen Sie SELinux einfach nur abschalten. Ich kann dieses Vorgehen nicht empfehlen, sondern Ihnen nur raten, sich mit dem System zu beschäftigen und es zu nutzen. Es wird die Sicherheit Ihres Systems enorm erhöhen.

Falls Sie sich dennoch dazu entscheiden, möchte ich Ihnen zeigen, wie Sie dies richtig bewerkstelligen. Eigentlich ist es ganz einfach. Es gibt drei Möglichkeiten:

- Sie können SELinux komplett abschalten.
- Sie können SELinux in einen Warnmodus versetzen.
- Sie können SELinux für einen einzelnen Dienst abschalten, wenn Sie die Targeted Policy verwenden.

Während die dritte Alternative im nächsten Abschnitt betrachtet wird, werden wir hier die ersten beiden Möglichkeiten besprechen. Die Konfiguration von SELinux erfolgt in der Datei `/etc/selinux/config`. Diese Datei enthält unter anderem die folgenden Zeilen:

```
# 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
```

Der Wert der Variable `SELINUX` bestimmt das Verhalten bei dem Boot:

- *Enforcing*: SELinux ist aktiv und setzt die Policy um. SELinux kann zur Laufzeit in den Modus *Permissive* versetzt werden.

- *Permissive*: SELinux ist aktiv. Verletzungen der Policy werden erlaubt, aber protokolliert. SELinux kann zur Laufzeit in den Modus Enforcing versetzt werden.
- *Disabled*: Dies schaltet SELinux komplett ab. Es kann zur Laufzeit nicht aktiviert werden. Ein Reboot ist hierzu erforderlich.

Um SELinux komplett abzuschalten, genügt daher die folgende Zeile in der Datei `/etc/selinux/config`:

```
SELINUX=disabled
```



Achtung

Es ist wichtig, dass Sie in dieser Zeile keine Leerzeichen einfügen.

Sie müssen aber berücksichtigen, dass nun ohne einen Neustart des Systems keine Aktivierung von SELinux möglich ist. SELinux kann nicht, wie zum Beispiel AppArmor, im laufenden Betrieb geladen werden. Außerdem verlangt ein späterer Start mit aktiviertem SELinux immer ein komplettes *Relabeling* (siehe Abschnitt 19.3) des Systems, so als ob SELinux nie auf dem System aktiv gewesen sei (siehe Kapitel 22).

19.2 Abschalten von SELinux für einen Dienst

In vielen Fällen entsteht der Wunsch, SELinux abzuschalten, wenn ein Dienst sich nicht wie erwartet verhält. Anstatt SELinux komplett abzuschalten, sollten Sie in diesem Fall abwägen, ob Sie zunächst SELinux nur für diesen Dienst oder das betroffene Programm abschalten. Dann können Sie später, nach der Lektüre dieses Buchs, mit mehr Verständnis, Wissen und Zeit, SELinux entsprechend anpassen und wieder für den Dienst aktivieren.

Wenn Sie die *Targeted-Policy* einsetzen, können Sie für jeden überwachten Dienst SELinux einzeln an- bzw. wieder abschalten.

Hierzu existiert für jeden Dienst eine boolesche Variable: `<dienst>_disable_trans`. Wenn Sie diese Variable setzen und den Dienst neu starten, wird der Dienst anschließend nicht von SELinux in seiner eigenen Domäne überwacht, sondern wird in der Domäne *unconfined_t* kontrolliert. Damit ist der Dienst faktisch ohne besondere Überwachung.

Für den Webserver lautet die Variable zum Beispiel: `httpd_disable_trans`.

19.3 Erneutes Labeln des Betriebssystems

Ein *Relabeling* des Systems ist nur in seltenen Fällen nötig. Hierbei handelt es sich um den Wechsel der Policy oder um den Neustart des Systems, nachdem es zwischen- durch ohne SELinux-Unterstützung betrieben wurde.

Jede Datei auf einem SELinux-System besitzt einen *Security-Context*. Dieser Security-Context bestimmt, wer diese Datei nutzen darf. Falls eine Datei keinen Security-Context besitzt, erhält sie automatisch einen Default-Context. Dieser hängt von der Policy ab. Mit dem Security-Context bestimmt SELinux, welche Prozesse auf die Datei zugreifen dürfen. In vielen Fällen führt diese Tatsache dazu, dass der gewünschte Zugriff auf die Datei nicht möglich ist. Handelt es sich nur um eine Datei, kann der Security-Context sehr einfach mit dem Befehl `restorecon` wiederhergestellt werden.

Wird das System mit einem Kernel gebootet, der kein SELinux unterstützt, oder wurde SELinux vorübergehend abgeschaltet, so sind während des Betriebs alle Dateien ohne einen Security-Context angelegt worden. Hierbei handelt es sich nun um zahlreiche über das gesamte System verstreute Dateien. Diese alle mit `restorecon` einzeln zu labeln würde recht lange dauern. Mit `find` können Sie zwar diese Dateien finden, jedoch ist das recht aufwendig.

Meist erkennt das Betriebssystem selbst, dass es vorübergehend ohne SELinux-Unterstützung betrieben wurde, und erzwingt das Relabeling. Wenn Sie es manuell anstoßen möchten, gibt es zwei Möglichkeiten:

- Sie können die Datei `/.autorelabel` anlegen. Findet der SysV-Init diese Datei, so führt er automatisch ein Relabeling bei dem Boot durch.
- Sie rufen den Befehl `fixfiles relabel` auf.

Wenn Sie mit dem Werkzeug `system-config-securitylevel` einer Fedora oder RHEL-Distribution die Policy wechseln, wird ebenfalls das *Relabeling* des gesamten Systems erzwungen. Wenn Sie manuell die Policy wechseln, indem Sie die Policy in der Datei `/etc/selinux/config` ändern, dann müssen Sie auch die Datei `/.autorelabel` erzeugen.

19.4 Programme in `unconfined_t` funktionieren nicht

SELinux beschränkt unter Fedora und RHEL in den neueren Versionen auch Programme in der Domäne `unconfined_t`. Dabei überwacht SELinux einige Speicheroperationen:

- `execmod`: Hierbei überwacht das System, ob ein Programm eine Speicherseite, die zuvor geschrieben (modifiziert) wurde, anschließend ausführen möchte. Da diese Funktionalität bei vielen Angriffen (z.B. Buffer-Overflows) benötigt wird, verhindert SELinux dies. Ein häufiger Grund für derartige Fehler sind jedoch Text-Relocations. Diese können Sie für eine Bibliothek erlauben. Allerdings sollten Sie anschließend einen Bug-Report verfassen, da es sich eigentlich um einen

Fehler handelt. Ulrich Drepper hat weitere Informationen auf seiner Homepage¹. Sie erkennen den Fehler an der folgenden Meldung:

```
error while loading shared libraries: /usr/lib/<bibliothek>.so
cannot restore segment prot after reloc: Permission denied
```

Um Text-Relocations zu erlauben, stellen Sie sicher, dass die betroffene Bibliothek den Typ *textrel_shlib_t* besitzt:

```
# /usr/sbin/semanage fcontext -a -t textrel_shlib_t ◀
    '/usr/lib/<bibliothek>.so'
# /sbin/restorecon -v /usr/lib/<bibliothek>.so
```

Zusätzlich kann die boolesche Variable *allow_execmod* aktiv sein. Dann dürfen in der *Targeted-Policy* alle Programme und Bibliotheken vom Typ *unconfined_t* im Vorfeld modifizierten Speicher ausführen.

- *execmem*: Dieser Fehler tritt auf, wenn eine Applikation ein Anonymous Mapping² als ausführbar kennzeichnet. Um dies zu erlauben, existiert die boolesche Variable *allow_execmem*, die dies für alle Applikationen erlaubt.
- *execstack*: Dieser Fehler tritt auf, wenn eine Applikation versucht, ihren Stack oder Teile davon ausführbar zu machen. Dies kann nur über boolesche Variablen erlaubt werden. Hierzu stehen unter Fedora 6 zwei Variablen zur Verfügung:

```
allow_execstack --> on
allow_java_execstack --> off
```

- *execheap*: Dieser Fehler tritt auf, wenn ein Programm Daten auf dem Heap ausführen möchte. Dies sollte bei sauberer Programmierung nie erforderlich sein. Dennoch können Sie es mit einer booleschen Variable erlauben: *allow_execheap*.

19.5 KDE-Programme

Nach der Installation und Aktivierung von SELinux funktioniert die grafische Oberfläche *KDE* möglicherweise nicht mehr wie erwartet. Wenn Sie auf Ihrem System zu einem späteren Zeitpunkt SELinux installieren und aktivieren und vorher bereits *KDE* benutzt haben, kann es zu Problemen kommen. *KDE* legt viele temporäre Dateien in */tmp* und */var/tmp* an. Diese können bei dem automatischen *Relabeling* nicht richtig erkannt werden und erhalten daher nicht den korrekten *Security-Context*. Bei einem erneuten Start von *KDE* kann es daher nicht auf die Dateien zugreifen und startet nicht korrekt. Löschen Sie daher einfach diese Dateien von *KDE*:

```
rm -rf /var/tmp/kdecache-<user>
```

¹ <http://people.redhat.com/drepper/textrelocs.html>

² Programme können Dateien in den Speicher »mappen«. Hierzu dient der Funktionsaufruf *mmap*. Wenn allgemein Speicher benötigt wird, ohne dass eine spezielle Datei genutzt wird, kann als Flag bei dem Aufruf *MAP_ANONYMOUS* angegeben werden. Dann wird lediglich eine bestimmte Menge Speicher zur Verfügung gestellt.

19.6 Start eines Dienstes mit ungewöhnlichem Port

Einige Administratoren möchten einen Netzwerkdienst auf einem Nicht-Standard-Port starten. SELinux verhindert dies für die überwachten Dienste in der *Targeted-Policy*. In der *Strict-Policy* wird das für jeden Dienst überwacht. Um einen weiteren Port zu erlauben, muss dieser SELinux bekannt gemacht werden. Um zum Beispiel den Webserver auf dem Port 8088 zu starten, ist es erforderlich, dass der Port über den entsprechenden Typ verfügt. Zunächst sollte geprüft werden, welchen Typ der Port benötigt. Normalerweise startet der *Apache* Webserver auf dem Port 80. Mit dem folgenden Befehl ermitteln Sie den Typ des Ports:

```
[root@supergrobi ~]# semanage port -l | grep 80
amanda_port_t          tcp      10080, 10081, 10082, 10083
amanda_port_t          udp      10080, 10081
hplip_port_t           tcp      1782, 2207, 2208, 8290, ◀
                    50000, 50002, 8292, 9100, 9101, 9102, 9220, ◀
                    9221, 9222, 9280, 9281, 9282, 9290, 9291, 9292
http_cache_port_t      tcp      8888, 3128, 8080, 8118
http_port_t            tcp      81, 80, 443, 488, 8008, ◀
                    8009, 8443
ocsp_port_t            tcp      9080
soundd_port_t          tcp      8000, 9433
transproxy_port_t      tcp      8081
xen_port_t              tcp      8002
zope_port_t            tcp      8021
```

Da Sie nun wissen, dass SELinux dem Webserver den Zugriff auf Ports vom Typ *http_port_t* erlaubt, können Sie mit *semanage* Ihren Port mit dem entsprechenden Typ versehen:

```
[root@supergrobi ~]# semanage port -a -p tcp -t http_port_t 8088
```

Leider kann ein Port nicht zwei Typen gleichzeitig erhalten. Ports, die durch die Policy definiert werden, können auch nicht modifiziert werden. Wenn Sie einen Port verwenden möchten, der bereits in der Policy anders definiert wurde, müssen Sie dem Webserver entweder erlauben, auf Ports mit diesem Typ zuzugreifen, oder die Policy modifizieren, indem Sie das entsprechende Policy-Modul entfernen oder die gesamte Policy neu bauen. Hinweise hierzu finden Sie in Abschnitt 16.5.

19.7 Verwendung einer SWAP-Datei

Wenn der SWAP-Speicher in Form von SWAP-Partitionen nicht reicht, kann auch eine Datei als SWAP genutzt werden. Hierzu muss die Datei lediglich erzeugt und entsprechend formatiert werden:

```
[root@supergrobi ~]# dd if=/dev/zero of=/swap bs=1M count=100
100+0 Datensätze ein
100+0 Datensätze aus
104857600 Bytes (105 MB) kopiert, 0,262746 Sekunden, 399 MB/s
[root@supergrobi ~]# mkswap /swap
Setting up swapspace version 1, size = 104853 kB
```

Damit SELinux die Verwendung als SWAP zulässt, muss die Datei nun auch noch den richtigen Typ erhalten. Hierzu benötigt sie den Typ *swapfile_t*. Dies erreichen Sie mit:

```
[root@supergrobi ~]# chcon -t swapfile_t /swap
```

Da der Typ *swapfile_t* zu den *Customizable Types* gehört, ist die Registrierung in der Policy mit *semanage* nicht zwingend erforderlich.

19.8 Apache Webserver

Viele Administratoren setzen ein exponiertes Linux-System als Betriebssystem für einen *Apache* Webserver ein. Hier ist ein hohes Maß an Sicherheit erforderlich. Dies trifft besonders bei Webservern mit dynamisch generierten Webseiten zu. Häufig gab es in der Vergangenheit Probleme und Sicherheitslücken, die einen Angriff ermöglichen. Dies wird in der Zukunft sicherlich weiterhin so sein.

Daher ist es besonders wichtig, den Webserver mit SELinux zu überwachen. Hier tauchen aber auch die meisten Probleme auf. Daher will ich Ihnen in den folgenden Abschnitten die wesentlichen Probleme und ihre Lösungen aufzeigen.

19.8.1 Neues DocumentRoot-Verzeichnis

Viele Benutzer legen für ihren Webserver ein eigenes *DocumentRoot*-Verzeichnis an. Dies trifft besonders in vielen *Shared-Hosting*-Umgebungen zu. Die SELinux-Policy berücksichtigt aber nur die Default-Verzeichnisse der Linux-Distributionen:

- `/var/www`
- `/srv/www`
- `/www`

Falls Sie Ihr *DocumentRoot*-Verzeichnis an einer anderen Stelle, z.B. `/home/www` ablegen möchten, so berücksichtigt die Policy nicht das Verzeichnis und erlaubt dem Apache Webserver nicht den Zugriff auf die dort gespeicherten Dateien.

Laut seiner Policy darf der Apache nur auf Dateien mit folgendem Typ zugreifen:

- *httpd_sys_content_t*: Alle Dateien mit diesem Typ dürfen von dem Webserver ausgeliefert werden. Auch *PHP*-Scripts benötigen diesen Typ, wenn sie mit *mod_php* ausgeliefert werden. Dies ist ein *Customizable Types*. Es ist daher nicht zwingend erforderlich, die Dateien in der Policy zu registrieren. Allerdings kann es auch nicht schaden. Um sicherzustellen, dass alle Dateien in dem Verzeichnis */web* diesen Typ erhalten, verwenden Sie:

```
[root@supergrobi ~]# semanage fcontext -a -t ◀
    httpd_sys_content_t '/web(/.*)?'
```

- *httpd_sys_script_exec_t*: CGI-Scripts benötigen diesen Typ. Sie dürfen dann auf alle Dateien mit dem Typ *httpd_sys_** zugreifen.
- *httpd_sys_script_ro_t*: Dateien mit diesem Typ dürfen nur von CGI-Scripts mit Typ *httpd_sys_script_t* gelesen werden.
- *httpd_sys_script_rw_t*: Dateien mit diesem Typ dürfen nur von CGI-Scripts mit dem Typ *httpd_sys_script_t* gelesen und geschrieben werden.
- *httpd_sys_script_ra_t*: Diese Dateien dürfen gelesen und im Append-Modus von CGI-Scripts mit dem Typ *httpd_sys_script_t* geschrieben werden.
- *httpd_unconfined_script_exec_t*: Dies ist die letzte Rettung für Scripts, die so kompliziert sind, dass eine Anpassung der Policy zu aufwendig ist. Bevor Sie die SELinux-Überwachung für den gesamten Webserver abschalten, können Sie einem Script diesen Typ zuweisen und damit die Überwachung nur für das Script abschalten.

19.8.2 Gleichzeitiger Zugriff per FTP, Samba etc.

Häufig werden die Webseiten, die der *Apache* Server ausliefern soll, von den Anwendern über eine Windows-Freigabe per *Samba* oder mithilfe eines *FTP*-Servers verwaltet. Sobald die Dateien aber mit *Samba* oder mithilfe eines *FTP*-Servers geschrieben werden, besitzen sie nicht den richtigen Typ für die Auslieferung durch den Webserver. Viele Benutzer schalten daher die Überwachung für den Webserver ab. Dies muss jedoch nicht sein, da die SELinux Policy diese gemeinsame Nutzung erlaubt.

Hierzu wurden in der Policy die Typen *public_content_t* und *public_content_rw_t* geschaffen. Die Dienste *Apache*, *FTP*, *Samba* und *Rsync* dürfen alle Dateien vom Typ *public_content_t* lesen. Wenn einer dieser Dienste auch Dateien schreiben soll, so müssen Sie zuvor das Verzeichnis, in dem der Dienst Schreibrechte erhalten soll, mit dem Typ *public_content_rw_t* versehen und für den Dienst die boolesche Variable *allow_<domain>_anon_write* setzen. Um CGI-Scripts Schreibrechte an diesen Dateien zu geben, verwenden Sie die boolesche Variable *allow_httpd_sys_script_anon_write*. Die so erzeugten Dateien können dann auch per *FTP* oder *Samba* zur Verfügung gestellt werden.

19.8.3 Zugriff auf eine MySQL-Datenbank

Häufig werden bei dem Betrieb eines Webservers PHP- oder Perl-Scripts eingesetzt, die Zugriff auf eine Datenbank oder andere Netzdienste benötigen. Normalerweise unterbindet die SELinux-Policy diese Zugriffe. Um diese Zugriffe ohne großen Aufwand zu erlauben, können Sie zwei boolesche Variablen verwenden. Hierbei handelt es sich um:

- `httpd_can_network_connect`: Diese Variable erlaubt dem Webserver den Aufbau von Netzwerkverbindungen zu beliebigen entfernten TCP-Diensten.
- `httpd_can_network_connect_db`: Diese Variable erlaubt dem Webserver den Aufbau von TCP-Verbindungen zu Ports, die als Typ `mysqld_port_t` oder `postgresql_port_t` aufweisen. Dies sind normalerweise die Ports 3306 und 5432. Wenn Ihre Datenbank einen anderen Port verwendet, können Sie den entsprechenden Port natürlich mit einem der beiden Typen versehen. Dann wird die Policy den Zugriff erlauben.

19.9 Sicherung (Backup)

Wenn Sie ein SELinux-System sichern, ist es wichtig, dass Sie auch die *Security-Contexts* der Dateien sichern. Kommandos wie `tar` können dies nicht.

Wenn Sie ein Ext3-Dateisystem einsetzen, können Sie die Kommandos `dump` und `restore` verwenden. Diese sichern die Dateien auf Dateisystemebene und berücksichtigen in neueren Versionen die erweiterten Attribute, in denen die SELinux-Contexts gespeichert werden.

Wenn Sie bisher das Kommando `tar` eingesetzt haben, können Sie dieses durch `star` ersetzen:

```
star --xattr -H=exustar -c -f sicherung.star /verz
```