Verwendung von SELinux

Table of Contents

Verhindern Sie, dass Benutzer und Prozesse unbefugte Interaktionen mit Dateien und Geräten durchführen, indem Sie Security-Enhanced Linux (SELinux) verwenden.

Durch die Konfiguration von SELinux können Sie die Sicherheit Ihres Systems erhöhen. SELinux ist eine Implementierung der Mandatory Access Control (MAC) und bietet eine zusätzliche Sicherheitsebene. Die SELinux-Richtlinie definiert, wie Benutzer und Prozesse mit den Dateien auf dem System interagieren können. Sie können steuern, welche Benutzer welche Aktionen ausführen können, indem Sie sie bestimmten auf SELinux beschränkten Benutzern zuordnen.

1. Erste Schritte mit SELinux

Security Enhanced Linux (SELinux) bietet eine zusätzliche Ebene der Systemsicherheit. SELinux beantwortet grundsätzlich die Frage:

Darf <subject> <action> to <object> tun?

Zum Beispiel: Darf ein Webserver auf Dateien in den Home-Verzeichnissen der Benutzer zugreifen?

1.1. Einführung in SELinux

Die standardmäßige Zugriffsrichtlinie, die auf Benutzer-, Gruppen- und anderen Berechtigungen basiert und als Discretionary Access Control (DAC) bekannt ist, ermöglicht es Systemadministratoren nicht, umfassende und fein abgestimmte Sicherheitsrichtlinien zu erstellen, wie etwa die Beschränkung bestimmter Anwendungen auf die ausschließliche Anzeige von Protokolldateien. Gleichzeitig können andere Anwendungen neue Daten an die Protokolldateien anhängen.

Security Enhanced Linux (SELinux) implementiert Mandatory Access Control (MAC). Jeder Prozess und jede Systemressource verfügt über ein spezielles Sicherheitsetikett, das als SELinux-Kontext bezeichnet wird. Ein SELinux-Kontext, manchmal auch als SELinux-Label bezeichnet, ist eine Kennung, die die Details auf Systemebene abstrahiert und sich auf die Sicherheitseigenschaften der Entität konzentriert. Dies bietet nicht nur eine konsistente Möglichkeit, Objekte in der SELinux-Richtlinie zu referenzieren, sondern beseitigt auch alle Unklarheiten, die bei anderen Identifizierungsmethoden auftreten können. Beispielsweise kann eine Datei auf einem System, das Bind-Mounts verwendet, mehrere gültige Pfadnamen haben.

Die SELinux-Richtlinie verwendet diese Kontexte in einer Reihe von Regeln, die definieren, wie Prozesse miteinander und mit den verschiedenen Systemressourcen interagieren können. Standardmäßig lässt die Richtlinie keine Interaktion zu, es sei denn, eine Regel gewährt explizit Zugriff

Denken Sie daran, dass SELinux-Richtlinienregeln nach DAC-Regeln überprüft werden. SELinux-Richtlinienregeln werden nicht verwendet, wenn DAC-Regeln den Zugriff zuerst verweigern. Dies bedeutet, dass keine SELinux-Verweigerung protokolliert wird, wenn die herkömmlichen DAC-Regeln den Zugriff verhindern.

SELinux-Kontexte verfügen über mehrere Felder

- Benutzer
- Rolle
- Typ

- Sicherheitsstufe

Die SELinux-Typinformationen sind möglicherweise die wichtigsten, wenn es um die SELinux-Richtlinie geht, da die häufigste Richtlinienregel, die die zulässigen Interaktionen zwischen Prozessen und Systemressourcen definiert, SELinux-Typen und nicht den vollständigen SELinux-Kontext verwendet.

SELinux-Typen enden mit _t. Der Typname für den Webserver lautet beispielsweise httpd_t. Der Typkontext für Dateien und Verzeichnisse, die normalerweise in /var/www/html/ zu finden sind, ist httpd_sys_content_t. Der Typkontext für Dateien und Verzeichnisse, der normalerweise in /tmp und /var/tmp/ zu finden ist, ist tmp_t. Der Typkontext für Webserver-Ports ist http_port_t.

Es gibt eine Richtlinienregel, die es Apache (dem Webserverprozess, der als httpd_t läuft) erlaubt, auf Dateien und Verzeichnisse mit einem Kontext zuzugreifen, der normalerweise in /var/www/html/ und anderen Webserververzeichnissen (httpd_sys_content_t) zu finden ist. Für Dateien, die sich normalerweise in /tmp und /var/tmp/ befinden, gibt es in der Richtlinie keine Zulassungsregel, sodass der Zugriff nicht gestattet ist. Selbst wenn Apache kompromittiert wird und ein bösartiges Skript Zugriff erhält, ist es mit SELinux immer noch nicht in der Lage, auf das Verzeichnis /tmp zuzugreifen.

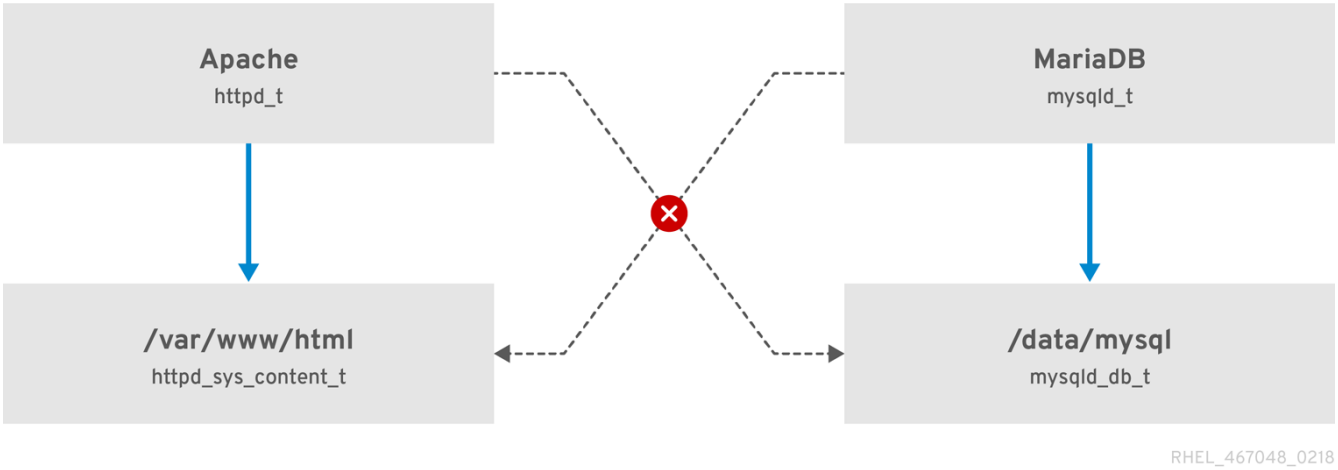

Figure 1. SELinux-Einführung: Apache und MariaDB

Wie das vorherige Schema zeigt, erlaubt SELinux dem Apache-Prozess, der als httpd_t ausgeführt wird, den Zugriff auf das Verzeichnis /var/www/html/ und verweigert demselben Prozess den Zugriff auf das Verzeichnis /data/mysql/, da es keine Zulassungsregel für httpd_t und gibt Kontexte vom Typ mysqld_db_t. Andererseits kann der MariaDB-Prozess, der als mysqld_t ausgeführt wird, auf das Verzeichnis /data/mysql/ zugreifen, und SELinux verweigert dem Prozess mit dem Typ mysqld_t ebenfalls ordnungsgemäß den Zugriff auf das Verzeichnis /var/www/html/ mit der Bezeichnung httpd_sys_content_t.

ZUSÄTZLICHE RESSOURCEN: Selinux(8) - **Manpage und Manpages**, die vom Befehl apropos selinux aufgelistet werden. Manpages, die vom Befehl man -k _selinux aufgelistet werden, wenn das Paket selinux-policy-doc installiert wird.

1.2. Vorteile der Ausführung von SELinux

SELinux bietet folgenden Vorteile:

- Alle Prozesse und Dateien sind gekennzeichnet. SELinux-Richtlinienregeln definieren, wie Prozesse mit Dateien interagieren und wie Prozesse miteinander interagieren. Der Zugriff ist nur zulässig, wenn eine SELinux-Richtlinienregel vorhanden ist, die ihn ausdrücklich zulässt.
- Fein abgestimmte Zugangskontrolle. SELinux-Zugriffsentscheidungen gehen über herkömmliche UNIX-Berechtigungen hinaus, die nach Ermessen des Benutzers gesteuert werden und auf Linux-Benutzer- und Gruppen-IDs basieren, und basieren auf allen verfügbaren Informationen, wie z. B. einem SELinux-Benutzer, einer Rolle, einem Typ und optional einer Sicherheitsstufe.
- Die SELinux-Richtlinie wird administrativ definiert und systemweit durchgesetzt.
- Verbesserte Abwehr von Privilege-Escalation-Angriffen. Prozesse laufen in Domänen ab und sind daher voneinander getrennt. SELinux-Richtlinienregeln definieren, wie Prozesse auf Dateien und andere Prozesse zugreifen. Wenn ein Prozess kompromittiert wird, hat der Angreifer nur Zugriff auf die normalen Funktionen dieses Prozesses und auf Dateien, auf die der Prozess Zugriff haben soll. Wenn beispielsweise der Apache-HTTP-Server kompromittiert ist, kann ein Angreifer diesen Prozess nicht zum Lesen von Dateien in Benutzer-Home-Verzeichnissen verwenden, es sei denn, eine bestimmte SELinux-Richtlinienregel wurde hinzugefügt oder konfiguriert, um einen solchen Zugriff zu ermöglichen.
- SELinux kann verwendet werden, um die Vertraulichkeit und Integrität von Daten durchzusetzen und Prozesse vor nicht vertrauenswürdigen Eingaben zu schützen.

SELinux ist jedoch nicht:

- Antiviren Software
- Ersatz für Passwörter, Firewalls und andere Sicherheitssysteme
- All-in-One-Sicherheitslösung

1.3. SELinux Beispiele

Die folgenden Beispiele zeigen, wie SELinux die Sicherheit erhöht:

- Die Standardaktion ist „Verweigern“. Wenn keine SELinux-Richtlinienregel vorhanden ist, um den Zugriff zu ermöglichen, beispielsweise für einen Prozess, der eine Datei öffnet, wird der Zugriff verweigert.
- SELinux kann Linux-Benutzer einschränken. In der SELinux-Richtlinie gibt es eine Reihe eingeschränkter SELinux-Benutzer. Linux-Benutzer können eingeschränkten SELinux-Benutzern zugeordnet werden, um die auf sie angewendeten Sicherheitsregeln und -mechanismen zu nutzen. Wenn Sie beispielsweise einen Linux-Benutzer dem SELinux-Benutzer user_u zuordnen, führt dies dazu, dass ein Linux-Benutzer keine Anwendungen wie sudo und su ausführen kann, sofern nicht anders konfiguriert und eine Benutzer-ID (setuid) festgelegt ist.
- SELinux trägt dazu bei, den durch Konfigurationsfehler verursachten Schaden zu mindern.

Domain Name System (DNS)-Server replizieren häufig Informationen untereinander im Rahmen einer Zonenübertragung. Angreifer können Zonenübertragungen nutzen, um DNS-Server mit falschen Informationen zu aktualisieren. Wenn die Berkeley Internet Name Domain (BIND) als DNS-Server in RHEL ausgeführt wird, verhindert die standardmäßige SELinux-Richtlinie Aktualisierungen für Zonendateien [1], die Zonenübertragungen verwenden, selbst wenn ein Administrator vergisst, einzuschränken, welche Server eine Zonenübertragung durchführen können der BIND benannte Daemon selbst und durch andere Prozesse.

- Ohne SELinux kann ein Angreifer eine Schwachstelle zum Path Traversal auf einem Apache-Webserver missbrauchen und über spezielle Elemente wie ../ auf im Dateisystem gespeicherte Dateien und Verzeichnisse zugreifen. Wenn ein Angreifer versucht, einen Server anzugreifen, auf dem SELinux im Erzwingungsmodus (enforcing mode) läuft, verweigert SELinux den Zugriff auf Dateien, auf die der httpd-Prozess nicht zugreifen darf. SELinux kann diese Art von Angriff nicht vollständig blockieren, schwächt ihn jedoch effektiv ab.
- Der boolesche Wert „deny_ptrace SELinux“ und der Modus „SELinux im Erzwingungsmodus“ schützen Systeme vor der Sicherheitslücke PTRACE_TRACEME. Eine solche Konfiguration verhindert Szenarien, in denen ein Angreifer Root-Rechte erlangen kann.
- Die booleschen Werte nfs_export_all_rw und nfs_export_all_ro von SELinux bieten ein benutzerfreundliches Tool, um Fehlkonfigurationen des Network File System (NFS) zu verhindern, wie z. B. die versehentliche Freigabe von /home-Verzeichnissen.

1.4. SELinux: Architektur und Pakete

SELinux ist ein Linux-Sicherheitsmodul (LSM), das in den Linux-Kernel integriert ist. Das SELinux-Subsystem im Kernel wird durch eine Sicherheitsrichtlinie gesteuert, die vom Administrator gesteuert und beim Booten geladen wird. Alle sicherheitsrelevanten Zugriffe auf Kernelebene auf das System werden von SELinux abgefangen und im Kontext der geladenen Sicherheitsrichtlinie untersucht. Wenn die geladene Richtlinie den Vorgang zulässt, wird sie fortgesetzt. Andernfalls wird der Vorgang blockiert und der Prozess erhält einen Fehler.

SELinux-Entscheidungen, wie z. B. das Zulassen oder Verbieten des Zugriffs, werden zwischengespeichert. Dieser Cache wird als Access Vector Cache (AVC) bezeichnet. Bei Verwendung dieser zwischengespeicherten Entscheidungen müssen SELinux-Richtlinienregeln weniger überprüft werden, was die Leistung erhöht. Denken Sie daran, dass SELinux-Richtlinienregeln keine Wirkung haben, wenn DAC-Regeln zuerst den Zugriff verweigern. Rohe Prüfmeldungen werden in /var/log/audit/audit.log protokolliert und beginnen mit der Zeichenfolge „type = AVC“.

In RHEL werden Systemdienste vom systemd-Daemon gesteuert. Systemd startet und stoppt alle Dienste und Benutzer und Prozesse kommunizieren mit systemd über das Dienstprogramm systemctl. Der systemd-Daemon kann die SELinux-Richtlinie konsultieren und die Bezeichnung des aufrufenden Prozesses sowie die Bezeichnung der Unit-Datei überprüfen, die der Aufrufer zu verwalten versucht, und dann SELinux fragen, ob dem Aufrufer der Zugriff gestattet ist oder nicht. Dieser Ansatz stärkt die Zugriffskontrolle auf kritische Systemfunktionen, zu denen das Starten und Stoppen von Systemdiensten gehört.

Der systemd-Daemon fungiert auch als SELinux Access Manager. Er ruft die Bezeichnung des Prozesses ab, der systemctl ausführt, oder des Prozesses, der eine D-Bus-Nachricht an systemd gesendet hat. Der Daemon sucht dann nach der Bezeichnung der Unit-Datei, die der Prozess

konfigurieren wollte. Schließlich kann systemd Informationen vom Kernel abrufen, wenn die SELinux-Richtlinie den spezifischen Zugriff zwischen der Prozessbezeichnung und der Einheitsdateibezeichnung zulässt. Das bedeutet, dass eine kompromittierte Anwendung, die für einen bestimmten Dienst mit systemd interagieren muss, nun von SELinux eingeschränkt werden kann. Richtlinienautoren können diese fein abgestimmten Kontrollen auch nutzen, um Administratoren einzuschränken.

Wenn ein Prozess eine D-Bus-Nachricht an einen anderen Prozess sendet und die SELinux-Richtlinie die D-Bus-Kommunikation dieser beiden Prozesse nicht zulässt, gibt das System eine USER_AVC-Verweigerungsnachricht aus und die D-Bus-Kommunikation läuft ab. Man beachte, dass die D-Bus-Kommunikation zwischen zwei Prozessen bidirektional funktioniert.

Um eine falsche SELinux-Bezeichnung und daraus resultierende Probleme zu vermeiden, stellen Sie sicher, dass Sie Dienste mit einem systemctl-Startbefehl starten.

RHEL 8 stellt die folgenden Pakete für die Arbeit mit SELinux bereit:

- policies: selinux-policy-targeted, selinux-policy-mls
- tools: policycoreutils, policycoreutils-gui, libselinux-utils, policycoreutils-python-utils, setools-console, checkpolicy

1.5. SELinux - Zustände und Modi

SELinux kann in einem von drei Modi ausgeführt werden:

- enforcing (erzwingend)
- permissive (freizügig)
- disabled (deaktiviert)

Die Modi dtailliert:

- Der Erzwingungsmodus ist der standardmäßige und empfohlene Betriebsmodus. Im Durchsetzungsmodus arbeitet SELinux normal und setzt die geladene Sicherheitsrichtlinie auf dem gesamten System durch.
- Im permissiven Modus verhält sich das System so, als würde SELinux die geladene Sicherheitsrichtlinie durchsetzen, einschließlich der Kennzeichnung von Objekten und der Ausgabe von Zugriffsverweigerungseinträgen in den Protokollen, verweigert jedoch tatsächlich keine Vorgänge. Obwohl es für Produktionssysteme nicht empfohlen wird, kann der permissive Modus für die Entwicklung und das Debuggen von SELinux-Richtlinien hilfreich sein.
- Vom deaktivierten Modus wird dringend abgeraten; Das System vermeidet nicht nur die Durchsetzung der SELinux-Richtlinie, sondern vermeidet auch die Kennzeichnung dauerhafter Objekte wie Dateien, was die zukünftige Aktivierung von SELinux erschwert.

Verwenden Sie das Dienstprogramm setenforce, um zwischen dem erzwingenden und dem permissiven Modus zu wechseln. Mit setenforce vorgenommene Änderungen bleiben bei Neustarts nicht bestehen. Um in den Erzwingungsmodus zu wechseln, geben Sie als Linux-Root-Benutzer den

Befehl setenforce 1 ein. Um in den permissiven Modus zu wechseln, geben Sie den Befehl setenforce 0 ein. Verwenden Sie das Dienstprogramm getenforce, um den aktuellen SELinux-Modus anzuzeigen:

```
# getenforce
Enforcing
```

```
# setenforce 0
# getenforce
Permissive
```

```
setenforce 1
getenforce
```

In Red Hat Enterprise Linux können Sie einzelne Domänen in den permissiven Modus versetzen, während das System im erzwingenden Modus läuft. So machen Sie beispielsweise die httpd_t-Domäne permissive:

```
# semanage permissive -a httpd_t
```

Beachten Sie, dass freizügige Domänen ein leistungsstarkes Tool sind, das die Sicherheit Ihres Systems gefährden kann. Red Hat empfiehlt, permissive Domänen mit Vorsicht zu verwenden, beispielsweise beim Debuggen eines bestimmten Szenarios.

2. Ändern von SELinux Zuständen und Modi

Wenn SELinux aktiviert ist, kann es in einem von zwei Modi ausgeführt werden: enforce oder permissive. Die folgenden Abschnitte zeigen, wie Sie dauerhaft in diese Modi wechseln.

2.1. Permanente Änderungen in SELinux

Wie in den SELinux -Zuständen und -Modi erläutert, kann SELinux aktiviert oder deaktiviert werden. Wenn SELinux aktiviert ist, verfügt es über zwei Modi: enforce und permissive. :w

Verwenden Sie die Befehle `getenforce` oder `sestatus`, um zu überprüfen, in welchem Modus SELinux ausgeführt wird. Der Befehl getenforce gibt Enforcing, Permissive oder Disabled zurück.

Der Befehl sestatus gibt den SELinux-Status und die verwendete SELinux-Richtlinie zurück:

```
$ 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
Memory protection checking:     actual (secure)
Max kernel policy version:      31
```

Wenn Systeme SELinux im permissiven Modus ausführen, beschriften Benutzer und Prozesse möglicherweise verschiedene Dateisystemobjekte falsch. Dateisystemobjekte, die erstellt werden, während SELinux deaktiviert ist, werden überhaupt nicht gekennzeichnet. Dieses Verhalten verursacht Probleme beim Wechsel in den Erzwingungsmodus, da SELinux auf korrekte Bezeichnungen von Dateisystemobjekten angewiesen ist.

Um zu verhindern, dass falsch gekennzeichnete und nicht gekennzeichnete Dateien Probleme verursachen, beschriftet SELinux Dateisysteme automatisch neu, wenn vom deaktivierten Zustand in den permissiven oder erzwingenden Modus gewechselt wird. Verwenden Sie den Befehl fixfiles -F onboot als Root, um die Datei /.autorelabel mit der Option -F zu erstellen, um sicherzustellen, dass Dateien beim nächsten Neustart neu gekennzeichnet werden.

Bevor Sie das System zum Umbenennen neu starten, stellen Sie sicher, dass das System im zulässigen Modus startet, indem Sie beispielsweise die Kernel-Option Freizeitcing = 0 verwenden. Dies verhindert, dass das System nicht startet, falls das System unbeschriftete Dateien enthält, die systemd vor dem Start des Selinux-Autorelabel-Dienstes benötigt.

2.2. Wechsel in den permissive Modus

Verwenden Sie das folgende Verfahren, um den SELinux-Modus dauerhaft in „permissiv“ zu ändern. Wenn SELinux im permissiven Modus ausgeführt wird, wird die SELinux-Richtlinie nicht erzwungen. Das System bleibt betriebsbereit und SELinux verweigert keine Vorgänge, sondern protokolliert nur AVC-Meldungen, die dann zur Fehlerbehebung, zum Debuggen und zur Verbesserung der SELinux-Richtlinien verwendet werden können. In diesem Fall wird jeder AVC nur einmal protokolliert.

Voraussetzungen

- Die Pakete „selinux-policy-based“, „libselinux-utils“ und „policycoreutils“ sind auf Ihrem System installiert.
- Die Kernel-Parameter selinux = 0 oder Torscing = 0 werden nicht verwendet.

Vorgehensweise

1. Öffnen Sie die Datei `/etc/selinux/config` in einem Texteditor Ihrer Wahl, zum Beispiel:

```
vi /etc/selinux/config
```

1. Konfigurieren Sie die `SELINUX=permissive` Option:

```
# 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=permissive
# SELINUXTYPE= can take one of these two values:
#       targeted - Targeted processes are protected,
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

1. Danach ist ein Restart erforderlich:

```
# reboot
```

Überprüfung

Bestätigen Sie nach dem Neustart des Systems, dass der Befehl `getenforce` Permissive zurückgibt:

```
$ getenforce
Permissive
```

2.3. Wechsel in den enforcing Modus

Verwenden Sie das folgende Verfahren, um SELinux in den enforcings Modus zu schalten. Wenn SELinux im enforcing Modus ausgeführt wird, erzwingt es die SELinux-Richtlinie und verweigert den Zugriff basierend auf den SELinux-Richtlinienregeln. In RHEL ist der Durchsetzungsmodus standardmäßig aktiviert, wenn das System zum ersten Mal mit SELinux installiert wurde.

Vorgehensweise

1. Öffnen Sie die Datei `/etc/selinux/config` in einem Texteditor Ihrer Wahl, zum Beispiel:

```
vi /etc/selinux/config
```

1. Konfigurieren Sie die `SELINUX=enforcing` Option:

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       enforcing - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.
SELINUX=enforcing
```

```
# SELINUXTYPE= can take one of these two values:
#       targeted - Targeted processes are protected,
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

1. Danach ist ein Restart erforderlich:

```
# reboot
```

Beim nächsten Start beschriftet SELinux alle Dateien und Verzeichnisse im System neu und fügt SELinux-Kontext für Dateien und Verzeichnisse hinzu, die erstellt wurden, als SELinux deaktiviert wurde.

Überprüfung

Bestätigen Sie nach dem Neustart des Systems, dass der Befehl `getenforce` enforcing zurückgibt:

```
$ getenforce
enforcing
```

Nach dem Wechsel in den Erzwingungsmodus verweigert SELinux möglicherweise einige Aktionen aufgrund falscher oder fehlender SELinux-Richtlinienregeln. Um anzuzeigen, welche Aktionen SELinux ablehnt, geben Sie als Root den folgenden Befehl ein:

```
# ausearch -m AVC,USER_AVC,SELINUX_ERR,USER_SELINUX_ERR -ts today
```

Geben Sie alternativ bei installiertem `setroubleshoot-server` Paket Folgendes ein:

```
# grep "SELinux is preventing" /var/log/messages
```

Wenn SELinux aktiv ist und der Audit-Daemon (auditd) nicht auf Ihrem System ausgeführt wird, suchen Sie in der Ausgabe des Befehls dmesg nach bestimmten SELinux-Meldungen:

```
# dmesg | grep -i -e type=1300 -e type=1400
```

2.4. Aktivieren von SELinux auf Systemen, auf denen es zuvor deaktiviert war

Um Probleme zu vermeiden, z. B. dass Systeme nicht mehr starten können oder Prozessfehler auftreten, befolgen Sie dieses Verfahren, wenn Sie SELinux auf Systemen aktivieren, auf denen es zuvor deaktiviert war.

Wenn Systeme SELinux im permissiven Modus ausführen, beschriften Benutzer und Prozesse möglicherweise verschiedene Dateisystemobjekte falsch. Dateisystemobjekte, die erstellt werden, während SELinux deaktiviert ist, werden überhaupt nicht gekennzeichnet. Dieses Verhalten verursacht Probleme beim Wechsel in den Erzwingungsmodus, da SELinux auf korrekte Bezeichnungen von Dateisystemobjekten angewiesen ist. Um zu verhindern, dass falsch gekennzeichnete und nicht gekennzeichnete Dateien Probleme verursachen, beschriftet SELinux Dateisysteme automatisch neu, wenn vom deaktivierten Zustand in den permissiven oder erzwingenden Modus gewechselt wird.

Vorgensweise

1. Aktivieren Sie SELinux im permissiven Modus
2. Rebooten Sie Ihr System

```
# reboot
```

1. Suchen Sie nach SELinux-Ablehnungsmeldungen (denial messages) Details in einem der nächsten Kapitel.
2. Stellen Sie sicher, dass die Dateien beim nächsten Neustart neu gekennzeichnet werden:

```
# fixfiles -F onboot
```

Dadurch wird die Datei `/.autorelabel` erstellt, die die Option `-F` enthält.

Wechseln Sie immer in den permissiven Modus, bevor Sie den Befehl fixfiles -F onboot eingeben. Dies verhindert, dass der Systemstart fehlschlägt, falls das System unbeschriftete Dateien enthält.

1. Wenn es keine Ablehnungen gibt, wechseln Sie in den enforcing mode.

Überprüfung

Bestätigen Sie nach dem Neustart des Systems, dass der Befehl `getenforce` enforcing zurückgibt:

```
$ getenforce
Enforcing
```

Um benutzerdefinierte Anwendungen mit SELinux im Erzwingungsmodus auszuführen, wählen Sie eines der folgenden Szenarios:

- Führen Sie Ihre Anwendung in der Domäne unconfined_service_t aus.
- Schreiben Sie eine neue Richtlinie für Ihre Bewerbung. Weitere Informationen finden Sie im Abschnitt „Schreiben einer benutzerdefinierten SELinux-Richtlinie“.

2.5. SELinux deaktivieren

Verwenden Sie das folgende Verfahren, um SELinux dauerhaft zu deaktivieren.

Wenn SELinux deaktiviert ist, wird die SELinux-Richtlinie überhaupt nicht geladen; Es wird nicht erzwungen und AVC-Nachrichten werden nicht protokolliert. Daher gehen alle Vorteile der Ausführung von SELinux verloren.

Red Hat empfiehlt dringend, den Permissive-Modus zu verwenden, anstatt SELinux dauerhaft zu deaktivieren.

Das Deaktivieren von SELinux mithilfe der Option `SELINUX=disabled` in `/etc/selinux/config` führt zu einem Prozess, bei dem der Kernel mit aktiviertem SELinux startet und später im Startvorgang in den deaktivierten Modus wechselt. Da es zu Speicherlecks und Race-Conditions kommen kann, die zu Kernel-Paniken führen können, sollten Sie SELinux lieber deaktivieren, indem Sie den Parameter „selinux=0“ zur Kernel-Befehlszeile hinzufügen.

Vorgehensweise

1. Öffnen Sie die Datei `/etc/selinux/config` in einem Texteditor Ihrer Wahl, zum Beispiel:

```
vi /etc/selinux/config
```

1. Konfigurieren Sie die `SELINUX=disabled` Option:

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       disabled - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
#       targeted - Targeted processes are protected,
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

1. Danach ist ein Restart erforderlich:

```
# reboot
```

Überprüfung

1. Vergewissern Sie sich nach dem Neustart, dass der Befehl `getenforce` „Deaktiviert“ zurückgibt:

```
$ getenforce
Disabled
```

2.6. Ändern der SELinux-Modi beim Booten

Beim Booten können Sie mehrere Kernel-Parameter festlegen, um die Art und Weise zu ändern, wie SELinux ausgeführt wird:

enforcing=0

Das Festlegen dieses Parameters führt dazu, dass das System im zulässigen Modus startet, was bei der Fehlerbehebung hilfreich ist. Wenn Ihr Dateisystem zu stark beschädigt ist, ist die Verwendung des Permissivmodus möglicherweise die einzige Möglichkeit, ein Problem zu erkennen. Darüber hinaus erstellt das System im Permissivmodus die Etiketten weiterhin korrekt. Die in diesem Modus erstellten AVC-Nachrichten können sich von denen im Erzwingungsmodus unterscheiden.

Im permissiven Modus wird nur die erste Ablehnung aus einer Reihe derselben Ablehnungen gemeldet. Im Durchsetzungsmodus erhalten Sie jedoch möglicherweise eine Ablehnung bezüglich des Lesens eines Verzeichnisses und eine Anwendung wird angehalten. Im permissiven Modus erhalten Sie dieselbe AVC-Meldung, aber die Anwendung liest weiterhin Dateien im Verzeichnis und Sie erhalten zusätzlich eine AVC für jede Ablehnung.

selinux=0

Dieser Parameter bewirkt, dass der Kernel die gesamte SELinux-Infrastruktur nicht lädt. Die Init-Skripte bemerken, dass das System mit dem Parameter „selinux = 0“ gestartet wurde, und aktualisieren die Datei `/.autorelabel`. Dies führt dazu, dass das System beim nächsten Start mit aktiviertem SELinux automatisch eine neue Bezeichnung vornimmt.

Red Hat empfiehlt die Verwendung des Parameters `selinux=0` nicht, um Ihr System zu debuggen, verwenden Sie lieber den permissiven Modus.

autorelabel=1 Dieser Parameter erzwingt eine Umbenennung durch das System, ähnlich wie bei den folgenden Befehlen:

```
# touch /.autorelabel
# reboot
```

Wenn ein Dateisystem eine große Menge falsch gekennzeichneter Objekte enthält, starten Sie das System im permissive Modus, damit der automatische Neukennzeichnungsprozess erfolgreich ist.

3. Verwaltung eingeschränkter und nicht eingeschränkter Benutzer

In den folgenden Abschnitten wird die Zuordnung von Linux-Benutzern zu SELinux-Benutzern

erläutert, die grundlegenden eingeschränkten Benutzerdomänen beschrieben und die Zuordnung eines neuen Benutzers zu einem SELinux-Benutzer veranschaulicht.

3.1. Eingeschränkte und uneingeschränkte Benutzer

Jeder Linux-Benutzer wird mithilfe der SELinux-Richtlinie einem SELinux-Benutzer zugeordnet. Dadurch können Linux-Benutzer die Einschränkungen für SELinux-Benutzer übernehmen.

Um die SELinux -Benutzerzuordnung auf Ihrem System anzuzeigen, verwenden Sie den Befehl `semanage login -l` als Root:

```
# semanage login -l
Login Name           SELinux User         MLS/MCS Range        Service

__default__          unconfined_u         s0-s0:c0.c1023       *
root                 unconfined_u         s0-s0:c0.c1023       *
```

In Red Hat Enterprise Linux werden Linux-Benutzer standardmäßig der SELinux-Standardanmeldung zugeordnet, die dem SELinux-Benutzer `unconfined_u` zugeordnet ist. Die folgende Zeile definiert die Standardzuordnung:

```
__default__          unconfined_u         s0-s0:c0.c1023       *
```

Eingeschränkte Benutzer (Confined users) werden durch SELinux-Regeln eingeschränkt, die explizit in der aktuellen SELinux-Richtlinie definiert sind. Uneingeschränkte (Unconfined) Benutzer unterliegen durch SELinux nur minimalen Einschränkungen.

Eingeschränkte und unbeschränkte Linux-Benutzer unterliegen Prüfungen des ausführbaren und beschreibbaren Speichers und werden auch durch MCS oder MLS eingeschränkt.

Geben Sie den folgenden Befehl ein, um die verfügbaren SELinux-Benutzer aufzulisten:

```
$ seinfo -u
Users: 8
   guest_u
   root
   staff_u
   sysadm_u
   system_u
   unconfined_u
   user_u
   xguest_u
```

Beachten Sie, dass der Befehl `seinfo` vom Paket `setools-console` bereitgestellt wird, das nicht standardmäßig installiert ist.

Wenn ein unbeschränkter Linux-Benutzer eine Anwendung ausführt, die in der SELinux-Richtlinie

als eine Anwendung definiert ist, die von der unconfined_t-Domäne zu ihrer eigenen begrenzten Domäne wechseln kann, unterliegt der unbeschränkte Linux-Benutzer weiterhin den Einschränkungen dieser begrenzten Domäne. Der Sicherheitsvorteil besteht darin, dass die Anwendung eingeschränkt bleibt, auch wenn ein Linux-Benutzer uneingeschränkt ausgeführt wird. Daher kann die Ausnutzung eines Fehlers in der Anwendung durch die Richtlinie eingeschränkt werden.

Ebenso können wir diese Prüfungen auf eingeschränkte Benutzer anwenden. Jeder eingeschränkte Benutzer ist durch eine eingeschränkte Benutzerdomäne eingeschränkt. Die SELinux-Richtlinie kann auch einen Übergang von einer begrenzten Benutzerdomäne zu einer eigenen begrenzten Zieldomäne definieren. In einem solchen Fall unterliegen eingeschränkte Benutzer den Einschränkungen dieser eingeschränkten Zieldomäne. Der Hauptpunkt besteht darin, dass den eingeschränkten Benutzern entsprechend ihrer Rolle besondere Privilegien zugeordnet sind.

3.2. Eingeschränkte Administratorrollen in SELinux

In SELinux gewähren eingeschränkte Administratorrollen den ihnen zugewiesenen Linux-Benutzern bestimmte Privilegien und Berechtigungen zum Ausführen bestimmter Aufgaben. Durch die Zuweisung separater, eingeschränkter Administratorrollen können Sie die Berechtigungen über verschiedene Domänen der Systemadministration auf einzelne Benutzer aufteilen. Dies ist in Szenarios mit mehreren Administratoren nützlich, von denen jeder eine eigene Domäne hat.

SELinux verfügt über die folgenden eingeschränkten Administratorrollen:

auditadm_r

Die Audit-Administratorrolle ermöglicht die Verwaltung des Audit-Subsystems.

dbadm_r

Die Datenbankadministratorrolle ermöglicht die Verwaltung von MariaDB- und PostgreSQL-Datenbanken.

logadm_r

Die Rolle des Protokolladministrators ermöglicht die Verwaltung von Protokollen.

webadm_r

Der Webadministrator ermöglicht die Verwaltung des Apache HTTP Servers.

secadm_r

Die Rolle des Sicherheitsadministrators ermöglicht die Verwaltung der SELinux-Datenbank.

sysadm_r

Die Systemadministratorrolle ermöglicht die Ausführung aller zuvor aufgeführten Rollen und verfügt über zusätzliche Berechtigungen. In nicht standardmäßigen Konfigurationen kann die Sicherheitsverwaltung von der Systemverwaltung getrennt werden, indem das Modul sysadm_secadm in der SELinux-Richtlinie deaktiviert wird.

ZUSÄTZLICHE RESSOURCEN

Weitere Informationen zu den einzelnen Rollen und den zugehörigen Typen finden Sie auf den entsprechenden Manpages:

```
auditadm_selinux(8)
```

```
dbadm_selinux (8)
```

```
logadm_selinux(8)
```

```
webadm_selinux(8)
```

```
secadm_selinux(8)
```

```
sysadm_selinux(8)
```

3.3. Benutzerfunktionen (capabilities)

ZUSÄTZLICHE RESSOURCEN

Die SELinux-Richtlinie ordnet jeden Linux-Benutzer einem SELinux-Benutzer zu. Dadurch können Linux-Benutzer die Einschränkungen von SELinux-Benutzern übernehmen.

Sie können die Berechtigungen für eingeschränkte Benutzer in Ihrer SELinux-Richtlinie an spezifische Anforderungen anpassen, indem Sie boolesche Werte in der Richtlinie anpassen. Sie können den aktuellen Status dieser booleschen Werte ermitteln, indem Sie den Befehl `semanage boolean -l` verwenden. Um alle SELinux-Benutzer, ihre SELinux-Rollen sowie MLS/MCS-Ebenen und Bereiche aufzulisten, verwenden Sie den Befehl `semanage user -l` als Root.

User	Defaul role	Additional roles
unconfined_u	unconfined_r	system_r
guest_u	guest_r	
xguest_u	xguest_r	
user_u	user_r	
staff_u	staff_r	sysadm_r
		unconfined_r
		system_r

User	Defaul role	Additional roles
sysadm_u	sysadm_r	

Beachten Sie, dass system_u eine spezielle Benutzeridentität für Systemprozesse und -objekte ist und system_r die zugehörige Rolle ist. Administratoren dürfen diesen system_u-Benutzer und die system_r-Rolle niemals einem Linux-Benutzer zuordnen. Darüber hinaus sind unconfined_u und root unbeschränkte Benutzer. Aus diesen Gründen sind die diesen SELinux-Benutzern zugeordneten Rollen nicht in der folgenden Tabelle „Typen und Zugriff von SELinux-Rollen“ enthalten.

Jede SELinux-Rolle entspricht einem SELinux-Typ und stellt spezifische Zugriffsrechte bereit.

Role	Type	Log in using X Window System	su and sudo	Execute in home directory and /tmp	Networking
unconfined_r	unconfined_t	yes	yes	yes	yes
guest_r	guest_t	no	no	yes	no
xguest_r	xguest_t	yes	no	yes	web browsers only
user_r	user_t	yes	no	yes	yes
staff_r	staff_t	yes	only sudo	yes	yes
auditadm_r	auditadm_t		yes	yes	yes
secadm_r	secadm_t		yes	yes	yes

- Linux-Benutzer in den Domänen „user_t“, „guest_t“ und „xguest_t“ können Anwendungen mit festgelegter Benutzer-ID (setuid) nur ausführen, wenn die SELinux-Richtlinie dies zulässt (z. B. passwd). Diese Benutzer können die Anwendungen su und sudo setuid nicht ausführen und können diese Anwendungen daher nicht verwenden, um Root zu werden. Linux-Benutzer in den Domänen sysadm_t, staff_t, user_t und xguest_t können sich über das X Window System und ein Terminal anmelden. Standardmäßig können Linux-Benutzer in den Domänen „staff_t“, „user_t“, „guest_t“ und „xguest_t“ Anwendungen in ihren Home-Verzeichnissen und in /tmp ausführen.
- Um zu verhindern, dass sie in Verzeichnissen, auf die sie Schreibzugriff haben, Anwendungen ausführen, die Benutzerberechtigungen erben, setzen Sie die booleschen Werte „guest_exec_content“ und „xguest_exec_content“ auf „off“. Dadurch wird verhindert, dass fehlerhafte oder bösartige Anwendungen die Dateien der Benutzer ändern.
- Der einzige Netzwerkzugriff, den Linux-Benutzer in der xguest_t-Domäne haben, ist Firefox, der eine Verbindung zu Webseiten herstellt. Der Benutzer sysadm_u kann sich nicht direkt über SSH anmelden. Um SSH-Anmeldungen für sysadm_u zu aktivieren, setzen Sie den booleschen Wert ssh_sysadm_login auf „on“:

```
# setsebool -P ssh_sysadm_login on
```

Neben den bereits erwähnten SELinux-Benutzern gibt es spezielle Rollen, die diesen Benutzern mit dem Befehl semanage user zugeordnet werden können. Diese Rollen bestimmen, was SELinux dem Benutzer erlaubt:

- webadm_r kann nur SELinux-Typen verwalten, die sich auf den Apache HTTP Server beziehen.
- dbadm_r kann nur SELinux-Typen verwalten, die sich auf die MariaDB-Datenbank und das PostgreSQL-Datenbankverwaltungssystem beziehen.
- logadm_r kann nur SELinux-Typen verwalten, die sich auf die Syslog- und Auditlog-Prozesse beziehen.
- secadm_r kann nur SELinux verwalten.
- auditadm_r kann nur Prozesse verwalten, die sich auf das Audit-Subsystem beziehen.

Um alle verfügbaren Rollen aufzulisten, geben Sie den Befehl `seininfo -r` ein:

```
$ seinfo -r
Roles: 14
   auditadm_r
   dbadm_r
   guest_r
   logadm_r
   nx_server_r
   object_r
   secadm_r
   staff_r
   sysadm_r
   system_r
   unconfined_r
   user_r
   webadm_r
   xguest_r
```

Beachten Sie, dass der Befehl seinfo vom Paket setools-console bereitgestellt wird, das nicht standardmäßig installiert ist.

ZUSÄTZLICHE RESSOURCEN

Die Manpages seinfo (1), semanage-login (8) und xguest_selinux (8).

3.4. Das Hinzufügen eines neuen Benutzers wird automatisch dem SELinux-Benutzer unconfined_u zugeordnet

Das folgende Verfahren zeigt, wie Sie dem System einen neuen Linux-Benutzer hinzufügen. Der Benutzer wird automatisch dem SELinux-Benutzer unconfined_u zugeordnet.

Voraussetzungen

1. Der Root-Benutzer läuft uneingeschränkt, wie es standardmäßig in Red Hat Enterprise Linux der Fall ist.

Vorgehensweise

1. Geben Sie den folgenden Befehl ein, um einen neuen Linux-Benutzer mit dem Namen example.user zu erstellen:

```
# useradd example.user
```

1. So weisen Sie dem Linux-Benutzer example.user ein Passwort zu:

```
# passwd example.user
Changing password for user example.user.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

1. Melden Sie sich von Ihrer aktuellen Sitzung ab.

Melden Sie sich als Linux-Benutzer example.user an. Wenn Sie sich anmelden, ordnet das PAM-Modul pam_selinux den Linux-Benutzer automatisch einem SELinux-Benutzer zu (in diesem Fall unconfined_u) und richtet den resultierenden SELinux-Kontext ein. Mit diesem Kontext wird dann die Shell des Linux-Benutzers gestartet.

Überprüfung

1. Wenn Sie als Benutzer example.user angemeldet sind, überprüfen Sie den Kontext eines Linux-Benutzers:

```
$ id -Z
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

ZUSÄTZLICHE RESSOURCEN

Pam_selinux (8) Manpage.

3.5. Hinzufügen eines neuen Benutzers als auf SELinux beschränkter Benutzer

Führen Sie die folgenden Schritte aus, um dem System einen neuen, auf SELinux beschränkten Benutzer hinzuzufügen. In diesem Beispielverfahren wird der Benutzer mit dem Befehl zum Erstellen des Benutzerkontos dem SELinux-Benutzerrecht „staff_u“ zugeordnet.

Voraussetzungen

- Der Root-Benutzer läuft uneingeschränkt, wie es standardmäßig in Red Hat Enterprise Linux der Fall ist.

Vorgehensweise

1. Geben Sie den folgenden Befehl ein, um einen neuen Linux-Benutzer mit dem Namen example.user zu erstellen und ihn dem SELinux-Benutzer „staff_u“ zuzuordnen:

```
# useradd -Z staff_u example.user
```

1. So weisen Sie dem Linux-Benutzer example.user ein Passwort zu:

```
# passwd example.user
Changing password for user example.user.
New password:
Retype new password:
passwd: all authentication tokens updated successfully
```

1. Melden Sie sich von Ihrer aktuellen Sitzung ab.
2. Melden Sie sich als Linux-Benutzer example.user an. Die Shell des Benutzers wird mit dem Kontext „staff_u“ gestartet.

Überprüfung

1. Wenn Sie als Benutzer example.user angemeldet sind, überprüfen Sie den Kontext eines Linux-Benutzers:

```
$ id -Z
uid=1000(example.user) gid=1000(example.user) groups=1000(example.user)
context=staff_u:staff_r:staff_t:s0-s0:c0.c1023
```

ZUSÄTZLICHE RESSOURCEN

Pam_selinux (8) Manpage.

3.6. Confining (Beschränkung) regulärer Benutzer

Sie können alle regulären Benutzer auf Ihrem System einschränken, indem Sie sie dem SELinux-Benutzer user_u zuordnen.

Standardmäßig werden alle Linux-Benutzer in Red Hat Enterprise Linux, einschließlich Benutzern mit Administratorrechten, dem uneingeschränkten SELinux-Benutzer unconfined_u zugeordnet. Sie können die Sicherheit des Systems verbessern, indem Sie Benutzer SELinux-beschränkten Benutzern zuweisen.

Vorgehensweise

1. Zeigen Sie die Liste der SELinux-Anmeldedatensätze an. Die Liste zeigt die Zuordnungen von Linux-Benutzern zu SELinux-Benutzern:

```
# semanage login -l
Login Name    SELinux User  MLS/MCS Range   Service
__default__   unconfined_u  s0-s0:c0.c1023       *
root          unconfined_u  s0-s0:c0.c1023       *
```

1. Ordnen Sie den Benutzer *default*, der alle Benutzer ohne explizite Zuordnung darstellt, dem SELinux-Benutzer user_u zu:

```
# semanage login -m -s user_u -r s0 __default__
```

Überprüfung

1. Überprüfen Sie, ob der Benutzer *default* dem SELinux-Benutzer user_u zugeordnet ist:

```
# semanage login -l

Login Name    SELinux User   MLS/MCS Range    Service

__default__   user_u         s0               *
root          unconfined_u   s0-s0:c0.c1023   *
```

Stellen Sie sicher, dass die Prozesse eines neuen Benutzers im SELinux-Kontext user_u: user_r: user_t: s0 ausgeführt werden.

Erstellen Sie einen neuen Benutzer:

```
# adduser example.user
```

Definieren Sie ein Passwort für example.user:

```
# Passwd example.user
```

Melden Sie sich als Root ab und als neuer Benutzer an. Zeigen Sie den Sicherheitskontext für die Benutzer-ID an:

```
[example.user@localhost ~]$ id -Z
user_u:user_r:user_t:s0
```

Zeigen Sie den Sicherheitskontext der aktuellen Prozesse des Benutzers an:

```
[example.user@localhost ~]$ ps axZ
LABEL                            PID TTY      STAT   TIME COMMAND
-                                  1 ?        Ss     0:05 /usr/lib/systemd/systemd
--switched-root --system --deserialize 18
-                               3729 ?        S      0:00 (sd-pam)
user_u:user_r:user_t:s0         3907 ?        Ss     0:00 /usr/lib/systemd/systemd
--user
-                               3911 ?        S      0:00 (sd-pam)
user_u:user_r:user_t:s0         3918 ?        S      0:00 sshd: example.user@pts/0
user_u:user_r:user_t:s0         3922 pts/0    Ss     0:00 -bash
user_u:user_r:user_dbusd_t:s0   3969 ?        Ssl    0:00 /usr/bin/dbus-daemon
--session --address=systemd: --nofork --nopidfile --systemd-activation --syslog-only
user_u:user_r:user_t:s0         3971 pts/0    R+     0:00 ps axZ
```

3.7. Confining eines Administrators durch Zuordnung zum sysadm_u

Sie können einen Benutzer mit Administratorrechten einschränken, indem Sie den Benutzer direkt dem SELinux-Benutzer sysadm_u zuordnen. Wenn sich der Benutzer anmeldet, wird die Sitzung im SELinux-Kontext sysadm_u: sysadm_r: sysadm_t ausgeführt.

Standardmäßig werden alle Linux-Benutzer in Red Hat Enterprise Linux, einschließlich Benutzern mit Administratorrechten, dem uneingeschränkten SELinux-Benutzer unconfined_u zugeordnet. Sie können die Sicherheit des Systems verbessern, indem Sie Benutzer SELinux-beschränkten Benutzern zuweisen.

Voraussetzung

- Der Root-Benutzer läuft uneingeschränkt. Dies ist die Standardeinstellung von Red Hat Enterprise Linux.

Vorgehensweise

1. Optional: Um sysadm_u-Benutzern die Verbindung mit dem System über SSH zu ermöglichen:

```
# setsebool -P ssh_sysadm_login on
```

1. Erstellen Sie einen neuen Benutzer, fügen Sie den Benutzer der Wheel-Benutzergruppe hinzu und ordnen Sie den Benutzer dem SELinux-Benutzer sysadm_u zu:

```
# adduser -G wheel -Z sysadm_u example.user
```

Optional: Ordnen Sie einen vorhandenen Benutzer dem SELinux-Benutzer sysadm_u zu und fügen Sie den Benutzer der Wheel-Benutzergruppe hinzu:

```
# Usermod -G Wheel -Z sysadm_u example.user
```

Überprüfung

1. Check that example.user is mapped to the sysadm_u SELinux user:

```
# semanage login -l | grep example.user
example.user     sysadm_u    s0-s0:c0.c1023   *
```

1. Melden Sie sich beispielsweise über SSH als example.user an und zeigen Sie den Sicherheitskontext des Benutzers an:

```
example.user@localhost ~]$ id -Z
sysadm_u:sysadm_r:sysadm_t:s0-s0:c0.c1023
```

1. Wechseln Sie zum Root-Benutzer:

```
$ Sudo -i
[Sudo] Passwort für example.user:
```

1. Stellen Sie sicher, dass der Sicherheitskontext unverändert bleibt:

```
# Id -Z
Sysadm_u: sysadm_r: sysadm_t: s0-s0: c0.c1023
```

1. Versuchen Sie es mit einer Verwaltungsaufgabe, zum Beispiel dem Neustart des SSHD-Dienstes:

```
# systemctl restart sshd
```

Erfolgt keine Ausgabe, wurde der Befehl erfolgreich abgeschlossen.

Wenn der Befehl nicht erfolgreich abgeschlossen wird, wird die folgende Meldung ausgegeben:

```
Failed to restart sshd.service: Access denied
See system logs and 'systemctl status sshd.service' for details.
```

3.8. Confining eines Administrators mit sudo und der rule sysadm_r

Sie können dem SELinux-Benutzer „staff_u“ einen bestimmten Benutzer mit Administratorrechten zuordnen und sudo so konfigurieren, dass der Benutzer die SELinux-Administratorrolle „sysadm_r“ erhalten kann. Mit dieser Rolle kann der Benutzer Verwaltungsaufgaben ohne SELinux-Ablehnungen ausführen. Wenn sich der Benutzer anmeldet, wird die Sitzung im SELinux-Kontext „staff_u: staff_r: staff_t“ ausgeführt. Wenn der Benutzer jedoch mit „sudo“ einen Befehl eingibt, wechselt die Sitzung in den Kontext „staff_u: sysadm_r: sysadm_t“.

Standardmäßig werden alle Linux-Benutzer in Red Hat Enterprise Linux, einschließlich Benutzern mit Administratorrechten, dem uneingeschränkten SELinux-Benutzer unconfined_u zugeordnet. Sie können die Sicherheit des Systems verbessern, indem Sie Benutzer SELinux-beschränkten Benutzern zuweisen.

Voraussetzungen

- Der Root-Benutzer läuft uneingeschränkt. Dies ist die Standardeinstellung von Red Hat Enterprise Linux. Verfahren
  1. Erstellen Sie einen neuen Benutzer, fügen Sie den Benutzer der Wheel-Benutzergruppe hinzu und ordnen Sie den Benutzer dem SELinux-Benutzer „staff_u“ zu:

```
e# adduser -G wheel -Z staff_u xample.user
```

1. Optional: Ordnen Sie einen vorhandenen Benutzer dem SELinux-Benutzer „staff_u“ zu und fügen Sie den Benutzer der Wheel-Benutzergruppe hinzu:

```
# usermod -G wheel -Z staff_u example.user
```

1. Damit example.user die SELinux-Administratorrolle erhalten kann, erstellen Sie eine neue Datei im Verzeichnis /etc/sudoers.d/, zum Beispiel:

```
# visudo -f /etc/sudoers.d/example.user
```

1. Fügen Sie der neuen Datei die folgende Zeile hinzu:

```
example.user ALL=(ALL) TYPE=sysadm_t ROLE=sysadm_r ALL
```

Überprüfung

1. Überprüfen Sie, ob example.user dem SELinux-Benutzer „staff_u“ zugeordnet ist:

```
# semanage login -l | grep example.user
example.user     staff_u    s0-s0:c0.c1023   *
```

1. Melden Sie sich beispielsweise über SSH als example.user an und wechseln Sie zum Root-Benutzer:

```
[example.user@localhost ~]$ sudo -i
[sudo] password for example.user:
```

1. Zeigen Sie den Root-Sicherheitskontext an:

```
# id -Z
staff_u:sysadm_r:sysadm_t:s0-s0:c0.c1023
```

1. Versuchen Sie es mit einer Verwaltungsaufgabe, zum Beispiel dem Neustart des SSHD-Dienstes:

```
# systemctl restart sshd
```

Erfolgt keine Ausgabe, wurde der Befehl erfolgreich abgeschlossen.

Wenn der Befehl nicht erfolgreich abgeschlossen wird, wird die folgende Meldung ausgegeben:

```
Failed to restart sshd.service: Access denied
See system logs and 'systemctl status sshd.service' for details.
```

4. Konfigurieren von SELinux für Anwendungen und Dienste mit nicht standardmäßigen Konfigurationen

Wenn sich SELinux im enforcing mode befindet, ist die Standardrichtlinie die Zielrichtlinie. Die folgenden Abschnitte enthalten Informationen zum Einrichten und Konfigurieren der SELinux-Richtlinie für verschiedene Dienste, nachdem Sie Konfigurationsstandardwerte wie Ports, Datenbankspeicherorte oder Dateisystemberechtigungen für Prozesse geändert haben.

Sie lernen, SELinux-Typen für nicht standardmäßige Ports zu ändern, falsche Beschriftungen für Änderungen von Standardverzeichnissen zu identifizieren und zu beheben und die Richtlinie mithilfe von SELinux-Booleschen Werten anzupassen.

4.1. Anpassen der SELinux-Richtlinie für den Apache HTTP-Server in einer nicht standardmäßigen Konfiguration

Sie können den Apache-HTTP-Server so konfigurieren, dass er einen anderen Port überwacht und Inhalte in einem nicht standardmäßigen Verzeichnis bereitstellt. Um Folgeverweigerungen von SELinux zu verhindern, befolgen Sie die Schritte in diesem Verfahren, um die SELinux-Richtlinie Ihres Systems anzupassen.

Voraussetzungen

- Das httpd-Paket ist installiert und der Apache-HTTP-Server ist so konfiguriert, dass er den TCP-Port 3131 überwacht und das Verzeichnis/var/test_www/anstelle des Standardverzeichnisses/var/www/verwendet.
- Die Pakete „policycoreutils-python-utils“ und „setroubleshoot-server“ sind auf Ihrem System installiert.

Vorgehensweise

1. Starten Sie den httpd-Dienst und prüfen Sie den Status:

```
# systemctl start httpd
# systemctl status httpd
...
httpd[14523]: (13)Permission denied: AH00072: make_sock: could not bind to address
[::]:3131
...
systemd[1]: Failed to start The Apache HTTP Server.
```

1. Die SELinux-Richtlinie geht davon aus, dass httpd auf Port 80 ausgeführt wird:

```
# semanage port -l | grep http
http_cache_port_t              tcp      8080, 8118, 8123, 10001-10010
http_cache_port_t              udp      3130
http_port_t                    tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t            tcp      5988
pegasus_https_port_t           tcp      5989
```

1. Ändern Sie den SELinux-Typ von Port 3131 so, dass er mit Port 80 übereinstimmt: ...

```
# semanage port -a -t http_port_t -p tcp 3131
```

1. Starten Sie httpd erneut:

```
# systemctl start httpd
```

1. Der Inhalt bleibt jedoch unzugänglich:

```
# wget localhost:3131/index.html
...
HTTP request sent, awaiting response... 403 Forbidden
...
```

1. Finden Sie den Grund mit dem Sealert-Tool:

```
# sealert -l "*"
...
SELinux is preventing httpd from getattr access on the file
/var/test_www/html/index.html.
...
```

1. Vergleichen Sie SELinux-Typen für den Standard- und den neuen Pfad mit dem Tool matchpathcon:

```
# matchpathcon /var/www/html /var/test_www/html
/var/www/html       system_u:object_r:httpd_sys_content_t:s0
/var/test_www/html  system_u:object_r:var_t:s0
```

1. Ändern Sie den SELinux-Typ des neuen Inhaltsverzeichnisses/var/test_www/html/in den Typ des Standardverzeichnisses/var/www/html:

```
# semanage fcontext -a -e /var/www /var/test_www
```

1. Benennen Sie das Verzeichnis /var rekursiv um:

```
# restorecon -Rv /var/
...
Relabeled /var/test_www/html from unconfined_u:object_r:var_t:s0 to
unconfined_u:object_r:httpd_sys_content_t:s0
Relabeled /var/test_www/html/index.html from unconfined_u:object_r:var_t:s0 to
unconfined_u:object_r:httpd_sys_content_t:s0
```

Überprüfung

1. Überprüfen Sie, ob der httpd-Dienst ausgeführt wird:

```
# systemctl status httpd
```

```
...
Active: active (running)
...
systemd[1]: Started The Apache HTTP Server.
httpd[14888]: Server configured, listening on: port 3131
...
```

1. Stellen Sie sicher, dass auf den vom Apache-HTTP-Server bereitgestellten Inhalt zugegriffen werden kann:

```
# wget localhost:3131/index.html
...
HTTP request sent, awaiting response... 200 OK
Length: 0 [text/html]
Saving to: 'index.html'
...
```

4.2. Anpassen der Richtlinie für die gemeinsame Nutzung von NFS- und CIFS-Volumes mithilfe boolescher SELinux-Werte

Sie können Teile der SELinux-Richtlinie zur Laufzeit mithilfe von Booleschen Werten ändern, auch ohne Kenntnisse über das Schreiben von SELinux-Richtlinien. Dies ermöglicht Änderungen, z. B. das Ermöglichen des Dienstzugriffs auf NFS-Volumes, ohne die SELinux-Richtlinie neu laden oder neu kompilieren zu müssen. Das folgende Verfahren veranschaulicht das Auflisten von SELinux-Booleschen Werten und deren Konfiguration, um die erforderlichen Änderungen in der Richtlinie zu erreichen.

NFS-Mounts auf der Clientseite werden mit einem Standardkontext gekennzeichnet, der durch eine Richtlinie für NFS-Volumes definiert wird. In RHEL verwendet dieser Standardkontext den Typ nfs_t. Darüber hinaus werden auf der Clientseite gemountete Samba-Freigaben mit einem durch die Richtlinie definierten Standardkontext gekennzeichnet. Dieser Standardkontext verwendet den Typ cifs_t. Sie können boolesche Werte aktivieren oder deaktivieren, um zu steuern, welche Dienste auf die Typen nfs_t und cifs_t zugreifen dürfen.

Führen Sie die folgenden Schritte aus, um dem Apache-HTTP-Serverdienst (httpd) den Zugriff auf und die Freigabe von NFS- und CIFS-Volumes zu ermöglichen:

Voraussetzungen

- Installieren Sie optional das Paket selinux-policy-devel, um klarere und detailliertere Beschreibungen der booleschen Werte von SELinux in der Ausgabe des Befehls semanage boolean -l zu erhalten. Verfahren

Vorgehemsweise

1. Identifizieren Sie SELinux-Boolesche Werte, die für NFS, CIFS und Apache relevant sind:

```
# semanage boolean -l | grep 'nfs\|cifs' | grep httpd
httpd_use_cifs                (off  ,  off)  Allow httpd to access cifs file systems
httpd_use_nfs                 (off  ,  off)  Allow httpd to access nfs file systems
```

1. Listen Sie den aktuellen Status der booleschen Werte auf:

```
$ getsebool -a | grep 'nfs\|cifs' | grep httpd
httpd_use_cifs --> off
httpd_use_nfs --> off
```

. Aktivieren Sie die identifizierten booleschen Werte:

```
# setsebool httpd_use_nfs on
# setsebool httpd_use_cifs on
```

Verwenden Sie „setsebool“ mit der Option „-P“, um die Änderungen über Neustarts hinweg dauerhaft zu machen. Der Befehl „setsebool -P“ erfordert eine Neuerstellung der gesamten Richtlinie. Dies kann je nach Konfiguration einige Zeit dauern.

Überprüfung

1. Überprüfen Sie, ob die booleschen Werte aktiviert sind:

```
$ getsebool -a | grep 'nfs\|cifs' | grep httpd
httpd_use_cifs --> on
httpd_use_nfs --> on
```

4.3. Finden des richtigen SELinux-Typs zum Verwalten des Zugriffs auf nicht standardmäßige Verzeichnisse

Wenn Sie Zugriffskontrollregeln festlegen müssen, die von der Standard-SELinux-Richtlinie nicht abgedeckt werden, suchen Sie zunächst nach einem Boolean-Wert, der Ihrem Anwendungsfall entspricht. Wenn Sie keinen passenden Boolean-Wert finden, können Sie einen passenden SELinux-Typ verwenden oder sogar ein lokales Richtlinienmodul erstellen.

Voraussetzungen

- Die Pakete selinux-policy-doc und setools-console sind auf Ihrem System installiert.

Vorgehensweise

1. Listen Sie alle SELinux-bezogenen Themen auf und beschränken Sie die Ergebnisse auf eine Komponente, die Sie konfigurieren möchten. Beispiel:

```
# man -k selinux | grep samba
samba_net_selinux (8) - Security Enhanced Linux Policy for the samba_net processes
samba_selinux (8)    - Security Enhanced Linux Policy for the smbd processes
…
```

1. Optional: Sie können die Standardzuordnung von Prozessen an Standardspeicherorten anzeigen, indem Sie den Befehl „semanage fcontext -l“ verwenden, zum Beispiel:

```
# semanage fcontext -l | grep samba
…
/var/cache/samba(/.*)?                             all files
system_u:object_r:samba_var_t:s0
…
/var/spool/samba(/.*)?                             all files
system_u:object_r:samba_spool_t:s0
```

1. Verwenden Sie den Befehl sesearch, um Regeln in der Standard-SELinux-Richtlinie anzuzeigen. Sie können den zu verwendenden Typ und Boolean-Wert finden, indem Sie die entsprechende Regel auflisten, zum Beispiel:

```
$ sesearch -A | grep samba | grep httpd
…
allow httpd_t cifs_t:dir { getattr open search }; [ use_samba_home_dirs &&
httpd_enable_homedirs ]:True
…
```

1. Ein SELinux-Boolescher Wert ist möglicherweise die einfachste Lösung für Ihr Konfigurationsproblem. Sie können alle verfügbaren Booleschen Werte und deren Werte mit dem Befehl getsebool -a anzeigen, zum Beispiel:

```
$ getsebool -a | grep homedirs
git_cgi_enable_homedirs --> off
git_system_enable_homedirs --> off
httpd_enable_homedirs --> off
mock_enable_homedirs --> off
mpd_enable_homedirs --> off
openvpn_enable_homedirs --> on
ssh_chroot_rw_homedirs --> off
```

1. Sie können mit dem Befehl „sesearch“ überprüfen, ob der ausgewählte Boolesche Wert genau das gewünschte Ergebnis liefert. Beispiel:

```
$ sesearch -A | grep httpd_enable_homedirs
…
allow httpd_suexec_t autofs_t:dir { getattr open search }; [ use_nfs_home_dirs &&
```

```
httpd_enable_homedirs ]:True
allow httpd_suexec_t autofs_t:dir { getattr open search }; [ use_samba_home_dirs &&
httpd_enable_homedirs ]:True
...
```

1. Wenn kein Boolean-Wert zu Ihrem Szenario passt, suchen Sie nach einem SELinux-Typ, der für Ihren Fall geeignet ist. Sie können einen Typ für Ihre Dateien finden, indem Sie mit sesearch eine entsprechende Regel aus der Standardrichtlinie abfragen, zum Beispiel:

```
$ sesearch -A -s httpd_t -c file -p read
...
allow httpd_t httpd_t:file { append getattr ioctl lock open read write };
allow httpd_t httpd_tmp_t:file { append create getattr ioctl link lock map open
```

1. Wenn keine der vorherigen Lösungen Ihr Szenario abdeckt, können Sie der SELinux-Richtlinie eine benutzerdefinierte Regel hinzufügen. Weitere Informationen finden Sie im Abschnitt Erstellen eines lokalen SELinux-Richtlinienmoduls.

4.4. Verwalten des Zugriffs auf nicht standardmäßige freigegebene Verzeichnisse für nicht privilegierte SELinux-Benutzer

Sie können den Zugriff auf ein nicht standardmäßiges freigegebenes Verzeichnis für den generischen, nicht privilegierten SELinux-Benutzer user_u konfigurieren, indem Sie den entsprechenden SELinux-Dateityp suchen und zuordnen. Der Benutzer user_u hat die Standardrolle user_r und die Standarddomäne user_t.

Voraussetzungen

Die Pakete selinux-policy-doc und setools-console sind auf Ihrem System installiert.

Vorgehensweise

- Öffnen Sie die Manpage user_selinux(8) in Ihrem Terminal:

```
$ man user_selinux
```

Suchen Sie im Abschnitt VERWALTETE DATEIEN nach einem Attribut oder Typ, der Ihrem Szenario entspricht. Beispielsweise das Attribut user_home_type.

- Optional: Um alle einem Attribut zugewiesenen Typen aufzulisten, verwenden Sie den Befehl seinfo mit den Optionen -x und -a, beispielsweise:

```
$ seinfo -x -a user_home_type
```

```
Type Attributes: 1
   attribute user_home_type;
...
    chrome_sandbox_home_t
    config_home_t
    cvs_home_t
    data_home_t
    dbus_home_t
    fetchmail_home_t
    gconf_home_t
    git_user_content_t
...
```

- Nachdem Sie einen Kandidaten für den entsprechenden Typ identifiziert haben, in diesem Beispiel den Typ data_home_t, überprüfen Sie seine SELinux-Zuordnung:

```
$ semanage fcontext -l | grep data_home_t
...
/root/\.local/share(/.*)?                         all files
system_u:object_r:da
```

- Map the corresponding type to a directory that you want to make accessible for user_u, for example, /shared-data:

```
$ semanage fcontext -a -t data_home_t '/shared-data(/.*)?'
```

Überprüfung

- Überprüfen Sie die Zuordnung des von Ihnen konfigurierten Verzeichnisses:

```
# semanage fcontext -l | grep "shared-data"
/shared-data(/.*)?                                all files
system_u:object_r:data_h
```

- Melden Sie sich als Linux-Benutzer an, der dem SELinux-Benutzer user_u zugeordnet ist, und überprüfen Sie, ob Sie auf das Verzeichnis zugreifen können.

5. Beheben von Problemen im Zusammenhang mit SELinux

Wenn Sie SELinux auf Systemen aktivieren möchten, auf denen es zuvor deaktiviert wurde, oder wenn Sie einen Dienst in einer nicht standardmäßigen Konfiguration ausführen, müssen Sie möglicherweise Situationen beheben, die möglicherweise durch SELinux blockiert werden. Beachten Sie, dass SELinux-Ablehnungen in den meisten Fällen Anzeichen einer Fehlkonfiguration

sind.

5.1. Identifizieren von SELinux-Ablehnungen

Befolgen Sie nur die notwendigen Schritte dieses Verfahrens. In den meisten Fällen müssen Sie nur Schritt 1 ausführen.

Vorgehensweise

- Wenn Ihr Szenario von SELinux blockiert wird, ist die Datei /var/log/audit/audit.log der erste Ort, an dem Sie nach weiteren Informationen zu einer Ablehnung suchen können. Verwenden Sie zum Abfragen von Audit-Protokollen das Tool ausearch. Da die SELinux-Entscheidungen, z. B. das Zulassen oder Verbieten des Zugriffs, zwischengespeichert werden und dieser Cache als Access Vector Cache (AVC) bezeichnet wird, verwenden Sie die Werte AVC und USER_AVC für den Nachrichtentypparameter, z. B.:

```
# ausearch -m AVC,USER_AVC,SELINUX_ERR,USER_SELINUX_ERR -ts recent
```

Wenn es keine Übereinstimmungen gibt, prüfen Sie, ob der Audit-Daemon ausgeführt wird. Wenn nicht, wiederholen Sie das abgelehnte Szenario, nachdem Sie auditd gestartet haben, und prüfen Sie das Audit-Protokoll erneut.

- Falls auditd ausgeführt wird, aber keine Übereinstimmungen in der Ausgabe von ausearch vorliegen, prüfen Sie die vom systemd-Journal bereitgestellten Nachrichten:

```
journalctl -t setroubleshoot
```

- Wenn SELinux aktiv ist und der Audit-Daemon nicht auf Ihrem System läuft, suchen Sie in der Ausgabe des dmesg-Befehls nach bestimmten SELinux-Meldungen:

```
dmesg | grep -i -e type=1300 -e type=1400
```

- Auch nach den vorherigen drei Prüfungen ist es möglich, dass Sie nichts gefunden haben. In diesem Fall können AVC-Ablehnungen aufgrund von Dontaudit-Regeln unterdrückt werden.

So deaktivieren Sie Dontaudit-Regeln vorübergehend, sodass alle Ablehnungen protokolliert werden:

```
semodule -DB
```

Nachdem Sie Ihr abgelehntes Szenario erneut ausgeführt und mithilfe der vorherigen Schritte Ablehnungsmeldungen gefunden haben, aktiviert der folgende Befehl die Dontaudit-Regeln in der Richtlinie wieder:

```
semodule -B
```

- Wenn Sie alle vier vorherigen Schritte ausführen und das Problem weiterhin nicht identifiziert wird, prüfen Sie, ob SELinux Ihr Szenario wirklich blockiert:

Wechseln Sie in den permissiven Modus:

```
# setenforce 0
$ getenforce
Permissive
```

Wenn das Problem weiterhin auftritt, blockiert etwas anderes als SELinux Ihr Szenario.

5.2. Analysieren von SELinux-Ablehnungsmeldungen

Nachdem Sie festgestellt haben, dass SELinux Ihr Szenario blockiert, müssen Sie möglicherweise die Grundursache analysieren, bevor Sie eine Lösung auswählen.

Voraussetzungen

Die Pakete policycoreutils-python-utils und setroubleshoot-server sind auf Ihrem System installiert.

Vorgehensweise

- Zeigen Sie weitere Details zu einer protokollierten Ablehnung mit dem Befehl sealert auf, zum Beispiel:

```
$ sealert -l "*"
SELinux is preventing /usr/bin/passwd from write access on the file
/root/test.

*****  Plugin leaks (86.2 confidence) suggests ****************************

If you want to ignore passwd trying to write access the test file,
because you believe it should not need this access.
Then you should report this as a bug.
You can generate a local policy module to dontaudit this access.
Do
# ausearch -x /usr/bin/passwd --raw | audit2allow -D -M my-passwd
# semodule -X 300 -i my-passwd.pp

*****  Plugin catchall (14.7 confidence) suggests **************************

...

Raw Audit Messages
type=AVC msg=audit(1553609555.619:127): avc:  denied  { write } for
pid=4097 comm="passwd" path="/root/test" dev="dm-0" ino=17142697
```

```
scontext=unconfined_u:unconfined_r:passwd_t:s0-s0:c0.c1023
tcontext=unconfined_u:object_r:admin_home_t:s0 tclass=file permissive=0

...

Hash: passwd,passwd_t,admin_home_t,file,write
```

Wenn die im vorherigen Schritt erhaltene Ausgabe keine klaren Vorschläge enthält:

- Aktivieren Sie die vollständige Pfadprüfung, um vollständige Pfade zu aufgerufenen Objekten anzuzeigen und zusätzliche Linux Audit-Ereignisfelder sichtbar zu machen:

```
auditctl -w /etc/shadow -p w -k shadow-write
```

Leeren Sie den Setroubleshoot-Cache:

```
rm -f /var/lib/setroubleshoot/setroubleshoot.xml
```

Reproduzieren Sie das Problem. Wiederholen Sie Schritt 1.

Nachdem Sie den Vorgang abgeschlossen haben, deaktivieren Sie die vollständige Pfadüberwachung:

```
auditctl -W /etc/shadow -p w -k shadow-write
```

Wenn sealert nur Catchall-Vorschläge zurückgibt oder vorschlägt, mit dem Tool audit2allow eine neue Regel hinzuzufügen, gleichen Sie Ihr Problem mit den in den SELinux-Ablehnungen im Audit-Protokoll aufgeführten und erläuterten Beispielen ab.

5.3. Beheben analysierter SELinux-Ablehnungen

In den meisten Fällen geben Ihnen die Vorschläge des Tools sealert die richtige Anleitung, wie Sie Probleme im Zusammenhang mit der SELinux-Richtlinie beheben können. Informationen zur Verwendung von sealert zur Analyse von SELinux-Ablehnungen finden Sie unter Analysieren von SELinux-Ablehnungsmeldungen.

Seien Sie vorsichtig, wenn das Tool vorschlägt, das Tool audit2allow für Konfigurationsänderungen zu verwenden. Sie sollten audit2allow nicht als erste Option verwenden, um ein lokales Richtlinienmodul zu generieren, wenn Sie eine SELinux-Ablehnung sehen. Die Fehlerbehebung sollte mit einer Überprüfung beginnen, ob ein Beschriftungsproblem vorliegt. Der zweithäufigste Fall ist, dass Sie eine Prozesskonfiguration geändert und vergessen haben, SELinux darüber zu informieren.

Beschriftungsprobleme (Labeling)

Eine häufige Ursache für Beschriftungsprobleme ist die Verwendung eines nicht standardmäßigen

Verzeichnisses für einen Dienst. Beispielsweise möchte ein Administrator für eine Website möglicherweise nicht /var/www/html/, sondern /srv/myweb/ verwenden. Unter Red Hat Enterprise Linux wird das Verzeichnis /srv mit dem Typ var_t beschriftet. In /srv erstellte Dateien und Verzeichnisse erben diesen Typ. Außerdem können neu erstellte Objekte in Verzeichnissen der obersten Ebene, wie z. B. /myserver, mit dem Typ default_t beschriftet werden. SELinux verhindert, dass der Apache-HTTP-Server (httpd) auf beide Typen zugreift. Um den Zugriff zu ermöglichen, muss SELinux wissen, dass die Dateien in /srv/myweb/ über httpd zugänglich sein sollen:

```
# semanage fcontext -a -t httpd_sys_content_t "/srv/myweb(/.*)?"
```

Dieser Semanage-Befehl fügt den Kontext für das Verzeichnis /srv/myweb/ und alle Dateien und Verzeichnisse darunter zur SELinux-Dateikontextkonfiguration hinzu. Das Semanage-Dienstprogramm ändert den Kontext nicht. Verwenden Sie als Root das Dienstprogramm restorecon, um die Änderungen anzuwenden:

```
# restorecon -R -v /srv/myweb
```

Falscher Kontext

Das Dienstprogramm matchpathcon prüft den Kontext eines Dateipfads und vergleicht ihn mit der Standardbezeichnung für diesen Pfad. Das folgende Beispiel zeigt die Verwendung von matchpathcon in einem Verzeichnis, das falsch bezeichnete Dateien enthält:

```
$ matchpathcon -V /var/www/html/*
/var/www/html/index.html hat den Kontext unconfined_u:object_r:user_home_t:s0, sollte
system_u:object_r:httpd_sys_content_t:s0 sein
/var/www/html/page1.html hat den Kontext unconfined_u:object_r:user_home_t:s0, sollte
system_u:object_r:httpd_sys_content_t:s0 sein
```

In diesem Beispiel sind die Dateien index.html und page1.html mit dem Typ user_home_t bezeichnet. Dieser Typ wird für Dateien in Benutzer-Home-Verzeichnissen verwendet. Wenn Sie den Befehl mv verwenden, um Dateien aus Ihrem Home-Verzeichnis zu verschieben, kann es sein, dass Dateien mit dem Typ user_home_t gekennzeichnet werden. Dieser Typ sollte außerhalb von Home-Verzeichnissen nicht existieren. Verwenden Sie das Dienstprogramm restorecon, um solche Dateien auf ihren richtigen Typ zurückzusetzen:

```
# restorecon -v /var/www/html/index.html
restorecon reset /var/www/html/index.html context
unconfined_u:object_r:user_home_t:s0->system_u:object_r:httpd_sys_content_t:s0
```

Um den Kontext für alle Dateien in einem Verzeichnis wiederherzustellen, verwenden Sie die Option -R:

```
# restorecon -R -v /var/www/html/
restorecon reset /var/www/html/page1.html context
```

```
unconfined_u:object_r:samba_share_t:s0->system_u:object_r:httpd_sys_content_t:s0
restorecon reset /var/www/html/index.html context
unconfined_u:object_r:samba_share_t:s0->system_u:object_r:httpd_sys_content_t:s0
```

Confined Anwendungen, die auf nicht standardmäßige Weise konfiguriert sind

Dienste können auf verschiedene Arten ausgeführt werden. Um dies zu berücksichtigen, müssen Sie angeben, wie Sie Ihre Dienste ausführen. Sie können dies durch SELinux-Boolesche Werte erreichen, die es ermöglichen, Teile der SELinux-Richtlinie zur Laufzeit zu ändern. Dies ermöglicht Änderungen, wie z. B. das Erlauben des Zugriffs von Diensten auf NFS-Volumes, ohne die SELinux-Richtlinie neu zu laden oder neu zu kompilieren. Außerdem erfordert das Ausführen von Diensten auf nicht standardmäßigen Portnummern eine Aktualisierung der Richtlinienkonfiguration mit dem Befehl „semanage“.

Um beispielsweise dem Apache-HTTP-Server die Kommunikation mit MariaDB zu ermöglichen, aktivieren Sie den Booleschen Wert httpd_can_network_connect_db:

```
# setsebool -P httpd_can_network_connect_db on
```

Beachten Sie, dass die Option -P die Einstellung über Neustarts des Systems hinweg persistent macht.

Wenn der Zugriff für einen bestimmten Dienst verweigert wird, verwenden Sie die Dienstprogramme getsebool und grep, um zu prüfen, ob Boolesche Werte verfügbar sind, um den Zugriff zu erlauben. Verwenden Sie beispielsweise getsebool -a | grep ftp-Befehl zum Suchen nach FTP-bezogenen Booleans:

```
$ getsebool -a | grep ftp
ftpd_anon_write --> aus
ftpd_full_access --> aus
ftpd_use_cifs --> aus
ftpd_use_nfs --> aus

ftpd_connect_db --> aus
httpd_enable_ftp_server --> aus
tftp_anon_write --> aus
```

Um eine Liste der Booleans zu erhalten und herauszufinden, ob sie aktiviert oder deaktiviert sind, verwenden Sie den Befehl getsebool -a. Um eine Liste der Booleans einschließlich ihrer Bedeutung zu erhalten und herauszufinden, ob sie aktiviert oder deaktiviert sind, installieren Sie das Paket selinux-policy-devel und verwenden Sie den Befehl semanage boolean -l als Root.

Portnummern

Je nach Richtlinienkonfiguration können Dienste nur auf bestimmten Portnummern ausgeführt werden dürfen. Der Versuch, den Port zu ändern, auf dem ein Dienst ausgeführt wird, ohne die Richtlinie zu ändern, kann dazu führen, dass der Dienst nicht gestartet wird. Führen Sie

beispielsweise den Befehl semanage port -l | grep http als Root aus, um HTTP-bezogene Ports aufzulisten:

```
# semanage port -l | grep http
http_cache_port_t tcp 3128, 8080, 8118
http_cache_port_t udp 3130
http_port_t tcp 80, 443, 488, 8008, 8009, 8443
pegasus_http_port_t tcp 5988
pegasus_https_port_t tcp 5989
```

Der Porttyp http_port_t definiert die Ports, auf denen der Apache HTTP-Server lauschen kann. In diesem Fall sind dies die TCP-Ports 80, 443, 488, 8008, 8009 und 8443. Wenn ein Administrator httpd.conf so konfiguriert, dass httpd auf Port 9876 lauscht (Listen 9876), die Richtlinie jedoch nicht entsprechend aktualisiert wird, schlägt der folgende Befehl fehl:

```
# systemctl start httpd.service
Job für httpd.service fehlgeschlagen. Weitere Informationen finden Sie unter
"systemctl status httpd.service" und "journalctl -xn".

# systemctl status httpd.service
httpd.service - Der Apache-HTTP-Server
Geladen: geladen (/usr/lib/systemd/system/httpd.service; deaktiviert)
Aktiv: fehlgeschlagen (Ergebnis: Exitcode) seit Do 15.08.2013 09:57:05 MESZ; Vor 59 s
Prozess: 16874 ExecStop=/usr/sbin/httpd $OPTIONS -k Graceful-Stop (Code=exited,
Status=0/ERFOLGREICH)
Prozess: 16870 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (Code=exited,
Status=1/FEHLER)
```

Eine SELinux-Ablehnungsmeldung ähnlich der folgenden wird in /var/log/audit/audit.log protokolliert:

```
type=AVC msg=audit(1225948455.061:294): avc: denied { name_bind } for pid=4997
comm="httpd" src=9876 scontext=unconfined_u:system_r:httpd_t:s0
tcontext=system_u:object_r:port_t:s0 tclass=tcp_socket
```

Um httpd das Abhören eines Ports zu ermöglichen, der nicht für den Porttyp http_port_t aufgeführt ist, verwenden Sie den Befehl semanage port, um dem Port eine andere Bezeichnung zuzuweisen:

```
# semanage port -a -t http_port_t -p tcp 9876
```

Die Option -a fügt einen neuen Datensatz hinzu, die Option -t definiert einen Typ und die Option -p definiert ein Protokoll. Das letzte Argument ist die hinzuzufügende Portnummer.

Sonderfälle, sich entwickelnde oder defekte Anwendungen und kompromittierte Systeme

Anwendungen können Fehler enthalten, die dazu führen, dass SELinux den Zugriff verweigert.

Außerdem entwickeln sich die SELinux-Regeln weiter – SELinux hat möglicherweise eine Anwendung nicht auf eine bestimmte Weise ausgeführt, was möglicherweise dazu führt, dass es den Zugriff verweigert, obwohl die Anwendung wie erwartet funktioniert. Wenn beispielsweise eine neue Version von PostgreSQL veröffentlicht wird, kann sie Aktionen ausführen, die die aktuelle Richtlinie nicht berücksichtigt, wodurch der Zugriff verweigert wird, obwohl der Zugriff erlaubt sein sollte.

Verwenden Sie für diese Situationen nach der Zugriffsverweigerung das Dienstprogramm audit2allow, um ein benutzerdefiniertes Richtlinienmodul zu erstellen, das den Zugriff erlaubt. Sie können fehlende Regeln in der SELinux-Richtlinie in Red Hat Bugzilla melden. Erstellen Sie für Red Hat Enterprise Linux 9 Fehler für das Produkt Red Hat Enterprise Linux 9 und wählen Sie die Komponente selinux-policy aus. Fügen Sie die Ausgabe der Befehle audit2allow -w -a und audit2allow -a in solche Fehlerberichte ein.

Wenn eine Anwendung wichtige Sicherheitsberechtigungen anfordert, kann dies ein Zeichen dafür sein, dass die Anwendung kompromittiert ist. Verwenden Sie Intrusion Detection Tools, um solches verdächtiges Verhalten zu untersuchen.

Die Solution Engine auf dem Red Hat Customer Portal kann auch Anleitungen in Form eines Artikels mit einer möglichen Lösung für dasselbe oder ein sehr ähnliches Problem wie Sie bereitstellen. Wählen Sie das entsprechende Produkt und die Version aus und verwenden Sie SELinux-bezogene Schlüsselwörter wie selinux oder avc zusammen mit dem Namen Ihres blockierten Dienstes oder Ihrer blockierten Anwendung, zum Beispiel: selinux samba.

5.4. Erstellen eines lokalen SELinux-Richtlinienmoduls

Durch das Hinzufügen bestimmter SELinux-Richtlinienmodule zu einer aktiven SELinux-Richtlinie können bestimmte Probleme mit der SELinux-Richtlinie behoben werden. Sie können dieses Verfahren verwenden, um ein bestimmtes bekanntes Problem zu beheben, das in den Versionshinweisen von Red Hat beschrieben wird, oder um eine bestimmte Red Hat-Lösung zu implementieren.

Verwenden Sie nur von Red Hat bereitgestellte Regeln. Red Hat unterstützt das Erstellen von SELinux-Richtlinienmodulen mit benutzerdefinierten Regeln nicht, da dies außerhalb des Umfangs des Produktionssupports liegt. Wenn Sie kein Experte sind, wenden Sie sich an Ihren Red Hat-Vertriebsmitarbeiter und fordern Sie Beratungsdienste an.

Voraussetzungen

Die Pakete „setools-console“ und „audit“ zur Überprüfung.

Vorgehensweise

- Öffnen Sie eine neue .cil-Datei mit einem Texteditor, zum Beispiel:

```
# vim <local_module>.cil
```

Um Ihre lokalen Module besser zu organisieren, verwenden Sie das Präfix „local_“ in den Namen lokaler SELinux-Richtlinienmodule.

- Fügen Sie die benutzerdefinierten Regeln aus einem bekannten Problem oder einer Red Hat-Lösung ein.

Schreiben Sie keine eigenen Regeln. Verwenden Sie nur die Regeln, die in einem bestimmten bekannten Problem oder einer Red Hat-Lösung bereitgestellt werden.

- Um beispielsweise die Lösung „SELinux verweigert cups-lpd den Lesezugriff auf cups.sock in RHEL“ zu implementieren, fügen Sie die folgende Regel ein:

```
(allow cupsd_lpd_t cupsd_var_run_t (sock_file (read)))
```

Die Beispiellösung wurde für RHEL in RHBA-2021:4420 dauerhaft behoben. Daher haben die für diese Lösung spezifischen Teile dieses Verfahrens keine Auswirkungen auf aktualisierte RHEL 8- und 9-Systeme und sind nur als Syntaxbeispiele enthalten.

Sie können eine der beiden SELinux-Regelsyntaxen verwenden, Common Intermediate Language (CIL) und m4. Beispielsweise entspricht (allow cupsd_lpd_t cupsd_var_run_t (sock_file (read))) in CIL dem Folgenden in m4:

```
module local_cupslpd-read-cupssock 1.0;

require {
    type cupsd_var_run_t;
    type cupsd_lpd_t;
    class sock_file read;
}

#============= cupsd_lpd_t ==============
allow cupsd_lpd_t cupsd_var_run_t:sock_file read;
```

- Speichern und schließen Sie die Datei.
- Installieren Sie das Richtlinienmodul:

```
semodule -i <local_module>.cil
```

Wenn Sie ein lokales Richtlinienmodul entfernen möchten, das Sie mit „semodule -i“ erstellt haben, verweisen Sie auf den Modulnamen ohne das Suffix „.cil“. Um ein lokales Richtlinienmodul zu entfernen, verwenden Sie „semodule -r <local_module>“.

- Starten Sie alle mit den Regeln verbundenen Dienste neu:

```
systemctl restart <service-name>
```

Überprüfung

- Listen Sie die in Ihrer SELinux-Richtlinie installierten lokalen Module auf:

```
# semodule -lfull | grep "local_"
400 local_module cil
```

Da lokale Module die Priorität 400 haben, können Sie sie auch mithilfe dieses Werts aus der Liste filtern, z. B. mit dem Befehl semodule -lfull | grep -v ^100.

- Durchsuchen Sie die SELinux-Richtlinie nach den relevanten Zulassungsregeln:

```
# sesearch -A --source=<SOURCENAME> --target=<TARGETNAME> --class=<CLASSNAME>
--perm=<P1>,<P2>
```

Wobei <SOURCENAME> der Quell-SELinux-Typ, <TARGETNAME> der Ziel-SELinux-Typ, <CLASSNAME> der Name der Sicherheitsklasse oder Objektklasse und <P1> und <P2> die spezifischen Berechtigungen der Regel sind.

Beispielsweise verweigert SELinux cups-lpd den Lesezugriff auf cups.sock in der RHEL-Lösung:

```
# sesearch -A --source=cupsd_lpd_t --target=cupsd_var_run_t --class=sock_file
--perm=read
allow cupsd_lpd_t cupsd_var_run_t:sock_file { append getattr open read write };
```

Die letzte Zeile sollte jetzt den Lesevorgang enthalten.

- Überprüfen Sie, ob der relevante Dienst durch SELinux eingeschränkt ausgeführt wird:

Identifizieren Sie den Prozess, der mit dem relevanten Dienst zusammenhängt:

```
$ systemctl status <service-name>
```

Überprüfen Sie den SELinux-Kontext des Prozesses, der in der Ausgabe des vorherigen Befehls aufgeführt ist:

```
$ ps -efZ | grep <Prozessname>
```

Überprüfen Sie, dass der Dienst keine SELinux-Ablehnungen verursacht:

```
# ausearch -m AVC -i -ts recent
<keine Übereinstimmungen>
```

Die Option -i interpretiert die numerischen Werte in einen für Menschen lesbaren Text.

5.5. SELinux-Ablehnungen im Audit-Protokoll

Das Linux Audit-System speichert Protokolleinträge standardmäßig in der Datei /var/log/audit/audit.log.

Um nur SELinux-bezogene Datensätze aufzulisten, verwenden Sie den Befehl ausearch mit dem Nachrichtentypparameter, der mindestens auf AVC und AVC_USER eingestellt ist, zum Beispiel: i

```
# ausearch -m AVC,USER_AVC,SELINUX_ERR,USER_SELINUX_ERR
```

Ein SELinux-Ablehnungseintrag in der Audit-Protokolldatei kann wie folgt aussehen:

```
type=AVC msg=audit(1395177286.929:1638): avc: denied { read } for pid=6591
comm="httpd" name="webpages" dev="0:37" ino=2112 scontext=system_u:system_r:httpd_t:s0
tcontext=system_u:object_r:nfs_t:s0 tclass=dir
```

Die wichtigsten Teile dieses Eintrags sind:

- avc: denied – die von SELinux ausgeführte und im Access Vector Cache aufgezeichnete Aktion (AVC)
- { read } – die verweigerte Aktion
- pid=6591 – die Prozesskennung des Subjekts, das versucht hat, die verweigerte Aktion auszuführen
- comm="httpd" – der Name des Befehls, der zum Aufrufen des analysierten Prozesses verwendet wurde
- httpd_t – der SELinux-Typ des Prozesses
- nfs_t – der SELinux-Typ des von der Prozessaktion betroffenen Objekts
- tclass=dir – die Zielobjektklasse

Der vorherige Protokolleintrag kann wie folgt übersetzt werden:

SELinux hat dem httpd-Prozess mit PID 6591 und dem Typ httpd_t das Lesen aus einem Verzeichnis mit dem Typ nfs_t verweigert.

Die folgende SELinux-Ablehnungsmeldung wird angezeigt, wenn der Apache-HTTP-Server versucht, auf ein Verzeichnis zuzugreifen, das mit einem Typ für die Samba-Suite gekennzeichnet ist:

```
type=AVC msg=audit(1226874073.147:96): avc: denied { getattr } for pid=2465
comm="httpd" path="/var/www/html/file1" dev=dm-0 ino=284133
scontext=unconfined_u:system_r:httpd_t:s0
tcontext=unconfined_u:object_r:samba_share_t:s0 tclass=file
```

- { getattr } – der getattr-Eintrag gibt an, dass der Quellprozess versucht hat, die

Statusinformationen der Zieldatei zu lesen. Dies geschieht vor dem Lesen von Dateien. SELinux lehnt diese Aktion ab, da der Prozess auf die Datei zugreift und diese keine entsprechende Bezeichnung hat. Häufig verwendete Berechtigungen sind getattr, read und write.

- path="/var/www/html/file1" – der Pfad zum Objekt (Ziel), auf das der Prozess zugreifen wollte.
- scontext="unconfined_u:system_r:httpd_t:s0" – der SELinux-Kontext des Prozesses (Quelle), der die verweigerte Aktion versucht hat. In diesem Fall ist es der SELinux-Kontext des Apache-HTTP-Servers, der mit dem Typ httpd_t ausgeführt wird.
- tcontext="unconfined_u:object_r:samba_share_t:s0" – der SELinux-Kontext des Objekts (Ziel), auf das der Prozess zugreifen wollte. In diesem Fall ist es der SELinux-Kontext von file1.

Diese SELinux-Verweigerung kann wie folgt übersetzt werden:

SELinux hat dem httpd-Prozess mit PID 2465 den Zugriff auf die Datei /var/www/html/file1 mit dem Typ samba_share_t verweigert, die für Prozesse, die in der Domäne httpd_t ausgeführt werden, nicht zugänglich ist, sofern nicht anders konfiguriert.

6. Verwenden von Multi-Level Security (MLS)

Die Multi-Level Security (MLS)-Richtlinie verwendet Freigabestufen, wie sie ursprünglich von der US-amerikanischen Verteidigungsgemeinschaft entwickelt wurden. MLS erfüllt einen sehr engen Satz von Sicherheitsanforderungen, die auf Informationsmanagement in streng kontrollierten Umgebungen wie dem Militär basieren.

Die Verwendung von MLS ist komplex und lässt sich nicht gut auf allgemeine Anwendungsszenarien übertragen.

6.1. Multi-Level Security (MLS)

Die Multi-Level Security (MLS)-Technologie klassifiziert Daten in einer hierarchischen Klassifizierung anhand von Informationssicherheitsstufen, zum Beispiel:

- [niedrigste] Nicht klassifiziert
- [niedrig] Vertraulich
- [hoch] Geheim
- [höchste] Streng geheim

Standardmäßig verwendet die MLS SELinux-Richtlinie 16 Vertraulichkeitsstufen:

- s0 ist die am wenigsten sensible.
- s15 ist die am meisten sensible.

MLS verwendet eine spezielle Terminologie, um Vertraulichkeitsstufen anzusprechen:

- Benutzer und Prozesse werden als Subjekte bezeichnet, deren Vertraulichkeitsstufe als Freigabe

bezeichnet wird.

- Dateien, Geräte und andere passive Komponenten des Systems werden als Objekte bezeichnet, deren Vertraulichkeitsstufe als Klassifizierung bezeichnet wird.

Um MLS zu implementieren, verwendet SELinux das Bell-La Padula-Modell (BLP). Dieses Modell gibt an, wie Informationen innerhalb des Systems fließen können, basierend auf Etiketten, die jedem Subjekt und Objekt zugeordnet sind.

Das Grundprinzip von BLP lautet „Kein Lesen, kein Aufschreiben“. Dies bedeutet, dass Benutzer nur Dateien mit ihrer eigenen Vertraulichkeitsstufe und niedriger lesen können und Daten nur von niedrigeren zu höheren Ebenen fließen können und nie umgekehrt.

Die MLS SELinux-Richtlinie, die die Implementierung von MLS auf RHEL darstellt, wendet ein modifiziertes Prinzip namens Bell-La Padula mit Schreibgleichheit an. Dies bedeutet, dass Benutzer Dateien auf ihrer eigenen Vertraulichkeitsstufe und niedriger lesen können, aber nur auf genau ihrer eigenen Stufe schreiben können. Dies verhindert beispielsweise, dass Benutzer mit niedriger Berechtigung Inhalte in streng geheime Dateien schreiben.

Beispielsweise gilt standardmäßig für einen Benutzer mit Berechtigungsstufe s2:

- Kann Dateien mit den Vertraulichkeitsstufen s0, s1 und s2 lesen.
- Kann keine Dateien mit der Vertraulichkeitsstufe s3 und höher lesen.
- Kann Dateien mit der Vertraulichkeitsstufe genau s2 ändern.
- Kann keine Dateien mit einer anderen Vertraulichkeitsstufe als s2 ändern.

Sicherheitsadministratoren können dieses Verhalten anpassen, indem sie die SELinux-Richtlinie des Systems ändern. Sie können Benutzern beispielsweise erlauben, Dateien auf niedrigeren Ebenen zu ändern, wodurch die Vertraulichkeitsstufe der Datei auf die Freigabestufe des Benutzers erhöht wird.

In der Praxis werden Benutzern normalerweise verschiedene Freigabestufen zugewiesen, z. B. s1-s2. Ein Benutzer kann Dateien mit niedrigeren Vertraulichkeitsstufen als der maximalen Stufe des Benutzers lesen und in alle Dateien innerhalb dieses Bereichs schreiben.

Beispielsweise gilt standardmäßig für einen Benutzer mit einer Freigabestufe von s1-s2:

- Kann Dateien mit den Vertraulichkeitsstufen s0 und s1 lesen.
- Kann keine Dateien mit der Vertraulichkeitsstufe s2 und höher lesen.
- Kann Dateien mit der Vertraulichkeitsstufe s1 ändern.
- Kann keine Dateien mit einer anderen Vertraulichkeitsstufe als s1 ändern.
- Kann die eigene Freigabestufe auf s2 ändern.

Der Sicherheitskontext für einen nicht privilegierten Benutzer in einer MLS-Umgebung ist beispielsweise:

```
user_u:user_r:user_t:s1
```

Dabei gilt:

user_u

Ist der SELinux-Benutzer.

user_r

Ist die SELinux-Rolle.

user_t

Ist der SELinux-Typ.

s1

Ist der Bereich der MLS-Vertraulichkeitsstufen.

Das System kombiniert immer MLS-Zugriffsregeln mit herkömmlichen Dateizugriffsberechtigungen. Wenn beispielsweise ein Benutzer mit der Sicherheitsstufe „Geheim“ Discretionary Access Control (DAC) verwendet, um den Zugriff anderer Benutzer auf eine Datei zu blockieren, können selbst „Streng geheime“ Benutzer nicht auf diese Datei zugreifen. Eine hohe Sicherheitsfreigabe erlaubt einem Benutzer nicht automatisch, das gesamte Dateisystem zu durchsuchen.

Benutzer mit Freigaben auf höchster Ebene erhalten nicht automatisch Administratorrechte auf mehrstufigen Systemen. Obwohl sie möglicherweise Zugriff auf alle vertraulichen Informationen über das System haben, ist dies etwas anderes als Administratorrechte.

Außerdem bieten Administratorrechte keinen Zugriff auf vertrauliche Informationen. Selbst wenn sich beispielsweise jemand als Root anmeldet, kann er streng geheime Informationen nicht lesen.

Sie können den Zugriff innerhalb eines MLS-Systems mithilfe von Kategorien weiter anpassen. Mit Multi-Category Security (MCS) können Sie Kategorien wie Projekte oder Abteilungen definieren. Benutzer dürfen dann nur auf Dateien in den Kategorien zugreifen, denen sie zugewiesen sind. Weitere Informationen finden Sie unter Verwenden von Multi-Category Security (MCS) zum Schutz der Datenvertraulichkeit.

6.2. SELinux roles in MLS

Die SELinux-Richtlinie ordnet jeden Linux-Benutzer einem SELinux-Benutzer zu. Dadurch können Linux-Benutzer die Beschränkungen von SELinux-Benutzern übernehmen.

Die MLS-Richtlinie enthält nicht das uneingeschränkte Modul, einschließlich uneingeschränkter Benutzer, Typen und Rollen. Daher können uneingeschränkte Benutzer, einschließlich Root, nicht auf alle Objekte zugreifen und alle Aktionen ausführen, die sie in der Zielrichtlinie ausführen könnten.

Sie können die Berechtigungen für eingeschränkte Benutzer in Ihrer SELinux-Richtlinie nach Ihren

spezifischen Anforderungen anpassen, indem Sie die Booleschen Werte in der Richtlinie anpassen. Sie können den aktuellen Status dieser Booleschen Werte mit dem Befehl „semanage boolean -l“ ermitteln. Um alle SELinux-Benutzer, ihre SELinux-Rollen und MLS/MCS-Ebenen und -Bereiche aufzulisten, verwenden Sie den Befehl „semanage user -l“ als Root.

Benutzer	Standardrole	Zusätzliche Rolle
guest_u	guest_r	
xguest_u	gues_r	
user_u	user_r	
staff_u	starr_r	auditadm_r
		secadm_r
		sysadm_r
		staff_r
sysadm_u	sysadm_r	
root	staff_r	auditadm_r
		secadm_r
		sysadm_r
		system_r
system_u	system_r	

Beachten Sie, dass system_u eine spezielle Benutzeridentität für Systemprozesse und -objekte und system_r die zugehörige Rolle ist. Administratoren dürfen diesen system_u-Benutzer und die system_r-Rolle niemals einem Linux-Benutzer zuordnen. Auch unconfined_u und root sind uneingeschränkte Benutzer. Aus diesen Gründen sind die diesen SELinux-Benutzern zugeordneten Rollen nicht in der folgenden Tabelle Typen und Zugriff von SELinux-Rollen enthalten.

Jede SELinux-Rolle entspricht einem SELinux-Typ und bietet bestimmte Zugriffsrechte.

Role	Type	Grafisches Login	su und sudo
guest_r	guest_t	no	no
xguest_r	xguest_t	yes	no
user_r	user_t	yes	no
staff_r	staff_t	yes	nur sudo
auditadm_r	auditatm_t		yes
secadm_r	auditam_t		yes
secadm_r	auditadm_t		yes
sysadm_r	secadm_t		yes

secadm_r	secadm_t	nur wenn xdm_sysadm_login ist an	yes

- Standardmäßig verfügt die Rolle sysadm_r über die Rechte der Rolle secadm_r, was bedeutet, dass ein Benutzer mit der Rolle sysadm_r die Sicherheitsrichtlinie verwalten kann. Wenn dies nicht Ihrem Anwendungsfall entspricht, können Sie die beiden Rollen trennen, indem Sie das Modul sysadm_secadm in der Richtlinie deaktivieren. Weitere Informationen finden Sie unter Trennen der Systemadministration von der Sicherheitsadministration in MLS.
- Die Nicht-Anmelderollen dbadm_r, logadm_r und webadm_r können für eine Teilmenge der Verwaltungsaufgaben verwendet werden. Standardmäßig sind diese Rollen keinem SELinux-Benutzer zugeordnet.

6.3. Umstellung der SELinux-Richtlinie auf MLS

Verwenden Sie die folgenden Schritte, um die SELinux-Richtlinie von gezielt auf Multi-Level Security (MLS) umzustellen.

Verwenden Sie die MLS-Richtlinie nicht auf einem System, auf dem das X Window System ausgeführt wird. Wenn Sie das Dateisystem mit MLS-Bezeichnungen neu kennzeichnen, kann das System außerdem den Zugriff auf eingeschränkte Domänen verhindern, wodurch Ihr System nicht ordnungsgemäß gestartet werden kann. Stellen Sie daher sicher, dass Sie SELinux in den permissiven Modus schalten, bevor Sie die Dateien neu kennzeichnen. Auf den meisten Systemen treten nach dem Wechsel zu MLS viele SELinux-Ablehnungen auf, und viele davon sind nicht leicht zu beheben.

Vorgehensweise

- Installieren Sie das Paket selinux-policy-mls:

```
# dnf install selinux-policy-mls
```

- Öffnen Sie die Datei /etc/selinux/config in einem Texteditor Ihrer Wahl, zum Beispiel:

```
# vi /etc/selinux/config
```

- Ändern Sie den SELinux-Modus von „enforcing“ (erzwingend) auf „permissive“ (freigebend) und wechseln Sie von der gezielten Richtlinie zu MLS:

```
SELINUX=permissive
SELINUXTYPE=mls
```

Speichern Sie die Änderungen und beenden Sie den Editor.

- Bevor Sie die MLS-Richtlinie aktivieren, müssen Sie jede Datei im Dateisystem mit einer MLS-Bezeichnung neu kennzeichnen:

```
# fixfiles -F onboot
Das System wird beim nächsten Start neu gekennzeichnet
```

- Starten Sie das System neu:

```
# reboot
```

- Suchen Sie nach SELinux-Ablehnungen:

```
# ausearch -m AVC,USER_AVC,SELINUX_ERR,USER_SELINUX_ERR -ts recent -i
```

Da der vorherige Befehl nicht alle Szenarien abdeckt, finden Sie unter Beheben von Problemen im Zusammenhang mit SELinux Anleitungen zum Identifizieren, Analysieren und Beheben von SELinux-Ablehnungen.

- Nachdem Sie sichergestellt haben, dass auf Ihrem System keine Probleme im Zusammenhang mit SELinux vorliegen, schalten Sie SELinux zurück in den Durchsetzungsmodus, indem Sie die entsprechende Option in /etc/selinux/config ändern:

```
SELINUX=enforcing
```

Starten Sie das System neu:

```
# reboot
```

Wenn Ihr System nicht startet oder Sie sich nach dem Wechsel zu MLS nicht anmelden können, fügen Sie den Parameter enforcing=0 zu Ihrer Kernel-Befehlszeile hinzu. Weitere Informationen finden Sie unter Ändern der SELinux-Modi beim Booten.

Beachten Sie auch, dass sich in MLS SSH-Anmeldungen als Root-Benutzer, der der SELinux-Rolle sysadm_r zugeordnet ist, von der Anmeldung als Root in staff_r unterscheiden. Bevor Sie Ihr System zum ersten Mal in MLS starten, sollten Sie SSH-Anmeldungen als sysadm_r zulassen, indem Sie den SELinux-Booleschen Wert ssh_sysadm_login auf 1 setzen. Um ssh_sysadm_login später, bereits in MLS, zu aktivieren, müssen Sie sich als Root in staff_r anmelden, mit dem Befehl newrole -r sysadm_r zu Root in sysadm_r wechseln und dann den Booleschen Wert auf 1 setzen.

Überprüfung

Überprüfen Sie, ob SELinux im Durchsetzungsmodus ausgeführt wird:

```
# getenforce
Enforcing
```

Überprüfen Sie, ob der Status von SELinux den mls-Wert zurückgibt:

```
# sestatus | grep mls
Loaded policy name: mls
```

6.4. Einrichten der Benutzerfreigabe in MLS

Nachdem Sie die SELinux-Richtlinie auf MLS umgestellt haben, müssen Sie Benutzern Sicherheitsfreigabestufen zuweisen, indem Sie sie eingeschränkten SELinux-Benutzern zuordnen. Standardmäßig gilt für einen Benutzer mit einer bestimmten Sicherheitsfreigabe:

- Kann keine Objekte mit einer höheren Vertraulichkeitsstufe lesen.
- Kann nicht in Objekte mit einer anderen Vertraulichkeitsstufe schreiben.

Voraussetzungen

- Die SELinux-Richtlinie ist auf „mls“ eingestellt.
- Der SELinux-Modus ist auf „erzwingen“ eingestellt.
- Das Paket policycoreutils-python-utils ist installiert.
- Ein Benutzer, der einem eingeschränkten SELinux-Benutzer zugewiesen ist:
  - Für einen nicht privilegierten Benutzer, zugewiesen zu user_u (example_user im folgenden Verfahren).
  - Für einen privilegierten Benutzer, zugewiesen zu staff_u (staff im folgenden Verfahren).

Stellen Sie sicher, dass die Benutzer erstellt wurden, als die MLS-Richtlinie aktiv war. In anderen SELinux-Richtlinien erstellte Benutzer können nicht in MLS verwendet werden.

Vorgehensweise

- Optional: Um zu verhindern, dass Ihrer SELinux-Richtlinie Fehler hinzugefügt werden, wechseln Sie in den permissiven SELinux-Modus, der die Fehlerbehebung erleichtert:

  ```
  # setenforce 0
  ```

Beachten Sie, dass SELinux im permissiven Modus die aktive Richtlinie nicht erzwingt, sondern nur Access Vector Cache (AVC)-Nachrichten protokolliert, die dann zur Fehlerbehebung und zum Debuggen verwendet werden können.

- Definieren Sie einen Freigabebereich für den SELinux-Benutzer staff_u. Beispielsweise legt

dieser Befehl den Freigabebereich von s1 bis s15 fest, wobei s1 die Standardfreigabestufe ist:

```
# semanage user -m -L s1 -r s1-s15 staff_u
```

- Generieren Sie SELinux-Dateikontextkonfigurationseinträge für Benutzer-Home-Verzeichnisse:

```
# genhomedircon
```

- Dateisicherheitskontexte auf Standard zurücksetzen:

```
 # restorecon -R -F -v /home/
Relabeled /home/staff from staff_u:object_r:user_home_dir_t:s0 to
staff_u:object_r:user_home_dir_t:s1
Relabeled /home/staff/.bash_logout from staff_u:object_r:user_home_t:s0 to
staff_u:object_r:user_home_t:s1
Relabeled /home/staff/.bash_profile from staff_u:object_r:user_home_t:s0 to
staff_u:object_r:user_home_t:s1
Relabeled /home/staff/.bashrc from staff_u:object_r:user_home_t:s0 to
staff_u:object_r:user_home_t:s1
```

- Assign a clearance level to the user:

```
# chcon -R -l s1 /home/example_user
```

- Optional: Wenn Sie zuvor in den permissiven SELinux-Modus gewechselt sind und überprüft haben, dass alles wie erwartet funktioniert, wechseln Sie zurück in den erzwingenden SELinux-Modus:

```
# setenforce 1
```

Überprüfungsschritte

- Überprüfen Sie, ob der Benutzer dem richtigen SELinux-Benutzer zugeordnet ist und die richtige Freigabestufe zugewiesen ist:

```
 # semanage login -l
Login Name      SELinux User         MLS/MCS Range        Service
__default__     user_u               s0-s0                *
example_user    user_u               s1                   *
...
```

- Melden Sie sich als Benutzer bei MLS an.
- Überprüfen Sie, ob die Sicherheitsstufe des Benutzers korrekt funktioniert:

Die Dateien, die Sie zur Überprüfung verwenden, sollten keine vertraulichen Informationen enthalten, für den Fall, dass die Konfiguration falsch ist und der Benutzer tatsächlich ohne Autorisierung auf die Dateien zugreifen kann.

- Stellen Sie sicher, dass der Benutzer eine Datei mit höherer Vertraulichkeitsstufe nicht lesen kann.
- Stellen Sie sicher, dass der Benutzer in eine Datei mit derselben Vertraulichkeitsstufe schreiben kann.
- Stellen Sie sicher, dass der Benutzer eine Datei mit niedrigerer Vertraulichkeitsstufe lesen kann.

6.5. Ändern der Freigabestufe eines Benutzers innerhalb des definierten Sicherheitsbereichs in MLS

6.6. Erhöhung der Dateisensitivität in MLS

6.7. Ändern der Dateiempfindlichkeit in MLS

6.8. Trennung der Systemadministration von der Sicherheitsadministration in MLS

6.9. Definieren eines sicheren Terminals in MLS

6.10. MLS-Benutzern das Bearbeiten von Dateien auf niedrigeren Ebenen ermöglichen

7. Verwendung von Multi-Category Security (MCS) für Datenvertraulichkeit

7.1. Multi-Category Security (MCS)

7.2. Konfigurieren der Multi-Category-Sicherheit für die Datenvertraulichkeit

7.3. Definieren von Kategoriebezeichnungen in MCS

7.4. Zuweisen von Kategorien zu Benutzern in MCS

10.4. Übertragen von SELinux-Einstellungen auf ein anderes System mit semanage

11. Legal Notice