Fedora 11

Security-Enhanced Linux

Gebruikers gids

Murray McAllister

Daniel Walsh

Dominick Grift

Eric Paris

James Morris

Scott Radvan

Fedora 11 Security-Enhanced Linux
Gebruikers gids
Uitgave 1.3

Auteur	Murray McAllister	mmcallis@redhat.com
Auteur	Daniel Walsh	dwalsh@redhat.com
Auteur	Dominick Grift	domg472@gmail.com
Auteur	Eric Paris	eparis@parisplace.org
Auteur	James Morris	jmorris@redhat.com
Auteur	Scott Radvan	sradvan@redhat.com

Dieze gids behandelt het beheer en gebruik van Security-Enhanced Linux.

Voorwoord

De Fedora 11 SELinux gebruikers gids is voor mensen met minimale of geen ervaring met SELinux. Hoewel systeembeheer ervaring niet noodzakelijk is, is de inhoud van deze gids geschreven voor systeembeheer taken. Deze gids biedt een inleiding voor fundamentele concepten en practische toepassingen van SELinux. Na lezing van deze gids moet je een redelijk begrip hebben van SELinux.

Dank je voor iedereen die aanmoediging, hulp en testen aanbood - het wordt zeer gewaardeerd. Met een speciale dank aan:

- Dominick Grift, Stephen Smalley, en Russell Coker voor hun bijdrages, hulp, en geduld.
- Karsten Wade voor zijn hulp, het toevoegen van een component voor deze gids aan *Red Hat Bugzilla*[1], en voor het uitzoeken van web onderdak op *http://docs.fedoraproject.org/*.
- Het *Fedora Infrastructure Team*[2] voor het geven van onderdak.
- Jens-Ulrik Petersen voor het zorgen dat Red Hat Brisbane up-to-date Fedora spiegels heeft.

1. Document Conventie

Dit handboek hanteert verscheidene conventies om bepaalde woorden of zinsdelen te benadrukken en aandacht te vestigen op specifieke delen van informatie.

In PDF en papieren edities gebruikt dit handboek *Liberation Fonts set*[3] lettertypen. Het Liberation lettertype wordt ook gebruikt in HTML-edities indien dit lettertype op uw computer geïnstalleerd is. Indien dat niet het geval is, worden alternatieve, gelijkwaardige lettertypen gebruikt. Noot: bij Red Hat Enterprise Linux 5 en later wordt de Liberation Font set standaard meegeleverd.

1.1. Typografische Conventies

Vier typografische conventies worden gebruikt om aandacht te vestigen op specifieke woorden en zinsdelen. Deze conventies -en de omstandigheden waaronder zij gebruikt worden- luiden als volgt:

`Mono-spaced Bold`

Wordt gebruikt om systeem input, waaronder shell commando's, bestandsnamen en paden aan te geven. Wordt ook gebruikt bij toetsaanduiding of toetsencombinaties. Voorbeeld:

> Om de inhoud van het bestand **`mijn_onwijsgoed_verkopende_boek`** in uw huidige directory te zien, voert u het commando **`cat mijn_onwijsgoed_verkopende_boek`** in bij de shell-prompt en drukt u op **`Enter`** om het commando uit te laten voeren.

Bovenstaande bevat een bestandsnaam, een shell-commando en een toetsaanduiding, alle getoond in Mono-spaced Bold en alle te onderscheiden dankzij hun context.

Toetsencombinaties kunnen worden onderscheiden van toetsaanduiding door het plusteken dat elk deel van een toetsencombinatie aan elkaar verbind. Voorbeeld:

> Druk op **`Enter`** om het commando te laten uitvoeren.

[3] https://fedorahosted.org/liberation-fonts/

> Druk op **Ctrl**+**Alt**+**F1** om naar de eerste virtuele terminal over te schakelen. Druk op **Ctrl**+**Alt**+**F7** om terug te keren naar uw X-Windows sessie.

De eerste zin benadrukt de bepaalde toets die moet worden ingedrukt. De tweede benadrukt twee reeksen van drie toetsen, waarbij de toetsen van elke reeks tegelijk moet worden ingedrukt.

Indien broncode wordt besproken, worden klassennamen, functies, variabele namen en resultaten die in een paragraaf worden genoemd, weergegeven als hier boven afgedrukt, namelijk in **Mono-spaced Bold**. Voorbeeld:

> Onder bestandsgerelateerde klassen vallen **filesystem** voor bestandsystemen, **file** voor bestanden, en **dir** voor directories. Elke klasse heeft haar eigen set van permissies.

Proportional Bold

Wordt gebruikt om woorden of zinsdelen op een systeem aan te duiden, waaronder applicatie namen, dialoogtekst-boxen, gelabelde toetsen, checkbox en radiobutton labels, menutitels en submenutitels. Voorbeeld:

> Kies **Systeem > Voorkeuren > Muis** uit de hoofdmenubalk om **Muis Voorkeuren** te openen. In de **Knoppen** tab, klik de **Linkshandige muis** checkbox aan en klik **Sluiten** om de primaire muisknop van links naar rechts te wisselen (waardoor de muis beter geschikt is geworden voor linkshandig gebruik).
>
> Om een speciaal teken in een **gedit** bestand op te nemen, kiest u **Toepassingen > Hulpmiddelen > Tekentabel** uit de hoofdmenubalk. Vervolgens kiest u **Zoeken > Find…** uit de **Tekentabel** menubalk, typ de naam van het teken in het **Zoek** veld en klik **Volgende**. Het teken dat u zoekt zal worden gemarkeerd in de **Tekentafel**. Dubbel-klik op dit teken om het in de **Te kopiëren tekst** veld op te nemen en klik dan de **Kopiëren** knop. Keer terug naar uw document en kies **Bewerken > Plakken** uit de **gedit** menubalk.

De bovenstaande tekst bevat applicatienamen, systeemwijde menunamen en onderdelen, applicatie specifieke menunamen, en knoppen en tekst van een GUI-interface, alle vertoond in Proportional Bold en alle te onderscheiden dankzij hun context.

Merk het **>**-teken op, gebruikt om aan te geven dat door een menu en sub-menu wordt gelopen. Dit voorkomt het gebruik van de nogal omslachtige 'Selecteer **Muis** van het **Voorkeuren** sub-menu in het **Systeem** menu uit de hoofdmenubalk'-omschrijvingen.

Mono-spaced Bold Italic of ***Proportional Bold Italic***

Mono-spaced Bold of Proportional Bold behandelt indien cursief gedrukt altijd vervangbare of wisselende teksten. Cursief wijst op niet letterlijke tekst of toont tekst dat wisselt naar omstandigheden. Voorbeeld:

> Om verbinding te maken met een andere computer met behulp van ssh, typt u **ssh *gebruikersnaam*@*domein.naam*** bij een shell prompt.
>
> Het **mount -o remount *file-system*** commando mount opnieuw het genoemde bestandsysteem. Om bijvoorbeeld het **/home** bestandsysteem opnieuw te mounten, gebruikt men het **mount -o remount /home** commando.

> Om de versie van een huidig geïnstalleerd pakket te zien, gebruikt u het **`rpm -q`** ***`package`*** commando. Dit zal het volgende resultaat opleveren: ***`package-version-release`*** .

Let op de woorden in bold italics in bovenstaande tekst — username, domain.name, file-system, package, version en release. Elk woord is een [plaatshouder], hetzij voor tekst dat u invult indien u een commando typt, hetzij voor tekst die door het systeem wordt getoond.

Buiten het standaard gebruik bij het presenteren van een titel van een werk, wordt cursief ingezet om het eerste gebruik van een nieuwe en belangrijke term te benadrukken. Voorbeeld:

> Wanneer de Apache HTTP Server verzoeken accepteert, zet het childprocessen of threads ter afhandeling in. Deze groep van childprocessen of threads staan bekend als een *server-pool*. Onder Apache HTTP Server 2.0 is de verantwoordelijkheid voor het creëren en onderhouden van deze server-pools toegewezen aan een groep modules genaamd *Multi-Processing Modules* (*MPMs*). Anders dan bij de andere modules kan slechts één module van de MPM groep door de Apache HTTP Server geladen zijn.

1.2. Pull-quote Conventies

Twee, normaal gesproken uit meerdere regels bestaande, datatypes worden visueel van de omringende tekst gescheiden.

Tekst gezonden naar een terminal wordt getoond in `Mono-spaced Roman` en als volgt gepresenteerd:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Opsommingen van broncode worden ook vertoond in `Mono-spaced Roman` maar worden alsvolgt gepresenteerd en benadrukt:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
   public static void main(String args[])
       throws Exception
   {
      InitialContext iniCtx = new InitialContext();
      Object         ref    = iniCtx.lookup("EchoBean");
      EchoHome       home   = (EchoHome) ref;
      Echo           echo   = home.create();

      System.out.println("Created Echo");

      System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
   }
```

```
}
```

1.3. Noten en waarschuwingen

Tenslotte gebruiken we drie visuele stijlen om aandacht te vestigen op informatie die anders misschien over het hoofd zou worden gezien.

Noot

A Note is a tip or shortcut or alternative approach to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

Belangrijk

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring Important boxes won't cause data loss but may cause irritation and frustration.

Waarschuwing

Een waarschuwing dient niet genegeerd te worden. Waarschuwingen negeren zal ongetwijfeld leiden tot data- en haarverlies.

2. We hebben terugkoppeling nodig!

Indien u een typografische fout in deze handleiding vindt, of u weet een manier om deze handleiding te verbeteren, zouden wij dat graag van u horen! Meldt u alstublieft fouten in de uitgave **Fedora Documentation** via Bugzilla: *http://bugzilla.redhat.com/bugzilla/*.

Indien u fouten meldt, vergeet dan alstublieft niet het kenmerk: *selinux-user-guide* te vermelden.

Indien u suggesties hebt om de documentatie te verbeteren, probeer dan zo duidelijk mogelijk deze suggesties te omschrijven. Indien u fouten hebt ontdekt, vermeldt dan alstublieft het sectienummer en wat omringende tekst, opdat we de fout makkelijker kunnen vinden.

Handelsmerk informatie

1.1. Source Code

The XML source for this guide is available at *http://svn.fedorahosted.org/svn/selinuxguide/*

Inleiding

Bestanden, zoals mappen en apparaten, worden objecten genoemd. Processen, zoals een gebruiker die een commando draait of de Mozilla® Firefox® toepassing worden subjecten genoemd. De meeste operating systemen gebruiken een Discretionary Access Control (DAC) (toegangscontrole naar goeddunken) systeem dat controleert hoe subjecten omgaan met objecten en hoe subjecten omgaan met elkaar. In operating systemen die DAC gebruiken, controleren gebruikers de toegangsrechten van bestanden (objecten) waarvan zij eigenaar zijn. Bijvoorbeeld, in Linux® operating systemen, kunnen gebruikers hun persoonlijke mappen leesbaar voor de wereld maken, en kunnen zij gebruikers en processen (subjects) toegang geven tot potentieel gevoelige informatie.

DAC mechanismes zijn fundamenteel onvoldoende voor een sterke systeem beveiliging. DAC toegangsbeslissingen zijn alleen gebaseerd op gebruikers identiteit en eigendom, en negeren andere informatie die relevant is voor beveiliging, zoals de rol van de gebruiker, de functie en betrouwbaarheid van het programma, en de gevoeligheid en integriteit van de data. Elke gebruiker heeft volledige vrijheid over zijn bestanden, wat het onmogelijk maakt om een systeem-brede beveiligings tactiek af te dwingen. Verder erft elk programma dat door een gebruiker gedraaid wordt alle toegangsrechten die aan die gebruiker gegeven zijn en kan het toegang tot de bestanden van de gebruiker veranderen, dus wordt er geen bescherming geboden tegen kwaadwillige software. Veel systeem voorzieningen en bevoorrechte programma's moeten draaien met grof-korrelige voorrechten die hun vereiste ver overschrijden, zodat een mankement in elk van deze programma's benut kan worden om complete systeem toegang te krijgen.[1]

Het volgende is een voorbeeld van rechten gebruikt in Linux operating systemen die niet mogelijk zijn in Security-Enhanced Linux (SELinux). De rechten in deze voorbeelden kunnen verschillen van jouw systeem. Gebruik het **`ls -l`** commando om de bestandsrechten te zien:

```
$ ls -l file1
-rwxrw-r-- 1 user1 group1 0 2009-04-30 15:42 file1
```

De eerste drie rechten bits, `rwx`, controleren de toegang die de Linux `user1` gebruiker (in dit geval de eigenaar) heeft voor **`file1`**. De volgende drie rechten bits, `rw-`, controleren de toegang die de Linux `group1` groep heeft voor **`file1`**. De laatste drie rechten bits, `r--`, controleren de toegang die alle anderen hebben voor **`file1`**, wat alle gebruikers en processen omvat.

Security-Enhanced Linux (SELinux) voegt Mandatory Access Control (MAC) (verplichte toegangs controle) toe aan de Linux kernel, en is standaard aangezet in Fedora. Een MAC architectuur voor algemene doeleinden heeft de mogelijkheid nodig om een door beheerders opgestelde beveiligings tactiek af te dwingen voor alle processen en bestanden op het systeem, waarbij beslissingen gebaseerd worden op labels die een verscheidenheid aan informatie bevatten die voor beveiling relevant is. Als dit correct geimplementeerd is, staat dit een systeem toe zich afdoende te verdedigen en biedt kritische ondersteuning voor beschermning tegen het knoeien met, of het omheen gaan van, beveiligde toepassingen. MAC biedt een sterke scheiding aan voor toepassingen wat het veilig uitvoeren van onbetrouwbare toepassingen toestaat. Zijn mogelijkheid om de rechten van draaiende processen te beperken, beperkt de reikwijdte van potentiele schade die het resultaat kan zijn van de uitbuiting van kwetsbaarheden in toepassingen en systeem voorzieningen. MAC staat toe dat

[1] "Integrating Flexible Support for Security Policies into the Linux Operating System", door Peter Loscocco en Stephen Smalley. Dit artikel was oorspronkelijk gemaakt voor de National Security Agency en is dus in het publieke domein. Refereer naar het *originele artikel* [http://www.nsa.gov/research/_files/selinux/papers/freenix01/index.shtml] voor details en het document zoals het eerst is vrijgegeven. Alle aanpassingen en veranderingen zijn gedaan door Murray McAllister.

informatie beschermd wordt tegen legitieme gebruikers met beperkte toestemming en ook tegen gemachtigde gebruikers die onbedoeld kwaadwillige toepassingen uitvoeren.[2]

Het volgende is een voorbeeld van de labels die beveilings-relevantie informatie bevatten die worden gebruikt voor processen, Linux gebruikers, en bestanden in Linux operating systemen die SELinux draaien. Deze informatie wordt de SELinux context genoemd, en kan getoond worden met gebruik van het **`ls -Z`** commando:

```
$ ls -Z file1
-rwxrw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0      file1
```

In dit voorbeeld, biedt SELinux een gebruiker (`unconfined_u`), een rol (`object_r`), een type (`user_home_t`), en een niveau (`s0`) aan. Deze informatie wordt gebruikt om toegangscontrole beslissingen te maken. Met DAC kan de toegang alleen maar gecontroleerd worden op basis van de gebruiker en groep ID's. De SELinux tactiekregels worden gecontroleerd na de DAC regels. SELinux tactiekregels worden niet gebruikt als DAC reeds toegang weigert.

Linux en SELinux gebruikers

Op Linux operating systemen die SELinux draaien, zijn er zowel Linux gebruikers als SELinux gebruikers. SELinux gebruikers zijn onderdeel van de SELinux tactiek. Linux gebruikers zijn afgebeeld op SELinux gebruikers. Om verwarring te vermijden, gebruikt deze gids "Linux gebruiker" en "SELinux gebruiker" om tussen deze twee verschil te maken.

2.1. Voordelen van het draaien van SELinux

- Alle processen en bestanden zijn gelabeld met een type. Een type definieert het domein voor processen en een type voor bestanden. Processen zijn van elkaar gescheiden door ze ieder in hun eigen domein te draaien, en SELinux tactiekregels bepalen hoe processen omgaan met bestanden, en ook hoe processen omgaan met elkaar. Toegang is alleen toegestaan als er een SELinux tactiekregel bestaat die dat specifiek toestaat.

- Fijn-korrelige toegangs controle. Door verder te gaan dan de traditionele UNIX® rechten die gecontroleerd worden door het oordeel van de gebruiker en gebaseerd zijn op Linux gebruiker en groep ID's, zijn SELinux toegangs beslissingen gebaseerd op alle beschikbare informatie, zoals een SELinux gebruiker, rol, type en, optioneel, een niveau.

- SELinux tactiek is gedefinieerd op beheersniveau, wordt systeem-breed afgedwongen, en wordt niet ingesteld door het oordeel van de gebruiker.

- Verminderde gevoeligheid voor rechten escalatie aanvallen. Een voorbeeld: omdat processen in domeinen draaien, en daarom van elkaar gescheiden zijn, en SELinux tactiek regels bepalen hoe processen toegang hebben tot bestanden en andere processen; als dan een proces gecompromitteerd wordt, heeft de aanvaller alleen toegang tot de normale functie van dat proces, en tot bestanden waarvoor het proces ingesteld is om toegang tot te hebben. Bijvoorbeeld, als de Apache HTTP server gecompromitteerd is, kan een aanvaller dat proces niet gebruiken om bestanden te lezen in persoonlijke mappen, behalve als een specifieke SELinux tactiekregel was toegevoegd of ingesteld die deze toegang toestaat.

[2] "Meeting Critical Security Objectives with Security-Enhanced Linux", door Peter Loscocco en Stephen Smalley. Dit artikel was oorspronkelijk gemaakt voor de National Security Agency en is dus in het publieke domein. Refereer naar het *originele artikel* [http://www.nsa.gov/research/_files/selinux/papers/ottawa01/index.shtml] voor details en het document zoals het eerst was vrijgegeven. Alle aanpassingen en veranderingen zijn gemaakt door Murray McAllister.

- SELinux kan gebruikt worden om data vertrouwelijkheid en integriteit af te dwingen, en ook om processen te beschermen voor niet vertrouwde input.

SELinux is niet:

- antivirus software.
- een vervanging voor wachtwoorden, firewall, of andere beveiligings systemen.
- een alles inbegrepen beveiligings oplossing.

SELinux is ontworpen om bestaande beveiligings oplossingen te versterken, niet om ze te vervangen. Zelfs als je SELinux draait, ga dan door met het opvolgen van goede beveilings praktijken, zoals de software up-to-date houden, moeilijk te raden wachtwoorden te gebruiken, firewall, enzovoort

2.2. Voorbeelden

De volgende voorbeelden laten zien hoe SELinux de beveiliging verbetert:

- de standaard actie is weigeren. Als er geen SELinux tactiek regel bestaat om toegang toe te staan, zoals voor een proces om een bestand te openen, wordt toegang geweigerd.
- SELinux kan Linux gebruikers beperken. Er bestaan een aantal beperkte SELinux gebruikers. Linux gebruikers kunnen afgebeeld worden op SELinux gebruikers om voordeel te hebben van beperkte SELinux gebruikers. Bijvoorbeeld, een Linux gebruiker afbeelden op de SELinux user_u gebruiker, resulteert in een Linux gebruiker die niet in staat is om instellen van user ID (setuid) toepassingen te draaien (tenzij anders geconfigureerd), zoals **sudo** en **su**, zowel als ze te verhinderen om bestanden en toepassingen in hun persoonlijke map uit te voeren - als dat ingesteld is, dit belet gebruikers om verdachte bestanden vanuit hun persoonlijke mappen op te starten.
- proces scheiding. Processen draaien in hun eigen domein, wat processen verhindert om toegang te krijgen tot bestanden die door andere processen gebruikt worden, en ook voorkomen dat processen toegang krijgen tot andere processen. Bijvoorbeeld, als SELinux gedraaid wordt, tenzij anders ingesteld, kan een aanvaller een Samba server niet compromitteren, en dan die Samba server gebruiken om te lezen en te schrijven naar bestanden in gebruik van andere processen, zoals een database gebruikt door MySQL®.
- helpt de schade te beperken veroorzaakt door configuratie vergissingen. *Domain Name System (DNS)*[3] servers kunnen een kopie maken van elkaars informatie. Dit staat bekend als zone transfer. Aanvallers kunnen zone transfer gebruiken om DNS servers te vernieuwen met verkeerde informatie. Als je de *Berkeley Internet Name Domain (BIND)*[4] DNS server in Fedora 11 draait, zelfs als een beheerder vergeet te beperken welke server een zone transfer kan uitvoeren, zal de standaard SELinux tactiek voorkomen dat zone bestanden [5] vernieuwd worden door zone transfers, het BIND `named` daemon, en andere processen.
- refereer naar het *Red Hat® Magazine*[6] artikel, *Risk report: Three years of Red Hat Enterprise Linux 4*[78], voor uitbuitingen die beperkt werden dankzij de standaard SELinux doel tactiek in Red Hat® Enterprise Linux® 4.
- refereer naar het *LinuxWorld.com*[9] artikel, *A seatbelt for server software: SELinux blocks real-world exploits*[1011], voor achtergrond informatie over SELinux, en informatie over verscheidene uitbuitingen die SELinux heeft voorkomen.

- refereer naar James Morris's *SELinux mitigates remote root vulnerability in OpenPegasus*[12] blog post, voor informatie over een uitbuiting in *OpenPegasus*[13] die was verlicht door SELinux zoals geleverd met Red Hat Enterprise Linux 4 en 5.

De *Tresys Technology*[14] website heeft een *SELinux Mitigation News*[15] sectie (aan de rechter kant) die recente uitbuitingen laat zien die verlicht of verhinderd zijn door SELinux.

2.3. SELinux architectuur

SELinux is een beveiligingsmodule die ingebouwd is in de Linux kernel. SELinux wordt bestuurt door laadbare tactiekregels. Als toegang plaats vindt die relevant is voor de beveiliging, zoals wanneer een proces probeert een bestand te openen, wordt de operatie onderschept in de kernel door SELinux. Als een SELinux tactiek regel de operatie toestaat, gaat het verder, anders wordt de operatie geblokkeerd en ontvangt het proces een fout.

SELinux beslissingen, zoals het toestaan of tegenhouden van toegang, worden opgeslagen. Deze opslag staat bekend als de Access Vector Cache (AVC). Het opslaan van beslissingen vermindert hoe vaak SELinux tactiekregels geraadpleegd moeten worden, wat de prestaties verbetert. SELinux tactiekregels hebben geen effect als DAC regels toegang als eerste weigeren.

2.4. SELinux in andere operating systemen

Refereer naar de volgende verwijzingen voor informatie over SELinux draaiende in operating systemen:

- Hardened Gentoo: *http://www.gentoo.org/proj/en/hardened/selinux/selinux-handbook.xml*.
- Debian: *http://wiki.debian.org/SELinux*.
- Ubuntu: *https://wiki.ubuntu.com/SELinux* en *https://help.ubuntu.com/community/SELinux*.
- Red Hat Enterprise Linux: *Red Hat Enterprise Linux Deployment Guide*[16] en *Red Hat Enterprise Linux 4 SELinux Guide*[17].
- Fedora: *http://fedoraproject.org/wiki/SELinux* en de *Fedora Core 5 SELinux FAQ*[18].

[14] http://www.tresys.com/
[15] http://www.tresys.com/innovation.php

SELinux context

Processen en bestanden worden gelabeld met een SELinux context die extra informatie bevat, zoals een SELinux gebruiker, rol, type, en, optioneel, een niveau. Als SELinux gedraaid wordt, wordt al deze informatie gebruikt om toegangs contole beslissingen te maken. In Fedora 11, biedt SELinux een combinatie van Role-Based Access Contol (RBAC) (toegangscontrole gebaseerd op rol), Type Enforcement® (TE) (type afdwinging), en, optioneel, Multi-Level Security (MLS) (multi-niveau beveiliging)

Het volgende is een voorbeeld SELinux context. SELinux context wordt gebruikt voor processen, Linux gebruikers, en bestanden in Linux operating systemen die SELinux draaien. Gebruik het **`ls -Z`** commando om de SELinux context van bestanden en mappen te zien:

```
$ ls -Z file1
-rwxrw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0      file1
```

SELinux context volgt de *SELinux gebruiker:rol:type:niveau* syntax:

SELinux gebruiker
: De SELinux gebruiker identiteit is een identiteit die bekend is aan de tactiek en die gemachtigd is voor een specifieke verzameling van rollen, en voor een specifieke MLS reeks. Iedere Linux gebruiker wordt afgebeeld op een SELinux gebruiker via SELinux tactiek. Dit staat Linux gebruikers toe om de beperkingen van SELinux gebruiker te erven. De afgebeelde SELinux identiteit wordt gebruikt in de SELinux context voor processen in die sessie, om te beperken welke rollen en niveau's ze kunnen betreden. Voer het **`semanage login -l`** commando uit als Linux root gebruiker om een lijst van afbeeldingen te zien tussen SELinux en Linux gebruikersaccounts:

```
# /usr/sbin/semanage login -l

Login Name                SELinux User              MLS/MCS Range

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

De output kan verschillen van systeem tot systeem. De `Login Name` kolom laat de Linux gebruikers zien, en de `SELinux User` kolom laat zien op welke SELinux gebruiker de Linux gebruiker is afgebeeld. Voor processen beperkt de SELinux gebruiker welke rollen en niveau's bereikbaar zijn. De laatste kolom, `MLS/MCS Range`, is het niveau gebruikt door Multi-Level Security (MLS) en Multi-Category Security (MCS) (multi-categorie beveiliging). Niveau's worden later kort besproken.

rol
: Onderdeel van SELinux is het Role-Based Access Control (RBAC) beveiligings model. De rol is een kenmerk van RBAC. SELinux gebruikers zijn gemachtigd voor rollen, en rollen zijn gemachtigd voor domeinen. De rol dient als een bemiddelaar tussen domeinen en SELinux gebruikers. De rollen die betreed kunnen worden bepalen welke domeinen betreed kunnen worden - uiteindelijk controleert dit tot welke object types toegang kan worden verkregen. Dit helpt de kwetsbaarheid te verminderen voor rechten escalatie aanvallen.

type
: Het type is een kenmerk van Type Enforcement. Het type definieert een domein voor processen, en een type voor bestanden. SELinux tactiekregels bepalen hoe types toegang hebben tot elkaar, of het nu een domein is die toegang heeft tot een type, of een domein die toegang heeft tot een ander domein. Toegang is alleen toegestaan als er een specifieke SELinux tactiekregel bestaat die dit toestaat.

niveau
: Het niveau is een kenmerk van MLS en Multi-Category Security (MCS). Een MLS reeks is een paar van niveau's, geschreven als *laagniveau-hoogniveau* als de niveau's verschillend zijn, of *laagniveau* als de niveau's identiek zijn (`s0-s0` is hetzelfde als `s0`). Elk niveau is een gevoeligheid-categorie paar, met categorie optioneel. Als er categorieën zijn, wordt het niveau geschreven als *gevoeligheid:categorie-set*. Als er geen categorieën zijn, wordt het geschreven als *gevoeligheid*.

  Als de categorie set een opeenvolgende reeks is, kan het afgekort worden. Bijvoorbeeld, `c0.c3` is hetzelfde als `c0,c1,c2,c3`. Het **`/etc/selinux/targeted/setrans.conf`** bestand beeldt niveau's (`s0:c0`) af op een voor mensen leesbare vorm (`CompanyConfidential`). Bewerk **`setrans.conf`** niet met een tekstverwerker: gebruik **`semanage`** om veranderingen te maken. Refereer naar de semanage(8) manual pagina voor meer informatie. De gerichte tactiek in Fedora 11 dwingt MCS af, en in MCS is een gevoeligheid, `s0`. MCS in Fedora 11 ondersteunt 1024 verschillende categorieën: `c0` tot en met `c1023`. `s0-s0:c0.c1023` is gevoeligheid `s0` en gemachtigd voor alle categorieën.

  MLS dwingt het *Bell-LaPadula Mandatory Access Model*[1] af, en wordt gebruikt in Labeled Security Protection Profile (LSPP) omgevingen. Om MLS beperkingen te gebruiken, installeer je het *selinux-policy-mls* pakket, en je stelt MLS in om de standaard SELinux tactiek te zijn. De MLS tactiek die onderdeel van Fedora is laat veel programma domeinen weg die geen onderdeel waren van de geevalueerde configuratie, en daarom is MLS niet bruikbaar op een bureau werkstation (geen ondersteuning voor het X Windows systeem); echter een MLS tactiek kan gemaakt worden van de *upstream SELinux Referentie Tactiek*[2] die alle programma domeinen bevat.

3.1. Domein overgangen

Een proces in een domein gaat over naar een ander domein door het uitvoeren van een toepassing die het `entrypoint` type heeft voor het nieuwe domein. De `entrypoint` toestemming wordt gebruikt in SELinux tactiek, en controleert welke toepassingen gebruikt kunnen worden om een domein in te gaan. Het volgende voorbeeld laat een domein overgang zien:

1. Een gebruiker wil zijn wachtwoord veranderen. Om zijn wachtwoord te veranderen, gebruikt hij de **`passwd`** toepassing. Het **`/usr/bin/passwd`** uitvoerbare programma is gelabeld met het `passwd_exec_t` type:

   ```
   $ ls -Z /usr/bin/passwd
   -rwsr-xr-x  root root system_u:object_r:passwd_exec_t:s0 /usr/bin/passwd
   ```

   De **passwd** toepassing heeft toegang tot **`/etc/shadow`**, welke gelabeld is met het `shadow_t` type:

[1] http://en.wikipedia.org/wiki/Bell-LaPadula_model
[2] http://oss.tresys.com/projects/refpolicy

```
$ ls -Z /etc/shadow
-r--------  root root system_u:object_r:shadow_t:s0    /etc/shadow
```

2. Een SELinux tactiekregel zegt dat het aan processen die in het `passwd_t` domein draaien toegestaan wordt om te lezen en schrijven naar bestanden gelabeld met het `shadow_t` type. Het `shadow_t` type wordt alleen toegepast bij bestanden die nodig zijn voor een verandering van wachtwoord. Dit omvat **`/etc/gshadow`**, **`/etc/shadow`**, en hun backup bestanden.

3. Een SELinux tactiekregel zegt dat het `passwd_t` domein `entrypoint` toestemming heeft voor het `passwd_exec_t` type.

4. Als een gebruiker de **`/usr/bin/passwd`** toepassing uitvoert, zal het shell proces van de gebruiker overgaan naar het `passwd_t` domein. Met SELinux, omdat de standaard actie weigeren is, en er een regel bestaat die toestaat (onder andere) dat toepassingen die draaien in het `passwd_t` domein toegang hebben tot bestanden gelabeld met het `shadow_t` type, is het de **passwd** toepassing toegestaan om toegang te hebben tot **`/etc/shadow`**, en dus om het wachtwoord van de gebruiker te vernieuwen.

Dit voorbeeld is niet volledig,en wordt gebruikt als een basis voorbeeld om domein overgangen uit te leggen. Hoewel er in werkelijkheid een regel is die subjects, die in het `passwd_t` domein draaien, toestaan om toegang te hebben tot objecten met het `shadow_t` bestandslabel type, moet er voldaan worden aan andere SELinux tactiekregels voordat het subject kan overgaan naar een nieuw domein. In dit voorbeeld, verzekert Type Enforcement dat:

- het `passwd_t` domein kan alleen betreden worden door het uitvoeren van een toepassing gelabeld met het `passwd_exec_t` type; kan alleen uitgevoerd worden met gemachtigde gedeelde bibliotheken, zoals het `lib_t` type; en kan niet uitgevoerd worden enig andere toepassing.

- alleen gemachtigde domeinen, zoals `passwd_t`, kunnen naar bestanden schrijven met het `shadow_t` type. Zelfs als andere processen draaien met root gebruiker rechten, kunnen deze processen niet schrijven naar bestanden gelabeld met het `shadow_t` type, omdat deze niet draaien in het `passwd_t` domein.

- alleen gemachtigde domeinen kunnen overgaan naar het `passwd_t` domein. Bijvoorbeeld, het `sendmail` proces draaiend in het `sendmail_t` domein heeft geen geldige reden om **`passwd`** uit te voeren; daarom kan het nooit overgaan naar het `passwd_t` domein.

- processen die draaien in het `passwd_t` domein kunnen alleen lezen en schrijven naar gemachtigde types, zoals bestanden gelabeld met `etc_t` of `shadow_t` types. Dit verhindert de **passwd** toepassing om misleid te worden om willekeurige bestanden te lezen of te schrijven.

3.2. SELinux context voor processen

Gebruikt het **`ps -eZ`** commando om de SELinux context voor processen te bekijken. Bijvoorbeeld:

1. Open een terminal, zoals **Toepassingen** → **Systeemgereedschappen** → **Terminal**.

2. Voer het **`/usr/bin/passwd`** commando uit. Vul geen nieuw wachtwoord in.

3. Open een nieuwe tab, of een andere terminal, en voer het **`ps -eZ | grep passwd`** commando uit. De output lijkt op het volgende:

```
unconfined_u:unconfined_r:passwd_t:s0-s0:c0.c1023 13212 pts/1 00:00:00
 passwd
```

4. Druk op **Ctrl+C** in de eerste tab om de **passwd** toepassing te stoppen.

Als in dit voorbeeld de **/usr/bin/passwd** toepassing (gelabeld met het passwd_exec_t type) wordt uitgevoerd, gaat het shell proces van de gebruiker over naar het passwd_t domein. Denk eraan: het type definieert een domein voor een proces, en een type voor bestanden.

Gebruik het **ps -eZ** commando om de SELinux context voor draaiende processen te zien. Het volgende is een beperkt voorbeeld van de output, en kan op jouw systeem anders zijn:

```
system_u:system_r:setroubleshootd_t:s0 1866 ?  00:00:08 setroubleshootd
system_u:system_r:dhcpc_t:s0     1869 ?        00:00:00 dhclient
system_u:system_r:sshd_t:s0-s0:c0.c1023 1882 ? 00:00:00 sshd
system_u:system_r:gpm_t:s0       1964 ?        00:00:00 gpm
system_u:system_r:crond_t:s0-s0:c0.c1023 1973 ? 00:00:00 crond
system_u:system_r:kerneloops_t:s0 1983 ?       00:00:05 kerneloops
system_u:system_r:crond_t:s0-s0:c0.c1023 1991 ? 00:00:00 atd
```

De system_r rol wordt gebruikt voor systeem processen, zoals daemons. Type Enforcement afzondert dan elk domein.

3.3. SELinux context voor gebruikers

Gebruik het **id -Z** commando om de SELinux context te zien die verbonden is met jouw Linux gebruiker:

```
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

In Fedora 11 draaien Linux gebruikers standaard als unconfined (onbeperkt). Deze SELinux context laat zien dat de Linux gebruiker is afgebeeld op de SELinux unconfined_u gebruiker, draait in de unconfined_r rol, en draait in het unconfined_t domein. s0-s0 is een MLS reeks, die in dit geval hetzelfde is als s0. De categorieën waar naar de gebruiker toegang heeft zijn gedefinieerd door c0.c1023, wat betekent alle categorieën (c0 tot en met c1023).

Gerichte tactiek

Gerichte tactiek is de standaard SELinux tactiek die gebruikt wordt in Fedora 11. Bij een gerichte tactiek draaien processen waarop gericht wordt in een beperkt domein, en draaien processen waar niet op gericht wordt in een onbeperkt domein. Bijvoorbeeld, standaard draaien ingelogde gebruikers in het `unconfined_t` domein, en systeem processen opgestart door init draaien in het `initrc_t` domein - deze beide domeinen zijn onbeperkt.

Unconfined domains (as well as confined domains) are subject to executable and writeable memory checks. By default, subjects running in an unconfined domain can not allocate writeable memory and execute it. This reduces vulnerability to *buffer overflow attacks*[1]. These memory checks are disabled by setting Booleans, which allow the SELinux policy to be modified at runtime. Boolean configuration is discussed later.

4.1. Beperkte processen

Bijna iedere service die luistert op een netwerk is beperkt in Fedora 11. Bovendien zijn de meeste processen beperkt die gedraaid worden als de Linux root gebruiker en taken uitvoeren voor gebruikers, zoals de **passwd** toepassing. Als een proces beperkt is, draait het in zijn eigen domein, zoals het `httpd` proces draait in het `httpd_t` domein. Als een beperkt proces in gevaar wordt gebracht door een aanvaller zal, afhankelijk van de SELinux tactiek instellingen, de toegang van de aanvaller naar hulpbronnen, en de mogelijke schade die aangericht kan worden, beperkt zijn.

Het volgende voorbeeld laat zien hoe SELinux voorkomt dat de Apache HTTP server (`httpd`) leest van bestanden die niet correct gelabeld zijn, zoals bestanden bedoelt voor gebruik met Samba. Dit is een voorbeeld, en moet niet gebruikt worden in een productieomgeving. Het neemt aan dat de *httpd*, *wget*, *setroubleshoot-server*, en *audit* pakketten geinstalleerd zijn, dat de SELinux gerichte tactiek wordt gebruikt, en dat SELinux draait in de afdwingende (enforcing) modus.

1. Voer het **`sestatus`** commando uit om te bevestigen dat SELinux is aangezet, het draait in de afdwingende modus, en dat de gerichte tactiek wordt gebruikt:

   ```
   $ /usr/sbin/sestatus
   SELinux status:                 enabled
   SELinuxfs mount:                /selinux
   Current mode:                   enforcing
   Mode from config file:          enforcing
   Policy version:                 23
   Policy from config file:        targeted
   ```

   `SELinux status: enabled` is returned when SELinux is enabled. `Current mode: enforcing` is returned when SELinux is running in enforcing mode. `Policy from config file: targeted` is returned when the SELinux targeted policy is used.

2. Als de Linux root gebruiker draai je het **`touch /var/www/html/testfile`** commando om een bestand aan te maken.

3. Draai het **`ls -Z /var/www/html/testfile`** commando om de SELinux context te bekijken:

[1] http://en.wikipedia.org/wiki/Buffer_overflow

```
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/
www/html/testfile
```

Standaard draaien Linux gebruikers onbeperkt in Fedora 11, daarom is het **testfile** bestand gelabeld met de SELinux `unconfined_u` gebruiker. RBAC wordt gebruikt voor processen, niet voor bestanden. Rollen hebben geen betekenis voor bestanden - de `object_r` rol is een algemene rol gebruikt voor bestanden (op blijvende opslag en netwerkbestandsystemen). In de **/proc/** map, kunnen bestanden die gerelateerd zijn aan processen de `system_r` rol gebruiken.[2] Het `httpd_sys_content_t` type staat het `httpd` proces toe om toegang te krijgen tot dit bestand.

4. Als de Linux root gebruiker, draai je het **service httpd start** commando om het `httpd` proces te starten. Als `httpd` met succes opstart verschijnt de volgende output:

```
# /sbin/service httpd start
Starting httpd:                                            [  OK  ]
```

5. Ga naar een map waar je Linux gebruiker schrijfrechten heeft, en draai het **wget http://localhost/testfile** commando. Behalve als er veranderingen in de standaard instelling gemaakt zijn zal dit commando slagen:

```
--2009-05-06 23:00:01--  http://localhost/testfile
Resolving localhost... 127.0.0.1
Connecting to localhost|127.0.0.1|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 0 [text/plain]
Saving to: `testfile'

[ <=>                                 ] 0     --.-K/s   in 0s

2009-05-06 23:00:01 (0.00 B/s) - `testfile' saved [0/0]
```

6. Het **chcon** commando herlabelt bestanden; zulke label veranderingen zullen echter niet blijven bestaan als het bestandssysteem opnieuw gelabeld wordt. Voor permanente veranderingen die een herlabeling van het bestandssysteem zullen overleven, gebruik je het **semanage** commando, dat later besproken wordt. Als de Linux root gebruiker draai je het volgende commando om het type te veranderen naar een type dat door Samba gebruikt wordt:

   chcon -t samba_share_t /var/www/html/testfile

   Draai het **ls -Z /var/www/html/testfile** commando om de veranderingen te bekijken:

```
-rw-r--r--  root root unconfined_u:object_r:samba_share_t:s0 /var/www/
html/testfile
```

7. Merk op: de huidige DAC rechten staan het `httpd` proces toegang toe tot**testfile**. Ga naar een map waar je Linux gebruiker schrijfrechten heeft, en voer het **wget http://localhost/testfile** commando uit. Behalve als er veranderingen in de standaard instelling gemaakt zijn zal dit commando falen:

```
--2009-05-06 23:00:54--  http://localhost/testfile
Resolving localhost... 127.0.0.1
Connecting to localhost|127.0.0.1|:80... connected.
HTTP request sent, awaiting response... 403 Forbidden
2009-05-06 23:00:54 ERROR 403: Forbidden.
```

8. Als de Linux root gebruiker draai je het **`rm -i /var/www/html/testfile`** commando om **`testfile`** te verwijderen.

9. Als het voor jou niet nodig is dat `httpd` draait, voer je als de Linux root gebruiker het **`service httpd stop`** commando uit om `httpd` te stoppen:

```
# /sbin/service httpd stop
Stopping httpd:                                            [  OK  ]
```

Dit voorbeeld laat de extra beveiliging zien die toegevoegd is door SELinux. Hoewel DAC regels het `httpd` proces toegang toestaan tot **`testfile`** in stap 7, zal SELinux toegang weigeren omdat het bestand gelabeld was met een type waarnaar het `httpd` proces geen toegang heeft. Na stap 7 wordt een fout weggeschreven naar **`/var/log/messages`** die lijkt op het volgende:

```
May 6 23:00:54 localhost setroubleshoot: SELinux is preventing httpd
 (httpd_t) "getattr"
to /var/www/html/testfile (samba_share_t). For complete SELinux messages.
run sealert -l c05911d3-e680-4e42-8e36-fe2ab9f8e654
```

Eerdere log bestanden kunnen een **`/var/log/messages.`*`YYYYMMDD`*** formaat gebruiken. Als **syslog-ng** draait, kunnen eerdere log bestanden een **`/var/log/messages.`*`X`*** formaat gebruiken. Als de `setroubleshootd` en `auditd` processen draaien, worden fouten naar **`/var/log/audit/audit.log`** weggeschreven likend op het volgende:

```
type=AVC msg=audit(1220706212.937:70): avc:  denied  { getattr }
 for  pid=1904 comm="httpd" path="/var/www/html/testfile"
 dev=sda5 ino=247576 scontext=unconfined_u:system_r:httpd_t:s0
 tcontext=unconfined_u:object_r:samba_share_t:s0  tclass=file

type=SYSCALL msg=audit(1220706212.937:70): arch=40000003 syscall=196
 success=no exit=-13 a0=b9e21da0 a1=bf9581dc a2=555ff4 a3=2008171 items=0
 ppid=1902 pid=1904 auid=500 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48
 sgid=48 fsgid=48 tty=(none) ses=1 comm="httpd" exe="/usr/sbin/httpd"
 subj=unconfined_u:system_r:httpd_t:s0 key=(null)
```

Er wordt ook een fout weggeschreven naar **`/var/log/httpd/error_log`** lijkend op het volgende:

```
[Wed May 06 23:00:54 2009] [error] [client 127.0.0.1] (13)Permission
 denied: access to /testfile denied
```

Opmerking

In Fedora 11 worden de *setroubleshoot-server* en *audit* pakketten standaard geinstalleerd. Deze pakketten bevatten respectievelijk de `setroubleshootd` en `auditd` daemons. Deze daemons draaien standaard. Het stoppen van een van deze daemons verandert waar de SELinux weigeringen weggeschreven worden. Refereer naar *Paragraaf 5.2, "Welk log bestand wordt gebruikt"* voor meer informatie.

4.2. Onbeperkte processen

Onbeperkte processen draaien in onbeperkte domeien, bijvoorbeeld, init programma's draaien in het onbeperkte `initrc_t` domein, onbeperkte kernel processen draaien in het `kernel_t` domein, en onbeperkte Linux gebruikers draaien in het `unconfined_t` domein. Voor onbeperkte processen worden SELinux tactiekregels toegepast, maar de bestaande tactiekregels staan processen die in onbeperkte domeinen draaien bijna alle toegang toe. Processen die draaien in onbeperkte domeinen vallen terug op het gebruik van alleen de DAC regels. Als een onbeperkt proces in gevaar wordt gebracht, verhindert SELinux een aanvaller niet om toegang te krijgen tot systeemhulpbronnen en data, maar de DAC regels worden natuurlijk nog gebruikt. SELinux is een beveiligingsverbetering boven op DAC regels - het vervangt deze niet.

Het volgende voorbeeld laat zien hoe de Apache HTTP Server (`httpd`) als het onbeperkt draait toegang kan krijgen tot data die bedoeld is voor gebruik met Samba. Merk op: in Fedora 11 draait het `httpd` proces standaard in het beperkte `httpd_t` domein. Dit is een voorbeeld en moet niet in een productieomgeving gebruikt worden. Het neemt aan dat de *httpd*, *wget*, *setroubleshoot-server*, en *audit* pakketten geinstalleerd zijn, dat de SELinux gerichte tactiek gebruikt wordt, en dat SELinux in de afdwingende modus is:

1. Voer het **`sestatus`** commando uit om te bevestigen dat SELinux is aangezet, het draait in de afdwingende modus, en dat de gerichte tactiek wordt gebruikt:

```
$ /usr/sbin/sestatus
SELinux status:                 enabled
SELinuxfs mount:                /selinux
Current mode:                   enforcing
Mode from config file:          enforcing
Policy version:                 23
Policy from config file:        targeted
```

   `SELinux status: enabled` is returned when SELinux is enabled. `Current mode: enforcing` is returned when SELinux is running in enforcing mode. `Policy from config file: targeted` is returned when the SELinux targeted policy is used.

2. Als de Linux root gebruikerm voer je het **`touch /var/www/html/test2file`** commando uit om een bestand te maken.

3. Voer het **`ls -Z /var/www/html/test2file`** commando uit om de SELinux context te zien:

```
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/
www/html/test2file
```

Standaard draaien Linux gebruikers onbeperkt in Fedora 11, daarom is het **test2file** bestand gelabeld met de SELinux unconfined_u gebruiker. RBAC wordt gebruikt voor processen, niet voor bestanden. Rollen hebben geen betekenis voor bestanden - de object_r rol is een algemene rol gebruikt voor bestanden (op blijvende opslag en netwerkbestandssystemen). In de **/proc/** map, kunnen bestanden die gerelateerd zijn aan processen de system_r rol gebruiken.[3] Het httpd_sys_content_t type staat het httpd proces toe om toegang te krijgen tot dit bestand.

4. Het **chcon** commando herlabelt bestanden; zulke label veranderingen zullen echter niet blijven bestaan als het bestandssysteem opnieuw gelabeld wordt. Voor permanente veranderingen die een herlabeling van het bestandssysteem zullen overleven, gebruik je het **semanage** commando, dat later besproken wordt. Als de Linux root gebruiker draai je het volgende commando om het type te veranderen naar een type dat door Samba gebruikt wordt:

   chcon -t samba_share_t /var/www/html/test2file

   Voer het **ls -Z /var/www/html/test2file** commando uit om de veranderingen te bekijken:

   ```
   -rw-r--r--  root root unconfined_u:object_r:samba_share_t:s0 /var/www/
   html/test2file
   ```

5. Voer het **service httpd status** commando uit om te bevestigen dat het httpd proces niet draait:

   ```
   $ /sbin/service httpd status
   httpd is stopped
   ```

   Als de output anders is, voer je het **service httpd stop** commando uit als de Linux root gebruiker om het httpd proces te stoppen:

   ```
   # /sbin/service httpd stop
   Stopping httpd:                                            [  OK  ]
   ```

6. Om het httpd proces onbeperkt te laten draaien, voer je het volgende commando uit als de Linux root gebruiker om het type van **/usr/sbin/httpd** te veranderen naar een type dat niet overgaat naar een beperkt domein:

   chcon -t unconfined_exec_t /usr/sbin/httpd

7. Voer het **ls -Z /usr/sbin/httpd** commando uit om te bevestigen dat **/usr/sbin/httpd** is gelabeld met het unconfined_exec_t type:

   ```
   -rwxr-xr-x  root root system_u:object_r:unconfined_exec_t /usr/sbin/
   httpd
   ```

8. Als de Linux root gebruiker, draai je het **service httpd start** commando om het httpd proces te starten. Als httpd met succes opstart verschijnt de volgende output:

   ```
   # /sbin/service httpd start
   ```

```
Starting httpd:                                            [  OK  ]
```

9. Voer het **ps -eZ | grep httpd** commando uit om `httpd` te zien draaien in het `unconfined_t` domein:

```
$ ps -eZ | grep httpd
unconfined_u:system_r:unconfined_t 7721 ?      00:00:00 httpd
unconfined_u:system_r:unconfined_t 7723 ?      00:00:00 httpd
unconfined_u:system_r:unconfined_t 7724 ?      00:00:00 httpd
unconfined_u:system_r:unconfined_t 7725 ?      00:00:00 httpd
unconfined_u:system_r:unconfined_t 7726 ?      00:00:00 httpd
unconfined_u:system_r:unconfined_t 7727 ?      00:00:00 httpd
unconfined_u:system_r:unconfined_t 7728 ?      00:00:00 httpd
unconfined_u:system_r:unconfined_t 7729 ?      00:00:00 httpd
unconfined_u:system_r:unconfined_t 7730 ?      00:00:00 httpd
```

10. Ga naar een map waar jouw Linux gebruiker schrijfrechten heeft, en voer het **wget http://localhost/test2file** commando uit. Behalve als er veranderingen in de standaard instelling gemaakt zijn, zal dit commando slagen:

```
--2009-05-07 01:41:10--  http://localhost/test2file
Resolving localhost... 127.0.0.1
Connecting to localhost|127.0.0.1|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 0 [text/plain]
Saving to: `test2file.1'

[ <=>                                ]--.-K/s   in 0s

2009-05-07 01:41:10 (0.00 B/s) - `test2file.1' saved [0/0]
```

Hoewel het `httpd` proces geen toegang heeft tot bestanden gelabeld met het `samba_share_t` type, draait `httpd` in het onbeperkte `unconfined_t` domein, en valt terug op het gebruiken van DAC regels, en dus zal het **wget** commando slagen. Als `httpd` in het beperkte `httpd_t` domein had gedraaid, dan zou het **wget** commando gefaald hebben.

11. Het **restorecon** commando herlaadt de standaard SELinux context voor bestanden. Als de Linux root gebruiker voet je het **restorecon -v /usr/sbin/httpd** commando uit om de standaard SELinux context voor **/usr/sbin/httpd** te herladen:

```
# /sbin/restorecon -v /usr/sbin/httpd
restorecon reset /usr/sbin/httpd context
 system_u:object_r:unconfined_notrans_exec_t:s0-
>system_u:object_r:httpd_exec_t:s0
```

Voer het **ls -Z /usr/sbin/httpd** commando uit om te bevestigen dat **/usr/sbin/httpd** is gelabeld met het `httpd_exec_t` type:

```
$ ls -Z /usr/sbin/httpd
```

```
-rwxr-xr-x  root root system_u:object_r:httpd_exec_t   /usr/sbin/httpd
```

12. Als de Linux root gebruiker voer je het **`/sbin/service httpd restart`** commando uit om `httpd` opnieuw te starten. Na het herstarten, voer je het **`ps -eZ | grep httpd`** commando uit om te bevestigen dat `httpd` in het beperkte `httpd_t` domein draait:

```
# /sbin/service httpd restart
Stopping httpd:                                            [  OK  ]
Starting httpd:                                            [  OK  ]
# ps -eZ | grep httpd
unconfined_u:system_r:httpd_t    8880 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8882 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8883 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8884 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8885 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8886 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8887 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8888 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8889 ?        00:00:00 httpd
```

13. Als de Linux root gebruiker voer je het **`rm -i /var/www/html/test2file`** commando uit om **`test2file`** te verwijderen.

14. Als het voor jou niet nodig is dat `httpd` draait, voer je als de Linux root gebruiker het **`service httpd stop`** commando uit om `httpd` te stoppen:

```
# /sbin/service httpd stop
Stopping httpd:                                            [  OK  ]
```

De voorbeelden in deze paragrafen laten zien hoe data kan worden beschermd voor een in gevaar gebracht beperkt proces (beschermd door SELinux), en ook hoe data beter bereikbaar is voor een aanvaller vanuit een in gevaar gebracht onbeperkt proces (niet beschermd door SELinux).

4.3. Beperkte en onbeperkte gebruikers

Elke Linux gebruiker wordt afgebeeld op een SELinux gebruiker met SELinux tactiek. Dit staat Linux gebruikers toe om de beperkingen voor SELinux gebruikers te erven. Deze Linux gebruiker afbeelding kan bekeken worden door het uitvoeren van het **`semanage login -l`** commando als de Linux root gebruiker:

```
# /usr/sbin/semanage login -l

Login Name                SELinux User              MLS/MCS Range

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

In Fedora 11 worden Linux gebruikers standaard afgebeeld op de __default__ login (welke is afgebeeld op de SELinux unconfined_u gebruiker). Het volgende definieert de standaard afbeelding:

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

Het volgende voorbeeld laat het toevoegen van een nieuwe Linux gebruiker zien, en het afbeelden van die Linux gebruiker op de SELinux unconfined_u gebruiker. Het neemt aan dat de Linux root gebruiker onbeperkt draait, wat standaard het geval is in Fedora 11:

1. Als de Linux root gebruiker voer je het **/usr/sbin/useradd newuser** commando uit om een nieuwe Linux gebruiker aan te maken met de naam newuser.

2. As the Linux root user, run the **passwd newuser** command to assign a password to the Linux newuser user:

   ```
   # passwd newuser
   Changing password for user newuser.
   New UNIX password: Enter a password
   Retype new UNIX password: Enter the same password again
   passwd: all authentication tokens updated successfully.
   ```

3. Log uit van je huidige sessie, en login als de Linux newuser gebruiker. Als je inlogt, beeldt pam_selinux de Linux gebruiker af op een SELinux gebruiker (in dit geval, unconfined_u), en stelt de daaruit volgende SELinux context in. De shell van de Lunix gebruiker wordt opgestart met deze context. Voer het **id -Z** commando uit om de context van een Linux gebruiker te bekijken:

   ```
   [newuser@localhost ~]$ id -Z
   unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
   ```

4. Log uit van de sessie van de Linux newuser, en log in met je eigen account. Als je de Linux newuser gebruiker niet wilt, voer je het **/usr/sbin/userdel -r newuser** commando uit als de Linux root gebruiker om het te verwijderen, te samen met de persoonlijke map van de Linux newuser.

Beperkte en onbeperkte Linux gebruikers zijn onderworpen aan uitvoerbaarheids en naar-geheugen-schrijven controles, en zijn ook beperkt door MCS (en MLS, als de MLS tactiek wordt gebruikt). Als onbeperkte Linux gebruikers een toepassing uitvoeren waarvan SELinux tactiek definieert dat het kan overgaan van het unconfined_t domein naar zijn eigen beperkt domein, zullen onbeperkte Linux gebruikers nog steeds vallen onder de beperkingen van dat beperkte domein. Het beveilings voordeel hiervan is dat, zelfs als een Linux gebruiker onbeperkt draait, de toepassing beperkt blijft, en dat daardoor de uitbuiting van een fout in de toepassing beperkt kan worden door de tactiek. Merk op: dit beschermt het systeem niet voor de gebruiker. In plaats daarvan worden de gebruiker en het systeem beschermd tegen mogelijke schade veroorzaakt door een fout in de toepassing.

De volgende beperkte SELinux gebruikers zijn beschikbaar in Fedora 11:

Gebruiker	Domein	X Window systeem	su en sudo	Uitvoeren in persoonlijke map en /tmp/	Netwerken
guest_u	guest_t	nee	nee	optioneel	nee

Gebruiker	Domein	X Window systeem	su en sudo	Uitvoeren in persoonlijke map en /tmp/	Netwerken
xguest_u	xguest_t	ja	nee	optioneel	alleen **Firefox**
user_u	user_t	ja	nee	optioneel	ja
staff_u	staff_t	ja	alleen **sudo**	optioneel	ja

Tabel 4.1. SELinux gebruiker eigenschappen

- Linux gebruikers in de `guest_t`, `xguest_t`, en `user_t` domeinen kunnen alleen set user ID (setuid) toepassingen draaien als SELinux tactiek dat toestaat (zoals **passwd**). Ze kunnen de **su** en **/usr/bin/sudo** setuid toepassingen niet draaien, en kunnen daarom deze toepassingen niet gebruiken om de Linux root gebruiker te worden.
- Linux gebruikers in het `guest_t` domein hebben geen netwerk toegang, en kunnen alleen inloggen met een terminal (inclusief `ssh`; ze kunnen inloggen met `ssh`, maar kunnen `ssh` niet gebruiken om te verbinden met andere systemen).
- De enigste netwerk toegang die Linux gebruikers in het `xguest_t` domein hebben is om met **Firefox** te verbinden met web pagina's.
- Linux gebruikers in de `xguest_t`, `user_t` en `staff_t` domeinen kunnen inloggen met het X Window systeem en een terminal.
- Standaard hebben Linux gebruikers in het `staff_t` domein geen rechten om toepassingen met **/usr/bin/sudo** uit te voeren. Deze rechten moeten ingesteld worden door een beheerder.

Standaard kunnen Linux gebruikers in de `guest_t` en `xguest_t` domeinen geen toepassingen in hun persoonlijke mappen of **/tmp/** uitvoeren, wat hun tegenhoudt om toepassingen op te starten (welke de rechten van de gebruiker erven) in mappen waartoe ze schrijftoegang hebben. Dit helpt om foutieve of kwaadwillige toepassingen te verhinderen om bestanden waarvan ze eigenaar zijn te veranderen.

Standaard kunnen Linux gebruikers in de `user_t` en `staff_t` domeinen toepassingen in hun persoonlijke mappen en **/tmp/** uitvoeren. Refereer naar *Paragraaf 6.6, “Booleans voor gebruikers die toepassingen uitvoeren”* voor informatie over het toestaan en tegenhouden van gebruikers om toepassingen in hun persoonlijke mappen en **/tmp/** uit te voeren.

Werken met SELinux

De volgende paragrafen geven een kort overzicht van de belangrijkste SELinux pakketten in Fedora 11; het installeren en vernieuwen van pakketten; welke log bestanden gebruikt worden; het belangrijkste SELinux configuratie bestand; SELinux aanzetten en uitzetten, SELinux modes; het instellen van Booleans; het tijdelijk en blijvend veranderen van bestand en map labels; het voorbij gaan aan bestandssysteem labels met het **`mount`** commando; het aankoppelen van NFS bestandssystemen; en hoe je SELinux context behoudt bij het kopiëren en archiveren van bestanden en mappen.

5.1. SELinux pakketten

In Fedora 11 worden de SELinux pakketten standaard geinstalleerd, behalve als ze handmatig uitgezonderd worden tijdens de installatie. Standaard wordt de SELinux gerichte tactiek gebruikt, en draait SELinux in de afdwingende modus. Het volgende is een korte beschrijving van de belangrijkste SELinux pakketten:

policycoreutils: biedt gereedschappen zoals **`semanage`**, **`restorecon`**, **`audit2allow`**, **`semodule`**, **`load_policy`**, en **`setsebool`**, voor het uitvoeren en beheren van SELinux.

policycoreutils-gui: biedt **`system-config-selinux`**, een grafisch gereedschap voor het beheren van SELinux.

selinux-policy: biedt de SELinux Referentie Tactiek. De SELinux Referentie Tactiek is een complete SELinux tactiek, en wordt gebruikt als de basis voor andere tactieken, zoals de SELinux gerichte tactiek. Refereer naar de Tresys Technology *SELinux Reference Policy*[1] voor verdere informatie. Het *selinux-policy-devel* pakket levelt ontwikkelgereedschappen, zoals **`/usr/share/selinux/devel/policygentool`** en **`/usr/share/selinux/devel/policyhelp`**, en ook voorbeeld tactiek bestanden. Dit pakket is opgegaan in het *selinux-policy* pakket.

selinux-policy-`policy`: levert SELinux tactieken. Voor gerichte tactiek, installeer je *selinux-policy-targeted*. Voor MLS, installeer je *selinux-policy-mls*. In Fedora 8 is de strikte tactiek opgegaan in de gerichte tactiek, wat toestaat dat beperkte en onbeperkte gebruikers tegelijk kunnen bestaan op hetzelfde systeem.

setroubleshoot-server: vertaalt weigeringsboodschappen, die gemaakt worden als toegang geweigerd wordt door SELinux, in gedetaileerde beschrijvingen die bekeken kunnen worden met **`sealert`** (welke door dit pakket geleverd wordt).

setools, *setools-gui*, en *setools-console*: deze pakketten leveren de *Tresys Technology SETools distributie*[2], een aantal gereedschappen en bibliotheken voor het analiseren en ondervragen van tactiek, controle log waarnemen en rapporteren, en bestandscontext beheer [3]. Het *setools* pakket is een meta-pakket voor SETools. Het *setools-gui* pakket levert de **`apol`**, **`seaudit`**, en **`sediffx`** gereedschappen. Het *setools-console* pakket levert de **`seaudit-report`**, **`sechecker`**, **`sediff`**, **`seinfo`**, **`sesearch`**, **`findcon`**, **`replcon`**, en **`indexcon`** commadoregel gereedschappen. Refereer naar de *Tresys Technology SETools*[4] pagina voor meer informatie over deze gereedschappen.

[1] http://oss.tresys.com/projects/refpolicy
[2] http://oss.tresys.com/projects/setools
[3] Brindle, Joshua. "Re: blurb for fedora setools packages" Email aan Murray McAllister. 1 November 2008. Elke bewerking en verandering in deze versie is gedaan door Murray McAllister.
[4] http://oss.tresys.com/projects/setools

libselinux-utils: levert de **`avcstat`**, **`getenforce`**, **`getsebool`**, **`matchpathcon`**, **`selinuxconlist`**, **`selinuxdefcon`**, **`selinuxenabled`**, **`setenforce`**, en **`togglesebool`** gereedschappen.

mcstrans: vertaalt niveau's, zoals `s0-s0:c0.c1023`, naar een eenvoudiger te lezen vorm, zoals `SystemLow-SystemHigh`. Dit pakket is standaard niet geinstalleerd.

Om pakketten in Fedora 11 te installeren, draai je als de Linux root gebruiker het **`yum install pakket-naam`** commando. Bijvoorbeeld, om het *mcstrans* pakket te installeren, voer je het **`yum install mcstrans`** commando uit. Om alle geinstalleerde pakketten in Fedora 11 te vernieuwen, voer je het **`yum update`** commando uit.

Refereer naar *Managing Software with yum*[56] voor meer informatie over het gebruik van **`yum`** om pakketten te beheren.

Opmerking

In vorige versies van Fedora is het *selinux-policy-devel* pakket nodig voor het maken van een locale tactiek module met **`audit2allow -M`**.

5.2. Welk log bestand wordt gebruikt

In Fedora 11 worden de *setroubleshoot-server* en *audit* pakketten geinstalleerd als ze niet verwijderd zijn van de standaard software selectie. Deze pakketten bevatten respectievelijk de `setroubleshootd` en `auditd` daemons. Deze daemons draaien standaard.

SELinux weigeringsboodschappen, zoals de volgende worden standaard naar **`/var/log/audit/audit.log`** geschreven:

```
type=AVC msg=audit(1223024155.684:49): avc:  denied  { getattr }
 for  pid=2000 comm="httpd" path="/var/www/html/file1"
 dev=dm-0 ino=399185 scontext=unconfined_u:system_r:httpd_t:s0
 tcontext=system_u:object_r:samba_share_t:s0 tclass=file
```

Als `setroubleshootd` draait, wat standaard het geval is, worden weigeringsboodschappen van **`/var/log/audit/audit.log`** ook vertaald naar een eenvoudiger te lezen vorm en naar **`/var/log/messages`** gestuurd:

```
May  7 18:55:56 localhost setroubleshoot: SELinux is preventing httpd
 (httpd_t) "getattr" to /var/www/html/file1 (samba_share_t). For complete
 SELinux messages. run sealert -l de7e30d6-5488-466d-a606-92c9f40d316d
```

Weigeringsboodschappen worden naar verscheidene locaties gestuurd, afhankelijk van welke daemons draaien:

Daemon	Log Location
auditd on	**`/var/log/audit/audit.log`**
auditd off; rsyslogd on	**`/var/log/messages`**

[5] http://docs.fedoraproject.org/yum/en/
[6] Managing Software with yum, geschreven door Stuart Ellis, bewerkt door Paul W. Frields, Rodrigo Menezes, en Hugo Cisneiros.

Daemon	Log Location
setroubleshootd, rsyslogd, and auditd on	**/var/log/audit/audit.log**. Easier-to-read denial messages also sent to **/var/log/messages**

Daemons automatisch opstarten

Om de `auditd`, `rsyslogd`, en `setroubleshootd` daemons in te stellen om automatisch op te starten bij het opstarten van het systeem, voer je de volgende commando's uit als de Linux root gebruiker:

```
/sbin/chkconfig --levels 2345 auditd on
```

```
/sbin/chkconfig --levels 2345 rsyslog on
```

```
/sbin/chkconfig --levels 345 setroubleshoot on
```

Gebruik het **service *service-naam* status** commando om te controleren of deze services draaien, bijvoorbeeld:

```
$ /sbin/service auditd status
auditd (pid  1318) is running...
```

Als de hierboven genoemde services niet draaien (*service-naam* `is stopped`), voer je het **service *service-naam* start** commando uit als de Liinux root gebruiker om ze te starten. Bijvoorbeeld:

```
# /sbin/service setroubleshoot start
Starting setroubleshootd:                                  [  OK  ]
```

5.3. Het hoofd configuratie bestand

Het **/etc/selinux/config** bestand is het hoofd SELinux configuratie bestand. Het controleert de SELinux modus en de te gebruiken SELinux tactiek:

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

`SELINUX=enforcing (afdwingend)`

De `SELINUX` optie stelt de modus in waarin SELinux draait. SELinux heeft drie modes: enforcing (afdwingend), permissive (toelatend), en disabled (uitgezet). Als de afdwingende modus gebruikt

wordt, wordt de SELinux tactiek afgedwongen, en SELinux verbiedt toegang gebaseerd op SELinux tactiekregels. Weigeringsboodschappen worden gelogd. Als de toelatende modus gebruikt wordt, wordt de SELinux tactiek niet afgedwongen. SELinux verbiedt geen toegang, maar weigeringen voor acties die geweigerd zouden zijn als SELinux in de afdwingende modus zou zijn, worden gelogd. In de uitgezette modus, is SELinux uitgezet (de SELinux module wordt niet registreerd bij de Linux kernel), en alleen DAC regels worden gebruikt.

`SELINUXTYPE=targeted (gericht)`
: De `SELINUXTYPE` optie stelt de te gebruiken SELinux tactiek in. Gerichte tactiek is de standaard tactiek. Verander deze optie alleen als je de MLS tactiek wilt gebruiken. Om de MLS tactiek te gebruiken, installeer je het *selinux-policy-mls* pakket, je configureert `SELINUXTYPE=mls` in **`/etc/selinux/config`**; en je start je systeem opnieuw op.

Belangrijk

Als systemen draaien in de toelatende of uitgezette modes, hebben gebruikers toestemming om bestanden verkeerd te labelen. Ook worden bestanden aangemaakt terwijl SELinux uitgezet is niet gelabeld. Dit veroorzaakt problemen als daarna de modus naar afdwingend wordt veranderd. Om te voorkomen dat verkeerd gelabelde of niet gelabelde bestanden problemen veroorzaken, worden bestandssystemen automatisch geherlabeld als de modus verandert van uitgezet naar de toelatende of afdwingende modus.

5.4. SELinux aanzetten en uitzetten

Gebruik de **`/usr/sbin/getenforce`** of **`/usr/sbin/sestatus`** commando's om de status van SELinux te controleren. Het **`getenforce`** commando geeft `Enforcing`, `Permissive`, of `Disabled` terug. Het **`getenforce`** geeft `Enforcing` terug als SELinux is aangezet (SELinux tactiekregels zijn afgedwongen):

```
$ /usr/sbin/getenforce
Enforcing
```

Het **`getenforce`** commando geeft `Permissive` terug als SELinux is aangezet, maar SELinux tactiekregels worden niet afgedwongen, en alleen DAC regels worden gebruikt. Het **`getenforce`** commando geeft `Disabled` terug als SELinux is uitgezet.

Het **`sestatus`** commando geeft de SELinux status en de gebruikte SELinux tactiek terug:

```
$ /usr/sbin/sestatus
SELinux status:                 enabled
SELinuxfs mount:                /selinux
Current mode:                   enforcing
Mode from config file:          enforcing
Policy version:                 23
Policy from config file:        targeted
```

`SELinux status: enabled` is returned when SELinux is enabled. `Current mode: enforcing` is returned when SELinux is running in enforcing mode. `Policy from config file: targeted` is returned when the SELinux targeted policy is used.

5.4.1. SELinux aanzetten

Op systemen waar SELinux is uitgezet, is de `SELINUX=disabled` optie ingesteld in **`/etc/selinux/config`**:

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

Dus het **`getenforce`** commando geeft `Disabled` terug:

```
$ /usr/sbin/getenforce
Disabled
```

Om SELinux aan te zetten:

1. Gebruik de **`rpm -qa | grep selinux`**, **`rpm -q policycoreutils`**, en **`rpm -qa | grep setroubleshoot`** commando's om te bevestigen dat de SELinux pakketten geinstalleerd zijn. Deze gids neemt aan dat de volgende pakketten geinstalleerd zijn: *selinux-policy-targeted*, *selinux-policy*, *libselinux*, *libselinux-python*, *libselinux-utils*, *policycoreutils*, *setroubleshoot*, *setroubleshoot-server* en *setroubleshoot-plugins*. Als deze pakketten niet geinstalleerd zijn, installeer je ze als de Linux root gebruiker met het **`yum install pakket-naam`** commando. De volgende pakketten zijn optioneel: *policycoreutils-gui*, *setroubleshoot*, *selinux-policy-devel*, en *mcstrans*.

   Na het installeren van het *setroubleshoot-server* pakket, gebruik je het **`/sbin/chkconfig --list setroubleshoot`** commando om te bevestigen dat `setroubleshootd` opstart als het systeem draait in runlevel[7] 3, 4, en 5:

   ```
   $ /sbin/chkconfig --list setroubleshoot
   setroubleshoot  0:off   1:off   2:off   3:on    4:on    5:on    6:off
   ```

   Als de output anders is, voer je als de Linux root gebruiker het **`/sbin/chkconfig --levels 345 setroubleshoot on`** commando uit. Hierdoor start `setroubleshootd` automatisch op als het systeem in runlevel 3, 4, en 5 draait.

2. Voordat SELinux wordt aangezet, moet elk bestand in het bestandssysteem gelabeld worden met een SELinux context. Voordat dit gebeurt, kunnen beperkte domeinen toegang geweigerd worden, wat je systeem ervan weerhoudt om correct op te starten. Om dit te voorkomen, configureer je `SELINUX=permissive` in **`/etc/selinux/config`**:

   ```
   # This file controls the state of SELinux on the system.
   # SELINUX= can take one of these three values:
   #       enforcing - SELinux security policy is enforced.
   #       permissive - SELinux prints warnings instead of enforcing.
   ```

```
#       disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of these two values:
#       targeted - Targeted processes are protected,
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

3. Als de Linux root gebruiker voer je het **`reboot`** commando uit om het systeem opnieuw op te starten. Tijdens de volgende start worden de bestandssystemen gelabeld. Het label proces labelt alle bestanden met een SELinux context:

   ```
   *** Warning -- SELinux targeted policy relabel is required.
   *** Relabeling could take a very long time, depending on file
   *** system size and speed of hard drives.
   ****
   ```

   Elke * karakter in de onderste regel representeerd 1000 bestanden die gelabeld zijn. In het bovenstaande voorbeeld, representeren vier * karakters 4000 bestanden die gelabeld zijn. De tijd die het duurt om alle bestanden te labelen hangt af van het aantal bestanden op het systeem, en de snelheid van de harde schijf stations. Op moderne systemen kan dit proces 10 minuten duren.

4. In de toelatende modus, wordt SELinux tactiek niet afgedwongen, maar weigeringen worden nog steeds gelogd voor acties die geweigerd zouden zijn als het systeem in de afdwingende modus zou draaien. Voordat je verandert naar de afdwingende modus, voer je als de Linux root gebruiker het **`grep "SELinux is preventing" /var/log/messages`** commando uit om te bevestigen dat SELinux geen acties heeft geweigerd tijdens het laatste opstarten. Als SELinux geen acties heeft geweigerd tijdens het laatste opstarten, geeft dit commando geen output terug. Refereer naar *Hoofdstuk 7, Foutzoeken* voor foutzoek informatie als SELinux toegang heeft geweigerd tijdens het opstarten.

5. Als er geen weigeringsboodschappen in **`/var/log/messages`** waren, configureer je `SELINUX=enforcing` in **`/etc/selinux/config`**:

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

6. Start je systeem opnieuw op. Na het opstarten, bevestig je dat het **`getenforce`** commando `Enforcing` terug geeft:

   ```
   $ /usr/sbin/getenforce
   Enforcing
   ```

7. Als de Linux root gebruiker voer je het **`/usr/sbin/semanage login -l`** commando uit op de afbeelding tussen SELinux en Linux gebruikers te bekijken. De output moet als volgt zijn:

```
Login Name                SELinux User              MLS/MCS Range

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

Als dit niet het geval is, voer je als de Linux root gebruiker de volgende commando's uit om de gebruikersafbeeldingen te herstellen. Het is veilig om de `SELinux-user` *`gebruikersnaam`* `is already defined` waarschuwingen te negeren als ze voorkomen, waarin *`gebruikersnaam`* `unconfined_u`, `guest_u`, of `xguest_u` kan zijn:

1.
```
/usr/sbin/semanage user -a -S targeted -P user -R "unconfined_r
 system_r" -r s0-s0:c0.c1023 unconfined_u
```

2.
```
/usr/sbin/semanage login -m -S targeted -s "unconfined_u" -r s0-
s0:c0.c1023 __default__
```

3.
```
/usr/sbin/semanage login -m -S targeted -s "unconfined_u" -r s0-
s0:c0.c1023 root
```

4.
```
/usr/sbin/semanage user -a -S targeted -P user -R guest_r guest_u
```

5.
```
/usr/sbin/semanage user -a -S targeted  -P user -R xguest_r xguest_u
```

Belangrijk

Als systemen draaien in de toelatende of uitgezette modes, hebben gebruikers toestemming om bestanden verkeerd te labelen. Ook worden bestanden aangemaakt terwijl SELinux uitgezet is niet gelabeld. Dit veroorzaakt problemen als daarna de modus naar afdwingend wordt veranderd. Om te voorkomen dat verkeerd gelabelde of niet gelabelde bestanden problemen veroorzaken, worden bestandssystemen automatisch geherlabeld als de modus verandert van uitgezet naar de toelatende of afdwingende modus.

5.4.2. SELinux uitzetten

Om SELinux uit te zetten, configureer je `SELINUX=disabled` in **`/etc/selinux/config`**:

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

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

Start je systeem opnieuw op. Na het opstarten bevestig je dat het **`getenforce`** commando `Disabled` terug geeft:

```
$ /usr/sbin/getenforce
Disabled
```

5.5. SELinux modes

SELinux heeft drie modes:

- Afdwingend: SELinux tactiek wordt afgedwongen. SELinux verbiedt toegang gebaseerd op SELinux tactiekreges.
- Toestaand: SELinux tactiek wordt niet afgedwongen. SELinux verbiedt geen toegang, maar weigeringen worden gelogd voor acties die verboden zouden zijn als de afdwingende modus actief zou zijn.
- Uitgezet: SELinux is uitgezet. Alleen DAC regels worden gebruikt.

Gebruik het **`/usr/sbin/setenforce`** commando om te veranderen tussen afdwingende en toelatende mode. Veranderingen gemaakt met **`/usr/sbin/setenforce`** zijn niet blijvend na een heropstarten van het systeem. Om naar de afdwingende modus te veranderen, voer je als de Linux root gebruiker het **`/usr/sbin/setenforce 1`** commando uit. Om naar de toelatende modus te veranderen, voer je het **`/usr/sbin/setenforce 0`** commando uit. Gebruik het **`/usr/sbin/getenforce`** commando om de huidige SELinux modus te bekijken.

Blijvende modus veranderingen worden behandeld in *Paragraaf 5.4, “SELinux aanzetten en uitzetten”*.

5.6. Booleans

Booleans staan toe dat onderdelen van SELinux tactiek veranderd worden tijdens het draaien, zonder dat enige kennis nodig is over het schrijven van SELinux tactiek. Dit staat toe om veranderingen te maken, zoals het toestaan van services toegang tot NFS bestandssystemen, zonder het herladen of hercompileren van SELinux tactiek.

5.6.1. Booleans laten zien

Voor een lijst van Booleans, een uitleg over wat ze zijn, en of ze aan of uit zijn, voer je het **`semanage boolean -l`** commando uit als de Linux root gebruiker. Het volgende voorbeeld laat niet alle Booleans zien:

```
# /usr/sbin/semanage boolean -l
SELinux boolean                          Description

ftp_home_dir                   -> off   Allow ftp to read and write files
 in the user home directories
```

```
xen_use_nfs                    -> off   Allow xen to manage nfs files
xguest_connect_network         -> on    Allow xguest to configure Network
 Manager
```

De `SELinux boolean` kolom laat de lijst van Boolean namen zien. De `Description` kolom laat zien of de Booleans aan of uit zijn, en wat ze doen.

In het volgende voorbeeld, is de `ftp_home_dir` Boolean uit, dit belet de FTP daemon (`vsftpd`) om bestanden in de persoonlijke mappen van de gebruiker te lezen of te schrijven:

```
ftp_home_dir                   -> off   Allow ftp to read and write files
 in the user home directories
```

Het **`getsebool -a`** commando laat een lijst zien van de Booleans, of ze nu aan of uit zijn, maar geeft geen beschrijving van elke Boolean. Het volgende voorbeeld laat niet alle Booleans zien:

```
$ /usr/sbin/getsebool -a
allow_console_login --> off
allow_cvs_read_shadow --> off
allow_daemons_dump_core --> on
```

Voer het **`getsebool`** ***`boolean-naam`*** commando uit om alleen de status van de *`boolean-naam`* Boolean te laten zien:

```
$ /usr/sbin/getsebool allow_console_login
allow_console_login --> off
```

Gebruik een met spaties gescheiden lijst om meerdere Booleans te laten zien:

```
$ getsebool allow_console_login allow_cvs_read_shadow
 allow_daemons_dump_core
allow_console_login --> off
allow_cvs_read_shadow --> off
allow_daemons_dump_core --> on
```

5.6.2. Booleans instellen

Het **`setsebool`** ***`boolean-naam x`*** commando zet Booleans aan of uit, waarin *`boolean-naam`* de naam van een Boolean is, en *`x`* of `on` is om de Boolean aan te zetten, of `off` is om hem uit te zetten.

Het volgende voorbeeld laat het instellen van de `httpd_can_network_connect_db` Boolean zien:

1. Standaard is de `httpd_can_network_connect_db` Boolean uit, wat Apache HTTP scripts en modules belet om te verbinden met database servers:

   ```
   $ /usr/sbin/getsebool httpd_can_network_connect_db
   httpd_can_network_connect_db --> off
   ```

2. Om Apache HTTP server scripts en modules tijdelijk toe te staan om te verbinden met database servers, voer je het **`setsebool httpd_can_network_connect_db on`** commando uit als de Linux root gebruiker.

3. Gebruik het **`getsebool httpd_can_network_connect_db`** commando om te bevestigen dat de Boolean aangezet is:

```
$ /usr/sbin/getsebool httpd_can_network_connect_db
httpd_can_network_connect_db --> on
```

   Dit staat Apache HTTP server scripts en modules toe om te verbinden met de datadase server.

4. Deze verandering is niet blijvend na een systeem herstart. Om de veranderingen blijvend te maken na het herstarten van het systeem, voer je het **`setsebool -P` *`boolean-naam`* `on`** commando uit als de Linux root gebruiker:

```
# /usr/sbin/setsebool -P httpd_can_network_connect_db on
```

5. Om tijdelijk terug te gaan naar het standaard gedrag, voer je als de Linux root gebruiker het **`setsebool httpd_can_network_connect_db off`** commando uit. Voor veranderingen die blijvend zijn na het herstarten, voer je het **`setsebool -P httpd_can_network_connect_db off`** commando uit.

5.6.3. Booleans voor NFS en CIFS

Standaard zijn NFS aankoppelingen op de client zijde gelabeld met een standaard context gedefinieerd door tactiek voor NFS bestandssystemen. In algemene tactieken gebruikt deze standaard context het `nfs_t` type. En ook zijn standaard Samba delingen aangekoppeld op de client zijde gelabeld met een standaard context gedefinieerd door tactiek. In algemene tactieken gebruikt deze standaard context het `cifs_t` type.

Afhankelijk van de tactiek instelling, kunnen services niet in staat zijn om bestanden gelabeld met de `nfs_t` of `cifs_t` types te lezen. Dit kan bestandssystemen die met deze types gelabeld zijn beletten om aangekoppeld te worden en daarna gelezen of geexporteerd te worden door andere services. Booleans kunnen aan of uit gezet worden om te bepalen welke services toestemming hebben om toegang te krijgen tot de `nfs_t` en`cifs_t` types.

De **`setsebool`** en **`semanage`** commando's moeten uitgevoerd worden als de Linux root gebruiker. Het **`setsebool -P`** commando maakt blijvende veranderingen. Gebruik de `-P` optie niet als je niet wilt dat veranderingen blijvend zijn na het opnieuw opstarten van het systeem:

Apache HTTP server

Om toegang toe te staan voor NFS bestandssystemen (bestanden gelabeld met het `nfs_t` type):

`/usr/sbin/setsebool -P httpd_use_nfs on`

Om toegang toe te staan tot Samba bestandsystemen (bestanden gelabeld met het `cifs_t` type):

`/usr/sbin/setsebool -P httpd_use_cifs on`

Samba

Om NFS bestandssystemen te exporteren:

`/usr/sbin/setsebool -P samba_share_nfs on`

FTP (vsftpd)

Om toegang toe te staan tot NFS bestandssystemen:

```
/usr/sbin/setsebool -P allow_ftpd_use_nfs on
```

Om toegang toe te staan tot Samba bestandssystemen:

```
/usr/sbin/setsebool -P allow_ftpd_use_cifs on
```

Andere services

Voor een lijst van aan NFS gerelateerde Booleans voor andere services:

```
/usr/sbin/semanage boolean -l | grep nfs
```

Voor een lijst van aan Samba gerelateerde Booleans voor andere services:

```
/usr/sbin/semanage boolean -l | grep cifs
```

Opmerking

Deze Booleans bestaan in SELinux tactiek zoals verstuurd met Fedora 11. Ze kunnen misschien niet bestaan in tactiek verstuurd met andere versies van Fedora of andere operating systemen.

5.7. SELinux context - Bestanden labelen

Op systemen die SELinux draaien, zijn alle processen en bestanden gelabeld met een label die informatie bevat die relevant is voor de beveiliging. Deze informatie wordt de SELinux context genoemd. Voor bestanden kan deze bekeken worden met het **`ls -Z`** commando:

```
$ ls -Z file1
-rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 file1
```

In dit voorbeeld biedt SELinux een gebruiker (`unconfined_u`), een rol (`object_r`), een type (`user_home_t`), en een niveau (`s0`). Deze informatie wordt gebruikt om toegangscontrole beslissingen te maken. Op DAC systemen wordt toegang gecontroleerd op basis van Linux gebruiker en groep ID's. SELinux tactiekregels worden toegepast na de DAC regels. SELinux tactiekregels worden niet gebruikt als DAC regels als eerste toegang weigeren.

Er zijn meerdere commando's voor het beheren van de SELinux context voor bestanden, zoals **`chcon`**, **`semanage fcontext`**, en **`restorecon`**.

5.7.1. Tijdelijke veranderingen: chcon

Het **chcon** commando verandert de SELinux context voor bestanden. Deze veranderingen overleven het herlabelen van een bestandssysteem niet, en ook het **`/sbin/restorecon`** commando niet. SELinux tactiek controleert welke gebruikers in staat zijn om de SELinux context voor elk bestand te veranderen. Als **chcon** gebruikt wordt, kunnen gebruikers alle of een deel van de SELinux context veranderen. Een foutief bestands type is een vaak voorkomende fout als SELinux toegang weigert.

Korte referentie

- Voer het **chcon -t *type bestandsnaam*** commando uit om het bestandstype te veranderen, waarin *type* een type is, zoals `httpd_sys_content_t`, en *bestandsnaam* een bestand of een map is.
- Voer het **chcon -R -t *type mapnaam*** commando uit om het type van een map en zijn inhoud te veranderen, waarin *type* een type is, zoals `httpd_sys_content_t`, en *mapnaam* een mapnaam is.

Het veranderen van het type van een bestand of map

Het volgende voorbeeld laat het veranderen van het type zien, alle andere attributen van de SELinux context blijven onveranderd:

1. Voer het **cd** commando uit zonder argumenten om naar je persoonlijke map te gaan.
2. Voer het **touch file1** commando uit om een nieuw bestand aan te maken. Gebruik het **ls -Z file1** commando om de SELinux context voor **file1** te zien:

   ```
   $ ls -Z file1
   -rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 file1
   ```

   In dit voorbeeld bevat de SELinux context voor **file1** de SELinux `unconfined_u` gebruiker, de `object_r` rol, het `user_home_t` type, en het `s0` niveau. Voor een beschrijving van ieder onderdeel van de SELinux context, refereer je naar *Hoofdstuk 3, SELinux context*.
3. Voer het **chcon -t samba_share_t file1** commando uit om het type te veranderen naar `samba_share_t`. De `-t` optie verandert alleen het type. Bekijk de verandering met **ls -Z file1**:

   ```
   $ ls -Z file1
   -rw-rw-r--  user1 group1 unconfined_u:object_r:samba_share_t:s0 file1
   ```
4. Gebruik het **/sbin/restorecon -v file1** commando om de SELinux context voor het **file1** bestand te herstellen. Gebruik de `-v` optie om te zien wat er verandert:

   ```
   $ /sbin/restorecon -v file1
   restorecon reset file1 context unconfined_u:object_r:samba_share_t:s0-
   >system_u:object_r:user_home_t:s0
   ```

   In dit voorbeeld wordt het vorige type, `samba_share_t`, hersteld naar het juiste `user_home_t` type. Als gerichte tactiek gebruikt wordt (de standaard SELinux tactiek in Fedora 11), leest het **/sbin/restorecon** commando de bestanden in de **/etc/selinux/targeted/contexts/files/** map om te zien welke SELinux context bestanden moeten hebben.

Het voorbeeld in deze paragraaf werkt hetzelfde voor mappen, bijvoorbeeld, als **file1** een map was.

Een map en zijn context types veranderen

Het volgende voorbeeld laat het aanmaken van een nieuwe map zien, en het veranderen van het bestandstype van de map (te samen met zijn inhoud) naar een type dat gebruikt wordt voor de

Apache HTTP server. De instelling in dit voorbeeld wordt gebruikt als je wilt dat de Apache HTTP server een ander document root gebruikt (in plaats van **`/var/www/html/`**):

1. Als de Linux root gebruiker voer je het **`mkdir /web`** commando uit om een nieuwe map aan te maken, en daarna het **`touch /web/file{1,2,3}`** commando om 3 lege bestanden aan te maken (**`file1`**, **`file2`**, en **`file3`**). De **`/web/`** map en zijn bestanden zijn gelabeld met het `default_t` type:

   ```
   # ls -dZ /web
   drwxr-xr-x  root root unconfined_u:object_r:default_t:s0 /web
   # ls -lZ /web
   -rw-r--r--  root root unconfined_u:object_r:default_t:s0 file1
   -rw-r--r--  root root unconfined_u:object_r:default_t:s0 file2
   -rw-r--r--  root root unconfined_u:object_r:default_t:s0 file3
   ```

2. Als de Linux root gebruiker voer je het **`chcon -R -t httpd_sys_content_t /web/`** commando uit om het type van de **`/web/`** map (en zijn inhoud) te veranderen naar `httpd_sys_content_t`:

   ```
   # chcon -R -t httpd_sys_content_t /web/
   # ls -dZ /web/
   drwxr-xr-x  root root unconfined_u:object_r:httpd_sys_content_t:s0 /web/
   # ls -lZ /web/
   -rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file1
   -rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file2
   -rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file3
   ```

3. Als de Linux root gebruiker voer je het **`/sbin/restorecon -R -v /web/`** commando uit om de standaard SELinux context te herstellen:

   ```
   # /sbin/restorecon -R -v /web/
   restorecon reset /web context
    unconfined_u:object_r:httpd_sys_content_t:s0-
   >system_u:object_r:default_t:s0
   restorecon reset /web/file2 context
    unconfined_u:object_r:httpd_sys_content_t:s0-
   >system_u:object_r:default_t:s0
   restorecon reset /web/file3 context
    unconfined_u:object_r:httpd_sys_content_t:s0-
   >system_u:object_r:default_t:s0
   restorecon reset /web/file1 context
    unconfined_u:object_r:httpd_sys_content_t:s0-
   >system_u:object_r:default_t:s0
   ```

Refereer naar de chcon(1) manual pagina voor verdere informatie over **`chcon`**.

Opmerking

Type Enforcement is de belangrijkste toestemmingscontrole gebruikt in SELinux gerichte tactiek. Voor het grootste deel kunnen SELinux gebruikers en rollen genegeerd worden.

5.7.2. Permanente veranderingen: semanage fcontext

Het **`/usr/sbin/semanage fcontext`** commando verandert de SELinux context voor bestanden. Als de gerichte tactiek gebruikt wordt, worden veranderingen die met dit commando gemaakt zijn toegevoegd aan het **`/etc/selinux/targeted/contexts/files/file_contexts`** bestand als de veranderingen gemaakt zijn voor bestanden die bestaan in **`file_contexts`**, of worden toegevoegd aan **`file_contexts.local`** voor nieuwe bestanden en mappen, zoals het aan maken van een **`/web/`** map. **`setfiles`**, welke gebruikt wordt als een bestandssysteem geherlabeld wordt, en **`/sbin/restorecon`**, welke de standaard SELinux context herstelt, lezen deze bestanden. Dit betekent dat veranderingen gemaakt door **`/usr/sbin/semanage fcontext`** blijvend zijn, zelfs als het bestandssysteem opnieuw gelabeld wordt. SELinux tactiek controleert of gebruikers in staat zijn de SELinux context van een bepaald bestand kunnen veranderen.

Korte referentie

Om SELinux context veranderingen te maken die het opnieuw labelen van een bestandssysteem overleven:

1. Voer het **`/usr/sbin/semanage fcontext -a`** ***`opties bestandsnaam|mapnaam`*** commando uit, waarbij je er aan moet denken om voor een bestand of map het volledige pad te gebruiken.

2. Voer het **`/sbin/restorecon -v`** ***`bestandsnaam|mapnaam`*** commando uit om de context veranderingen toe te passen.

Het veranderen van een bestandstype

Het volgende voorbeeld laat het veranderen van een bestandstype zien, waarbij geen andere attributen van de SELinux context veranderen:

1. Als de Linux root gebruiker, voer je het **`touch /etc/file1`** commando uit om een nieuw bestand aan te maken. Standaard worden nieuw aangemaakte bestanden in de **`/etc/`** gelabeld met het `etc_t` type:

```
# ls -Z /etc/file1
-rw-r--r--  root root unconfined_u:object_r:etc_t:s0       /etc/file1
```

2. Als de Linux root gebruiker voer je het **`/usr/sbin/semanage fcontext -a -t samba_share_t /etc/file1`** commando uit om het type van **`file1`** te veranderen naar `samba_share_t`. De `-a` optie voegt een nieuwe optekening toe, en de `-t` optie definieert een type (`samba_share_t`). Merk op: het uitvoeren van dit commando verandert niet direct het type - **`file1`** is nog steeds gelabeld met het `etc_t` type:

```
# /usr/sbin/semanage fcontext -a -t samba_share_t /etc/file1
# ls -Z /etc/file1
```

```
-rw-r--r--  root root unconfined_u:object_r:etc_t:s0       /etc/file1
```

Het **/usr/sbin/semanage fcontext -a -t samba_share_t /etc/file1** commando voegt de volgende regel toe aan **/etc/selinux/targeted/contexts/files/file_contexts.local**:

```
/etc/file1    unconfined_u:object_r:samba_share_t:s0
```

3. Als de Linux root gebruiker voer je het **/sbin/restorecon -v /etc/file1** commando uit om het type te veranderen. Omdat het **semanage** commando een regel aan **file.contexts.local** heeft toegevoegd voor **/etc/file1**, verandert het **/sbin/restorecon** commando het type naar `samba_share_t`:

```
# /sbin/restorecon -v /etc/file1
restorecon reset /etc/file1 context unconfined_u:object_r:etc_t:s0-
>system_u:object_r:samba_share_t:s0
```

4. Als de Linux root gebruiker voer je het **rm -i /etc/file1** commando uit om **file1** te verwijderen.

5. Als de Linux root gebruiker voer je het **/usr/sbin/semanage fcontext -d /etc/file1** commando uit om de context toegevoegd voor **/etc/file1** te verwijderen. Als de context verwijderd is, zal het uitvoeren van **restorecon** het type veranderen naar `etc_t`, in plaats van `samba_share_t`.

Het veranderen van het type van een map

Het volgende voorbeeld laat het aanmaken van een nieuwe map zien en het veranderen van het bestandstype van die map naar een type dat gebruikt wordt door de Apache HTTP server:

1. Als de Linux root gebruiker voer je het **mkdir /web** commando uit om een nieuwe map aan te maken. Deze map is gelabeld met het `default_t` type:

```
# ls -dZ /web
drwxr-xr-x  root root unconfined_u:object_r:default_t:s0 /web
```

De **ls** `-d` optie laat **ls** informatie tonen over een map, in plaats van zijn inhoud, en de `-Z` optie laat **ls** de SELinux context tonen (in dit voorbeeld, `unconfined_u:object_r:default_t:s0`).

2. Als de Linux root gebruiker voer je het **/usr/sbin/semanage fcontext -a -t httpd_sys_content_t /web** commando uit om het type van **/web/** te veranderen naar `httpd_sys_content_t`. De `-a` optie voegt een nieuwe aantekening toe, en de `-t` optie definieert een type (`httpd_sys_content_t`). Merk op: het uitvoeren van dit commando verandert niet direct het type - **/web/** is nog steeds gelabeld met het `default_t` type:

```
# /usr/sbin/semanage fcontext -a -t httpd_sys_content_t /web
# ls -dZ /web
drwxr-xr-x  root root unconfined_u:object_r:default_t:s0   /web
```

Het **/usr/sbin/semanage fcontext -a -t httpd_sys_content_t /web** commmando voegt de volgende regel toe aan **/etc/selinux/targeted/contexts/files/file_contexts.local**:

```
/web    unconfined_u:object_r:httpd_sys_content_t:s0
```

3. Als de Linux root gebruiker voer je het **/sbin/restorecon -v /web** commando uit om het type te veranderen. Omdat het **semanage** commando een regel aan **file.contexts.local** heeft toegevoegd voor **/web**, verandert het **/sbin/restorecon** commando het type naar `httpd_sys_content_t`:

```
# /sbin/restorecon -v /web
restorecon reset /web context unconfined_u:object_r:default_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
```

Standaard erven nieuw aangemaakte bestanden en mappen het SELinux type van hun ouders map. Als dit voorbeeld gebruikt wordt, en voordat de SELinx context toegevoegd voor **/web/** is verwijderd, zijn bestanden en mappen aangemaakt in de **/web/** map gelabeld met het `httpd_sys_content_t` type.

4. Als de Linux root gebruiker voer je het **/usr/sbin/semanage fcontext -d /web** commando uit om de toegevoegde context voor **/web/** te verwijderen.

5. Als de Linux root gebruiker voer je het **/sbin/restorecon -v /web** commando uit om de standaard SELinux context te herstellen.

Een map en zijn context types veranderen

Het volgende voorbeeld laat het aanmaken van een nieuwe map zien, en het veranderen van het bestandstype van de map (te samen met zijn inhoud) naar een type dat gebruikt wordt door de Apache HTTP server. De instelling in dit voorbeeld wordt gebruikt als je wilt dat de Apache HTTP server een andere document root gaat gebruiken (in plaats van **/var/www/html/**):

1. Als de Linux root gebruiker voer je het **mkdir /web** commando uit om een nieuwe map aan te maken, en daarna het **touch /web/file{1,2,3}** commando om 3 lege bestanden aan te maken (**file1**, **file2**, en **file3**). De **/web/** map en zijn bestanden zijn gelabeld met het `default_t` type:

```
# ls -dZ /web
drwxr-xr-x  root root unconfined_u:object_r:default_t:s0 /web
# ls -lZ /web
-rw-r--r--  root root unconfined_u:object_r:default_t:s0 file1
-rw-r--r--  root root unconfined_u:object_r:default_t:s0 file2
-rw-r--r--  root root unconfined_u:object_r:default_t:s0 file3
```

2. Als de Linux root gebruiker voer je het **/usr/sbin/semanage fcontext -a -t httpd_sys_content_t "/web(/.*)?"** commando uit om het type van de **/web/**, en de bestanden hierin, te veranderen naar `httpd_sys_content_t`. De `-a` optie voegt een nieuwe aantekening toe, en de `-t` optie definieert een type (httpd_sys_content_t). De `"/web(/.*)?"` reguliere expressie laat het **semanage** commando veranderingen maken naar de **/web/** map, en

naar de bestanden daar in. Merk op: het uitvoeren van dit commando verandert niet direct het type - **/web/** en de bestanden hierin zijn nog steeds gelabeld met het `default_t` type:

```
# ls -dZ /web
drwxr-xr-x  root root unconfined_u:object_r:default_t:s0 /web
# ls -lZ /web
-rw-r--r--  root root unconfined_u:object_r:default_t:s0 file1
-rw-r--r--  root root unconfined_u:object_r:default_t:s0 file2
-rw-r--r--  root root unconfined_u:object_r:default_t:s0 file3
```

Het **/usr/sbin/semanage fcontext -a -t httpd_sys_content_t "/web(/.*)?"** commando voegt de volgende regel toe aan **/etc/selinux/targeted/contexts/files/file_contexts.local**:

```
/web(/.*)?    system_u:object_r:httpd_sys_content_t:s0
```

3. Als de Linux root gebruiker voer je het **/sbin/restorecon -R -v /web** commando uit om het type van de **/web/** map, en alle bestanden hierin, te veranderen. De `-R` optie staat voor recursief, wat betekent dat alle bestanden en mappen onder de **/web/** map gelabeld worden met het `httpd_sys_content_t` type. Omdat het **semanage** commando een regel toevoegde aan **file.contexts.local** voor `/web(/.*)?`, verandert het **/sbin/restorecon** commando de types naar `httpd_sys_content_t`:

```
# /sbin/restorecon -R -v /web
restorecon reset /web context unconfined_u:object_r:default_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
restorecon reset /web/file2 context unconfined_u:object_r:default_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
restorecon reset /web/file3 context unconfined_u:object_r:default_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
restorecon reset /web/file1 context unconfined_u:object_r:default_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
```

Standaard erven nieuw aangemaakte bestanden en mappen het SELinux type van hun ouders. In dit voorbeeld, zullen bestanden en mappen aangemaakt in de **/web/** map gelabeld worden met het `httpd_sys_content_t` type.

4. Als de Linux root gebruiker voer je het **/usr/sbin/semanage fcontext -d "/web(/.*)?"** commando uit om de context toegevoegd voor `"/web(/.*)?"` te verwijderen.

5. Als de Linux root gebruiker voer je het **/sbin/restorecon -R -v /web** om de standaard SELinux context te herstellen.

Een toegevoegde context verwijderen

Het volgende voorbeeld laat het toevoegen en verwijderen van een SELinux context zien:

1. Als de Linux root gebruiker voer je het **/usr/sbin/semanage fcontext -a -t httpd_sys_content_t /test** commando uit. De **/test/** map hoeft niet te bestaan. Dit commando voegt de volgende context toe aan **/etc/selinux/targeted/contexts/files/file_contexts.local**:

```
/test    system_u:object_r:httpd_sys_content_t:s0
```

2. Om de context te verwijderen, voer je als de Linux root gebruiker het **`/usr/sbin/semanage fcontext -d bestandsnaam|mapnaam`** commando uit, waarin *`bestandsnaam|mapnaam`* het eerste onderdeel is in **`file_contexts.local`**. Het volgende is een voorbeeld van een context in **`file_contexts.local`**:

   ```
   /test    system_u:object_r:httpd_sys_content_t:s0
   ```

   Waarin het eerste onderdeel `/test` is. Om te voorkomen dat de **`/test/`** map gelabeld wordt met `httpd_sys_content_t` na het draaien van **`/sbin/restorecon`**, of na een bestandssysteem herlabeling, voer je het volgende commando uit als de Linux root gebruiker om de context van **`file_contexts.local`** te verwijderen:

   `/usr/sbin/semanage fcontext -d /test`

Als de context onderdeel is van een reguliere expressie, bijvoorbeeld, `/web(/.*)?`, gebruik je aanhalingstekens om de reguliere expressie:

`/usr/sbin/semanage fcontext -d "/web(/.*)?"`

Refereer naar de semanage(8) manual pagina voor verdere informatie over **`/usr/sbin/semanage`**.

Belangrijk

Als de SELinux context veranderd wordt met **`/usr/sbin/semanage fcontext -a`**, gebruik dan het volledige pad naar het bestand of de map om te voorkomen dat bestanden verkeerd worden gelabeld na een systeem herlabeling, of nadat het **`/sbin/restorecon`** commando is uitgevoerd.

5.8. De file_t en default_t types

Als op bestandsystemen die uitgebreide attributen ondersteunen een bestand op de schijf wordt benaderd die geen SELinux context heeft, dan wordt dit bestand behandeld alsof het een standaard context heeft zoals gedefinieerd door SELinux tactiek. In gewone tactieken gebruikt deze standaard context het `file_t` type. Dit moet het enigste gebruik zijn van dit type, zodat bestanden zonder context op een schijf onderscheiden kunnen worden in de tactiek, en in het algemeen onbereikbaar worden gehouden voor beperkte domeinen. Het `file_t` type moet niet bestaan op correct gelabelde bestandssystemen, omdat alle bestanden op een systeem dat SELinux draait en SELinux context moeten hebben, en het `file_t` type wordt nooit gebruikt in bestandscontext configuratie[8].

Het `default_t` type wordt gebruikt voor bestanden die niet passen bij elk ander patroon in de bestandscontext configuratie, zodat zulke bestanden onderscheiden kunnen worden van bestanden op schijf die geen context hebben, en ze worden in het algemeen onbereikbaar gehouden voor beperkte domeinen. Als je een nieuwe top-niveau map aanmaakt, zoals **`/mijnmap/`**, zal deze misschien gelabeld worden met het `default_t` type. Als services toegang nodig hebben naar zo'n map, vernieuw dan de bestandscontext voor deze loocatie. Refereer naar *Paragraaf 5.7.2,*

[8] Bestanden in **`/etc/selinux/targeted/contexts/files/`** definieren context voor bestanden en mappen. Bestanden in deze map worden gelezen door **`restorecon`** en **`setfiles`** om bestanden en mappen te herstellen naar hun standaard context.

"Permanente veranderingen: semanage fcontext" voor details over het toevoegen van een context aan de bestandscontext configuratie.

5.9. Het aankoppelen van bestandssystemen

Als een bestandssysteem dat uitgebreide attributen ondersteund wordt aangekoppeld, wordt standaard de beveiligingscontext voor ieder bestand verkregen van de *security.selinux* uitgebreide attribuut van het bestand. Bestanden in bestandssystemen die uitgebreide attributen niet ondersteunen krijgen een enkele, standaard beveiliging van de tactiek configuratie, gebaseerd op het bestandssysteem type.

Gebruik het **`mount -o context`** commando om bestaande uitgebreide attributen ter zijde te schuiven, of om een andere, standaard context op te geven voor bestandssystemen die geen uitgebreide attributen ondersteunen. Dit is nuttig als je niet vertrouwt dat een bestandssysteem de juiste attriburen levert, bijvoorbeeld, verwijderbare media gebruikt in meerdere systemen. Het **`mount -o context`** commando kan ook gebruikt worden om labeling te ondersteunen voor bestandssystemen die geen uitgebreide attributen ondersteunen, zoals de File Allocation Table (FAT) of NFS bestandssystemen. De context opgegeven met de `context` optie wordt niet naar schijf geschreven, de originele context blijft bewaard, en kan gezien worden door aan te koppelen zonder een `context` optie (als het bestandssysteem om te beginnen uitgebreide attributen heeft).

Voor meer informatie over bestandssysteem labeling, refereer je naar het artikel van James Morris over "Filesystem Labeling in SELinux": *http://www.linuxjournal.com/article/7426*.

5.9.1. Context aankoppelingen

Om een bestandssysteem met een gespecificeerde context aan te koppelen, waarbij voorbij gegaan wordt aan bestaande context als die er is, of om een andere, standaard context op te geven voor een bestandssysteem dat geen uitgebreide attributen ondersteund, gebruik je als de Linux root gebruiker het **`mount -o context=`*`SELinux_user:role:type:level`*** commando om het gewenste bestandssysteem aan te koppelen. Context veranderingen worden niet naar schijf geschreven. Standaard worden NFS aankoppelingen op de client zijde gelabeld met een standaard context gedefinieerd door tactiek voor NFS bestandssystemen. In algemene tactieken gebruikt deze standaard context het `nfs_t` type. Zonder extra aankoppel opties, kan dit het delen van NFS bestandssytemen via andere services beletten, zoals de Apache HTTP server. Het volgende voorbeeld koppelt een NFS bestandssysteem zodanig aan dat het gedeeld kan worden via de Apache HTTP server:

```
# mount server:/export /local/mount/point -o\
context="system_u:object_r:httpd_sys_content_t:s0"
```

Nieuw aangemaakte bestanden en mappen in dit bestandssysteem lijken de SELinux context te hebben opgegeven met de `-o context` optie; omdat echter de context veranderingen niet naar schijf geschreven worden in deze situaties, wordt de context opgegeven met de `context` optie alleen vastgehouden als de `context` optie wordr gebruikt bij de volgende aankoppeling, en als dezelfde context wordt opgegeven.

Type Enforcement is de belangrijkste rechten controle die gebruikt wordt in de SELinux gerichte tactiek. Voor het grootste deel kunnen SELinux gebruikers en rollen genegeerd worden, dus als je de SELinux context terzijde schuift met de `-o context` optie, gebruik dan de SELinux `system_u` gebruiker en de `object_r` rol, en concentreer je op het type. Als je de MLS tactiek of multi-categorie beveiliging niet gebruikt, gebruik je het `s0` niveau.

Opmerking

Als een bestandssysteem wordt aangekoppeld met een `context` optie, zijn context veranderingen (door gebruikers en processen) verboden. Bijvoorbeeld, het uitvoeren van **chcon** op een bestandssysteem aangekoppeld met een `context` optie resulteert in een `Operation not supported` fout.

5.9.2. De standaard context veranderen

Zoals vermeldt in *Paragraaf 5.8, "De file_t en default_t types"*, worden benaderde bestanden, die geen SELinux context op schijf hebben, in bestandssystemen die uitgebreide attributen ondersteunen, behandeld alsof het een standaard context heeft zoals gedefinieerd door SELinux tactiek. In algemene tactieken gebruikt deze standaard context het `file_t` type. Als het wenselijk is om een andere standaard context te gebruiken, koppel dan het bestandssysteem aan met de `defcontext` optie.

Het volgende voorbeeld koppelt een nieuw aangemaakt bestandssysteem (op **/dev/sda2**) aan naar de nieuw aangemaakte **/test/** map. Het veronderstelt dat er geen regels zijn in **/etc/selinux/targeted/contexts/files/** die een context definieren voor de **/test/** map:

```
# mount /dev/sda2 /test/ -o defcontext="system_u:object_r:samba_share_t:s0"
```

In dit voorbeeld:

- de `defcontext` optie definieert dat `system_u:object_r:samba_share_t:s0` "de standaard beveiligingscontext voor niet gelabelde bestanden"[9] is.
- als het aangekoppeld is, wordt de root map (**/test/**) van het bestandssysteem behandeld alsof het is gelabeld met de context opgegeven door `defcontext` (dit label wordt niet op schijf bewaard). Dit heeft gevolgen voor het labelen van bestanden die in **/test/** aangemaakt worden: nieuwe bestanden erven het `samba_share_t` type en deze labels worden op schijf bewaard.
- bestanden aangemaakt in **/test/** terwijl het bestandssysteem was aangekoppeld met een `defcontext` optie houden hun labels vast.

5.9.3. Het aankoppelen van een NFS bestandssysteem

Standaard worden NFS aankoppelingen op de client zijde gelabeld met een standaard context gedefinieerd door tactiek voor NFS bestandssystemen. In algemene tactieken gebruikt deze standaard context het `nfs_t` type. Alfhankelijk van de tactiek instelling, zullen services, zoals de Apache HTTP server en MySQL, misschien niet in staat zijn om bestanden te lezen die gelabeld zijn met het `nfs_t` type. Dit kan verhinderen dat bestandssystemen die met dit type gelabeld zijn aangekoppeld worden en daarna gelezen of geexporteerd door andere services.

Als je een NFS bestandssysteem wilt aankoppelen en dat bestandssysteem wilt lezen of exporteren met een andere service, gebruik je de `context` optie tijdens het aankoppelen om het `nfs_t` type terzijde te schuiven. Gebruik de volgende context optie om NFS bestandssystemen aan te koppelen zodat ze gedeeld kunnen worden via de Apache HTTP server:

```
mount server:/export /local/mount/point -o\
context="system_u:object_r:httpd_sys_content_t:s0"
```

Omdat voor deze situaties context veranderingen niet naar schijf worden geschreven, wordt de context opgegeven met de `context` optie alleen behouden als de `context` optie wordt gebruikt tijdens de volgende aankoppeling, en als dezelfde context wordt opgegeven.

Als een alternatief voor het aankoppelen van bestandssystemen met `context` opties, kunnen Booleans aangezet worden om services toe te staan om toegang te hebben tot bestandssystemen gelabeld met het `nfs_t` type. Refereer naar *Paragraaf 5.6.3, "Booleans voor NFS en CIFS"* voor instructies over het instellen van Booleans om services toegang te geven tot het `nfs_t` type.

5.9.4. Meerdere NFS aankoppelingen

Als hetzelfde NFS bestandssysteem meerdere keren aangekoppeld wordt, en je probeert bij iedere aankoppeling de SELinux context terzijde te schuiven met een andere context, heeft dat als resultaat dat opvolgende aankoppel commando's zullen mislukken. In het volgende voorbeeld heeft de NFS server een enkele export, **`/export`**, welke twee submappen heeft, **`web/`** en **`database/`**. De volgende commando's proberen twee aankoppelingen te maken van een enkele NFS export, en proberen iedere keer de context terzijde te schuiven:

```
# mount server:/export/web /local/web -o\
context="system_u:object_r:httpd_sys_content_t:s0"

# mount server:/export/database /local/database -o\
context="system_u:object_r:mysqld_db_t:s0"
```

Het tweede aankoppel commando mislukt, en de volgende boodschap wordt naar **`/var/log/messages`** geschreven:

```
kernel: SELinux: mount invalid.  Same superblock, different security
 settings for (dev 0:15, type nfs)
```

Om meerdere aankoppelingen te maken naar een enkele NFS export, waarbij iedere aankoppeling een andere context heeft, gebruik je de `-o nosharecache,context` opties. Het volgende voorbeeld koppelt meerdere aankoppelingen voor een enkele NFS export, met een andere context voor iedere aankoppeling (wat een enkele service toestaat om toegang tot iedere te hebben):

```
# mount server:/export/web /local/web -o\
nosharecache,context="system_u:object_r:httpd_sys_content_t:s0"

# mount server:/export/database /local/database -o\
nosharecache,context="system_u:object_r:mysqld_db_t:s0"
```

In dit voorbeeld wordt `server:/export/web` locaal aangekoppeld naar **`/local/web/`**, waarbij alle bestanden gelabeld worden met het `httpd_sys_content_t` type, wat Apache HTTP server toegang toestaat. `server:/export/database` wordt locaal aangekoppeld naar **`/local/database`**, waarbij alle bestanden gelabeld worden met het `mysqld_db_t` type, wat MySQL toegang toestaat. Deze veranderingen worden niet naar schijf geschreven.

Belangrijk

De `nosharecache` opties laten je dezelfde submap van een export meerdere keren aankoppelen met verschillende contexten (bijvoorbeeld, het meerdere keren aankoppelen

van **/export/web**). Koppel dezelfde submap van een export niet meerdere keren aan met verschillende contexten, omdat dit een overlappende aankoppeling maakt, waarbij bestanden bereikbaar zijn met twee verschillende contexten.

5.9.5. Maak de context aankoppelingen blijvend

Om de context aankoppelingen blijven te maken voor opnieuw aankoppelen en opnieuw opstarten, voeg je regels toe voor de bestandssystemen in **/etc/fstab** of een automounter map, en gebruik de gewenste context als een aankoppel optie. Het volgende voorbeeld voegt een regel toe aan **/etc/fstab** voor een NFS context aankoppeling:

```
server:/export /local/mount/ nfs
 context="system_u:object_r:httpd_sys_content_t:s0" 0 0
```

Refereer naar *Red Hat Enterprise Linux 5 Deployment Guide, Section 19.2. "NFS Client Configuration"*[10] voor informatie over het aankoppelen van NFS bestandssystemen.

5.10. Het onderhouden van SELinux labels

Deze paragrafen behandelen wat er gebeurt met SELinux context tijdens het kopiëren, verplaatsen, en archiveren van bestanden en mappen. Ze leggen ook uit hoe je de context kunt behouden tijdens het kopiëren en archiveren.

5.10.1. Bestanden en mappen kopiëren

Als een bestand of map wordt gekopieerd, wordt een nieuw bestand of map aangemaakt als het nog niet bestaat. De context van dat nieuwe bestand of map is gebaseerd op standaard labelingregels, en niet de context van het originele bestand of map (behalve als er opties zijn gebruikt om de originele context te behouden). Bijvoorbeeld, bestanden aangemaakt in de persoonlijke mappen van de gebruiker worden gelabeld met het `user_home_t` type:

```
$ touch file1
$ ls -Z file1
-rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 file1
```

Als zo'n bestand gekopieerd wordt naar een andere map, zoals **/etc/**, wordt het nieuwe bestand aangemaakt overeenkomstig de standaard labelingsregels voor de **/etc/** map. Een bestand kopiëren (zonder extra opties) hoeft de originele context niet te behouden:

```
$ ls -Z file1
-rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 file1
# cp file1 /etc/
$ ls -Z /etc/file1
-rw-r--r--  root root unconfined_u:object_r:etc_t:s0   /etc/file1
```

Als **file1** gekopieerd wordt naar **/etc/**, en als **/etc/file1** niet bestaat, wordt **/etc/file1** aangemaakt als een nieuw bestand. Zoals in het voorbeeld hierboven is getoond, wordt **/etc/file1** gelabeld met het `etc_t` type, overeenkomstig de standaard labelingsregels.

[10] http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5.2/html/Deployment_Guide/s1-nfs-client-config.html

Als een bestand wordt gekopieerd over een bestaand bestand, wordt de context van het bestaande bestand gehandhaafd, behalve als de gebruiker **cp** opties op heeft gegeven om de context van het originele bestand te behouden, zoals `--preserve=context`. SELinux tactiek kan beletten dat contexten behouden blijven tijdens het kopiëren.

Kopiëren zonder behoud van SELinux context

Als een bestand gekopieerd wordt met het **cp** commando, en er worden geen opties meegegeven, wordt het type geerfd van de doel, ouders map:

```
$ touch file1
$ ls -Z file1
-rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 file1
$ ls -dZ /var/www/html/
drwxr-xr-x  root root system_u:object_r:httpd_sys_content_t:s0 /var/www/
html/
# cp file1 /var/www/html/
$ ls -Z /var/www/html/file1
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/
www/html/file1
```

In dit voorbeeld, wordt **file1** aangemaakt in de persoonlijke map van de gebruiker, en wordt gelabeld met het `user_home_t` type. De **/var/www/html/** map is gelabeld met het `httpd_sys_content_t` type, zoals getoond met het **ls -dZ /var/www/html/** commando. Als **file1** wordt gekopieerd naar **/var/www/html/**, erft het het `httpd_sys_content_t` type, zoals het **ls -Z /var/www/html/file1** commando laat zien.

De SELinux context behouden tijdens het kopiëren

Gebruik het **cp --preserve=context** commando om de context te behouden tijdens het kopiëren:

```
$ touch file1
$ ls -Z file1
-rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 file1
$ ls -dZ /var/www/html/
drwxr-xr-x  root root system_u:object_r:httpd_sys_content_t:s0 /var/www/
html/
# cp --preserve=context file1 /var/www/html/
$ ls -Z /var/www/html/file1
-rw-r--r--  root root unconfined_u:object_r:user_home_t:s0 /var/www/html/
file1
```

In dit voorbeeld, wordt **file1** aangemaakt in de persoonlijke map van de gebruiker, en wordt gelabeld met het `user_home_t` type. De **/var/www/html/** map is gelabeld met het `httpd_sys_content_t` type, zoals het **ls -dZ /var/www/html/** coomado laat zien. Het gebruik van de `--preserve=context` optie behoudt de SELinux context tijdens het kopiëren. Zoals het **ls -Z /var/www/html/file1** commando laat zien is het `user_home_t` type van **file1** behouden toen het bestand gekopieerd werd naar **/var/www/html/**.

Kopiëren en de context veranderen

Gebruik het **`cp -Z`** commando om de context van de doel kopie te veranderen. Het volgende voorbeeld is uitgevoerd in de persoonlijke map van de gebruiker:

```
$ touch file1
$ cp -Z system_u:object_r:samba_share_t:s0 file1 file2
$ ls -Z file1 file2
-rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 file1
-rw-rw-r--  user1 group1 system_u:object_r:samba_share_t:s0 file2
$ rm file1 file2
```

In dit voorbeeld, wordt de context gedefinieerd met de `-Z` optie. Zonder de `-Z` optie, zou **`file2`** gelabeld zijn met de `unconfined_u:object_r:user_home_t` context.

Kopiëren over een bestaand bestand

Als een bestand gekopieerd wordt over een bestaand bestand, wordt de context van het bestaande bestand behouden (behalve als een optie gebruikt wordt om de context te behouden). Bijvoorbeeld:

```
# touch /etc/file1
# ls -Z /etc/file1
-rw-r--r--  root root unconfined_u:object_r:etc_t:s0   /etc/file1
# touch /tmp/file2
# ls -Z /tmp/file2
-rw-r--r--  root root unconfined_u:object_r:user_tmp_t:s0 /tmp/file2
# cp /tmp/file2 /etc/file1
# ls -Z /etc/file1
-rw-r--r--  root root unconfined_u:object_r:etc_t:s0   /etc/file1
```

In dit voorbeeld worden twee bestanden aangemaakt: **`/etc/file1`**, gelabeld met het `etc_t` type, en **`/tmp/file2`**, gelabeld met het `user_tmp_t` type. Het **`cp /tmp/file2 /etc/file1`** commando overschrijft **`file1`** met **`file2`**. Na het kopiëren, laat het **`ls -Z /etc/file1`** zien dat **`file1`** gelabeld is met het `etc_t` type, niet het `user_tmp_t` type van **`/tmp/file2`** welke **`/etc/file1`** heeft vervangen.

Belangrijk

Kopieer bestanden en mappen, inplaats van ze te verplaatsen. Dit helpt om er zeker van te zijn dat ze gelabeld zijn met de juiste SELinux context. Foutieve SELinux contexten kunnen voorkomen dat processen toegang hebben tot zulke bestanden en mappen.

5.10.2. Bestanden en mappen verplaatsen

Bestanden en mappen behouden hun huidige SELinux context als ze verplaats worden. In veel gevallen is dit niet juist voor de locatie waarheen ze verplaatst zijn. Het volgende voorbeeld laat het verplaatsen van een bestand van de persoonlijke map van een gebruiker zien naar **`/var/www/html/`**, welke gebruikt wordt door de Apache HTTP server. Omdat het bestand is verplaatst, erft het niet de juiste SELinux context:

1. Voer het **cd** commando uit zonder argumenten om naar je persoonlijke map te gaan. Als je daar bent, voer je het **touch file1** commando uit om een bestand aan te maken. Dit bestand is gelabeld met het user_home_t type:

```
$ ls -Z file1
-rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 file1
```

2. Voer het **ls -dZ /var/www/html/** commando uit om de SELinux context van de **/var/www/html/** map te zien:

```
$ ls -dZ /var/www/html/
drwxr-xr-x  root root system_u:object_r:httpd_sys_content_t:s0 /var/www/
html/
```

   Standaard wordt de **/var/www/html/** map gelabeld met het httpd_sys_content_t type. Bestanden en mappen aangemaakt in de **/var/www/html/** map erven dit type, en dus zijn ze ermee gelabeld.

3. Als de Linux root gebruiker voer je het **mv file1 /var/www/html/** commando uit om **file1** te verplaatsen naar de **/var/www/html/** map. Omdat het bestand is verplaatst behoudt het zijn huidige user_home_t type:

```
# mv file1 /var/www/html/
# ls -Z /var/www/html/file1
-rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 /var/www/
html/file1
```

Standaard kan de Apache HTTP server geen bestanden lezen die gelabeld zijn met het user_home_t type. Als alle bestanden die een web pagina vormen gelabeld zijn met het user_home_t type, of een ander type dat de Apache HTTP server niet kan lezen, wordt toegang geweigerd als je probeert ze te benaderen met Firefox of op tekst gebaseerde webbrowsers.

Belangrijk

Het verplaatsen van bestanden en mappen met het **mv** commando kan een verkeerde SELinux context tot gevolg hebben, en processen, zoals de Apache HTTP server en Samba, beletten om toegang te krijgen tot die bestanden en mappen.

5.10.3. Het controleren van de standaard SELinux context

Gebruik het **/usr/sbin/matchpathcon** commando om te controleren of bestanden en mappen de juiste SELinux context hebben. Van de matchpathcon(8) manual pagina: "**matchpathcon** ondervraagt de systeem tactiek en levert de standaard beveiligings context behorend bij het bestandspad."[11]. Het volgende voorbeeld laat het gebruik van het **/usr/sbin/matchpathcon** commando zien om te bevestigen dat bestanden in de **/var/www/html/** map correct gelabeld zijn:

[11] De matchpathcon(8) manual pagina, zoals verstuurd met het *libselinux-utils* pakket Fedora, is geschreven door Daniel Walsh. Alle bewerkingen of veranderingen in deze versie zijn gemaakt door Murray McAllister.

1. Als de Linux root gebruiker, voer je het **`touch /var/www/html/file{1,2,3}`** commando uit om drie bestanden (**`file1`**, **`file2`**, en **`file3`**) aan te maken. Deze bestanden erven het `httpd_sys_content_t` type van de **`/var/www/html/`** map:

```
# touch /var/www/html/file{1,2,3}
# ls -Z /var/www/html/
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file1
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file2
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file3
```

2. Als de Linux root gebruiker, voer je het **`chcon -t samba_share_t /var/www/html/file1`** commando uit om het type van **`file1`** te veranderen naar `samba_share_t`. Merk op: de Apache HTTP server kan geen bestanden of mappen lezen die gelabeld zijn met het `samba_share_t` type.

3. Het **`/usr/sbin/matchpathcon`** `-V` optie vergelijkt de huidige SELinux contect met de juiste, standaard context in SELinux tactiek. Voer het **`/usr/sbin/matchpathcon -V /var/www/html/*`** commando uit om alle bestanden in de **`/var/www/html/`** map te controleren:

```
$ /usr/sbin/matchpathcon -V /var/www/html/*
/var/www/html/file1 has context unconfined_u:object_r:samba_share_t:s0,
 should be system_u:object_r:httpd_sys_content_t:s0
/var/www/html/file2 verified.
/var/www/html/file3 verified.
```

De volgende output van het **`/usr/sbin/matchpathcon`** commando legt uit dat **`file1`** is gelabeld met het `samba_share_t` type, maar het zou gelabeld moeten zijn met het `httpd_sys_content_t` type:

```
/var/www/html/file1 has context unconfined_u:object_r:samba_share_t:s0,
 should be system_u:object_r:httpd_sys_content_t:s0
```

Om het label probleem op te lossen, en de Apache HTTP server toegang te geven tot **`file1`**, voer je als de Linux root gebruiker het **`/sbin/restorecon -v /var/www/html/file1`** commando uit:

```
# /sbin/restorecon -v /var/www/html/file1
restorecon reset /var/www/html/file1 context
 unconfined_u:object_r:samba_share_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
```

5.10.4. Bestanden archiveren met tar

`tar` behoudt standaard geen uitgebreide attributen. Omdat SELinux context bewaard wordt in uitgebreide attributen, kan de context verloren raken als bestanden gearchiveerd worden. Gebruik **`tar --selinux`** om archieven te maken die de context behouden. Als een Tar archief bestanden zonder uitgebreide attributen bevat, of je wilt dat de uitgebreide attributen overeenkomen met de systeemstandaard, voer het archiveren dan uit met **`/sbin/restorecon`**:

```
$ tar -xvf archive.tar | /sbin/restorecon -f -
```

Merk op: afhankelijk van de map kan het nodig zijn dat je de Linux root gebruiker bent om het **/sbin/restorecon** commando uit te voeren.

Het volgende voorbeeld laat het maken van een Tar archief zien dat de SELinux context behoudt:

1. Als de Linux root gebruiker, voer je het **touch /var/www/html/file{1,2,3}** commando uit om drie bestanden (**file1**, **file2**, en **file3**) aan te maken. Deze bestanden erven het `httpd_sys_content_t` type van de **/var/www/html/** map:

```
# touch /var/www/html/file{1,2,3}
# ls -Z /var/www/html/
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file1
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file2
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file3
```

2. Voer het **cd /var/www/html/** commando uit om naar de **/var/www/html/** map te gaan. Als je in die map bent, voer je als de Linux root gebruiker het **tar --selinux -cf test.tar file{1,2,3}** uit om een Tar archief met de naam **test.tar** te maken.

3. Als de Linux root gebruiker voer je het **mkdir /test** commando uit om een nieuwe map aan te maken, en daarna voer je het **chmod 777 /test/** commando uit om alle gebruikers volledige toegang tot de **/test/** map te geven.

4. Voer het **cp /var/www/html/test.tar /test/** commando uit om het **test.tar** bestand naar de **/test/** map te kopiëren.

5. Voer het **cd /test/** comando uit om naar de **/test/** map te gaan. Als je in die map bent, voer je het **tar -xvf test.tar** commando uit om het Tar archief uit te pakken.

6. Voer het **ls -lZ /test/** commando uit om de SELinux context te bekijken. Het `httpd_sys_content_t` type is behouden gebleven, in plaats van veranderd te zijn naar `default_t`, wat gebeurt zou zijn als de `--selinux` optie niet gebruikt was:

```
$ ls -lZ /test/
-rw-r--r--  user1 group1 unconfined_u:object_r:httpd_sys_content_t:s0
 file1
-rw-r--r--  user1 group1 unconfined_u:object_r:httpd_sys_content_t:s0
 file2
-rw-r--r--  user1 group1 unconfined_u:object_r:httpd_sys_content_t:s0
 file3
-rw-r--r--  user1 group1 unconfined_u:object_r:default_t:s0 test.tar
```

7. Als de **/test/** map niet langer nodig is, voer je als de Linux root gebruiker het **rm -ri /test/** commando uit om het te verwijderen te samen met de bestanden er in.

Refereer naar de tar(1) manual pagina voor meer informatie over **tar**, zoals de `--xattrs` optie die alle uitgebreide attributen behoudt.

5.10.5. Bestanden archiveren met star

star behoudt standaard geen uitgebreide attributen. Omdat SELinux context bewaard wordt in uitgebreide attributen, kan de context verloren gaan als de bestanden gearchiveerd worden. Gebruik

star -xattr -H=exustar om archieven te maken die context behouden. Het *star* pakket is standaard niet geinstalleerd. Om **star** te installeren, voer je als de Linux root gebruiker het **yum install star** commando uit.

Het volgende voorbeeld laat het maken van een Star archief zien dat de SELinux context behoudt:

1. Als de Linux root gebruiker, voer je het **touch /var/www/html/file{1,2,3}** commando uit om drie bestanden (**file1**, **file2**, en **file3**) aan te maken. Deze bestanden erven het `httpd_sys_content_t` type van de **/var/www/html/** map:

```
# touch /var/www/html/file{1,2,3}
# ls -Z /var/www/html/
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file1
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file2
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file3
```

2. Voer het **cd /var/www/html/** commando uit om naar de **/var/www/html/** map te gaan. Als je in die map bent, voer je als de Linux root gebruiker het **star -xattr -H=exustar -c -f=test.star file{1,2,3}** commando uit om een Star archief met de naam **test.star** te maken:

```
# star -xattr -H=exustar -c -f=test.star file{1,2,3}
star: 1 blocks + 0 bytes (total of 10240 bytes = 10.00k).
```

3. Als de Linux root gebruiker voer je het **mkdir /test** commando uit om een nieuwe map aan te maken, en daarna voer je het **chmod 777 /test/** commando uit om alle gebruikers volledige toegang tot de **/test/** map te geven.

4. Voer het **cp /var/www/html/test.star /test/** commando uit om het **test.star** bestand te kopiëren naar de **/test/** map.

5. Voer het **cd /test/** commando uit om naar de **/test/** map te gaan. Als je in die map bent, voer je het **star -x -f=test.star** commando uit om het Star archief uit te pakken:

```
$ star -x -f=test.star
star: 1 blocks + 0 bytes (total of 10240 bytes = 10.00k).
```

6. Voer het **ls -lZ /test/** commando uit om de SELinux context te bekijken. Het `httpd_sys_content_t` type is behouden gebleven, in plaats van veranderd te zijn naar `default_t`, wat gebeurt zou zijn als de `--selinux` optie niet gebruikt was:

```
$ ls -lZ /test/
-rw-r--r--  user1 group1 unconfined_u:object_r:httpd_sys_content_t:s0
 file1
-rw-r--r--  user1 group1 unconfined_u:object_r:httpd_sys_content_t:s0
 file2
-rw-r--r--  user1 group1 unconfined_u:object_r:httpd_sys_content_t:s0
 file3
-rw-r--r--  user1 group1 unconfined_u:object_r:default_t:s0 test.star
```

7. Als de **/test/** map niet langer nodig is, voer je als de Linux root gebruiker het **rm -ri /test/** commando uit om het te verwijderen te samen met de bestanden er in.

8. Als **star** niet langer nodig is, voer je als de Linux root gebruiker het **yum remove star** commando uit om het pakket te verwijderen.

Refereer naar de star(1) manual pagina voor meer informatie over **star**.

Gebruikers beperken

Een aantal beperkte SELinux gebruikers zijn beschikbaar in Fedora 11. Elke Linux gebruiker wordt afgebeeld op een SELinux gebruiker met SELinux tactiek, wat Linux gebruikers de beperkingen van SELinux gebruikers laat erven, bijvoorbeeld (afhankelijk van de gebruiker), het niet in staat zijn om: het X Window systeem te draaien, het netwerk te gebruiken, setuid toepassingen draaien (tenzij SELinux tactiek dit toestaat), of de **su** en **sudo** commando's uit te voeren om de Linux root gebruiker te worden. Dit helpt om het systeem te beschermen tegen de gebruiker. Refereer naar *Paragraaf 4.3, "Beperkte en onbeperkte gebruikers"* voor meer informatie over beperkte gebruikers in Fedora 11.

6.1. Linix en SELinux gebruiker afbeelding

Als de Linux root gebruiker, voer je het **`semanage login -l`** commando uit om de afbeelding van Linux gebruikers op SELinux gebruikers te zien:

```
# /usr/sbin/semanage login -l

Login Name                SELinux User              MLS/MCS Range

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

In Fedora 11 worden Linux gebruikers standaard afgebeeld op de SELinux `__default__` (welke is afgebeeld op de SELinux `unconfined_u` gebruiker). Als een Linux gebruiker wordt aangemaakt met het **`useradd`** commando, worden ze afgebeeld op de SELinux `unconfined_u` gebruiker indien er geen opties zijn opgegeven. Het volgende definieert de standaard afbeelding:

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

6.2. Nieuwe Linux gebruikers beperken: useradd

Linux gebruikers die afgebeeld zijn op de SELinux `unconfined_u` gebruiker draaien in het `unconfined_t` domein. Dit kan getoond worden door het uitvoeren van het **`id -Z`** commando terwijl je ingelogd bent als een Linux gebruiker afgebeeld op `unconfined_u`:

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

Als Linux gebruikers in het `unconfined_t` domein draaien, worden SELinux tactiekregels toegepast, maar er bestaan tactiekregels die Linux gebruikers die in het `unconfined_t` domein draaien bijna alle toegang toestaan. Als onbeperkte Linux gebruikers een toepassing uitvoeren waarvan de SELinux tactiek definieert dat het kan overgaan van het `unconfined_t` domein naar zijn eigen beperkte domein, zijn onbeperkte Linux gebruikers nog steeds onderworpen aan de beperkingen van het beperkte domein. Het beveiligings voordeel hiervan is dat, zelfs als een Linux gebruiker onbeperkt draait, de toepassing beperkt blijft, en dat daardoor de uitbuiting van een fout in de toepassing beperkt kan worden door tactiek. Merk op: dit beschermt het systeem niet tegen de gebruiker. In plaats daarvan worden de gebruiker en het systeem beschermd tegen mogelijke schade veroorzaakt door een fout in de toepassing.

Als Linux gebruikers aangemaakt worden met **useradd**, gebruik je de `-Z` optie om op te geven op welke SELinux gebruiker ze afgebeeld worden. Het volgende voorbeeld maakt een nieuwe Linux gebruiker aan, useruuser, en beeldt die gebruiker af op de SELinux `user_u` gebruiker. Linux gebruikers afgebeeld op de SELinux `user_u` gebruiker draaien in het `user_t` domein. In dit domein zijn Linux gebruikers niet in staat om setuid toepassingen te draaien tenzij SELinux tactiek dit toestaat (zoals **passwd**), en kunnen ze **su** of **sudo** niet uitvoeren, wat hen tegenhoudt om met deze commando's de Linux root gebruiker te worden.

1. Als de Linux root gebruiker voer je het **/usr/sbin/useradd -Z user_u useruuser** commando uit om een nieuwe Linux gebruiker (useruuser) aan te maken die afgebeeld wordt op de SELinux `user_u` gebruiker.

2. Als de Linux root gebruiker voer je het **semanage login -l** commando uit om de afbeelding van de Linux `useruuser` gebruiker op `user_u` te zien:

```
# /usr/sbin/semanage login -l

Login Name                SELinux User              MLS/MCS Range

__default__               unconfined_u              s0-s0:c0.c1023
root                      unconfined_u              s0-s0:c0.c1023
system_u                  system_u                  s0-s0:c0.c1023
useruuser                 user_u                    s0
```

3. Als de Linix root gebruiker voer je het **passwd useruuser** commando uit om een wachtwoord toe te kennen aan de Linux useruuser gebruiker:

```
# passwd useruuser
Changing password for user useruuser.
New UNIX password: Enter a password
Retype new UNIX password: Enter the same password again
passwd: all authentication tokens updated successfully.
```

4. Log uit van de huidige sessie, en log in als de Linux useruuser gebruiker. Als je inlogt, beeldt pam_selinux de Linux gebruiker af op een SELinux gebruiker (in dit geval, `user_u`), en stelt de bijbehorende SELinux context in. De shell van de Linux gebruiker wordt dan opgestart met deze context. Voer het **id -Z** commando uit om de context van een Linux gebruiker te bekijken:

```
[useruuser@localhost ~]$ id -Z
user_u:user_r:user_t:s0
```

5. Log uit van de sessie van de Linux useruuuser, en log weer in met je account. Als je de Linux useruuser niet wilt houden, voer je het **/usr/sbin/userdel -r useruuser** commando uit als de Linux root gebruiker om het te verwijderen te samen met zijn persoonlijke map.

6.3. Bestaande Linux gebruikers beperken: semanage login

Als een Linux gebruiker is afgebeeld op de SELinux `unconfined_u` gebruiker (het standaard gedrag), en je wilt de SELinux gebruiker waarop ze afgebeeld zijn veranderen, gebruik je het

semanage login commando. Het volgende voorbeeld maakt een nieuwe Linux gebruiker aan met de naam newuser, en beeldt dan die Linux gebruiker af op de SELinux `user_u` gebruiker:

1. Als de Linux root gebruiker voer je het **/usr/sbin/useradd newuser** commando uit om een nieuwe Linux gebruiker (newuser) aan te maken. Omdat deze gebruiker de standaard afbeelding gebruikt, verschijnt deze niet in de de **/usr/sbin/semanage login -l** output:

   ```
   # /usr/sbin/semanage login -l

   Login Name                SELinux User              MLS/MCS Range

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

2. Om de Linux newuser gebruiker af te beelden op de SELinux `user_u` gebruiker, voer je het volgende commando uit als de Linux root gebruiker:

   /usr/sbin/semanage login -a -s user_u newuser

   De `-a` optie voegt een nieuwe aantekening toe, en de `-s` optie specificeert de SELinux gebruiker waarop de Linux gebruiker afgebeeld wordt. Het laatse argument, `newuser`, is de Linux gebruiker die je wilt afbeelden op de opgegeven SELinux gebruiker.

3. Om de afbeelding van de Linux newuser gebruiker op `user_u` te bekijken, voer je het **semanage login -l** commando uit als de Linux root gebruiker:

   ```
   # /usr/sbin/semanage login -l

   Login Name                SELinux User              MLS/MCS Range

   __default__               unconfined_u              s0-s0:c0.c1023
   newuser                   user_u                    s0
   root                      unconfined_u              s0-s0:c0.c1023
   system_u                  system_u                  s0-s0:c0.c1023
   ```

4. As the Linux root user, run the **passwd newuser** command to assign a password to the Linux newuser user:

   ```
   # passwd newuser
   Changing password for user newuser.
   New UNIX password: Enter a password
   Retype new UNIX password: Enter the same password again
   passwd: all authentication tokens updated successfully.
   ```

5. Log uit van je huidige sessie, en log in als de Linux newuser gebruiker. Voer het **id -Z** commando uit om de SELinux context van newuser te bekijken:

   ```
   [newuser@rlocalhost ~]$ id -Z
   user_u:user_r:user_t:s0
   ```

6. Log uit van de sessie van de Linux newuser, en log weer in met je account. Als je de Linux newuser gebruiker niet wilt behouden, voer je het **`userdel -r newuser`** commando uit als de Linux root gebruiker om het te verwijderen, te samen met zijn persoonlijke map. Ook de afbeelding van de Linux newuser gebruiker op `user_u` wordt verwijderd:

```
# /usr/sbin/userdel -r newuser
# /usr/sbin/semanage login -l

Login Name                SELinux User              MLS/MCS Range

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

6.4. De standaard afbeelbing veranderen

Standaard worden in Fedora 11 Linux gebruikers afgebeeld op de SELinux `__default__` login (welke afgebeeld wordt op de SELinux `unconfined_u` gebruiker). Als je nieuwe Linux gebruikers, en Linux gebruikers die niet specifiek afgebeeld zijn op een SELinux gebruiker, standaard wilt instellen om beperkt te zijn, verander je de standaard afbeelding met het **`semanage login`** commando.

Bijvoorbeeld, voer het volgende commando uit als de Linux root gebruiker om de standaard afbeelding te veranderen van `unconfined_u` naar `user_u`:

`/usr/sbin/semanage login -m -S targeted -s "user_u" -r s0 __default__`

Voer het **`semanage login -l`** commando uit als de Linux root gebruiker om te bevestigen dat de `__default__` login is afgebeeld op `user_u`:

```
# /usr/sbin/semanage login -l

Login Name                SELinux User              MLS/MCS Range

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

Als een nieuwe Linux gebruiker wordt aangemaakt en er wordt geen SELinux gebruiker opgegeven, of als een bestaande Linux gebruiker inlogt en niet overeenkomt met een specifieke regel in de **`semanage login -l`** output, worden ze afgebeeld op `user_u`, via de `__default__` login.

Om weer terug te gaan naar het standaard gedrag, voer je het volgende commando uit als de Linux root gebruiker om de `__default__` login af te beelden op de SELinux `unconfined_u` gebruiker:

```
/usr/sbin/semanage login -m -S targeted -s "unconfined_u" -r\
s0-s0:c0.c1023 __default__
```

6.5. xguest: kiosk modus

Het *xguest* pakket biedt een kiosk gebruikers accont aan. Dit account wordt gebruikt om machines te beveiligen waar mensen naar toe gaan en gebruiken, zoals in bibliotheken, banken, vliegvelden,

informatie kiosken, en koffie shops. Het kiosk gebruikers account is zwaar vergrendeld: in wezen staat het gebruikers alleen toe om in te loggen en **Firefox** te gebruiken om Internet websites te bekijken. Elke verandering die gemaakt wordt onder dit account, zoals bestanden aanmaken of instellingen veranderen, gaan verloren zodra je uitlogt.

Het kiosk account instellen:

1. Als de Linux root gebruiker voer je het **`yum install xguest`** commando uit om het *xguest* pakket te installeren. Installeer ook de benodigde afhankelijkheden.

2. Om toe te staan dat het kiosk account gebruikt wordt door verschillende mensen, wordt het account niet beschermd met een wachtwoord, en daarom kan het account alleen maar beschermd worden als SELinux in de afdwingende modus draait. Voordat je inlogt met dit account, gebruik je het **`getenforce`** commando om er zeker van te zijn dat SELinux in de afdwingende modus draait:

   ```
   $ /usr/sbin/getenforce
   Enforcing
   ```

   Als dat niet het geval is, refereer je naar *Paragraaf 5.5, "SELinux modes"* voor informatie over het veranderen van de modus naar afdwingend. Het is niet mogelijk om met dit account in te loggen als SELinux in de toelatende modus is of uitstaat.

3. Je kunt alleen inloggen met dit account in de GNOME Display Manager (GDM). Zodra het *xguest* pakket geinstalleerd is, wordt een `Guest` account toegevoegd aan GDM. Om in te loggen klik je op het `Guest` account:

6.6. Booleans voor gebruikers die toepassingen uitvoeren

Het niet toestaan aan Linux gebruikers om toepassingen (welke de rechten van de gebruiker erven) uit te voeren in hun persoonlijke mappen en **`/tmp/`**, waarnaar ze schrijfrechten hebben, helpt om te voorkomen van foutieve of kwaadwillige toepassingen bestanden veranderen waarvan de gebruiker eigenaar is. In Fedora 11 kunnen Linux gebruikers in de `guest_t` en `xguest_t` domeinen standaard geen toepassingen in hun persoonlijke mappen of in **`/tmp/`**; Linux gebruikers in de `user_t` en `staff_t` domeinen kunnen dit echter wel.

Booleans zijn beschikbaar om dit gedrag te veranderen, en deze worden ingesteld met het **setsebool** commando. Het **setsebool** commando moet uitgevoerd worden als de Linux root gebruiker. Het **setsebool -P** commando maakt de veranderingen blijvend. Geruik de `-P` optie niet als je niet wilt dat veranderingen blijvend zijn na een nieuwe systeem opstart:

guest_t

Om Linux gebruikers in het `guest_t` domein *toe te staan* toepassingen uit te voeren in hun persoonlijke mappen en **/tmp/**:

```
/usr/sbin/setsebool -P allow_guest_exec_content on
```

xguest_t

Om Linux gebruikers in het `xguest_t` domein *toe te staan* om toepassingen uit te voeren in hun persoonlijke mappen en **/tmp/**:

```
/usr/sbin/setsebool -P allow_xguest_exec_content on
```

user_t

Om Linux gebruikers in het `user_t` domein te *beletten* om toepassingen uit te voeren in hun persoonlijke mappen en **/tmp/**:

```
/usr/sbin/setsebool -P allow_user_exec_content off
```

staff_t

Om Linux gebruikers in het `staff_t` domein te *beletten* toepassingen uit te voeren in hun persoonlijke mappen en **/tmp/**:

```
/usr/sbin/setsebool -P allow_staff_exec_content off
```

Foutzoeken

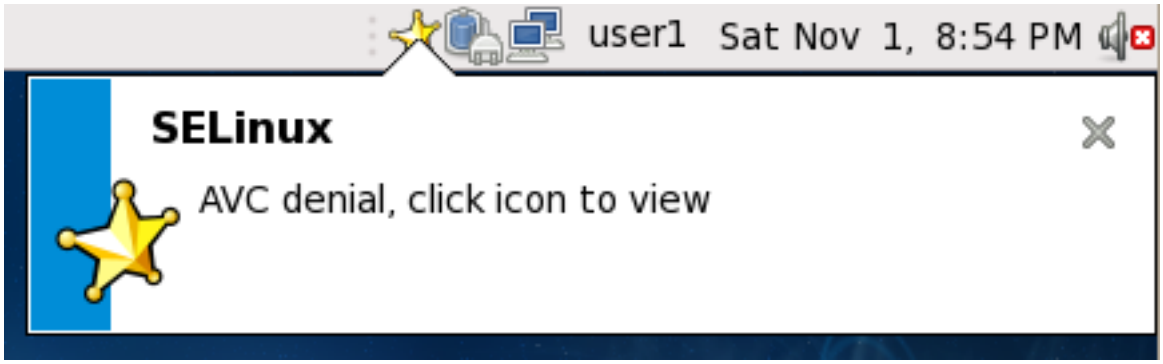
Het volgende hoofdstuk beschrijft wat er gebeurt als SELinux toegang weigert; de top drie oorzaken van problemen; waar informatie te vinden over correcte labels; SELinux weigeringen onderzoeken; en het maken aangepaste tactiekregels met **`audit2allow`**.

7.1. Wat gebeurt er als toegang wordt geweigerd

SELinux beslissingen, zoals het toestaan of weigeren van toegang, worden bewaard. Deze opslag staat bekend als de Access Vector Cache (AVC). Weigeringsboodschappen worden gelogd als SELinux toegang weigert. Deze weigeringen staan ook bekend als "AVC denials", en worden gelogd op een andere plaats, afhankelijk van welke daemons draaien:

Daemon	Log Location
auditd on	**`/var/log/audit/audit.log`**
auditd off; rsyslogd on	**`/var/log/messages`**
setroubleshootd, rsyslogd, and auditd on	**`/var/log/audit/audit.log`**. Easier-to-read denial messages also sent to **`/var/log/messages`**

Als je het X Window systeem draait, en de *setroubleshoot* en *setroubleshoot-server* zijn geinstalleerd, en de `setroubleshootd` en `auditd` daemons draaien, dan worden een gele ster en een waarschuwing getoond als toegang wordt geweigerd door SELinux:



Klikken op de ster presenteert een gedetaileerde analyse over waarom SELinux de toegang weigerde, en een mogelijke oplossing voor het toestaan van toegang. Als je hetX Window systeem niet draait, is het minder duidelijk wanneer toegang is geweigerd door SELinux. Bijvoorbeel, gebruikers die jouw website bezoeken kunnen een fout krijgen die lijkt op het volgende:

```
Forbidden

You don't have permission to access file name on this server
```

Voor deze situaties, als DAC regels (standaard L:innux rechten) toegang toestaan, controleer dan **`/var/log/messages`** en **`/var/log/audit/audit.log`** voor respectievelijk "`SELinux is preventing`" en "`denied`" fouten. Dit kan gedaan worden met het uitvoeren van de volgende commando's als de Linux root gebruiker:

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

grep "denied" /var/log/audit/audit.log
```

7.2. De top drie oorzaken van problemen

De volgende paragrafen beschrijven de top drie oorzaken van problemen: labelings problemen, het instellen van Booleans en poorten voor services, en het ontwikkelen van SELinux regels.

7.2.1. Labelings problemen

Op systemen die SELinux draaien worden alle processen en bestanden gelabeld met een label dat beveiligings relevante informatie bevat. Deze informatie wordt de SELinux context genoemd. Als deze labels verkeerd zijn, kan toegang geweigerd worden. Als een toepassing niet juist is gelabeld, kan het proces dat het voortbrengt een verkeerde label hebben, wat mogelijk veroorzaakt dat SELinux toegang weigert, en het proces is in staat om verkeerd gelabelde bestanden te maken.

Een veel voorkomende oorzaak van labelings problemen is het gebruiken van een niet-standaard map voor een sevice. Bijvoorbeeld, in plaats van het gebruiken van **`/var/www/html/`** voor een website, wil een beheerder **`/srv/myweb/`** gebruiken. In Fedora 11 is de **`/srv/`** map gelabeld met het `var_t` type. Bestanden en mappen aangemaakt in **`/srv/`** erven dit type. Ook kunnen nieuw aangemaakte top-niveau mappen (zoals **`/myserver/`**) gelabeld zijn met het `default_t` type. SELinux belet de Apache HTTP server (`httpd`) toegang tot beide deze types. Om toegang toe te staan, moet SELinux weten dat de bestanden in **`/srv/myweb/`** toegankelijk moeten zijn voor `httpd`:

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

Dit **`semanage`** commando voegt de context voor de **`/srv/myweb/`** map (en alle bestanden en mappen er in) toe aan de SELinux bestandscontext configuratie[1]. Het **`semanage`** commando verandert de context niet. Als de Linux root gebruiker voer je het **`restorecon`** commando uit om de veranderingen toe te passen:

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

Refereer naar *Paragraaf 5.7.2, “Permanente veranderingen: semanage fcontext”* voor meer informatie over het toevoegen van context aan de bestandscontext configuratie.

7.2.1.1. Wat is de juiste context?

Het **`matchpathcon`** commando controleert de context van een bestandspad en vergelijkt het met het standaard label voor dat pad. Het volgende voorbeeld laat het gebruik van **`matchpathcon`** zien voor een map die verkeerd gelabelde bestanden bevat:

```
$ /usr/sbin/matchpathcon -V /var/www/html/*
/var/www/html/index.html has context unconfined_u:object_r:user_home_t:s0,
 should be system_u:object_r:httpd_sys_content_t:s0
/var/www/html/page1.html has context unconfined_u:object_r:user_home_t:s0,
 should be system_u:object_r:httpd_sys_content_t:s0
```

In dit voorbeeld zijn de **`index.html`** en **`page1.html`** bestanden gelabeld met het `user_home_t` type. Dit type wordt gebruikt voor bestanden in persoonlijke mapen van gebruikers. Het gebruik

[1] Bestanden in **`/etc/selinux/targeted/contexts/files/`** definieren context voor bestanden en mappen. Bestanden in deze map worden gelezen door **`restorecon`** en **`setfiles`** om de standaard context van bestanden en mappen te herstellen.

van het **mv** commando om bestanden te verplaatsen vanuit je persoonlijke map kan als gevolg hebben dat de bestanden gelabeld zijn met het `user_home_t` type. Dit type moet niet bestaan buiten persoonlijke mappen. Gebruik het **restorecon** commando om het correcte type van zulke bestanden te herstellen:

```
# /sbin/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
```

Om de context voor alle bestanden in een map te herstellen, gebruik je de `-R` optie:

```
# /sbin/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
```

Refereer naar *Paragraaf 5.10.3, “Het controleren van de standaard SELinux context”* voor een uigebreider voorbeeld van **matchpathcon**.

7.2.2. Hoe draaien beperkte services?

Services kunnen op verschillende manieren gedraaid worden. Om hier rekening mee te houden moet je SELinux laten weten hoe je services draait. Dit kan bereikt worden met Booleans die toestaan dat onderdelen van SELinux tactiek veranderd worden tijdens het draaien, zonder enige kennis van het schrijven van SELinux tactiekregels. Dit staat veranderingen toe, zoals het toestaan aan services van toegang tot NFS bestandssystemen, zonder het herladen of hercompileren van SELinux tactiek. Ook vereisen services die draaien op niet-standaard poorten dat de tactiek configuratie vernieuwd wordt met het **semanage** commando.

Bijvoorbeeld, om de Apache HTTP server toe te staan te communiceren met MySQL, zet je de `httpd_can_network_connect_db` Boolean aan:

```
# /usr/sbin/setsebool -P httpd_can_network_connect_db on
```

Als toegang geweigerd wordt voor een bepaalde service, gebruik je de **getsebool** en **grep** commando's om te zien of er Booleans beschikbaar zijn die toegang toestaan. Bijvoorbeeld, gebruik het **getsebool -a | grep ftp** commando om te zoeken naar Booleans gerelateerd met FTP:

```
$ /usr/sbin/getsebool -a | grep ftp
allow_ftpd_anon_write --> off
allow_ftpd_full_access --> off
allow_ftpd_use_cifs --> off
allow_ftpd_use_nfs --> off
ftp_home_dir --> off
httpd_enable_ftp_server --> off
tftp_anon_write --> off
```

Voor een lijst met Booleans en te zien of ze aan of uit zijn, voer je het **`/usr/sbin/getsebool -a`** commando uit. Voor een lijst met Booleans, en uitleg over wat ieder is, en of ze aan of uit zijn, voer je het **`/usr/sbin/semanage boolean -l`** commando uit als de Linux root gebruiker. Refereer naar *Paragraaf 5.6, "Booleans"* voor informatie over het tonen en instellen van Booleans.

Poort nummers

Afhankelijk van de tactiek instelling, is het aan services toegestaan om alleen maar op bepaalde poort nummers te draaien. Het proberen om de poort te veranderen waarop een service draait zonder het veranderen van tactiek kan als resultaat hebben dat de service niet start. Bijvoorbeeld, voer het **`semanage port -l | grep http`** commando uit als de Linux root gebruiker om een lijst te zien van de aan `http` gerelateerde poorten:

```
# /usr/sbin/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
```

Het `http_port_t` poort type definieert de poorten waarnaar de Apache HTTP server kan luisteren, welke in dit geval zijn, de TCP poorten 80, 443, 488, 8008, 8009, en 8443. Als een beheerder **`httpd.conf`** zodanig instelt dat `httpd` luistert op poort 9876 (`Listen 9876`), maar tactiek is niet vernieuwd om dit te weten, zal het **`service httpd start`** commando falen:

```
# /sbin/service httpd start
Starting httpd: (13)Permission denied: make_sock: could not bind to address
 [::]:9876
(13)Permission denied: make_sock: could not bind to address 0.0.0.0:9876
no listening sockets available, shutting down
Unable to open logs
                                                           [FAILED]
```

Een SELinux weigering lijkend op de volgende wordt gelogd naar **`/var/log/audit/audit.log`**:

```
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
```

Om `httpd` toe te staan te luisteren op een poort dat niet getoond wordt voor het `http_port_t` poort type, voer je het **`semanage port`** commando uit om een poort toe te voegen aan de tactiek configuratie[2]:

```
# /usr/sbin/semanage port -a -t http_port_t -p tcp 9876
```

De `-a` optie voegt een nieuwe aantekening toe, de `-t` optie definieert een type, en de `-p` option definieert een protocol. Het laatste argument is het poortnummer dat toegevoegd moet worden.

[2] Het **`semanage port -a`** commando voegt een regle toe aan het **`/etc/selinux/targeted/modules/active/ports.local`** bestand. merk op: standaard kan dit bestand alleen bekeken worden door de Linux root gebruiker.

7.2.3. Het ontwikkelen van regels en gebrekkige toepassingen

Toepassingen kunnen gebreken hebben, wat veroorzaakt dat SELinux toegang weigert. Ook zijn SELinux regels in ontwikkeling - SELinux heeft misschien een applicatie niet op een bepaalde manier zien werken, en kan mogelijk toegang weigeren, zelfs als de toepassing werkt zoals verwacht. Bijvoorbeeld, als een nieuwe versie van PostgreSQL wordt vrijgegeven, kan deze acties uitvoeren die de huidige tactiek nog nooit heeft gezien, wat een toegangsweigering veroorzaakt, zelfs al zou de toegang toegestaan moeten worden.

Voor deze situaties, als toegang is geweigerd, gebruik je **`audit2allow`** om een aangepaste tactiek module te maken om toegang toe te staan. Refereer naar *Paragraaf 7.3.8, “Toegang toestaan: audit2allow”* voor informatie over het gebruik van **`audit2allow`**.

7.3. Problemen herstellen

De volgende paragrafen helpen met foutzoek zaken. Ze behandelen: het controleren van Linux rechten, welke toegepast worden voor de SELinux regels; mogelijke oorzaken voor toegangsweigering door SELinux, waarvan geen details gelogd zijn; manual pagina's voor services, welke informatie bevatten over labeling en Booleans; toestemmende domeinen, om een proces toe te staan toestemmend te draaien, in plaats van het hele systeem; hoe te zoeken naar weigeringsbooschappen en hoe ze te bekijken; het analyseren van weigeringen; en het maken van aangepaste tactiek modules met **`audit2allow`**.

7.3.1. Linux rechten

Als toegang geweigerd wordt, controleer dan de standaard Linux rechten. Zoals genoemd in *Hoofdstuk 2, Inleiding*, gebruiken de meeste operating systemen een Discretionary Access Control (DAC) (toegangscontrole naar goeddunken) systeem om toegang te controleren, welke gebruikers toestaat de rechten te controleren van bestanden waarvan ze eigenaar zijn. SELinux tactiekregels worden gecontroleerd na de DAC regels. SELinux tactiekregels worden niet gebruikt als de DAC regels als eerste toegang weigeren.

Als toegang wordt geweigerd en er zijn geen SELinux weigeringen gelogd, gebruik je het **`ls -l`** commando om de standaard Linux rechten te zien:

```
$ ls -l /var/www/html/index.html
-rw-r----- 1 root root 0 2009-05-07 11:06 index.html
```

In dit voorbeeld zijn de root gebruiker en groep eigenaar van **`index.html`**. De root gebruiker heeft lees en schrijf rechten (`-rw`), en leden van de root groep hebben lees rechten (`-r-`). Alle anderen hebben geen toegang (`---`). Standaard laten zulke rechten niet toe dat `httpd` dit bestand leest. Om dit op te lossen, gebruik je het **`chown`** commando om de eigenaar en de groep te veranderen. Dit commando moet uitgevoerd worden als de Linux root gebruiker:

```
# chown apache:apache /var/www/html/index.html
```

Dit neemt de standaard configuratie aan, waarin `httpd` draait als de Linux apache gebruiker. Als je `httpd` draait met een andere gebruiker, vervang je `apache:apache` met die gebruiker.

Refereer naar de *Fedora Documentation Project "Permissions"*[3] concept voor informatie over het beheren van Linux rechten.

[3] http://fedoraproject.org/wiki/Docs/Drafts/AdministrationGuide/Permissions

7.3.2. Mogelijke oorzaken van stille weigeringen

In bepaalde situaties worden AVC weigeringen misschien niet gelogd als SELinux toegang weigert. Toepassingen en systeembibliotheek functies onderzoeken vaak voor meer toegang dan nodig is voor het uitvoeren van hun taak. Om zo weinig mogelijk rechten te onderhouden, zonder de log te vullen met AVC weigeringen voor onschuldig onderzoeken door toepassingen, kan de tactiek AVC weigeringen stil houden zonder een recht toe te staan door het gebruiken van `dontaudit` regels. Deze regels komen veel voor in standaard tactiek. Het nadeel van `dontaudit` is dat, hoewel SELinux toegang weigert, weigeringsboodschappen niet worden gelogd, wat het foutzoeken moeilijk maakt.

Om tijdelijk `dontaudit` regels uit te zetten, zodat alle weigeringen gelogd worden, voer je het volgende commando uit als de Linux root gebruiker:

`/usr/sbin/semodule -DB`

De `-D` zet `dontaudit` regels uit; de `-B` optie bouwt de tactiek opnieuw op. Na het uitvoeren van **`semodule -DB`**, probeer je de toepassing opnieuw die toestemmigs problemen vertoonde, en kijk of er nu SELinux weigeringen — relevant voor de toepassing — zijn gelogd. Wees voorzichtig met het beslissen welke weigeringen toegestaan moeten worden, omdat sommige genegeerd moeten worden en afgehandeld met `dontaudit` regels. In geval van twijfel, of als je hulp zoekt, neem dan contact op met andere SELinux gebruikers en ontwikkelaars op een SELinux lijst, zoals *fedora-selinux-list*[4].

Om de tactiek opnieuw te bouwen en `dontaudit` regels weer aan te zetten, voer je het volgende commando uit als de Linux root gebruiker:

`/usr/sbin/semodule -B`

Dit brengt de tactiek terug naar zijn originele toestand. Voor een volledige lijst van `dontaudit` regels, voer je het **`sesearch --dontaudit`** commando uit. Versmal het zoeken met gebruik van de `-s` *`domein`* optie en het **`grep`** commando. Bijvoorbeeld:

```
$ sesearch --dontaudit -s smbd_t | grep squid
WARNING: This policy contained disabled aliases; they have been removed.
dontaudit smbd_t squid_port_t : tcp_socket name_bind ;
dontaudit smbd_t squid_port_t : udp_socket name_bind ;
```

Refereer naar *Paragraaf 7.3.6, “Ruwe audit boodschappen”* en *Paragraaf 7.3.7, “sealert boodschappen”* voor informatie over het onderzoeken van weigeringen.

7.3.3. Manual pagina's voor services

Manual pagina's voor services bevatten waardevolle informatie, zoals welk bestandstype gebruikt moet worden voor een bepaalde situatie, en Booleans om de toegang die een sevice heeft te veranderen (zoals toegang tot NFS bestandssystemen voor `httpd`). Deze informatie kan in de standaard manual pagina zijn, of in een manual pagina met `selinux` voor of achter de servicenaam.

Bijvoorbeeld, de httpd_selinux(8) manual pagina heeft informatie over het te gebruiken bestandstype in een bepaalde situatie, en ook Booleans voor het toestaan van scripts, bestanden delen, toegang krijgen tot mappen in de persoonlijke mappen van gebruikers, enzovoort. Andere manual pagina's met SELinux informatie voor services zijn:

- Samba: de samba_selinux(8) manual pagina beschrijft dat bestanden en mappen die met Samba geexporteerd worden gelabeld moeten zijn met het `samba_share_t` type, en ook Booleans die

[4] http://www.redhat.com/mailman/listinfo/fedora-selinux-list

toestaan dat bestanden gelabeld met andere types dan `samba_share_t` met Samba geexporteerd kunnen worden.

- NFS: de nfs_selinux(8) manual pagina beschrijft dat standaard bestandssystemen niet met NFS geexporteerd kunnen worden, en om toe te staan dat bestanden geexporteerd kunnen worden, Booleans zoals `nfs_export_all_ro` of `nfs_export_all_rw` aangezet moeten worden.

- Berkeley Internet Name Domain (BIND): de named(8) manual pagina beschrijft welke bestandstype te gebruiken voor een bepaalde situatie (zie de `Red Hat SELinux BIND Security Profile` paragraaf). De named_selinux(8) manual pagina beschrijf dat `named` standaard niet naar master zone bestanden kan schrijven, en om deze toegang toe te staan, de `named_write_master_zones` Boolean aangezet moet worden.

De informatie in manual pagina's helpt je de juiste bestandstypes en Booleans in te stellen, wat helpt voorkomen dat SELinux toegangs weigeringen geeft

7.3.4. Toelatende domeinen

Als SELinux in de toelatende modus draait, weigert SELinux geen toegang, maar weigeringen worden gelogd voor acties die geweigerd zouden zijn als SELinux in de afdwingende modus draaide. Vroeger was het niet mogelijk om een enkel domein toelatend te maken (denk eraan: processen draaien in domeinen). In bepaalde situaties leidde dit er toe dat het hele systeem toelatend werd gemaakt voor foutzoek doeleinden.

Fedora 11 introduceert toelatende domeinen, waarbij een beheerder een enkel proces (domein) toelatend kan draaien, in plaats van het hele systeem toelatend te maken. SELinux controles worden nog steeds gedaan voor toelatende domeinen, de kernel laat echter toegang toe en rapporteert een AVC weigering voor situaties waar SELinux toegang geweigerd zou hebben. Toelatende domeinen zijn ook beschikbaar in Fedora 9 (als de laatste vernieuwingen toegepast zijn).

In Red Hat Enterprise Linux 4 en 5 zijn *`domain`*`_disable_trans` Booleans beschikbaar om een toepassing te beletten om over te gaan naar een beperkt domein, en daarom draait het proces in een onbeperkt domein, zoals `initrc_t`. Het aanzetten van zulke Booleans kan grote problemen veroorzaken. Bijvoorbeeld, als de `httpd_disable_trans` Boolean aangezet is:

- draait `httpd` in het onbeperkte `initrc_t` domein. Bestanden gemaakt door processen die in het `initrc_t` domein draaien hebben misschien niet dezelfde labelingsregels toegepast gekregen als bestanden die draaien in het `httpd_t` domein, wat potentieel toestaat dat processen verkeerd gelabelde bestanden maken. Dit veroorzaakt later toegangs problemen.

- beperkte domeinen die toegestaan worden om te communiceren met `httpd_t` kunnen niet communiceren met `initrc_t`, wat potentieel extra fouten veroorzaakt.

De *`domain`*`_disable_trans` Booleans zijn verwijderd in Fedora 7, hoewel er geen vervanger voor was. Toelatende domeinen lossen de hierboven beschreven problemen op: overgangs regels worden toegepast, en bestanden worden aangemaakt met de juiste labels.

Toelatende domeinen kunnen gebruikt worden voor:

- maak een enkel proces (domein) toelatend om een probleem op te lossen, in plaats van het gehele systeem kwetsbaar te maken door het hele systeem toelatend te maken.

- het maken van tactiek voor nieuwe toepassingen. Vroeger was het aanbevolen dat een minmale tactiek werd gemaakt, en dat daarna de hele machine in de toelatende modus werd gebracht, zodat de toepassing kon draaien, maar SELinux weigeringen nog steeds gelogd werden. **`audit2allow`**

kon daarna gebruikt worden om te helpen met het schrijven van de tactiek. Dit maakte het gehele systeem kwetsbaar. Met toelatende domeinen, wordt alleen het domein in de nieuwe tactiek toelatend, zonder het hele systeem kwetsbaar te maken.

7.3.4.1. Een domein toelatend maken

Om een domein toelatend te maken voer je het **semanage permissive -a *domein*** commando uit, waarin *domein* het domein is dat je toelatend wilt maken. Bijvoorbeeld, voer het volgende commando uit als de Linux root gebruiker om het `httpd_t` domein (het domein waarin de Apache HTTP sever draait) toelatend te maken:

/usr/sbin/semanage permissive -a httpd_t

Om een lijst te zien van de domeinen die je toelatend hebt gemaakt, voer je het **semodule -l | grep permissive** commando uit als de Linux root gebruiker. Bijvoorbeeld:

```
# /usr/sbin/semodule -l | grep permissive
permissive_httpd_t      1.0
```

Als je een domein niet langer toelatend wilt laten zijn, voer je het **semanage permissive -d *domein*** commando uit als de Linux root gebruiker. Bijvoorbeeld:

/usr/sbin/semanage permissive -d httpd_t

7.3.4.2. Weigeringen voor toelatende domeinen

De SYSCALL boodschap is verschillend voor toelatende domeinen. Het volgende is een voorbeeld AVC weigering (en de bijbehorende systeem aanroep) van de Apache HTTP server:

```
type=AVC msg=audit(1226882736.442:86): avc:  denied  { getattr }
 for  pid=2427 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

type=SYSCALL msg=audit(1226882736.442:86): arch=40000003 syscall=196
 success=no exit=-13 a0=b9a1e198 a1=bfc2921c a2=54dff4 a3=2008171 items=0
 ppid=2425 pid=2427 auid=502 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48
 sgid=48 fsgid=48 tty=(none) ses=4 comm="httpd" exe="/usr/sbin/httpd"
 subj=unconfined_u:system_r:httpd_t:s0 key=(null)
```

Standaard is het `httpd_t` domein niet toelatend, en daarom wordt de actie geweigerd, en de SYSCALL boodschap bevat `success=no`. Het volgende is een voorbeeld AVC weigering voor dezelfde actie, behalve dat het **semanage permissive -a httpd_t** commando is uitgevoerd om het `httpd_t` domein toelatend te maken:

```
type=AVC msg=audit(1226882925.714:136): avc:  denied
 { read } for  pid=2512 comm="httpd" name="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

type=SYSCALL msg=audit(1226882925.714:136): arch=40000003 syscall=5
 success=yes exit=11 a0=b962a1e8 a1=8000 a2=0 a3=8000 items=0 ppid=2511
```

```
pid=2512 auid=502 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48
sgid=48 fsgid=48 tty=(none) ses=4 comm="httpd" exe="/usr/sbin/httpd"
subj=unconfined_u:system_r:httpd_t:s0 key=(null)
```

In dit geval, hoewel er een AVC weigering gelogd is, was toegang niet geweigerd, zoals getoond wordt door `success=yes` in de `SYSCALL` boodschap.

Refereer naar de blog van Dan Walsh *"Permissive Domains"*[5] voor meer informatie over toelatende domeinen.

7.3.5. Zoeken naar en het bekijken van weigeringen

Deze paragraaf neemt aan dat de *setroubleshoot*, *setroubleshoot-server*, en *audit* pakketten geinstalleerd zijn, en dat de `auditd`, `rsyslogd`, en `setroubleshootd` daemons draaien. Refereer naar *Paragraaf 5.2, "Welk log bestand wordt gebruikt"* voor informatie over het starten van deze daemons. Een aantal gereedschappen zijn beschikbaar voor het zoeken naar en bekijken van SELinux weigeringen, zoals **`ausearch`**, **`aureport`**, en **`sealert`**.

ausearch

Het *audit* pakket bevat **`ausearch`**. Uit de ausearch(8) manual pagina: "**`ausearch`** is een gereedschap dat de daemon logs kan onderzoeken voor gebeurtenissen gebaseerd op verscheidene zoek criteria"[6]. Het **`ausearch`** gereedschap heeft toegang tot **`/var/log/audit/audit.log`**, en moet daarom als de Linux root gebruiker uitgevoerd worden:

Zoeken naar	Commando
alle weigeringen	**`/sbin/ausearch -m avc`**
weigeringen voor vandaag	**`/sbin/ausearch -m avc -ts today`**
weigeringen van de laatste 10 minuten	**`/sbin/ausearch -m avc -ts recent`**

Om te zoeken naar SELinux weigeringen voor een bepaalde service, gebruik je de `-c` *`comm-naam`* optie, waarin *`comm-naam`* de naam van het programma is[7], bijvoorbeeld, `httpd` voor de Apache HTTP server, en `smbd` voor Samba:

```
/sbin/ausearch -m avc -c httpd
```

```
/sbin/ausearch -m avc -c smbd
```

Refereer naar de ausearch(8) manual pagina voor meer **`ausearch`** opties.

aureport

Het *audit* pakket bevat **`aureport`**. Uit de aureport(8) manual pagina: "**`aureport`** is een gereedschap dat samenvattingsrapporten maakt van de systeem logs"[8]. Het **`aureport`** gereedschap heeft toegang tot **`/var/log/audit/audit.log`**, en moet daarom als de Linux root gebruiker uitgevoerd worden. Om een lijst te zien van SELinux weigeringen en hoe vaak ze zijn voorgekomen, voer je het **`aureport -a`** commando uit. Het volgende is een vorbeeld output die twee weigeringen bevat:

[5] http://danwalsh.livejournal.com/24537.html
[6] Van de ausearch(8) manual pagina, zoals verstuurd met het *audit* pakket in Fedora 11.
[7] Van de ausearch(8) manual pagina, zoals verstuurd met het *audit* pakket in Fedora 11.
[8] Uit de aureport(8) manual pagina, zoals verstuurd met het *audit* pakket in Fedora 11.

```
# /sbin/aureport -a

AVC Report
============================================================
# date time comm subj syscall class permission obj event
============================================================
1. 05/01/2009 21:41:39 httpd unconfined_u:system_r:httpd_t:s0 195 file
 getattr system_u:object_r:samba_share_t:s0 denied 2
2. 05/03/2009 22:00:25 vsftpd unconfined_u:system_r:ftpd_t:s0 5 file read
 unconfined_u:object_r:cifs_t:s0 denied 4
```

Refereer naar de aureport(8) manual pagina voor meer **`aureport`** opties.

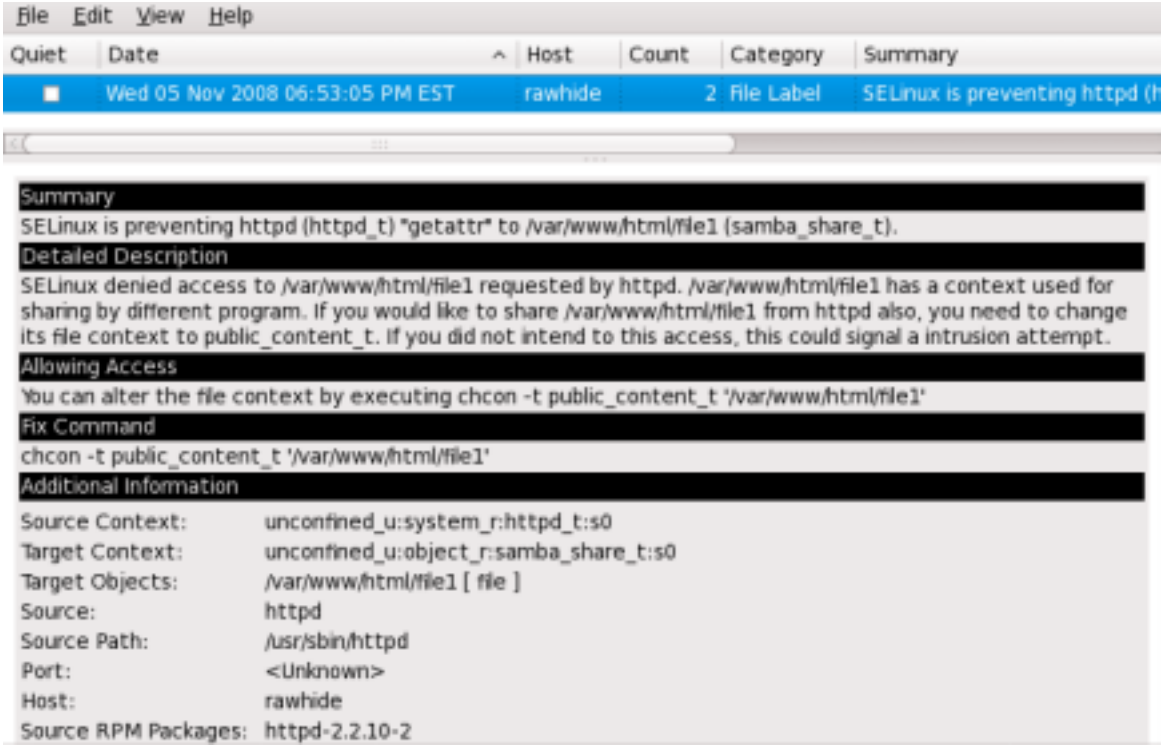
sealert

Het *setroubleshoot-server* pakket bevat **`sealert`**, welke weigerings boodschappen leest die vertaald zijn door *setroubleshoot-server*. Weigeringen krijgen ID's toegekend, zoals te zien in **`/var/log/messages`**. Het volgende is een voorbeeld weigering van **`messages`**:

```
setroubleshoot: SELinux is preventing httpd (httpd_t) "getattr" to /var/
www/html/file1 (samba_share_t). For complete SELinux messages. run sealert
 -l 84e0b04d-d0ad-4347-8317-22e74f6cd020
```

In dit voorbeeld is de ID van de weigering `84e0b04d-d0ad-4347-8317-22e74f6cd020`.
De `-l` optie neemt een ID als argument. Het uitvoeren van het **`sealert -l 84e0b04d-d0ad-4347-8317-22e74f6cd020`** commando laat een gedetaileerde analyse zien waarom SELinux toegang weigerde, en een mogelijke oplossing om toegang toe te staan.

Als je het X Window systeen draait, je de *setroubleshoot* en *setroubleshoot-server* pakketten geinstalleerd hebt, en de `setroubleshootd` en `auditd` daemons draaien, worden een gele ster en een waarschuwing getoond als toegang geweigerd wordt door SELinux. Het klikken op de ster start de **`sealert`** GUI, en laat de weigeringen zien in HTML output:



- Voer het **`sealert -b`** commando uit om de **`sealert`** GUI op te starten.
- Voer het **`sealert -l \*`** commando uit om een gedetaileerde analyse van alle weigeringen te zien.
- Als de Linux root gebruiker voer je het **`sealert -a /var/log/audit/audit.log -H > audit.html`** commando uit om een HTML versie te maken van de **`sealert`** analyse, zoals te zien is met de **`sealert`** GUI.

Refereer naar de sealert(8) manual pagina voor meer **`sealert`** opties.

7.3.6. Ruwe audit boodschappen

Ruwe audit boodschappen worden gelogd naar **`/var/log/audit/audit.log`**. Het volgende is een voorbeeld AVC weigering (en de bijbehorende systeem aanroep) die optrad toen de Apache HTTP server (draaiende in het `httpd_t` domein) probeerde om toegang te krijgen tot het **`/var/www/html/file1`** bestand (gelabeld met het `samba_share_t` type):

```
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

type=SYSCALL msg=audit(1226874073.147:96): arch=40000003 syscall=196
 success=no exit=-13 a0=b98df198 a1=bfec85dc a2=54dff4 a3=2008171 items=0
 ppid=2463 pid=2465 auid=502 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48
 sgid=48 fsgid=48 tty=(none) ses=6 comm="httpd" exe="/usr/sbin/httpd"
 subj=unconfined_u:system_r:httpd_t:s0 key=(null)
```

{ getattr }
: Het item tussen accolades geeft het recht aan dat geweigerd was. `getattr` geeft aan dat het bron proces probeerde om de status informatie van het doel bestand te lezen. Dit treedt op voor het lezen van bestanden. Deze actie is geweigerd omdat toegang werd geprobeerd naar het bestand dat een verkeerd label had. Vaak geziene rechten zijn `getattr`, `read`, en `write`.

comm="*httpd*"
: Het programma dat het proces heeft opgestart. Het volledige pad van het programma wordt gevonden in de `exe=` sectie van de systeem aanroep (SYSCALL) boodschap, welke in dit geval `exe="/usr/sbin/httpd"` is.

path="*/var/www/html/file1*"
: Het pad naar het object (doel) waar het proces toegang naar probeerde te krijgen.

scontext="*unconfined_u:system_r:httpd_t:s0*"
: De SELinux context van het proces dat de geweigerde actie probeerde uit te voeren. In dit geval is het de SELinux context van de Apache HTTP server, welke draait in het `httpd_t` domein.

tcontext="*unconfined_u:object_r:samba_share_t:s0*"
: De SELinux context van het object (doel) waarnaar het proces toegang probeerde te krijgen. In dit geval is het de SELinux context van **`file1`**. Merk op: het `samba_share_t` type is niet bereikbaar voor processen die draaien in het `httpd_t` domein.

  In bepaalde situaties kan de `tcontext` overeenkomen met de `scontext`, bijvoorbeeld, als een proces probeert een systeem service uit te voeren dat de eigenschappen van dat draaiende proces zal veranderen, zoals een gebruikers ID. Ook kan de `tcontext` overeenkomen met de `scontext` als een proces probeert meer hulpbronnen (zoals geheugen) te gebruiken dan de normale limieten toestaan, wat resulteert in een beveiligings controle om te zien of dat proces gemachtigd is om die limieten te doorbreken.

Van de systeem aanroep (SYSCALL) boodachap, zijn twee items interessant:

- `success=`*no*: geeft aan of de weigering (AVC) afdwingend was of niet. `success=no` geeft aan dat de system aanroep gefaald heeft (SELinux weigerde toegang). `success=yes` geeft aan dat de systeem aanroep gelukt is - dit kan voorkomen bij toelatende domeinen of onbeperkte domeinen, zoals `initrc_t` en `kernel_t`.
- `exe="`*`/usr/sbin/httpd`*`"`: het volledige pad naar het programma dat het proces heeft opgestart, welke in dit geval `exe="/usr/sbin/httpd"` is.

Een foutief bestandstype is een veel voorkomende oorzaak voor het weigeren van toegang door SELinux. Om te beginnen met foutzoeken, vergelijk je de bron context (`scontext`) met de doel context (`tcontext`). Moet het proces (`scontext`) toegang hebben tot zo'n object (`tcontext`)? Bijvoorbeeld, de Apache HTTP server (`httpd_t`) moet alleen toegang hebben tot types gespecificeerd in de httpd_selinux(8) manual pagina, zoals `httpd_sys_content_t`, `public_content_t`, enzovoort, behalve als iets anders ingesteld was.

7.3.7. sealert boodschappen

Weigeringen krijgen ID's toegewezen, zoals te zien is in **`/var/log/messages`**. Het volgende is een voorbeeld AVC weigering (gelogd naar **`messages`**) die optrad toen de Apache HTTP server (draaiende in het `httpd_t` domein) probeerde toegang te krijgen tot het **`/var/www/html/file1`** bestand (gelabeld met het `samba_share_t` type):

```
hostname setroubleshoot: SELinux is preventing httpd (httpd_t) "getattr"
 to /var/www/html/file1 (samba_share_t). For complete SELinux messages. run
 sealert -l 84e0b04d-d0ad-4347-8317-22e74f6cd020
```

Zoals aangeraden, voer je het **`sealert -l 84e0b04d-d0ad-4347-8317-22e74f6cd020`** commando uit om de complete boodschap te zien. Dit commando werkt alleen op de locale machine, en presenteert dezelfde informatie als de **`sealert`** GUI:

```
$ sealert -l 84e0b04d-d0ad-4347-8317-22e74f6cd020

Summary:

SELinux is preventing httpd (httpd_t) "getattr" to /var/www/html/file1
(samba_share_t).

Detailed Description:

SELinux denied access to /var/www/html/file1 requested by httpd.
/var/www/html/file1 has a context used for sharing by different program. If
 you
would like to share /var/www/html/file1 from httpd also, you need to change
 its
file context to public_content_t. If you did not intend to this access,
 this
could signal a intrusion attempt.

Allowing Access:

You can alter the file context by executing chcon -t public_content_t
'/var/www/html/file1'

Fix Command:

chcon -t public_content_t '/var/www/html/file1'

Additional Information:

Source Context                unconfined_u:system_r:httpd_t:s0
Target Context                unconfined_u:object_r:samba_share_t:s0
Target Objects                /var/www/html/file1 [ file ]
Source                        httpd
Source Path                   /usr/sbin/httpd
Port                          <Unknown>
Host                          hostname
Source RPM Packages           httpd-2.2.10-2
Target RPM Packages
Policy RPM                    selinux-policy-3.5.13-11.fc11
Selinux Enabled               True
Policy Type                   targeted
MLS Enabled                   True
Enforcing Mode                Enforcing
```

```
Plugin Name                   public_content
Host Name                     hostname
Platform                      Linux hostname 2.6.27.4-68.fc11.i686 #1 SMP
 Thu Oct
30 00:49:42 EDT 2008 i686 i686
Alert Count                   4
First Seen                    Wed Nov  5 18:53:05 2008
Last Seen                     Wed Nov  5 01:22:58 2008
Local ID                      84e0b04d-d0ad-4347-8317-22e74f6cd020
Line Numbers

Raw Audit Messages

node=hostname type=AVC msg=audit(1225812178.788:101): avc:  denied
  { getattr } for  pid=2441 comm="httpd" path="/var/www/html/file1"
 dev=dm-0 ino=284916 scontext=unconfined_u:system_r:httpd_t:s0
 tcontext=unconfined_u:object_r:samba_share_t:s0 tclass=file

node=hostname type=SYSCALL msg=audit(1225812178.788:101): arch=40000003
 syscall=196 success=no exit=-13 a0=b8e97188 a1=bf87aaac a2=54dff4
 a3=2008171 items=0 ppid=2439 pid=2441 auid=502 uid=48 gid=48 euid=48
 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=3 comm="httpd"
 exe="/usr/sbin/httpd" subj=unconfined_u:system_r:httpd_t:s0 key=(null)
```

Summary - samenvatting
: Een korte samenvatting van de geweigerde actie. Dit is hetzelfde als de weigering in **`/var/log/messages`**. In dit voorbeeld, werd het `httpd` proces toegang geweigerd tot een bestand (**`file1`**), welke gelabeld is met het `samba_share_t` type.

Detailed Description - Gedetaileerde beschrijving
: Een langere beschrijving. In dit voorbeeld is **`file1`** gelabeld met het `samba_share_t` type. Dit type wordt gebruikt voor bestanden en mappen die je wilt exporteren met Samba. De beschrijving suggereert het veranderen van het type naar een type dat bereikt kan worden door de Apache HTTP server en Samba, als die toegang gewenst is.

Allowing Acces - Toegang toestaan
: Een suggestie voor het toestaan van de toegang. Dit kan het herlabelen van bestanden zijn, het aanzetten van een Boolean, of het maken van een locale tactiek module. In dit geval is de aanbeveling om het bestand te labelen met een type waarnaar zowel de Apache HTTP server en Samba toegang naar hebben.

Fix Command - Herstel commando
: Een gesuggereerd commando om toegang toe te staan en de weigering op te lossen. In dit voorbeeld is dat het commando om het type van **`file1`** te veranderen naar `public_content_t`, welke toegankelijk is voor de Apache HTTP server en Samba.

Additional Information - Extra informatie
: Informatie die nuttig is voor foutrapporten, zoals de naam en versie van het tactiek pakket (`selinux-policy-3.5.13-11.fc11`), maar die niet zal helpen om de reden van de weigering op te lossen.

Raw Audit Messages - Ruwe audit boodschappen

De ruwe audit boodschappen van **/var/log/audit/audit.log** die verbonden zijn met de weigering. Refereer naar *Paragraaf 7.3.6, "Ruwe audit boodschappen"* voor informatie over ieder item in de AVC weigering.

7.3.8. Toegang toestaan: audit2allow

Gebruik het voorbeeld in deze paragraaf niet in een productieomgeving. Het wordt alleen gebruikt om het gebruik van **audit2allow** te laten zien.

Uit de audit2allow(1) manual pagina: "**audit2allow** - maak SELinux tactiek toelatingsregels van de logs van geweigerde operaties"[9]. Na het onderzoeken van weigeren zoals in *Paragraaf 7.3.7, "sealert boodschappen"*, en als label veranderingen of Booleans aanzetten geen toegang toestaan, gebruik je **audit2allow** om een locale tactiek module te maken. Nadat toegang geweigerd is door SELinux, presenteert het draaien van het **audit2allow** commando Type Enforcement regels die de hiervoor geweigerde toegang toestaan.

Het volgende voorbeeld laat het gebruik van **audit2allow** zien om een tactiek module te maken:

1. Een weigering en de bijbehorende systeem aanroep zijn gelogd naar **/var/log/audit/audit.log**:

```
type=AVC msg=audit(1226270358.848:238): avc:  denied
 { write } for  pid=13349 comm="certwatch" name="cache"
 dev=dm-0 ino=218171 scontext=system_u:system_r:certwatch_t:s0
 tcontext=system_u:object_r:var_t:s0 tclass=dir

type=SYSCALL msg=audit(1226270358.848:238): arch=40000003 syscall=39
 success=no exit=-13 a0=39a2bf a1=3ff a2=3a0354 a3=94703c8 items=0
 ppid=13344 pid=13349 auid=4294967295 uid=0 gid=0 euid=0 suid=0 fsuid=0
 egid=0 sgid=0 fsgid=0 tty=(none) ses=4294967295 comm="certwatch" exe="/
usr/bin/certwatch" subj=system_u:system_r:certwatch_t:s0 key=(null)
```

   In dit voorbeeld was **certwatch** (`comm="certwatch"`) schrijf toegang (`{ write }`) geweigerd naar een map gelabeld met het `var_t` type (`tcontext=system_u:object_r:var_t:s0`). Analyseer de weigering zoals in *Paragraaf 7.3.7, "sealert boodschappen"*. Als label veranderingen of Booleans aanzetten, geen toegang toestaan, gebruik je **audit2allow** om een locale tactiek module te maken.

2. Met een gelogde weigering, zoals de `certwatch` weigering in stap 1, voer je het **audit2allow -w -a** commando uit om een leesbare beschrijving te maken van de reden van de weigering. De `-a` optie zorgt ervoor dat alle logs gelezen worden. De `-w` optie produceert een leesbare beschrijving. Het **audit2allow** programma neemt toegang tot **/var/log/audit/audit.log**, en moet daarom uitgevoerd worden als de Linux root gebruiker:

```
# audit2allow -w -a
type=AVC msg=audit(1226270358.848:238): avc:  denied
 { write } for  pid=13349 comm="certwatch" name="cache"
 dev=dm-0 ino=218171 scontext=system_u:system_r:certwatch_t:s0
 tcontext=system_u:object_r:var_t:s0 tclass=dir
```

[9] Uit de audit2allow(1) manual pagina, zoals verstuurd met het *policycoreutils* pakket in Fedora 11.

```
        Was caused by:
                Missing type enforcement (TE) allow rule.

        You can use audit2allow to generate a loadable module to allow
 this access.
```

Zoals getoond was toegang geweigerd door een ontbrekende Type Enforcement regel.

3. Voer het **`audit2allow -a`** commando uit om de Type Enforcement regel te bekijken toe de geweigerde toegang toestaat:

```
# audit2allow -a


#============= certwatch_t ==============
allow certwatch_t var_t:dir write;
```

Belangrijk

Ontbrekende Type Enforcement regels worden gewoonlijk veroorzaakt door fouten in SELinux tactiek, en moeten gemeld worden in *Red Hat Bugzilla*[10]. Voor Fedora, maak je een foutrapport voor het `Fedora` product, en je selecteert de `selinux-policy` component. Voeg de output van de **`audit2allow -w -a`** en **`audit2allow -a`** commando's toe in zo'n foutrapport.

4. Om de regel te gebruiken die **`audit2allow -a`** liet zien, voer je het **`audit2allow -a -M mycertwatch`** commando uit als de Linux root gebruiker om de aangepaste module te maken. De `-M` optie maakt een Type Enforcement bestand (**`.te`**) met de naam opgegeven met de `-M` in je huidige werkmap:

```
# audit2allow -a -M mycertwatch

****************** IMPORTANT ***********************
To make this policy package active, execute:

semodule -i mycertwatch.pp

# ls
mycertwatch.pp  mycertwatch.te
```

`audit2allow` compileert de Type Enforcement regel ook naar een tactiek pakket (**`.pp`**). Om de module te installeren, voer je het **`/usr/sbin/semodule -i mycertwatch.pp`** commando uit als de Linux root gebruiker.

Belangrijk

Modules gemaakt met **`audit2allow`** kunnen meer toegang toestaan dan nodig is. Het wordt aanbevolen dat tactiek die gemaakt is met **`audit2allow`** voor bespreking

opgestuurd wordt naar een SELinux lijst, zoals *fedora-selinux-list*[11]. Als je denkt dat er een fout is in de tactiek, maak dat een foutrapport aan in *Red Hat Bugzilla*[12].

Als je meerdere weigeringen hebt van verschillende processen, maar je wilt alleen een aangepaste tactiek maken voor een enkel proces, gebruik je het **grep** commando om de input voor **audit2allow** te versmallen. Het volgende voorbeeld laat het gebruik van **grep** zien om alleen weigeringen te versturen naar **audit2allow** die gerelateerd zijn aan **certwatch**:

```
# grep certwatch /var/log/audit/audit.log | audit2allow -M mycertwatch2
******************** IMPORTANT ***********************
To make this policy package active, execute:

# /usr/sbin/semodule -i mycertwatch2.pp
```

Refereer naar de blog van Dan Walsh *"Using audit2allow to build policy modules. Revisited."*[13] voor meer informatie over het gebruik van **audit2allow** om tactiek modules te maken.

[13] http://danwalsh.livejournal.com/24750.html

Verdere informatie

8.1. Contributors

- *Geert Warrink*[1] (translation - Dutch)
- *Domingo Becker*[2] (translation - Spanish)
- *Daniel Cabrera*[3] (translation - Spanish)

8.2. Other Resources

De National Security Agency (NSA)

Van de NSA *Bijdragers aan SELinux*[4] pagina:

Onderzoekers in het National Information Assurance Research Laboratory (NIARL) van NSA ontwikkelden en implementeerden flexibele verplichte toegangscontrole in de belangrijkste subsystemen van de Linux kernel en implementeerden de nieuwe operating systeem onderdelen aangeboden door de Flask architectuur, namelijk de beveiligings server en de toegangs vector cache. De NSA onderzoekers pasten de op LSM gebaseerde SELinux aan om opgenomen te worden in Linux 2.6. NSA heeft ook de ontwikkeling geleid van soortgelijke controles voor het X Window systeem (XACE/XSELinux) en voor Xen (XSM/Flask).

- Hoofd SELinux website: *http://www.nsa.gov/research/selinux/index.shtml*.
- SELinux documentatie: *http://www.nsa.gov/research/selinux/docs.shtml*.
- SELinux achtergrond: *http://www.nsa.gov/research/selinux/background.shtml*.

Tresys Technology

Tresys Technology[5] is de upstream ontwikkeling voor:

- *SELinux gebruikersomgeving bibliotheken en gereedschappen*[6].
- *SELinux Referentie Tactiek*[7].

SELinux News

- Nieuws: *http://selinuxnews.org/wp/*.
- Planet SELinux (blogs): *http://selinuxnews.org/planet/*.

SELinux Project Wiki

- Hoofd pagina: *http://selinuxproject.org/page/Main_Page*.
- Gebruikers hulpbronnen, inclusief verwijzingen naar documentatie, maillijsten, websites, en gereedschappen: *http://selinuxproject.org/page/User_Resources*.

[4] http://www.nsa.gov/research/selinux/contrib.shtml
[5] http://www.tresys.com/

Red Hat Enterprise Linux

- De *Red Hat Enterprise Linux Deployment Guide*[8] bevat een SELinux *Referenties*[9] sectie, die verwijzingen heeft naar SELinux handleidingen, algemene informatie, en de technologie achter SELinux.
- De *Red Hat Enterprise Linux 4 SELinux Guide*[10].

Fedora

- Hoofd pagina: *http://fedoraproject.org/wiki/SELinux*.
- Foutoplossen: *http://fedoraproject.org/wiki/SELinux/Troubleshooting*.
- Fedora Core 5 SELinux FAQ: *http://docs.fedoraproject.org/selinux-faq-fc5/*.

De officieuze SELinux FAQ

http://www.crypt.gen.nz/selinux/faq.html

IRC

Op *Freenode*[11]:

- #selinux
- #fedora-selinux

[11] http://freenode.net/

Bijlage A. Revision History

Herziening 1.3 Tue May 12 2009 Scott Radvan *sradvan@redhat.com*

Heziening voor Fedora 11

Herziening 1.2 Mon Jan 19 2009 Murray McAllister *mmcallis@redhat.com*

Hyperlinks naar NSA websites vernieuwt

Herziening 1.1 Sat Dec 6 2008 Murray McAllister *mmcallis@redhat.com*

Red Hat Bugzilla #472986, "httpd does not write to /etc/httpd/logs/"[1] opgelost
Paragraaf toegevoegd: "6.6. Booleans for Users Executing Applications"
Kleine tekst veranderingen

Herziening 1.0 Tue Nov 25 2008 Murray McAllister *mmcallis@redhat.com*

Eerste inhoud vrijgave op *http://docs.fedoraproject.org/*