

Einführung in SELinux



FrOSCon 2011, Sankt Augustin

Robert Scheck

fedora[™] 

Robert Scheck


Fedora Package Maintainer und Provenpackager
Fedora Ambassador und Ambassador Mentor
Unterstützung der Website-/Übersetzungsteams
Open Source Contributor und Software-Entwickler

Mail: robert@fedoraproject.org

Web: <http://fedoraproject.org/wiki/RobertScheck>



Was ist SELinux?

- ▶ Security-Enhanced Linux
- ▶ Implementation des FLASK-Konzepts (Flux Advanced Security Kernel) 
- ▶ Zugriffskontrolle auf Ressourcen im Sinne von Mandatory Access Control (MAC)
- ▶ Maßgebliche Entwicklung: NSA und Red Hat
- ▶ Lizenziert unter GNU General Public License



Linux-Zugriffskontrolle

- ▶ Linux-Zugriffskontrolle kümmert sich um
 - ▶ Kernel-Überwachung
 - ▶ Prozesse (laufende Programme) und Zugriff auf
 - ▶ Ressourcen (Dateien, Verzeichnisse, Sockets, ...)
- ▶ Zum Beispiel:
 - ▶ Webserver-Prozesse können Webseiten lesen,
 - ▶ aber nicht `/etc/shadow`
- ▶ Wie werden diese Entscheidungen getroffen?



Standard-Zugriffskontrolle

- ▶ Prozesse und Dateien haben Attribute
 - ▶ Prozesse: Benutzer/Gruppe (real und effektiv)
 - ▶ Ressourcen: Benutzer/Gruppe und Zugriffsbits
 - ▶ Lesen, schreiben, ausführen für Benutzer, Gruppe, andere
- ▶ Richtlinie ist im Kernel hardcoded
- ▶ Beispiel: Kann Firefox meinen privaten SSH-Schlüssel lesen?

▶ robert 3127 1 5 10:00 ? 00:00:29 firefox

▶ -rw----- 1 robert users 993 6. Feb 2005 id_rsa



Gängige Sicherheitsprobleme

- ▶ Zugriff basiert auf Benutzerrechten
- ▶ Beispiel: Firefox kann SSH-Schlüssel lesen
 - ▶ Ist zwar normalerweise nicht der Fall, aber:
 - ▶ Wenn kompromittiert – potentiell desaströs
- ▶ Grundlegendes Problem:
 - ▶ Sicherheitsattribute nicht spezifisch genug
 - ▶ Kernel kann nicht zwischen Anwendung und Benutzer unterscheiden



Gängige Sicherheitsprobleme

- ▶ Prozesse können Sicherheitsattribute ändern
- ▶ Beispiel: E-Mail-Dateien nur für mich lesbar
 - ▶ Evolution kann diese für alle lesbar machen
- ▶ Grundlegendes Problem:
 - ▶ Benutzerbestimmbare Zugriffskontrolle, auch genannt Discretionary Access Control (DAC)
 - ▶ Prozesse können Sicherheitsrichtlinien anpassen bzw. ignorieren



Gängige Sicherheitsprobleme

- ▶ Zwei Berechtigungsstufen: Benutzer und Root
- ▶ Beispiel: Rechteausweitung im Apache
 - ▶ Apache-Fehler ermöglicht Root-Rechte
 - ▶ Gesamtes Linux-System ist kompromittiert
- ▶ Grundlegendes Problem:
 - ▶ Zu einfache Sicherheitsrichtlinie
 - ▶ Keine Möglichkeit minimale Berechtigungen zu erzwingen



Lösung: SELinux

- ▶ Zusätzliche Zugriffskontrollen durch SELinux
 - ▶ Neue Sicherheitsattribute für Prozesse/Ressourcen
 - ▶ Flexible Sicherheitsrichtlinie, die anpassbar ist
- ▶ Erzwingung durch Kernel und Anwendungen
- ▶ Adressiert Sicherheitsprobleme von Haus aus
 - ▶ Zwingend erforderlich (Mandatory Access Control, „MAC“), fein granuliert, minimale Berechtigungen
 - ▶ Kein allmächtiger Root-Benutzer
- ▶ Transparent für Anwendungen



SELinux-Zugriffskontrolle

- ▶ SELinux hat 3 Formen der Zugriffskontrolle
 - ▶ Type Enforcement (TE), primärer Mechanismus
 - ▶ Rollenbasierte Zugriffskontrolle (RBAC)
 - ▶ Multi-Level-Sicherheit (MLS)
- ▶ Konfigurierbar über Richtlinienensprache
 - ▶ Zentrale Konfigurationsdateien für alle Zugriffe
 - ▶ Verschiedene Richtlinien (targeted, mls, minimum)
- ▶ Jeder Zugriff wird standardmäßig verweigert



SELinux-Sicherheitsattribute

- ▶ Prozesse und Dateien mit Sicherheitskontext
 - ▶ `robert_u:staff_r:firefox_t:s0`
 - ▶ `robert_u:object_r:user_home_t:s0`
 - ▶ Benutzer:Rolle:Typ:Level
- ▶ Schlüsselfeld ist der Typ
 - ▶ Verwendet für Einführung von Type Enforcement
- ▶ Restliche Felder für RBAC und MLS
 - ▶ Hierzu später mehr



SELinux-Sicherheitskontext

- ▶ Verschiedene Tools für SELinux modifiziert
- ▶ „-Z“-Option üblicherweise für Kontextanzeige
- ▶ Beispiele:
 - ▶ `ps auxZ` (zeigt Kontexte von Prozessen)
 - ▶ `ls -laZ` (zeigt Kontexte von Dateien/Verzeichnissen)
- ▶ Ausgabebeispiele von „ls -Z“:
 - ▶ `----- . system_u:object_r:shadow_t:s0 /etc/shadow`
 - ▶ `-rwxr-xr-x. system_u:object_r:udev_exec_t:s0 /sbin/udev`



Einführung: Type Enforcement

- ▶ Basiert auf einzelner Sicherheitsattribut: Typ
 - ▶ Wird auf alle Prozesse & Ressourcen angewendet
 - ▶ Repräsentiert sicherheitsrelevante Informationen
- ▶ Typ wird Prozessen & Ressourcen zugewiesen
 - ▶ Apache-Prozesse → `httpd_t`
 - ▶ `/var/www/html/index.html` → `httpd_sys_content_t`
- ▶ Zugriff zwischen Typen wird erlaubt,
 - ▶ z.B. `httpd_t` kann `httpd_sys_content_t` lesen



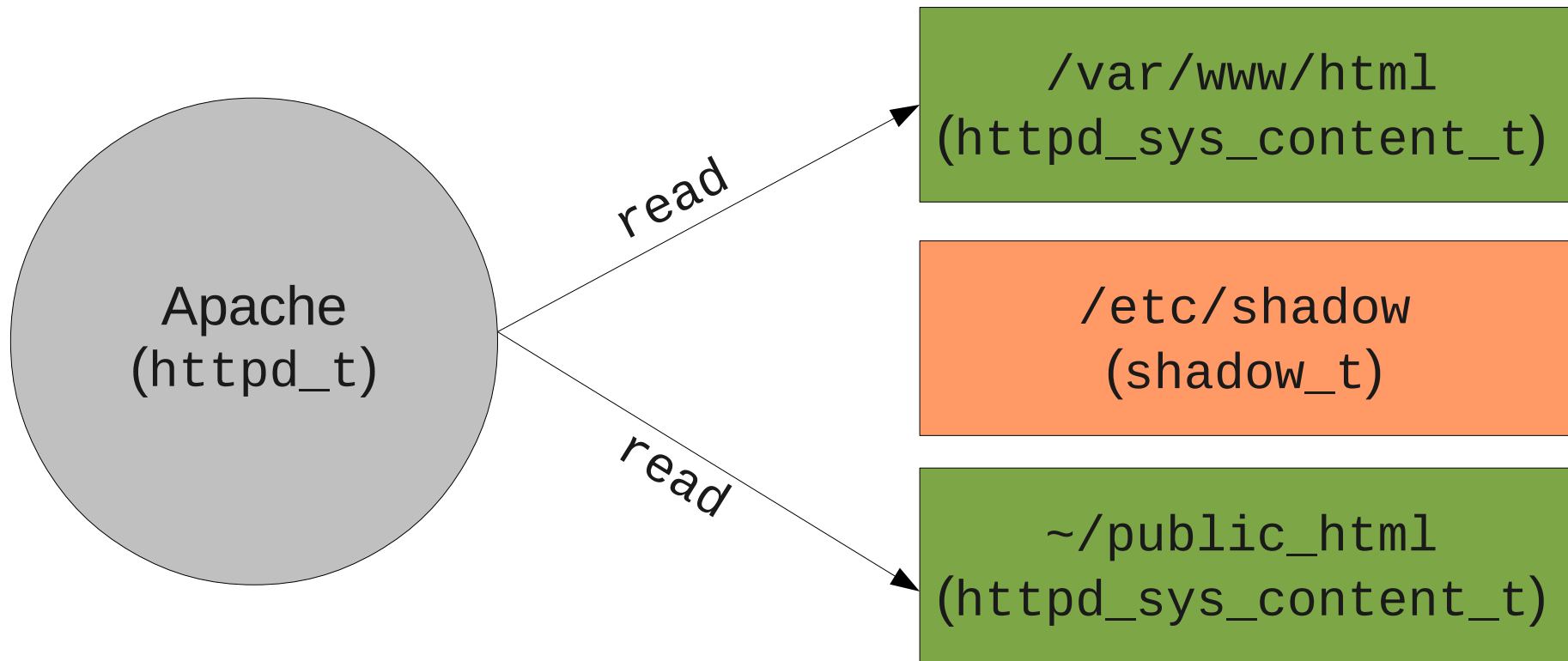
Einführung: Objektklassen

- ▶ Objektklassen spezifizieren die Zugriffsdetails
- ▶ Ressourcen werden in Klassen unterteilt,
 - ▶ z.B. `file`, `lnk_file`, `dir`, `socket`, `process`
- ▶ Jede Klasse hat Berechtigungen,
 - ▶ z.B. für `file`: `read`, `write`, `execute`, `getattr`
- ▶ Lesezugriff über Type Enforcement:
 - ▶ `allow httpd_t httpd_sys_content_t:file read;`



Übersicht: Type Enforcement

```
allow httpd_t httpd_sys_content_t:file read;
```



fedora^f

Konzept: Type Enforcement

- ▶ Zugriff wird ausschließlich über Typ erlaubt
 - ▶ Viele Prozesse & Ressourcen haben gleichen Typ
 - ▶ Vereinfacht die Richtlinien durch Gruppierung
 - ▶ Prozesse mit gleichem Typ haben gleiche Rechte
 - ▶ Gleiches gilt für Ressourcen (Dateien)
- ▶ Prozesstypen werden auch „Domains“ genannt
 - ▶ Manchmal für Ressourcen verwendet, z.B. Sockets
- ▶ Unterschiedliche Ressourcen können gleiche Typen haben



Initiale Typenzuweisung

- ▶ Dateien und Verzeichnisse:
 - ▶ Konfigurationsdatei definiert Standardkontexte
 - ▶ Sogenannte „file contexts“ (*.fc) – Dateikontexte
 - ▶ Reguläre Ausdrücke, /usr/(.*)?bin(/.*)? → bin_t
 - ▶ Vererbung vom Elternverzeichnis zur Laufzeit
- ▶ Anwendungen können Kontext explizit setzen
 - ▶ chcon – Tool für Kontextanpassungen (→ chown)
 - ▶ passwd – Verwaltet den Kontext von /etc/shadow



Zuweisung von Prozesstypen

- ▶ Prozesstypen werden
 - ▶ (standardmäßig) vom Elternprozess vererbt
 - ▶ durch Richtlinie gesetzt (Regel für Type Transition)
 - ▶ von der Anwendung gesetzt (z.B. `login`)
- ▶ Beispiele:
 - ▶ `bash (user_t) → ls (user_t)`
 - ▶ `init (init_t) → httpd-Initkript (initrc_t)`
→ `httpd (httpd_t)`
 - ▶ `login (login_t) → bash (user_t)`



Regeln für Type Transition

- ▶ Type Transition-Regeln setzen Prozesstypen:
 - ▶ Anhand Elternprozesstyp und Dateityp des Diensts
 - ▶ Ähnlich wie `setuid()`
- ▶ Beispiel: Nameserver starten
 - ▶ Richtlinien-Regel:
 - ▶ `domain_auto_trans(initrc_t, named_exec_t, named_t)`
 - ▶ Elternprozess (`initrc_t`)
 - ▶ Dateityp des Diensts (`named_exec_t`)
 - ▶ Ergebnis: `named_t`



Hinweise zur Type Transition

- ▶ Hauptgründe für das Setzen von Prozesstypen
 - ▶ Stellt sicher: Anwendung läuft in korrekter Domain
 - ▶ Benötigt keine Anpassung der Anwendung
- ▶ Muss durch Richtlinie erlaubt sein
 - ▶ z.B. Apache kann keine Prozesse in `init_t` starten
 - ▶ Verhindert Rechteauserweiterung für Anwendungen
- ▶ Bindet bestimmte ausführbare Dateien an eine bestimmte Domain
 - ▶ z.B. nur `/usr/bin/passwd` darf als `passwd_t` laufen



Benutzerfeld im Kontext

- ▶ `robert_u:staff_r:firefox_t:s0`
- ▶ Muss nicht dem Linux-Benutzer entsprechen
- ▶ Endet oft mit „_u“: `system_u`, `user_u`
- ▶ Keine Verwendung in der „targeted“-Richtlinie
- ▶ Dateien und Verzeichnisse:
 - ▶ Benutzer wird vom Prozess geerbt
 - ▶ Systemprozesse erzeugen Dateien mit dem Dateikontext `system_u`



Rollenfeld im Kontext

- ▶ `robert_u:staff_r:firefox_t:s0`
- ▶ Für rollenbasierte Zugriffskontrolle (RBAC)
 - ▶ Rolle schränkt Type Transitions weiter ein
 - ▶ Zusammen mit Type Enforcement (`user_r/user_t`)
- ▶ Endet üblicherweise mit „_r“
- ▶ Ressourcen erhalten standardmäßig `object_r`
- ▶ Verwendung in „mls“-Richtlinie
 - ▶ `user_r`, `staff_r`, `secadmin_r`



Details zum MCS-Levelfeld

- ▶ `robert_u:staff_r:firefox_t:s0`
- ▶ Verwendet für Multilevel-Sicherheit, kurz: MLS (oder für Multikategorie-Sicherheit, kurz: MCS)
- ▶ Oft unsichtbar in „targeted“-Richtlinie
- ▶ Gibt ein Level oder einen Bereich an
 - ▶ Einzelnes Level: `s0`
 - ▶ Bereich: `s0-s15:c0.c1023`
- ▶ Üblicherweise mit Labels übersetzt
 - ▶ `s15:c0.c1023` → „SystemHigh“



SELinux-Sicherheitsvorteile

- ▶ Typen für wichtige Sicherheitsinformationen
 - ▶ Zugriff basiert auf Benutzer- *und* Anwendungsfunktion bzw. -berechtigung
 - ▶ Transitions kümmern sich um Prozessketten
- ▶ Prozesse mit minimalen Berechtigungen
 - ▶ nur was für den Typ erlaubt ist
 - ▶ z.B. `httpd_t` kann nur Webseiten lesen
- ▶ Rechteauserweiterung streng kontrolliert
 - ▶ Kompromittierung des Apache durch Richtlinie begrenzt



Die „mls“-Richtlinie

- ▶ Richtlinie mit Bell-LaPadula-Unterstützung
 - ▶ Modell: Vertrauliche Informationen sollen nicht an nicht vertrauenswürdige Personen weitergegeben werden können
- ▶ Nur für Server-Betriebssysteme gedacht
 - ▶ Keine Unterstützung von X
 - ▶ Nur für bestimmte Pakete/Dienste
- ▶ Zertifizierung von Red Hat Enterprise Linux in 2007 (mit IBM) gegen LSPP, RBACPP & CAPP auf EAL 4+



Die „targeted“-Richtlinie

- ▶ Prozesse sind standardmäßig uneingeschränkt
 - ▶ Nur „targeted“ Prozesse sind eingeschränkt
- ▶ Uneingeschränkte Domains
 - ▶ Benutzerprozesse standardmäßig in `unconfined_t`
 - ▶ Systemprozesse in `initrc_t`
 - ▶ Uneingeschränkte Prozesse mit gleichem Zugriff als würden sie ohne SELinux laufen
- ▶ Dienste mit Richtlinie haben eine Transition von `unconfined_t` nach z.B. `httpd_t` (begrenzter Zugriff)



Konfigurationsdateien

▶ SELinux-Konfiguration in /etc/selinux

```
-rw-r--r--. 1 root root 458 26. Aug 2010 config
-rw-r--r--. 1 root root 2271 22. Jul 2010 semanage.conf
drwxr-xr-x. 5 root root 4096 7. Jun 01:53 mls
drwxr-xr-x. 5 root root 4096 7. Jun 01:53 targeted
```

▶ /etc/selinux/config – Richtlinie und Modus

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



Konfigurationsdateien

- ▶ contexts: Standardkontexte für das System
- ▶ modules: Module zum Bauen der Richtlinie
- ▶ policy: Kompilierte SELinux-Richtlinie
- ▶ setrans.conf: MLS/MCS-Übersetzungen
- ▶ seusers: Mapping Linux-/SELinux-Benutzer

```
$ ls -l /etc/selinux/targeted/  
drwxr-xr-x. 4 root root 4096 7. Jun 01:53 contexts  
drwxr-xr-x. 3 root root 4096 7. Jun 01:53 modules  
drwxr-xr-x. 2 root root 4096 7. Jun 01:53 policy  
-rw-r--r--. 1 root root 607 27. Mai 15:44 setrans.conf  
-rw-r--r--. 1 root root 176 7. Jun 01:53 seusers  
$
```



Boot-Parameter für den Kernel

- ▶ Kernel-Parameter übersteuern Einstellungen in `/etc/selinux/config`
- ▶ `selinux=0`
 - ▶ Startet den Kernel mit deaktiviertem SELinux
 - ▶ Alle neu erstellen Dateien haben keinen Kontext
 - ▶ Spätere SELinux-Nutzung erfordert Relabeling
- ▶ `enforcing=0`
 - ▶ Startet den Kernel im „permissive“ Modus
 - ▶ Eventuell abweichende Fehlermeldungen zu „enforced“



„man pages“ für „targeted“

```
httpd_selinux(8)      httpd Selinux Policy documentation      httpd_selinux(8)
```

```
NAME
    httpd_selinux - Security Enhanced Linux Policy for the httpd daemon

DESCRIPTION
    Security-Enhanced Linux secures the httpd server via flexible mandatory
    access control.

FILE_CONTEXTS
    SELinux requires files to have an extended attribute to define the file
    type. Policy governs the access daemons have to these files. SELinux
    httpd policy is very flexible allowing users to setup their web services
    in as secure a method as possible.

    The following file contexts types are defined for httpd:

    httpd_sys_content_t
    - Set files with httpd_sys_content_t if you want httpd_sys_script_exec_t
    scripts and the daemon to read the file, and disallow other non sys
    scripts from access.

    httpd_sys_script_exec_t
    - Set cgi scripts with httpd_sys_script_exec_t to allow them to run with
```

Modifizierte Systemwerkzeuge

- ▶ „Z“ ist die Antwort für SELinux
 - ▶ `ls -Z`
 - ▶ `id -Z`
 - ▶ `ps auxZ`
 - ▶ `lsof -Z`
 - ▶ `netstat -Z`
 - ▶ `find / -context=`



Modifizierte Systemwerkzeuge

- ▶ „Z“ ist die Antwort für SELinux
 - ▶ `ls -Z`
 - ▶ `id -Z`
 - ▶ `ps auxZ`
 - ▶ `lsof -Z`
 - ▶ `netstat -Z`
 - ▶ `find / -context=`



Modifizierte Systemwerkzeuge

- ▶ `cp`

- ▶ Erbt Kontext vom Elternverzeichnis oder setzt den Kontext basierend auf dem System-Standard
- ▶ Option „-a“ behält den ursprünglichen Kontext bei

- ▶ `mv`

- ▶ Behält weiterhin den ursprünglichen Kontext bei

- ▶ `install`

- ▶ Setzt Kontext basierend auf dem System-Standard

- ▶ Ausnahmen durch `restorecond`



SELinux-Pakete & -Werkzeuge

- ▶ libselinux mit Bibliothek für Anwendungen
- ▶ libselinux-utils
 - ▶ getenforce: Meldet enforcing/permissive/disabled
 - ▶ setenforce 0/1: Setzt permissive/enforcing
 - ▶ selinuxenabled: SELinux-Status für Skripte
 - ▶ matchpathcon: Zeigt standardmäßigen Kontext
 - ▶ avcstat: Zeigt SELinux AVC-Statistiken
- ▶ libselinux-python und libselinux-ruby
 - ▶ API-Schnittstellen



Policycoreutils

- ▶ genhomedircon, fixfiles, setfiles, chcat, restorecon, restorecond
- ▶ audit2allow, audit2why
 - ▶ SELinux AVC-Meldungen anzeigen/verstehen
- ▶ secon
 - ▶ Zeigt Kontext von Ressourcen und Programmen
- ▶ semodule, semodule_deps, semodule_link, semodule_expand, semodule_package
 - ▶ Verwaltung von Modulen



Policycoreutils

- ▶ genhomedircon, fixfiles, setfiles, chcat, restorecon, restorecond
- ▶ audit2allow, audit2why
 - ▶ SELinux AVC-Meldungen anzeigen/verstehen
- ▶ secon
 - ▶ Zeigt Kontext von Ressourcen und Programmen
- ▶ semodule, semodule_deps, semodule_link, semodule_expand, semodule_package
 - ▶ Verwaltung von Modulen



SELinux-Meldungen verstehen

- ▶ Access Vector Cache (AVC)
 - ▶ /var/log/messages (ohne auditd)
 - ▶ /var/log/audit/audit.log (mit auditd)

```
type=AVC msg=audit(1140184056.443:78): avc: denied { use } for ↵  
pid=2185 comm="mingetty" name="ptmx" dev=tmpfs ino=699 ↵  
scontext=system_u:system_r:getty_t:s0 ↵  
tcontext=system_u:system_r:kernel_t:s0 tclass=fd
```

```
type=AVC msg=audit(1166017682.366:876): avc: denied { getattr } for ↵  
pid=23768 comm="httpd" name="index.html" dev=dm0 ino=7996439 ↵  
scontext=user_u:system_r:httpd_t:s0 ↵  
tcontext=user_u:object_r:user_home_t:s0 tclass=file
```



SELinux-Meldungen verstehen

- ▶ AVC-Meldungen entstehen aus verschiedenen Gründen
 - ▶ Datei mit falschem Kontext
 - ▶ Prozess läuft unter falschem Kontext
 - ▶ Fehler in der SELinux-Richtlinie
 - ▶ Noch nie getestete Funktionalität oder Kombination wurde erstmals genutzt
 - ▶ Ein Einbruchversuch



SELinux-Meldungen verstehen

▶ audit2allow

- ▶ Werkzeug zum Erzeugen von „allow“-Regeln anhand der nicht gestatteten Zugriffe der Logs

- ▶ `audit2allow -i /var/log/audit/audit.log`

- ▶ `allow httpd_t user_home_t:file getattr;`

▶ audit2why

- ▶ Liefert eine Beschreibung für AVC-Meldungen, warum der Zugriff nicht gestattet wurde

- ▶ Begrenzt hilfreich für Einsteiger, eher benutzt von Richtlinien-Entwicklern



AVC-Meldungen analysieren

- ▶ AVC-Meldungen für Dateien mit `*:file_t`
 - ▶ Schwerwiegendes Kontextproblem
 - ▶ Datei wurde während `selinux=0` angelegt
 - ▶ Relabeling des Dateisystems durchführen
 - ▶ `touch /.autorelabel; reboot`
 - ▶ Neue Festplatte? `restorecon -R -v /<mnt>`
- ▶ AVC-Meldungen, die `default_t` enthalten
 - ▶ Vermutlich ein Labeling-Problem
 - ▶ Relabeling mit `chcon` oder s.o.



AVC-Meldungen analysieren

- ▶ Viele ähnliche Nachrichten zur gleichen Datei
 - ▶ Typischerweise ein Labeling-Problem
 - ▶ Beispiel:
 - ▶ Datei `/home/robert/resolv.conf` anlegen
 - ▶ `mv /home/robert/resolv.conf /etc/`
 - ▶ `ls -lZ /etc/resolv.conf`
 - ▶ Eingeschränkte Domains melden Zugriffsfehler auf `user_home_t`
 - ▶ `restorecon /etc/resolv.conf`



SELinux-Troubleshoot-Tool

- ▶ setroubleshoot
 - ▶ Dienst wartet AVC-Meldungen des Audit-Diensts
 - ▶ Plugin-Datenbank für bekannte Probleme
 - ▶ `/usr/share/setroubleshoot/plugins/`
 - ▶ Zeigt mögliche Abhilfemöglichkeiten/Lösungen
 - ▶ `sealert` startet Browser bzw. Protokollanalyse
 - ▶ E-Mail-Benachrichtigung durch Konfiguration
 - ▶ `/etc/setroubleshoot/setroubleshoot.conf`



SELinux Alert Browser

SELinux has detected a problem. Would you like to receive alerts? Yes No

The source process: smbd Di Aug 16, 2011 23:19 CEST
 Attempted this access: read
 On this directory: privat

Troubleshoot

If you were trying to...	Then this is the solution.
If you want to allow samba to share any file/directory read only.	You must tell SELinux about this by enabling the 'samba_export_all_ro' SELinux boolean. <pre>setsebool -P samba_export_all_ro 1</pre> <input type="button" value="Plugin Details"/>
If you want to allow samba to share any file/directory read/write.	You must tell SELinux about this by enabling the 'samba_export_all_rw' SELinux boolean. <pre>setsebool -P samba_export_all_rw 1</pre> <input type="button" value="Plugin Details"/>
If you want to allow want to treat privat as public content	You need to change the label on privat to public_content_t or public_content_t. <pre># semanage fcontext -a -t public_content_t 'privat' # restorecon -v 'privat'</pre> <input type="button" value="Plugin Details"/>
If you believe that smbd should be allowed read access on the privat directory by default.	You should report this as a bug. You can generate a local policy module to allow this access. Allow this access for now by executing: <pre># grep smbd /var/log/audit/audit.log audit2allow -M mypol # semodule -i mypol.pp</pre> <input type="button" value="Plugin Details"/> <input type="button" value="Report Bug"/>

Alert 1 of 2



Fehlende AVC-Meldungen

- ▶ Keine AVC-Meldung bei Anwendungsfehler
 - ▶ Mit `setenforce 0` versuchen – funktioniert es?
- ▶ `dontaudit`-Regeln verhindern AVC-Meldungen
- ▶ Fedora 14+ und Red Hat Enterprise Linux 6
 - ▶ `semodule -DB # --disable_dontaudit --build`
- ▶ Red Hat Enterprise Linux 5
 - ▶ `semodule -b /usr/share/selinux/targeted/enableaudit.pp`
 - ▶ `semodule -b /usr/share/selinux/targeted/base.pp`



Dateikontexte verwalten

- ▶ chcon
 - ▶ Grundlegendes Werkzeug für Kontextänderungen
 - ▶ `chcon -R -t httpd_sys_script_rw_t \`
`/var/www/myapp/data`
 - ▶ `chcon -t httpd_sys_script_t \`
`/var/www/cgi-bin/myapp`
 - ▶ Aufbau an `chmod` angelehnt
 - ▶ Anpassbare Typen: Kein Relabeling
 - ▶ `/etc/selinux/targeted/contexts/customizable_types`
- ▶ `touch /.autorelabel; reboot`
- ▶ Vollständiges Relabeling



Dateikontexte verwalten

- ▶ `restorecon`
 - ▶ Setzt eine Datei auf den Standardkontext zurück
 - ▶ Arbeitet auf Verzeichnis-/Dateiebene
- ▶ `setfiles`
 - ▶ Für System-Initialisierung, auf Dateisystemebene
 - ▶ Erwartet `file_contexts`-Datei als Spezifikation
- ▶ `fixfiles`
 - ▶ Shellskript-Wrapper um `setfiles` und `restorecon`
 - ▶ RPM-Name als Argument für Relabeling der Dateien im Paket



Dateikontexte verwalten

- ▶ `matchpathcon`
 - ▶ Zeigt Standardkontext der Ressource
- ▶ `semanage`
 - ▶ Standardkontext für Ressourcen anzeigen/ändern
 - ▶ Verwendet reguläre Ausdrücke für Pfadangaben
 - ▶ Viele weitere Funktionalitäten
- ▶ `system-config-selinux`
 - ▶ Grafisches Frontend für viele CLI-Werkzeuge
 - ▶ Etwa `semanage`-Funktionalität



SELinux-Booleans

- ▶ Booleans für if/else-Anweisungen in Richtlinie
- ▶ Konfiguration ohne Richtlinie zu bearbeiten
- ▶ `getsebool`
 - ▶ `getsebool -a`
- ▶ `setsebool`
 - ▶ `setsebool -P -allow=[1|0]`
- ▶ `system-config-selinux`
- ▶ Aktiviert/deaktiviert Teile der Richtlinie
 - ▶ `setsebool -P virt_use_usb 1`



Auswählen:

Status

Boolesche Variablen

Dateikennzeichnung

Benutzerzuordnung

SELinux-Benutzer

Netzwerk-Port

Richtlinienmodul

Prozess-Domainn



Zurücksetzen Angepasst Sperren...

Filter

Active	Module	Description	Name
<input type="checkbox"/>	abrt	Allow ABRT to modify public files used for public file tr	abrt_anon_write
<input type="checkbox"/>	apache	Allow httpd scripts and modules execmem/execstack	httpd_execmem
<input type="checkbox"/>	apache	Allow Apache to execute tmp content.	httpd_tmp_exec
<input type="checkbox"/>	apache	Allow httpd to access nfs file systems	httpd_use_nfs
<input type="checkbox"/>	apache	Allow httpd to read user content	httpd_read_user_c
<input checked="" type="checkbox"/>	apache	Unify HTTPD to communicate with the terminal. Need	httpd_tty_comm
<input type="checkbox"/>	apache	Allow HTTPD scripts and modules to connect to the ne	httpd_can_network
<input checked="" type="checkbox"/>	apache	Allow httpd to use built in scripting (usually php)	httpd_builtin_scrip
<input type="checkbox"/>	apache	Unify HTTPD handling of all content files.	httpd_unified
<input type="checkbox"/>	apache	Allow httpd to access cifs file systems	httpd_use_cifs
<input checked="" type="checkbox"/>	apache	Allow Apache to communicate with avahi service via	httpd_dbus_avahi
<input type="checkbox"/>	apache	Allow apache scripts to write to public content. Direc	allow_httpd_sys_sc
<input type="checkbox"/>	apache	Allow httpd to read home directories	httpd_enable_hom
<input type="checkbox"/>	apache	Allow Apache to modify public files used for public file	allow_httpd_anon_
<input type="checkbox"/>	apache	Allow Apache to use mod_auth_pam	allow_httpd_mod_a
<input checked="" type="checkbox"/>	apache	Allow httpd to execute cgi scripts	httpd_enable_cgi
<input type="checkbox"/>	apache	Allow httpd to run gpg in gpg-web domain	httpd_use_gpg
<input type="checkbox"/>	apache	Allow HTTPD scripts and modules to connect to datab	httpd_can_network
<input type="checkbox"/>	apache	Allow httpd to act as a relay	httpd_can_network
<input type="checkbox"/>	apache	Allow http daemon to send mail	httpd_can_sendma

SELinux-Module

- ▶ Modulare Richtlinie
 - ▶ Modulkonzept seit Fedora Core 5
- ▶ `semodule`-Befehl:
 - ▶ Kopiert „Policy Package“ (* .pp) ins Verzeichnis `/etc/selinux/targeted/modules/active/modules`
 - ▶ Kompiliert alle installierten * .pp-Dateien als Datei `/etc/selinux/targeted/policy/policy.24`
 - ▶ Erzeugt die Dateien `file_context` und `file_context.homedirs` neu
 - ▶ Lädt eine neue Richtlinie



SELinux-Module

- ▶ `semodule-Befehl:`
 - ▶ `semodule -l`
 - ▶ Zeigt alle zur Zeit geladenen SELinux-Module
 - ▶ `semodule -i /usr/share/selinux/targeted/gpg.pp`
 - ▶ `semodule -i meinmodul.pp`
 - ▶ Lädt (installiert) ein „Policy Package“
 - ▶ `semodule -r meinmodul`
 - ▶ Entlädt (entfernt) ein „Policy Package“



Auswählen:

- Status
- Boolesche Variablen
- Dateikennzeichnung
- Benutzerzuordnung
- SELinux-Benutzer
- Netzwerk-Port
- Richtlinienmodul**
- Prozess-Domainn



Neu



Hinzufügen



Entfernen



Audit aktivieren

Filter

Modulname ▾ | Version

abrt	1.1.1
accountsd	1.0.0
ada	1.4.0
afs	1.6.1
aiccu	1.0.0
aide	1.5.0
aisexec	1.0.0
ajaxterm	1.0.0
amanda	1.12.1
amavis	1.11.0
amtu	1.2.0
apache	2.2.0
apcupsd	1.7.0
arpwatch	1.9.1
asterisk	1.8.0
audioentropy	1.6.0
automount	1.13.0
avahi	1.12.0
awstats	1.2.1
bind	1.11.0
.....	---



Richtlinien-Module erstellen

- ▶ Richtlinien-Modul besteht aus 3 Dateien
 - ▶ Type Enforcement-Datei (* .te)
 - ▶ Enthält allow-Regeln und Interface-Aufrufe, die mit der eingeschränkten Domain zusammenhängen
 - ▶ File Context-Datei (* .fc)
 - ▶ Enthält alle Ressourcen-Labels des Moduls
 - ▶ Interface-Datei (* .if)
 - ▶ Enthält alle Schnittstellen, für andere Domains zur Interaktion mit dieser eingeschränkten Domain
 - ▶ DOMAIN_domtrans,
DOMAIN_read_config



Richtlinie mit audit2allow

- ▶ Für kleine Anpassungen an der Richtlinie
- ▶

```
grep httpd_t /var/log/audit/audit.log \  
| audit2allow -M mypolicy
```

 - ▶ Erstellt eine *.te-Datei und kompiliert diese in eine *.pp-Binärdatei
- ▶

```
semodule -i mypolicy.pp
```



SELinux-Systeme verwalten

- ▶ semanage-Framework seit Fedora Core 5
- ▶ Vermeidet viele eigene Richtlinien/Module
- ▶ Beispiel:
 - ▶ Ohne semanage-Framework:
 - ▶ Apache erlauben sich an Port 81 zu binden
 - ▶ Benötigt eine eigene Richtlinie/Modul
 - ▶ Mit semanage-Framework:
 - ▶ `semanage port -a -t http_port_t -P tcp 81`



semanage-Befehle/-Parameter

- ▶ SELinux-Benutzer
 - ▶ `semanage user -l`
 - ▶ `semanage user -a guest_u`
- ▶ Linux- auf SELinux-Benutzer-Zuordnung
 - ▶ `semanage login -a -s guest_u robert`
- ▶ Dateikontexte
 - ▶ `semanage fcontext -a -t \
httpd_bugzilla_script_exec_t \
'/usr/share/bugzilla/cgi(/.*)?'`



Auswählen:

- Status
- Boolesche Variablen
- Dateikennzeichnung
- Benutzerzuordnung
- SELinux-Benutzer
- Netzwerk-Port**
- Richtlinienmodul
- Prozess-Domainn

Hinzufügen
 Eigenschaften
 Löschen
 Listenansicht
 Angepasst

Filter

SELinux Port-Typ	Protokoll	Port
afs_bos_port_t	udp	7007
afs_client_port_t	udp	7001
afs_fs_port_t	tcp	2040
afs_fs_port_t	udp	7000, 7005
afs_ka_port_t	udp	7004
afs_pt_port_t	udp	7002
afs_vl_port_t	udp	7003
agentx_port_t	tcp	705
agentx_port_t	udp	705
ajaxterm_port_t	tcp	8022
amanda_port_t	tcp	10080-10083
amanda_port_t	udp	10080-10082
amavisd_recv_port_t	tcp	10024
amavisd_send_port_t	tcp	10025
amqp_port_t	tcp	5671-5672
amqp_port_t	udp	5671-5672
aol_port_t	tcp	5190-5193
aol_port_t	udp	5190-5193
apcupsd_port_t	tcp	3551
apcupsd_port_t	udp	3551



Datei Hilfe

Auswählen:

- Status
- Boolesche Variablen
- Dateikennzeichnung
- Benutzerzuordnung
- SELinux-Benutzer
- Netzwerk-Port**
- Richtlinienmodul
- Prozess-Domainn

Hinzufügen
 Eigenschaften
 Löschen
 Gruppenansicht
 Angepasst

Filter

SELinux Port-Typ	Protokoll	MLS/MCS-Stufe	Port
http_cache_port_t	udp	s0	3130
http_cache_port_t	tcp	s0	8080
http_cache_port_t	tcp	s0	8118
http_cache_port_t	tcp	s0	10001-10010
http_port_t	tcp	s0	80
http_port_t	tcp	s0	443
http_port_t	tcp		
http_port_t	tcp		
http_port_t	tcp		
http_port_t	tcp		
i18n_input_port_t	tcp		
imaze_port_t	tcp		
imaze_port_t	udp		
inetd_child_port_t	udp		
inetd_child_port_t	tcp		
inetd_child_port_t	udp	s0	13
inetd_child_port_t	tcp	s0	13
inetd_child_port_t	udp	s0	9
inetd_child_port_t	tcp	s0	9

Modify Netzwerk-Po

Port-Nummer

Protokoll

SELinux-Typ

MLS/MCS-Stufe

Auditierung

- ▶ Audit-System empfängt SELinux-Ereignisse
- ▶ Kein auditd-Prozess
 - ▶ AVCs in `/var/log/messages` und `dmesg`
- ▶ Mit auditd-Prozess
 - ▶ AVCs in `/var/log/audit/audit.log`
- ▶ Vollständige Auditierung mit Kernel-Parameter
 - ▶ `audit=1`



LSPP, CAPP & RBAC: EAL 4+

- ▶ Labeled Security Protection Profile (LSPP)
 - ▶ Schutzprofil mit MLS/MCS und MAC (→ B1)
- ▶ Controlled Access Protection Profile (CAPP)
 - ▶ Schutzprofil mit Benutzern/Authentifikation (→ C1)
- ▶ Role-Based Access Control (RBACPP)
 - ▶ Schutzprofil mit rollenbasierter Zugriffskontrolle
- ▶ Evaluation Assurance Level (EAL 4+)
 - ▶ Level der Tests und Dokumentation
 - ▶ Methodisch entwickelt, getestet und durchgesehen



aureport

- ▶ Zusammenfassung aus Audit-Protokollen
 - ▶ `-a` – Bericht über AVC-Meldungen
 - ▶ `-i` – Numerische Felder menschenlesbar darstellen
 - ▶ `-ts „Startzeit“ -te „Endzeit“`
 - ▶ `aureport -a -ts 1:00:00`
- ▶ `--success / --failed` – Standard ist beides
- ▶ `--summary` – Gesamtergebnis der Ereignisse



ausearch

- ▶ Protokolle des Audit-Diensts durchsuchen
 - ▶ `-m avc` – Ereignistyp, z.B. AVC-Meldungen
 - ▶ `-ts` – Startzeit der Suche
 - ▶ `-x` – Ausführbare Datei
 - ▶ `ausearch -m avc -ts 1:00:00 -x named`



Fazit

- ▶ SELinux
 - ▶ einfach benutzen
 - ▶ bitte nicht deaktivieren
 - ▶ schützt wirklich vor Systemeintrüben
 - ▶ ist kostenlose Sicherheit auf NSA-Niveau



fedora^f

Weiterführende Ressourcen

- ▶ Informationen
 - ▶ <http://www.nsa.gov/research/selinux>
 - ▶ http://docs.fedoraproject.org/en-US/Fedora/13/html/Security-Enhanced_Linux/ ←
 - ▶ <http://fedoraproject.org/wiki/SELinux>
- ▶ Mailinglisten
 - ▶ selinux@tycho.nsa.gov
 - ▶ selinux@lists.fedoraproject.org



Fragen?



fedora™

Vielen Dank!

