SELinux

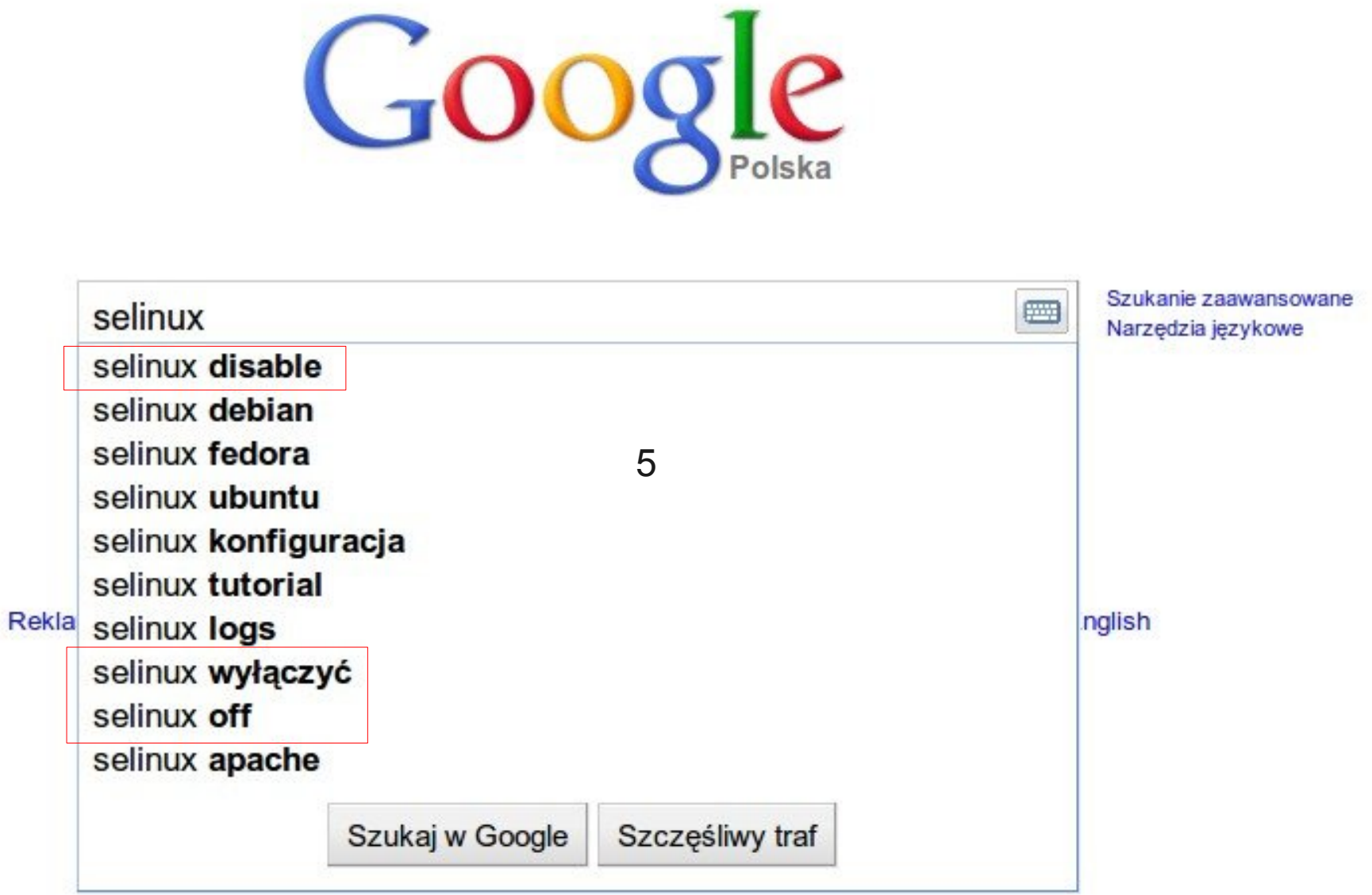
Robert Jaroszuk
<zim@iq.pl>

Zimowisko TLUG, 2011.

Wstęp – Co ludzie chcą wiedzieć o SELinux

- Co ludzie chcą wiedzieć o SELinux?

Wstęp – Co ludzie chcą wiedzieć o SELinux



5

Wstęp – Kontrola dostępu w Linuksie

- Kontrola dostępu w Linuksie:

Kontrola dostępu w Linuksie:

- Kernel
- Proces (program)
- Zasób (plik, katalog, socket, etc.)
- (Obok tego istnieją mechanizmy kontroli dostępu wewnątrz aplikacji)

Wstęp – Kontrola dostępu w Linuksie

Kontrola dostępu w Linuksie:

- Kernel
- Proces (program)
- Zasób (plik, katalog, socket, etc.)
- (Obok tego istnieją mechanizmy kontroli dostępu wewnątrz aplikacji)

Przykład:

- Serwer www może czytać pliki w /var/www i pisać do /var/log
- Nie może czytać /etc/shadow

Wstęp – Kontrola dostępu w Linuksie

Kontrola dostępu w Linuksie:

- Kernel
- Proces (program)
- Zasób (plik, katalog, socket, etc.)
- (Obok tego istnieją mechanizmy kontroli dostępu wewnątrz aplikacji)

Przykład:

- Serwer www może czytać pliki w /var/www i pisać do /var/log
- Nie może czytać /etc/shadow

Jak system podejmuje decyzję o dostępie do zasobu?

Wstęp – Kontrola dostępu w Linuksie

- Procesy i zasoby posiadają atrybuty:
  - procesy: użytkownik + grupa (rzeczywiste i efektywne)
  - zasoby: użytkownik + grupa + atrybuty dostępu (rwx)

Wstęp – Kontrola dostępu w Linuksie

- Procesy i zasoby posiadają atrybuty:
  - procesy: użytkownik + grupa (rzeczywiste i efektywne)
  - zasoby: użytkownik + grupa + atrybuty dostępu (rwx)

- Kernel wyposażony jest w stosowną politykę dostępu.

Wstęp – Kontrola dostępu w Linuksie

- Procesy i zasoby posiadają atrybuty:
  - procesy: użytkownik + grupa (rzeczywiste i efektywne)
  - zasoby: użytkownik + grupa + atrybuty dostępu (rwx)
- Kernel wyposażony jest w stosowną politykę dostępu.
- Przykład:

```
-r-------- 1 root root 1361 Dec 24 09:04 /etc/shadow
root      1769  0.0  0.0   5380  1184 ?        Ss    2010   0:01 crond
```

Wstęp – Kontrola dostępu w Linuksie

- Procesy i zasoby posiadają atrybuty:
  - procesy: użytkownik + grupa (rzeczywiste i efektywne)
  - zasoby: użytkownik + grupa + atrybuty dostępu (rwx)
- Kernel wyposażony jest w stosowną politykę dostępu.
- Przykład:

```
-r-------- 1 root root 1361 Dec 24 09:04 /etc/shadow
root      1769  0.0  0.0   5380  1184 ?        Ss    2010   0:01 crond
```

Czy crond może czytać /etc/shadow ?

- **Wszystko bazuje na uprawnieniach użytkownika**

- **Wszystko bazuje na uprawnieniach użytkownika**
  - crond może czytać /etc/shadow

- **Wszystko bazuje na uprawnieniach użytkownika**
  - crond może czytać /etc/shadow
  - (oczywiście nie ma ku temu powodu)

- **Wszystko bazuje na uprawnieniach użytkownika**
  - crond może czytać /etc/shadow
  - (oczywiście nie ma ku temu powodu)
  - (ale gdy zostanie skutecznie zaatakowany – MOŻE).

- **Wszystko bazuje na uprawnieniach użytkownika**
  - crond może czytać /etc/shadow
  - (oczywiście nie ma ku temu powodu)
  - (ale gdy zostanie skutecznie zaatakowany – MOŻE).
- Podstawowy problem

Wstęp – Typowe problemy z bezpieczeństwem w systemie Linux

- **Wszystko bazuje na uprawnieniach użytkownika**
  - crond może czytać /etc/shadow
  - (oczywiście nie ma ku temu powodu)
  - (ale gdy zostanie skutecznie zaatakowany – MOŻE).
- Podstawowy problem
  - atrybuty bezpieczeństwa nie są wystarczająco precyzyjne

- **Wszystko bazuje na uprawnieniach użytkownika**
  - crond może czytać /etc/shadow
  - (oczywiście nie ma ku temu powodu)
  - (ale gdy zostanie skutecznie zaatakowany – MOŻE).
- Podstawowy problem
  - atrybuty bezpieczeństwa nie są wystarczająco precyzyjne
  - (nawet chattr nam tutaj nie pomoże...)

Wszystko bazuje na uprawnieniach użytkownika

- crond może czytać /etc/shadow
- (oczywiście nie ma ku temu powodu)
- (ale gdy zostanie skutecznie zaatakowany – MOŻE).

Podstawowy problem

- atrybuty bezpieczeństwa nie są wystarczająco precyzyjne
- (nawet chattr nam tutaj nie pomoże...)
- kernel nie odróżnia aplikacji od użytkownika...

- **Procesy mogą zmienić uprawnienia do zasobów.**

- **Procesy mogą zmienić uprawnienia do zasobów.**
- Przykład:
  - /etc/shadow – do odczytu tylko dla root

- **Procesy mogą zmienić uprawnienia do zasobów.**
- Przykład:
  - /etc/shadow – do odczytu tylko dla root
  - crond może zrobić:

```
chmod("/etc/shadow", 0666);
```

- **Procesy mogą zmienić uprawnienia do zasobów.**
- Przykład:
  - /etc/shadow – do odczytu tylko dla root
  - crond może zrobić:

```
chmod("/etc/shadow", 0666);
```

- Podstawowy problem
  - podstawowy mechanizm bezpieczeństwa to tylko DAC

- **Procesy mogą zmienić uprawnienia do zasobów.**
- Przykład:
  - /etc/shadow – do odczytu tylko dla root
  - crond może zrobić:

```
chmod("/etc/shadow", 0666);
```

- Podstawowy problem
  - podstawowy mechanizm bezpieczeństwa to tylko DAC
  - użytkownik może wszystko (w obrębie swoich zasobów)

- **Procesy mogą zmienić uprawnienia do zasobów.**
- Przykład:
  - /etc/shadow – do odczytu tylko dla root
  - crond może zrobić:

```
chmod("/etc/shadow", 0666);
```

- Podstawowy problem
  - podstawowy mechanizm bezpieczeństwa to tylko DAC
  - użytkownik może wszystko (w obrębie swoich zasobów)
  - aplikacja to też... użytkownik (i też może zrobić wszystko).

- **Tylko dwa typy poziomów dostępu: user i root**

- **Tylko dwa typy poziomów dostępu: user i root**
- Przykład:
  - skuteczny atak na apache umożliwia uzyskanie rootshella,

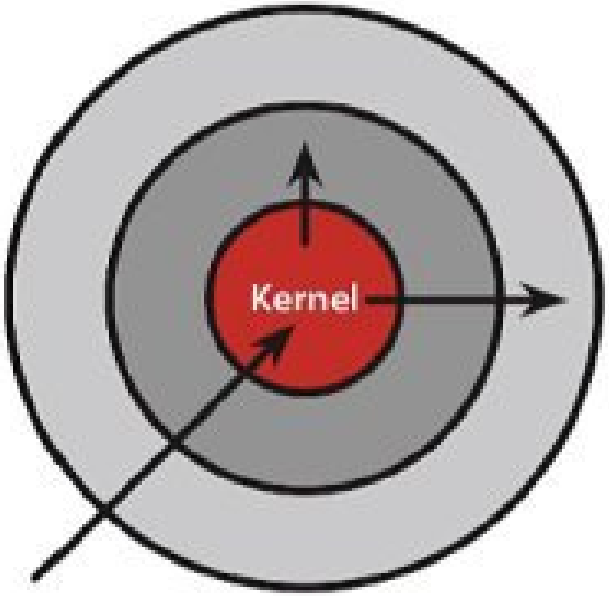
- **Tylko dwa typy poziomów dostępu: user i root**
- Przykład:
  - skuteczny atak na apache umożliwia uzyskanie rootshella,
  - następuje kompromitacja całego systemu.

- **Tylko dwa typy poziomów dostępu: user i root**
- Przykład:
  - skuteczny atak na apache umożliwia uzyskanie rootshella,
  - następuje kompromitacja całego systemu.
- Podstawowy problem
  - niewystarczające polityki bezpieczeństwa

- **Tylko dwa typy poziomów dostępu: user i root**
- Przykład:
  - skuteczny atak na apache umożliwia uzyskanie rootshella,
  - następuje kompromitacja całego systemu.

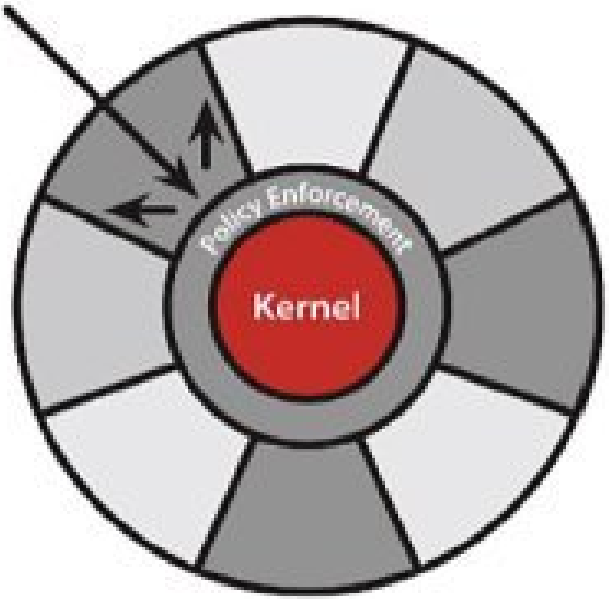
- Podstawowy problem
  - niewystarczające polityki bezpieczeństwa
  - nie ma możliwości wymuszenia podejścia least-privilege

Wstęp – DAC vs MAC



Discretionary Access Control

Once a security exploit gains access to priveleged system component, the entire system is compromised.



Mandatory Access Control

Kernel policy defines application rights, firewalling applications from compromising the entire system.

(RedHat Magazine)

SELinux - wstęp

Security-Enhanced Linux (SELinux) is a Linux feature that provides a mechanism for supporting access control security policies, including United States Department of Defense style mandatory access controls, through the use of Linux Security Modules (LSM) in the Linux kernel. It is not a Linux distribution, but rather a set of modifications that can be applied to Unix-like operating system kernels, such as Linux and that of BSD.

– (Wikipedia)

Czym jest SELinux ?

- SELinux dostarcza dodatkowe mechanizmy kontroli

Czym jest SELinux ?

- SELinux dostarcza dodatkowe mechanizmy kontroli
  - nowe atrybuty bezpieczeństwa dla procesów i zasobów

Czym jest SELinux ?

- SELinux dostarcza dodatkowe mechanizmy kontroli
  - nowe atrybuty bezpieczeństwa dla procesów i zasobów
  - elastyczna polityka bezpieczeństwa

Czym jest SELinux ?

- SELinux dostarcza dodatkowe mechanizmy kontroli
  - nowe atrybuty bezpieczeństwa dla procesów i zasobów
  - elastyczna polityka bezpieczeństwa
- Polityki bezpieczeństwa egzekwowane z poziomu jądra i aplikacji

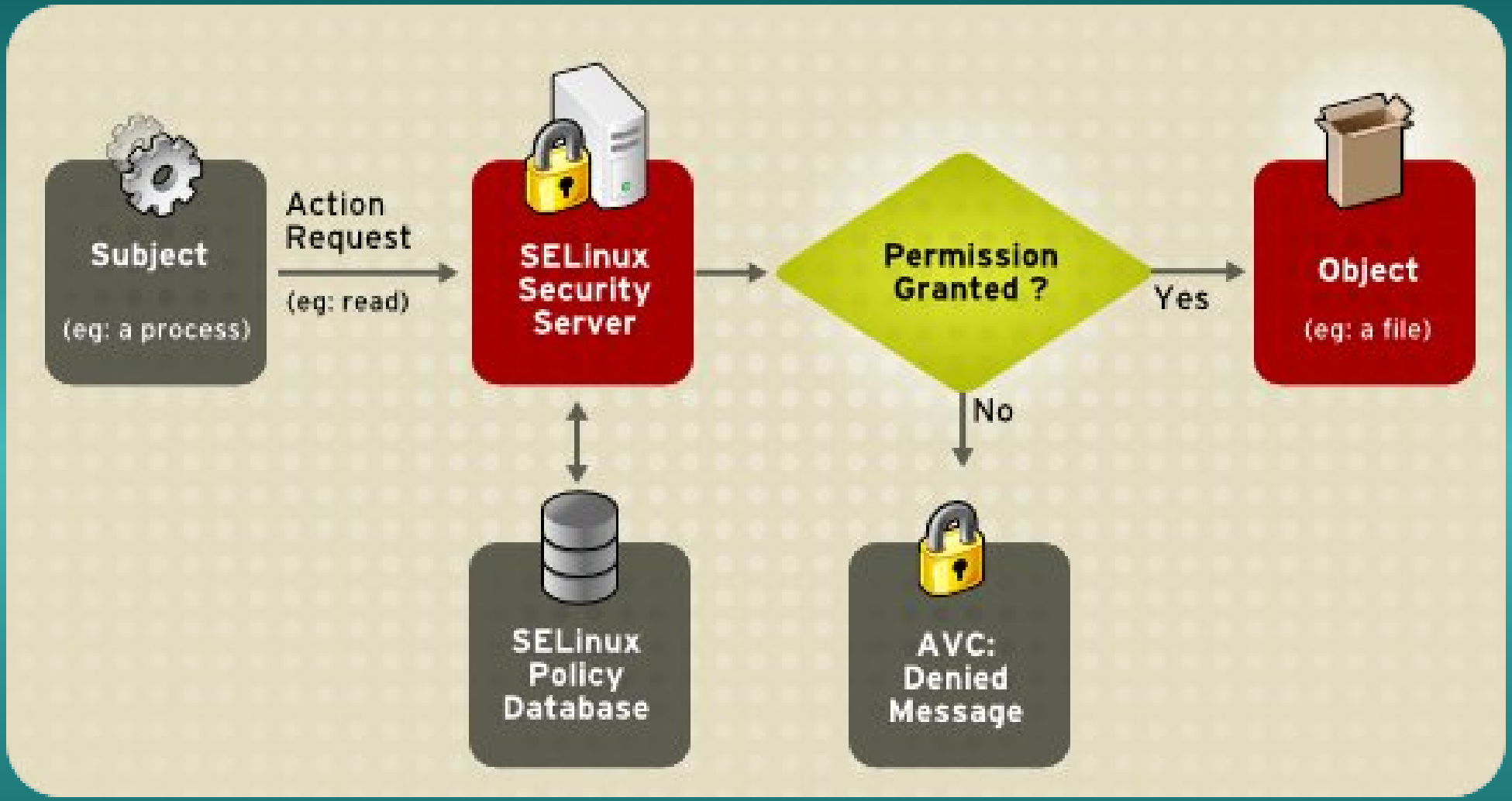
Czym jest SELinux ?

- SELinux dostarcza dodatkowe mechanizmy kontroli
  - nowe atrybuty bezpieczeństwa dla procesów i zasobów
  - elastyczna polityka bezpieczeństwa
- Polityki bezpieczeństwa egzekwowane z poziomu jądra i aplikacji
- Koncentruje się na problemach bezpieczeństwa
  - obowiązkowe przestrzeganie polityk
  - podejście typu „least-privilege”
  - precyzyjne polityki
  - koniec z rootem który może wszystko

Czym jest SELinux ?

- SELinux dostarcza dodatkowe mechanizmy kontroli
  - nowe atrybuty bezpieczeństwa dla procesów i zasobów
  - elastyczna polityka bezpieczeństwa
- Polityki bezpieczeństwa egzekwowane z poziomu jądra i aplikacji
- Koncentruje się na problemach bezpieczeństwa
  - obowiązkowe przestrzeganie polityk
  - podejście typu „least-privilege”
  - precyzyjne polityki
  - koniec z rootem który może wszystko
- SELinux jest transparentny dla aplikacji

Jak SELinux podejmuje decyzje?



(http://www.centos.org/docs/5/html/Deployment_Guide-en-US/ch-selinux.html)

SELinux – kontrola dostępu

- SELinux ma trzy formy kontroli dostępu:
- TE – Type Enforcement – mechanizm podstawowy
- RBAC – Role-Based Access Control
- MLS – Multi Level Security

SELinux – kontrola dostępu

- SELinux ma trzy formy kontroli dostępu:
  - TE – Type Enforcement – mechanizm podstawowy
  - RBAC – Role-Based Access Control
  - MLS – Multi Level Security
- Dostępne jest kilka różnych polityk
  - targeted
  - strict
  - mls

SELinux – kontrola dostępu

- SELinux ma trzy formy kontroli dostępu:
  - TE – Type Enforcement – mechanizm podstawowy
  - RBAC – Role-Based Access Control
  - MLS – Multi Level Security
- Dostępne jest kilka różnych polityk
  - targeted
  - strict
  - mls
- Wszystko jest zabronione, chyba że dozwolone

SELinux – security context

- Pliki i procesy posiadają tzw. security context:

```
system_u:system_r:syslogd_t:s0
system_u:object_r:shadow_t:s0
user:role:type:level
```

SELinux – security context

- Pliki i procesy posiadają tzw. security context:

```
system_u:system_r:syslogd_t:s0
system_u:object_r:shadow_t:s0
user:role:type:level
```

- Security context pozwala stwierdzić typ procesu/zasobu i zastosować odpowiednie polityki.

SELinux – security context

- Pliki i procesy posiadają tzw. security context:

```
system_u:system_r:syslogd_t:s0
system_u:object_r:shadow_t:s0
user:role:type:level
```

- Security context pozwala stwierdzić typ procesu/zasobu i zastosować odpowiednie polityki.

- Parametr „-Z” pozwala na podgląd security contextu
  - ps auxZ
  - ls -laZ
  - id -Z

system_u:system_r:syslogd_t:s0

- system_u → user (niezależny od tych w systemie)
- nazwy userów kończą się _u: system_u, user_u
- nie używa się ich w targeted policy
- user jest dziedziczony z procesu parenta

(na przykład plik utworzony przez proces system_u:x_r:y_t:s0 będzie utworzony jako system_u:....)

SELinux – security context

system_u:**system_r**:syslogd_t:s0

- system_r → rola
- używane do RBAC

(type transitions ograniczane w oparciu o role)

- nazwy ról kończą się _r: system_r, object_r
- wykorzystywane w politykach strict i MLS:

user_r, staff_r, secadm_r itp.

system_u:system_r:syslogd_t:**s0**

- używane do MLS i MCS

(type transitions ograniczane w oparciu o role)

- nazwy ról kończą się _r: system_r, object_r
- wykorzystywane w politykach strict i MLS:

user_r, staff_r, secadm_r itp.

Nie będziemy się tym dzisiaj zajmować.

system_u:system_r:**syslogd_t**:s0

SELinux – Type Enforcement

- Type Enforcement opiera się na... typie ;-)

SELinux – Type Enforcement

- Type Enforcement opiera się na... typie ;-)
  - typy mogą być nadawane zasobom i procesom

SELinux – Type Enforcement

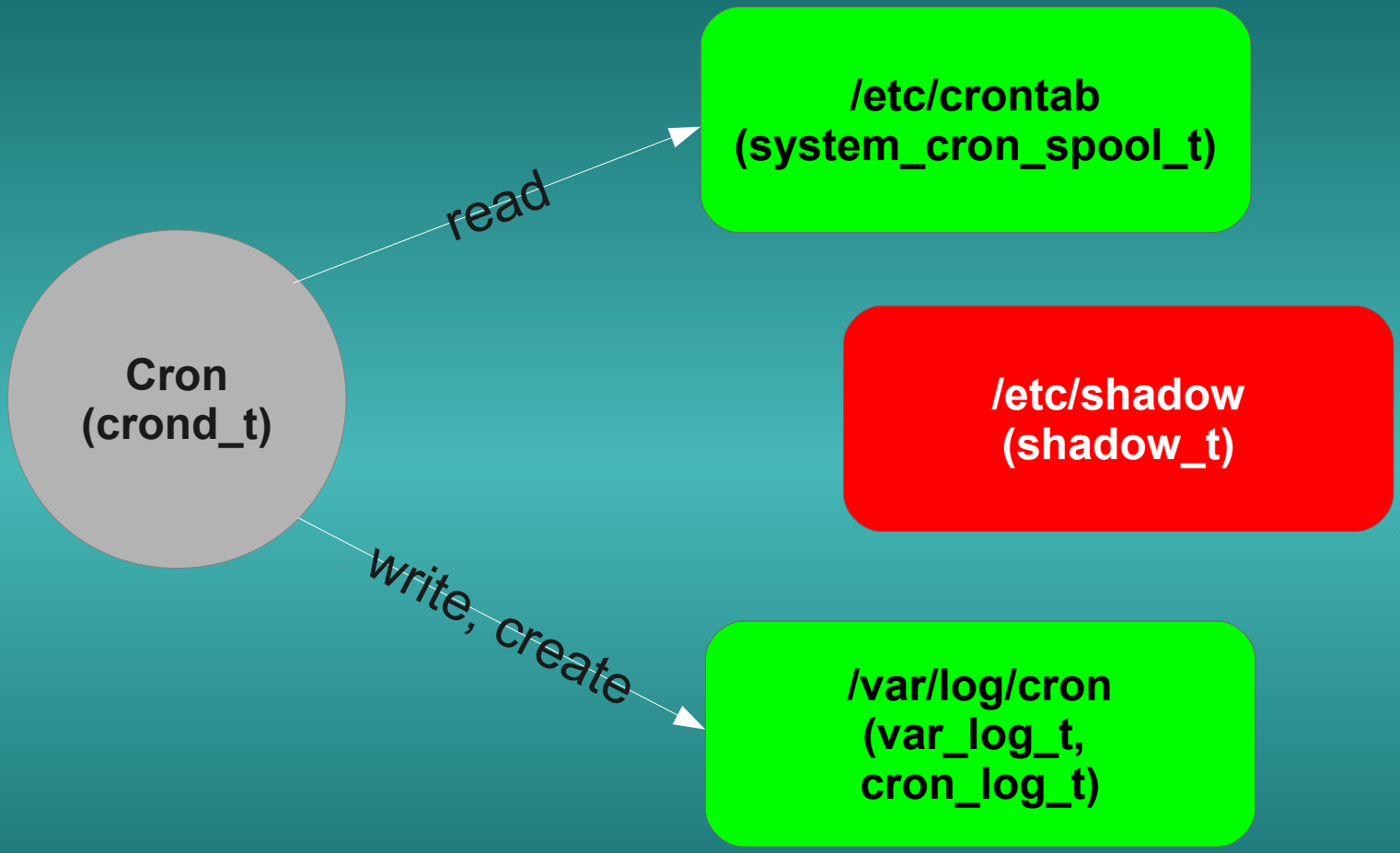
- Type Enforcement opiera się na... typie ;-)
- typy mogą być nadawane zasobom i procesom
- reprezentują wszystko co potrzebne z punktu widzenia bezpieczeństwa

SELinux – Type Enforcement

- Type Enforcement opiera się na... typie ;-)
  - typy mogą być nadawane zasobom i procesom
  - reprezentują wszystko co potrzebne z punktu widzenia bezpieczeństwa
- Przykład: cron
  - proces crond → crond_t
  - /etc/crontab → system_cron_spool_t

- Type Enforcement opiera się na... typie ;-)
  - typy mogą być nadawane zasobom i procesom
  - reprezentują wszystko co potrzebne z punktu widzenia bezpieczeństwa
- Przykład: cron
  - proces crond → crond_t
  - /etc/crontab → system_cron_spool_t
- W politykach definiuje się akcje pomiędzy typami
  - crond_t może czytać system_cron_spool_t

SELinux – Type Enforcement



SELinux – file context / process domain

- Procesy → zamiast terminu „typ” mówi się „domena”.
  - domyślnie child dziedziczy po parencie
  - ustawiany przez polityki (type transition rule)
  - ustawiany przez proces (np. login)

init (init_t) → httpd init script (initrc_t) → httpd (httpd_t)

SELinux – file context / process domain

- Procesy → zamiast terminu „typ” mówi się „domena”.
  - domyślnie child dziedziczy po parencie
  - ustawiany przez polityki (type transition rule)
  - ustawiany przez proces (np. login)

init (init_t) → httpd init script (initrc_t) → httpd (httpd_t)

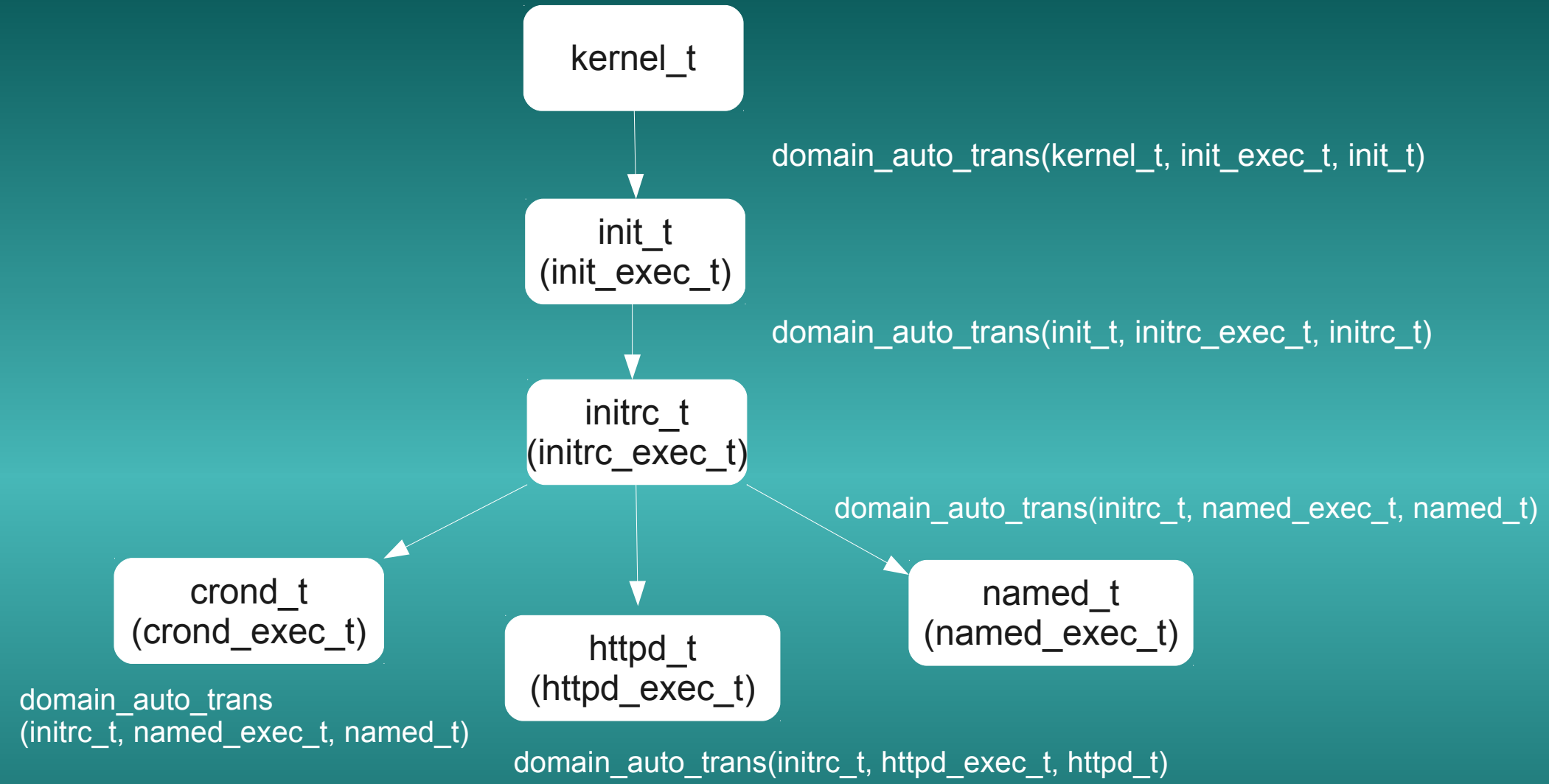
- Pliki → zamiast terminu „typ” mówi się „file context”.
  - domyślnie plik dziedziczy po katalogu w którym się znajduje
  - narzucany z poziomu plików konfiguracyjnych SELinux
  - chcon = chmod

SELinux – Type Transition

Type Transition

- sens działania podobny do setuid();
- typ procesu rodzica i typ pliku programu
- domain_auto_trans(initrc_t, crond_exec_t, crond_t)

SELinux – Type Transition



Robert Jaroszuk <zim@iq.pl>

SELinux – Type Transition

- Co daje nam Type Transition ?

SELinux – Type Transition

Co daje nam Type Transition ?

- zapewnia, że aplikacja pracuje we właściwej (osobnej) domenie

SELinux – Type Transition

Co daje nam Type Transition ?

- zapewnia, że aplikacja pracuje we właściwej (osobnej) domenie
- nie ma możliwości zwiększenia uprawnień z poziomu procesu

Co daje nam Type Transition ?

- zapewnia, że aplikacja pracuje we właściwej (osobnej) domenie
- nie ma możliwości zwiększenia uprawnień z poziomu procesu
- konkretna domena powiązana jest z konkretnym programem – tylko /usr/bin/passwd może pracować jako passwd_t i czytać /etc/shadow

Co daje nam Type Transition ?

- zapewnia, że aplikacja pracuje we właściwej (osobnej) domenie
- nie ma możliwości zwiększenia uprawnień z poziomu procesu
- konkretna domena powiązana jest z konkretnym programem – tylko /usr/bin/passwd może pracować jako passwd_t i czytać /etc/shadow
- nie jest wymagana modyfikacja kodu aplikacji

Co daje nam Type Transition ?

- zapewnia, że aplikacja pracuje we właściwej (osobnej) domenie
- nie ma możliwości zwiększenia uprawnień z poziomu procesu
- konkretna domena powiązana jest z konkretnym programem – tylko /usr/bin/passwd może pracować jako passwd_t i czytać /etc/shadow
- nie jest wymagana modyfikacja kodu aplikacji

(oczywiście są wyjątki: passwd, useradd, groupadd, openssh)

SELinux – Korzyści

- Dostęp nadawany jest w oparciu o usera i aplikację.

SELinux – Korzyści

- Dostęp nadawany jest w oparciu o usera i aplikację.
- Procesy pracują z najniższymi niezbędnymi uprawnieniami
  - httpd_t może czytać tylko pliki stron i pisać do logów

SELinux – Korzyści

- Dostęp nadawany jest w oparciu o usera i aplikację.
- Procesy pracują z najniższymi niezbędnymi uprawnieniami
  - httpd_t może czytać tylko pliki stron i pisać do logów
- Nawet jeżeli ktoś skutecznie zaatakuje apache, będzie mógł zrobić tylko to, na co pozwala polityka.

SELinux – Konfiguracja

- Polityka Strict – wszystko jest denied.
- Dostępna w RHEL 5.5 i starszych. Od RHEL 6.0 targeted jest już wystarczająco rozwinięta.
- Polityki mają tylko allowy
- Minimalne uprawnienia dla każdego demona
- Trudna do zarządzania na typowym serwerze

SELinux – Konfiguracja

- Polityka MLS – strict + Bell-LaPadula
- no read up, no write down
- tylko dla serwerów (nie ma supportu dla X i wielu innych)
- We're excited to announce that Red Hat has entered into an agreement with atsec information security to certify Red Hat Enterprise Linux 6 under Common Criteria at Evaluation Assurance Level (EAL) 4+,

Polityka Targeted

- tylko procesy objęte politykami są chronione, reszta jest „unconfined” - domyślnie są to procesy użytkowników,
- procesy „unconfined” funkcjonują tak, jakby nie było SELinux,
- procesy chronione pracują w swoich domenach i są ograniczane przez polityki,
- RHEL 4 → ~15 targetów
- RHEL 5.5 → 320+ targetów
- Fedora 13 → 600+
- Każdy program „out of box” zabezpieczony politykami SELinux

- SELinux trzyma pliki konfiguracyjne w /etc/selinux

```
-rw-r--r--  1 root root  512 Oct 22 22:17 config
-rw-------  1 root root  234 Apr  3  2010 restorecond.conf
-rw-r--r--  1 root root 1752 Sep  4  2009 semanage.conf
drwxr-xr-x  5 root root 4096 Dec 25 23:35 targeted
```

Najważniejszy plik to /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 - SELinux is fully disabled.
SELINUX=enforcing
# SELINUXTYPE= type of policy in use. Possible values are:
#   targeted - Only targeted network daemons are protected.
#   strict - Full SELinux protection.
SELINUXTYPE=targeted

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

- Parametry kernela (przydatne gdy coś się popsuje)

SELinux – Konfiguracja

- Parametry kernela (przydatne gdy coś się popsuje)

- selinux=0
  - Uruchamia system z wyłączonym SELinux
  - Tworzone pliki nie posiadają file contextów
  - Po ponownym włączeniu SELinux wymagany będzie relabeling.

SELinux – Konfiguracja

- Parametry kernela (przydatne gdy coś się popsuje)

- selinux=0
  - Uruchamia system z wyłączonym SELinux
  - Tworzone pliki nie posiadają file contextów
  - Po ponownym włączeniu SELinux wymagany będzie relabeling.

- enforcing=0
  - SELinux włączony w trybie permissive
  - Labeling działa prawidłowo

- Przydatne polecenia:

- Przydatne polecenia:

„-Z”

- id -Z
- ls -alZ
- ps auxZ
- netstat -Z
- lsof -Z

SELinux – Polecenia

- getenforce – wyświetla status enforcingu
- setenforce 0/1 – włącza/wyłącza enforcing

```
# getenforce
Enforcing
# setenforce 0
# getenforce
Permissive
# setenforce 1
# getenforce
Enforcing
#
```

SELinux – Polecenia

- selinuxenabled – zwraca 0 gdy włączony 1 gdy wyłączony (skrypty)
- chcon / chcat – change context, change category

```
# touch /tmp/test.html
# ls -Z /tmp/test.html
-rw-r--r--  root root user_u:object_r:tmp_t:s0   /tmp/test.html
# pwd
/var/www/html
# mv /tmp/test.html .
# ls -Z test.html
-rw-r--r--  root root user_u:object_r:tmp_t:s0   test.html
# chcon -t httpd_sys_content_t test.html
# ls -Z test.html
-rw-r--r--  root root user_u:object_r:httpd_sys_content_t:s0 test.html
```

SELinux – Polecenia

- fixfiles, restorecon – przywraca domyślne filecontexty

```
# ls -Z test.html
-rw-r--r--  root root user_u:object_r:tmp_t:s0         test.html
# restorecon -v test.html
restorecon reset /var/www/html/test.html context
user_u:object_r:tmp_t:s0->system_u:object_r:httpd_sys_content_t:s0
#
```

restorecon czyta /etc/selinux/POLICYTYPE/contexts/files/file_contexts*, gdzie przechowywane są wyrażenia regularne mapujące pliki na security contexty.

fixfiles to skrypt który wykorzystuje restorecon i pozwala stosować go dla pakietów rpm, albo dla całego systemu z pominięciem filesystemów sieciowych itp.

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

SELinux – Polecenia

- audit2why – pomaga zrozumieć komunikaty AVC zapisywane w logach
- audit2allow – generowanie reguł w oparciu o komunikaty AVC

```
type=AVC msg=audit(1293314294.051:3306): avc:  denied  { write } for
pid=3566 comm="httpd" name="mod_jk.shm.3566" dev=sda5 ino=130630
scontext=user_u:system_r:httpd_t:s0
tcontext=user_u:object_r:httpd_log_t:s0 tclass=file
```

```
type=AVC msg=audit(1293314294.052:3307): avc:  denied  { unlink } for
pid=3566 comm="httpd" name="mod_jk.shm.3566" dev=sda5 ino=130630
scontext=user_u:system_r:httpd_t:s0
tcontext=user_u:object_r:httpd_log_t:s0 tclass=file
```

```
type=AVC msg=audit(1293314482.430:3314): avc:  denied
{ name_connect } for  pid=3575 comm="httpd" dest=5080
scontext=user_u:system_r:httpd_t:s0
tcontext=system_u:object_r:port_t:s0 tclass=tcp_socket
```

SELinux – Polecenia

```
# cat log| audit2allow -m przyklad > przyklad.te
# cat przyklad.te
module przyklad 1.0;

require {
    type httpd_log_t;
    type httpd_t;
    type port_t;
    class tcp_socket name_connect;
    class file { write unlink };
}

#============= httpd_t ==============
allow httpd_t httpd_log_t:file { write unlink };
allow httpd_t port_t:tcp_socket name_connect;
#
```

SELinux – Polecenia

```
# checkmodule -M -m -o przyklad.mod przyklad.te
checkmodule:  loading policy configuration from przyklad.te
checkmodule:  policy configuration loaded
checkmodule:  writing binary representation (version 6) to
przyklad.mod
# semodule_package -o przyklad.pp -m przyklad.mod
# semodule -i przyklad.pp
```

SELinux – Polecenia

```
# cat log |audit2allow -M przyklad2
******************** IMPORTANT ***********************
To make this policy package active, execute:

semodule -i przyklad2.pp

# ls -la przyklad2.*
-rw-r--r-- 1 root root 1166 Jan 13 22:01 przyklad2.pp
-rw-r--r-- 1 root root  280 Jan 13 22:01 przyklad2.te
#
```

SELinux – Polecenia

- semanage – manipulowanie wieloma ustawieniami SELinux, bez potrzeby ręcznego rekompilowania polityk.

```
# semanage port -l|grep http_port_5
http_port_t                    tcp    80, 443, 488, 8008, 8009, 8443
pegasus_http_port_t            tcp    5988
# semanage port -a -p tcp -t http_port_t 8444
# semanage port -l|grep http_port_t
http_port_t                    tcp    8444, 80, 443, 488, 8008, 8009, 8443
pegasus_http_port_t            tcp    5988
#
```

SELinux – Polecenia

- sestatus – szczegółowe informacje o statusie SELinux

```
# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /selinux
Current mode:                   enforcing
Mode from config file:          enforcing
Policy version:                 21
Policy from config file:        targeted
#
```

SELinux – Polecenia

- setsebool (-P), getsebool – pobieranie/ustawianie booleanów SELinux
- man -k selinux

```
# getsebool klogd_disable_trans
klogd_disable_trans --> off
# ps axZ|grep klogd
system_u:system_r:klogd_t:s0     1517 ?        Ss     0:00 klogd -x
# setsebool klogd_disable_trans 1
# service syslog restart
Shutting down kernel logger:                               [  OK  ]
Shutting down system logger:                               [  OK  ]
Starting system logger:                                    [  OK  ]
Starting kernel logger:                                    [  OK  ]
# ps axZ|grep klog
user_u:system_r:initrc_t:s0      878 ?        Ss     0:00 klogd -x
#
```

SELinux – Polecenia

- semanage permissive

```
# /usr/sbin/semanage permissive -a httpd_t
# /usr/sbin/semodule -l | grep permissive
permissive_httpd_t      1.0
# /usr/sbin/semanage permissive -d httpd_t
```

- setroubleshoot – graficzny interpreter komunikatów AVC

- system-config-selinux – graficzny konfigurator + generator polityk

SELinux – Dodatkowe informacje

Q/A ?

- NSA SELinux – http://www.nsa.gov/research/selinux/index.shtml
- NSA SELinux FAQ – http://www.nsa.gov/research/selinux/faqs.shtml
- SELinux wiki — http://selinuxproject.org/
- http://fedora.redhat.com/projects/selinux/
- http://fedoraproject.org/wiki/SELinux
- http://danwalsh.livejournal.com/

IRC - irc.freenode.net, #fedora-selinux and #selinux

Robert Jaroszuk <zim@iq.pl>