SELinux dla dociekliwych
czyli co można z tym zrobić, jak i dlaczego?

```
# setenforce 1
```

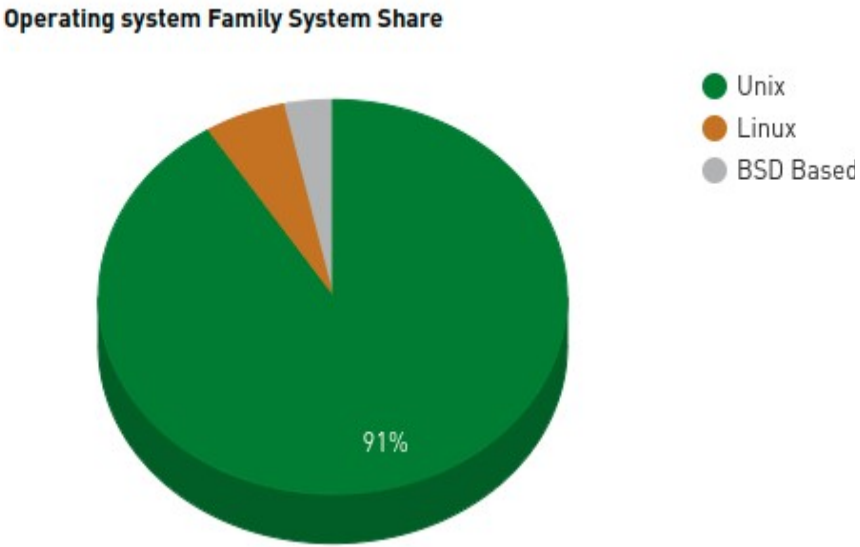
Tomasz Barbaszewski

Dlaczego nikt mnie nie lubi ???

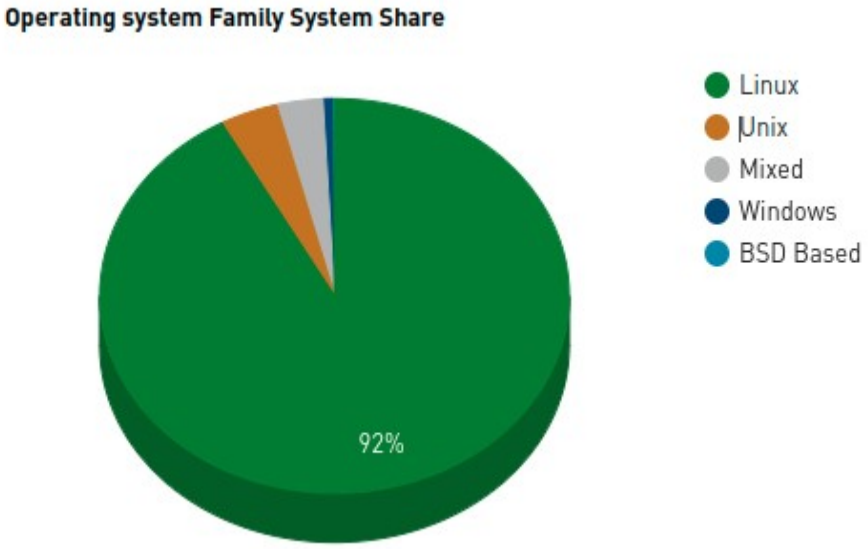
By Máirín Duffy (mentioned in the SELinux wiki.) - https://www.deviantart.com/pookstar/art/SELinuxNews-Logo-29645462, CC BY-SA 2.5, https://commons.wikimedia.org/w/index.php?curid=76206065

Trusted operating systems

Linux – “młodszy brat” Uniksa, który na przełomie XX I XXI w zajął jego miejsce:



Rok 2000



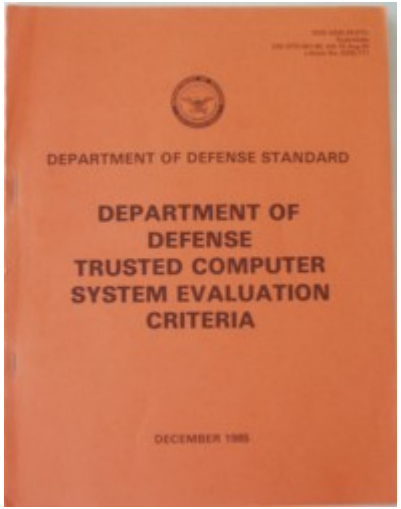
Rok 2010

Dziś udział Linuksa wynosi 100% (źródło top500.org)

Trusted operating systems

Unix – wyzwania związane z bezpieczeństwem systemu
- protected subsystems
- Trusted Computing Base (Rushby, 1981) – AIX, SCO UNIX...
- Pierwsza formalizacja US DoD (1985 “Orange Book”) :

D – ochrona minimalna (ang. Minimal Protection)
C1 – ochrona uznaniowa (ang. Discretionary Protection)
C2 – ochrona z kontrolą dostępu (ang. Controlled Access Protection)
B1 – ochrona z etykietowaniem (ang. Labeled Security Protection)
B2 – ochrona strukturalna (ang. Structured Protection)
B3 – ochrona przez podział (ang. Security Domains)
A1 – konstrukcja zweryfikowana (ang. Verified Design)



Systemy klasy Unix certyfikowane na poziom C2 stają się powszechnie dostępne w latach 1990

Trusted operating systems

Agencje rządowe oraz business żądają coraz większego bezpieczeństwa - Powstaje pojęcie “Mission Critical Systems”

Wymagania – klasa C2 (uznaniowa kontrola dostępu DAC oraz kontrola procesów logowania) +

- wprowadzenie etykietowania danych (Labeled Protection - klasa B1)
- dołączenie sformalizowanej polityki bezpieczeństwa (Security Policy – klasa B2)
- separacja zadań (Security Domains – klasa B3)

Firmy komercyjne zaczynają oferować systemy spełniające wymagania klasy B, lecz są one bardzo kosztowne. Na początku lat 1990 zostaje opracowany system Trusted XENIX (IBM i Trusted Information Systems), który w wyniku formalnych badań otrzymuje certyfikat klasy DoD TCSEC B2.

Trusted operating systems

Problemy narastają:

Powstaje seria publikacji znanych jako “Rainbow Books”

Wykorzystywanie systemów komputerowych przez agencje rządowe – np. Department of Defense lub inne o podobnym znaczeniu wymaga spełnienia formalnych reguł ochrony informacji niejawnych.

Business również zaczął się poważnie interesować systemami komputerowymi godnymi zaufania (“mission critical”).

Wprowadzenie oprogramowania Open Source w systemach realizujących krytyczne zadania (Mission Critical) staje się uwarunkowane wprowadzeniem i sformalizowaniem zaawansowanych mechanizmów bezpieczeństwa.

Trusted operating systems

Integrating Flexible Support for Security Policies into the Linux Operating System

Decision to move to Linux

- Recognized need to move to a mainstream platform
- Past strategies not producing desired results
- National Security Council interest in Open Source
- Technology transfer opportunities
- Linux chosen as best alternative

Źródło: NSA – Information Assurance Research Group (2003)

Wszystko ma etykietę!

Etykiety SELinux wykorzystują przestrzeń nazw (namespace) rozszerzonych atrybutów (Extended Attributes) do przechowywania dodatkowych metadanych realizując w ten sposób “Labeled Security Protection”:

```
[kazio@localhost ~]$
getfattr -m "-" file
# file: file
security.selinux
system.posix_acl_access
user.checksum.sha256
user.comment

[kazio@localhost ~]$ getfattr -n security.selinux file
# file: file
security.selinux="unconfined_u:object_r:user_home_t:s0"
```

Wszystko ma etykietę!

Usługi (services) zostały skojarzone z domenami oraz przyporządkowano im role:

```
[root@localhost ~]# ps -efZ |grep sshd
system_u:system_r:sshd_t:s0-s0:c0.c1023 root 1061      1  0 14:26 ?
00:00:00 /usr/sbin/sshd
[root@localhost ~]# ps -efZ | grep httpd
system_u:system_r:httpd_t:s0    root        2152       1  0 14:27 ?
00:00:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache      2153    2152  0 14:27 ?
00:00:00 /usr/sbin/httpd -DFOREGROUND ...
```

Usługa (program) ma dostęp jedynie do obiektów o określonym typie (TE) – np.

```
[root@localhost ~]# ls -lZ /var/www/
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0  6
06-15 14:29 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0     24
10-01 11:48 html
```

Trusted operating systems – podejście “ogólnowojskowe”

Wydzielenie zamkniętych stref

Obligatoryjna kontrola dostepu – MAC. Zastrzeżone akcje (np. testy samolotu SR-71 BlackBird itp.) mogą być prowadzone tylko w tych strefach przez upoważnionych

SELinux – targeted policy.
Zdefiniowane domeny – np. httpd:

```
[tomekb@localhost ~]$ ps -efZ|grep http
system_u:system_r:httpd_t:s0    root        2926       1  0 07:08 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache      2940    2926  0 07:08 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
```

```
[tomekb@localhost ~]$ ls -lZ /usr/sbin/httpd
-rwxr-xr-x. 1 root root system_u:object_r:httpd_exec_t:s0 583960 Jun 15 14:27 /usr/sbin/httpd
```

```
[tomekb@localhost ~]$ ls -lZ /var/www
total 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 Jun 15 14:29 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0     6 Jun 15 14:29 html
```

Trusted operating systems – podejście “ustawowe”

Klasyfikacja (classification) informacji

Udzielanie dostępu (clearance)

Może być łączone z wprowadzeniem zamkniętej strefy przetwarzania danych (np. Kancelaria tajna)

SELinux: Polityka MLS z regułami dostępu według modelu Bell-LaPadula:
No read UP – not write DOWN (praktyka wywiadu – raportuje się do Szefa!)

Uzupełnienie - write-equality:
Zapisywane mogą być tylko obiekty o takim samym poziomie klasyfikacji jak poziom dostępu (clearance) wykonywanej akcji (można to zmienić)

Domeny usług (akcji) oraz kategorie MCS są uwzględniane

W “naszym fachu” najlepiej wiedzieć jak najmniej!

S2 – Szef szefów otrzymuje komplet informacji (od wielu siatek szpiegowskich i podejmuje decyzje.
Nie udostępnia żadnych informacji szefom, siatek! Może znać tylko ich pseudonimy.

S1 – otrzymuje informacje od agentów, wstępnie je przetwarza i przesyła do Szefa Szefów.
Nie udostępnia żadnych informacji agentom.
Może znać tylko ich pseudonimy.

S0 – zbierają informacje I przesyłają je do Szefa Siatki.

SELinux – NSA

W SELinux NSA wprowadziła klasyczne reguły ochrony informacji:

- **Klasyfikacja** (Unclassified, Confidential, Secret, Top Secret):
  dotyczy informacji (w SELinux “objects”)

- **Poziom dostępu** “poświadczenie bezpieczeństwa” (clearance):
  dotyczy użytkowników oraz procesów (w SELinux “subjects”)

- **Certyfikacja** może dotyczyć organizacji, stref przetwarzania oraz systemów:
  poziomy zaufania (Assurance Levels), reguł dystrybucji i przekazywania informacji
  Dla informacji niejawnych w Polsce certyfikaty wydają ABW oraz SKW

Informacje niejawne (Polska)

Poziomy (klauzule) informacji niejawnych:

Ściśle tajne – zagrożenie suwerenności, niepodległości lub integralności terytorialnej,

Tajne – zagrożenie poważnej szkody dla Polski, porządku konstytucyjnego, realizacji zadań związanych z obronnością,

Poufne – ich ujawnienie może spowodować szkodę lub utrudni prowadzenie polityki zagranicznej Polski,

Zastrzeżone – ich ujawnienie może mieć szkodliwy wpływ na wykonywanie zadań związanych z obronnością Polski.

Przykłady w SELinux:

S15 – TopSecret
(najwyższy poziom)

S8 – Confidential
(np. poufne)

S0 – Unclassified
(ogólnie dostępne)

Nadana klauzula ochrony jest nieodłączalną etykietą informacji!

Wszystko ma etykietę – ale jej znaczenie może być różne

Subjects labels (clearance) – usługa:

```
[root@localhost ~]# ps -eZ|grep sshd
system_u:system_r:sshd_t:s0-s0:c0.c1023 1045 ?   00:00:00 sshd
```

Subjects labels (clearance) – users:

```
Login Name           SELinux User         MLS/MCS Range        Service

__default__          unconfined_u         s0-s0:c0.c1023       *
asia                 staff_u              s0:c1                *
root                 unconfined_u         s0-s0:c0.c1023       *
tomekb               secoff_u             s0:c3                *
```

Default unconfined user:

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

Confined user – Security Officer:

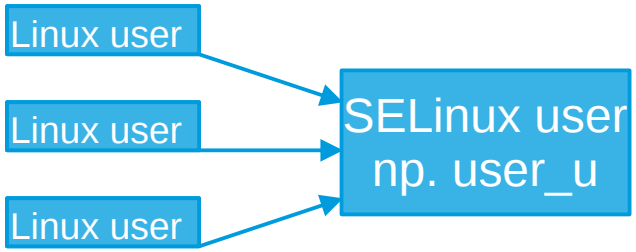
```
[tomekb@localhost ~]$ id -Z
secoff_u:staff_r:staff_t:s0:c3
```

Użytkownicy i role

Linux user – klasycznie rozumiany użytkownik Linuksa, którego nazwa (login name) decyduje o uprawnieniach w systemie Linux (DAC, ACL itp.).

SELinux user – np. user_u, staff_u itp. nazwa wykorzystywana do utworzenia etykiety;

```
[tomekb@localhost ~]$ id -Z
user_u:user_r:user_t:s0-s0:c0.c1023
[tomekb@localhost ~]$
```



Użytkownik SELinux określa uprawnienia w systemie SELinux.
Mapowanie użytkowników Linuksa na użytkowników SELinuksa umożliwia tworzenie grup mogących realizować określone role (RBAC) I uprawnienia (MAC)

Użytkownicy i role

Uwagi:

Domyślnie nowi użytkownicy w systemie (useradd) są przypisywani do etykiety:

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

Taką samą domyślną (__default__) etykietę otrzymuje użytkownik root.

Oznacza to, że SELinux nie ogranicza ich działań.

Ze względów bezpieczeństwa do administracji systemem powinno się wykorzystywać przeznaczone do tego role – sysadm_r, secadm_r, logadm_r, auditadm_r, dbadm_r lub webadm_r.

Użytkownicy i role

Celowa jest zmiana ustawienia domyślnego dla nowo zakładanych użytkowników:

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

Użytkownik SELinux user_u jest przeznaczony dla standardowego użytkownika systemu Linux i nie może wykorzytywać su ani sudo

Jeśli jest taka potrzeba można wykorzystać użytkownika staff_u, który może korzystać z sudo (ale nie z su):

```
useradd (usermod) -Z staff_u <linux_user>
```

Nie wpływa to na etykietę użytkownika root:

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

Użytkownicy i role

Dalsze uwagi:

Modyfikujemy przyporządkowanie użytkownika Linuksa <asia>

```
usermod -Z staff_u asia
```

Przyporządkowujemy użytkownikowi asia wymagany typ i rolę:

```
[root@localhost sudoers.d]# cat asia
asia ALL=(ALL) TYPE=sysadm_t ROLE=sysadm_r ALL

$ asia@localhost
sudo -i
[sudo] hasło użytkownika asia:
[root@localhost ~]# id -Z
staff_u:sysadm_r:sysadm_t:s0-s0:c0.c1023
```

Użytkownicy i role

Dalsze uwagi:

Modyfikujemy przyporządkowanie użytkownika Linuksa <asia>

```
usermod -Z staff_u asia
```

Przyporządkowujemy użytkownikowi asia wymagany typ i rolę:

```
[root@localhost sudoers.d]# cat asia
asia ALL=(ALL) TYPE=sysadm_t ROLE=sysadm_r ALL

$ asia@localhost
sudo -i
[sudo] hasło użytkownika asia:
[root@localhost ~]# id -Z
staff_u:sysadm_r:sysadm_t:s0-s0:c0.c1023
```

Użytkownicy i role

Dalsze uwagi:

Modyfikujemy przyporządkowanie użytkownika Linuksa <jurek>

```
setsebool -P ssh_sysadm_login on
usermod -G wheel -Z sysadm_u jurek
```

Wykorzystujemy standardową w /etc/sudoers grupę wheel:

```
%wheel  ALL=(ALL)       ALL
```

```
$ jurek@localhost
sudo -i
Multiple identities can be used for authentication:
 1.  jurek # jurek może korzystać z su i sudo
 2.  asia # asia NIE może korzystać z su, tylko z sudo
Choose identity to authenticate as (1-2): 1
[sudo] hasło użytkownika jurek:
[root@localhost ~]# id -Z
sysadm_u:sysadm_r:sysadm_t:s0-s0:c0.c1023
```

Użytkownicy SELinux I ich role

SELinux users i powiązane z nimi role:

```
[root@localhost ~]# semanage user -l

                Labeling   MLS/       MLS/
SELinux User    Prefix     MCS Level  MCS Range                SELinux Roles

guest_u         user       s0         s0                       guest_r
root            user       s0         s0-s0:c0.c1023           staff_r sysadm_r system_r unconfined_r
secoff_u        user       s0         s0:c3                    sysadm_r
staff_u         user       s0         s0-s0:c0.c1023           staff_r sysadm_r unconfined_r
sysadm_u        user       s0         s0-s0:c0.c1023           sysadm_r
system_u        user       s0         s0-s0:c0.c1023           system_r unconfined_r
unconfined_u    user       s0         s0-s0:c0.c1023           system_r unconfined_r
user_u          user       s0         s0                       user_r
xguest_u        user       s0         s0                       xguest_r
```

Wprowadzenie użytkowników SELinuksa oraz powiązanych z nimi ról, typów i zakresów MLS i MCS pozwala na budowę hierarchicznej struktury uprawnień.

Dostępne role:

```
seinfo -r

Roles: 14

auditadm_r   dbadm_r   guest_r   logadm_r   nx_server_r   object_r
secadm_r      staff_r       sysadm_r   system_r   unconfined_r
user_r
webadm_r      xguest_r
```

To samo dla wybranej roli # Uwaga lista może być długa!

```
[root@localhost ~]# seinfo -r webadm_r -x

Roles: 1
   role webadm_r types webadm_t;
```

Wszystko ma etykietę – ale jej znaczenie może być różne

Znaczenie etykiet obiektów jest inne – hierarchia typów katalogów

```
[root@localhost ~]# ls -ldZ /
dr-xr-xr-x. 17 root root system_u:object_r:root_t:s0 224 Sep 16 09:58 /
[root@localhost ~]# ls -ldZ /root
dr-xr-x---. 8 root root system_u:object_r:admin_home_t:s0 4096 Sep 25 09:16 /root
[root@localhost ~]# ls -ldZ /home
drwxr-xr-x. 6 root root system_u:object_r:home_root_t:s0 58 Sep 24 20:50 /home
[root@localhost ~]# ls -ldZ /home/tomekb
drwx------. 16 tomekb tomekb unconfined_u:object_r:user_home_dir_t:s0 4096 Sep 25 08:52 /home/tomekb
[root@localhost ~]# ls -ldZ /home/asia
drwx------. 4 asia asia unconfined_u:object_r:user_home_dir_t:s0 125 Sep 21 15:16 /home/asia
```

```
[tomekb@localhost ~]$ id -Z
system_u:system_r:unconfined_service_t:s0
[tomekb@localhost ~]$ touch jajo
[tomekb@localhost ~]$ ls -lZ jajo
-rw-rw-r--. 1 tomekb tomekb system_u:object_r:user_home_t:s0 0 Sep 25 08:52 jajo
```

Rola object_r jest domyślną rolą dla obiektów (także dla portów):

```
[root@localhost ~]# semanage port -l | grep ssh
ssh_port_t                     tcp      22
```

Wszystko ma etykietę – ale jej znaczenie może być różne

Format etykiety SELinux polityka MLS/MCS:

<użytkownik>_u:<rola>_r:<typ>_t:s0-s15:c0.c1023

s0-s15 – dla programów i użytkowników poziom dostępu (clearance)

s0-s15 – dla obiektów klasyfikacja ochrony (classification)

c0.c1023 – kategorie MCS (mogą być wykorzystywane także przez targeted policy)

```
[tomekb@localhost ~]$ ps -efZ|grep systemd
system_u:system_r:init_t:s0-s15:c0.c1023 root  1       0  0 14:52 ?        00:00:06 /usr/lib/systemd/systemd rhgb --switched-roo
t --system --deserialize 31
system_u:system_r:syslogd_t:s15:c0.c1023 root 630      1  0 14:52 ?        00:00:02 /usr/lib/systemd/systemd-journald
system_u:system_r:udev_t:s0-s15:c0.c1023 root 643       1  0 14:52 ?        00:00:00 /usr/lib/systemd/systemd-udevd
system_u:system_r:systemd_logind_t:s0-s15:c0.c1023 root 738 1  0 14:52 ?    00:00:00 /usr/lib/systemd/systemd-logind
user_u:user_r:user_t:s3         tomekb      5451       1  0 20:25 ?        00:00:00 /usr/lib/systemd/systemd --user
```

Polityki bezpieczeństwa

Dostępne polityki bezpieczeństwa w SELinux (implementacja RedHat)

- Targeted Policy
Domyślna polityka bezpieczeństwa, wykorzystująca odrębne domeny dla poszczególnych serwisów – np. httpd:

Usługa httpd musi mieć dostęp do określonego portu – np. 80, bo inaczej nie wystartuje:

- setenforce 1
- zmieniamy port dla usługi httpd np. na 180
- systemctl start httpd zwraca błąd: “control process excited with error code…”
- SELinux nie dopuszcza wykorzystania portu 180 przez usługę httpd:

```
3    httpd    name_bind    port 180    1    2022-10-01 12:06:31  Notify
```

Dostosowywanie polityki bezpieczeństwa

```
3    httpd    name_bind    port 180    1    2022-10-01 12:06:31  Notify
```

Mamy dwa wyjścia:

1. Zmienić “Listen port” w pliku httpd.conf na standardowy (np. Listen 80) i wykonać restart httpd

2. Dodać port 180 do akceptowanych przez SELinux dla usługi http:

```
[root@localhost conf]# semanage port -l|grep http
http_cache_port_t              tcp      8080, 8118, 8123, 10001-10010
http_cache_port_t              udp      3130
http_port_t                    tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000

[root@localhost ~]# semanage port --add -t http_port_t -p tcp 180

root@localhost conf]# semanage port -l|grep http
http_cache_port_t              tcp      8080, 8118, 8123, 10001-10010
http_cache_port_t              udp      3130
http_port_t                    tcp      180, 80, 81, 443, 488, 8008, 8009, 8443, 900
```

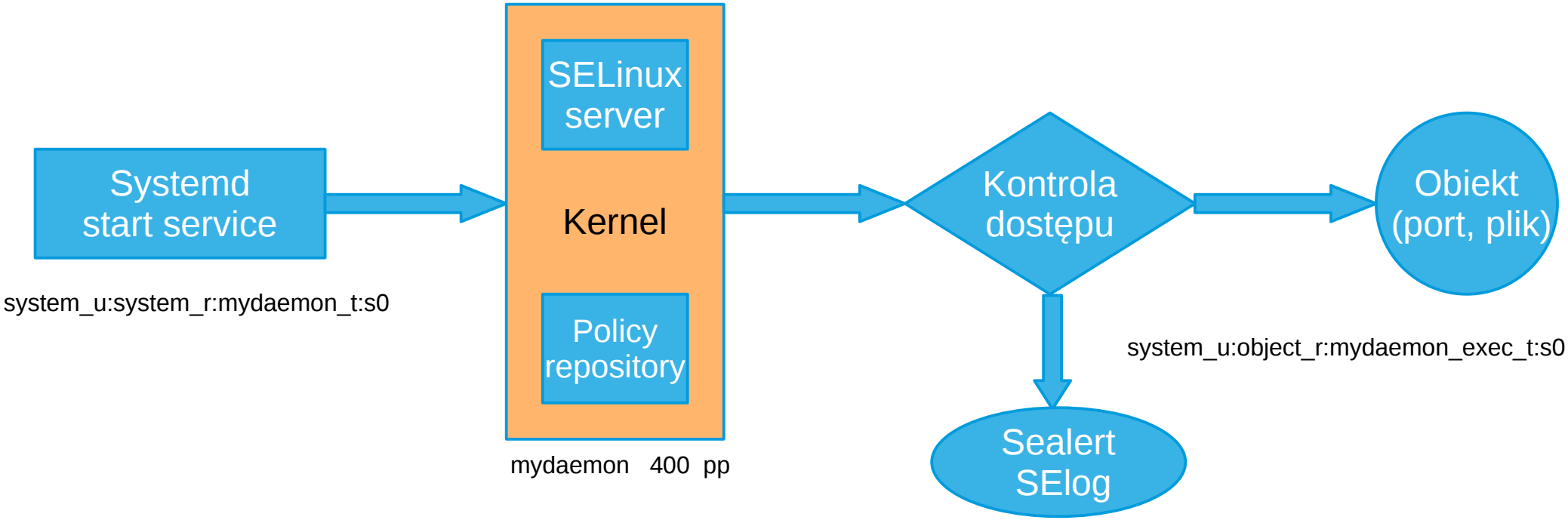
Targeted policy

Domyślna polityka bezpieczeństwa jest przede wszystkim przeznaczona do zabezpieczenia usług systemowych. Podstawą tej polityki jest wykorzystywanie izolowanych stref, do których należą elementy określone poprzez fragment etykiety TE (Type Enforcement).

Polityka jest tworzona w oparciu o zestaw plików – np. oto przykład mydaemon.te
(przygotowywanie własnej polityki bezpieczeństwa szczegółowo omówię później)

```
...
type mydaemon_t;
type mydaemon_exec_t;
init_daemon_domain(mydaemon_t, mydaemon_exec_t)

permissive mydaemon_t;
...
```

Targeted policy



Targeted policy

Targeted policy ma określone zadanie – realizować uruchomioną usługę (service) do do zasobów należących do określonej domeny skojarzonej z usługą.

Dostęp do obiektów jest po prostu określony przez ich typ – np.:

```
system_u:object_r:httpd_sys_content_t:s0
```

W przypadku konieczności wywołania jakiegoś programu z wnętrza innej usługi (np. mail z httpd) musimy pozwolić na domain transition:

```
[root@localhost www]# sepolicy transition -s httpd_t -t system_mail_t
httpd_t @ courier_exec_t --> system_mail_t -- Dozwolone False [ httpd_can_sendmail=0 ]
httpd_t @ exim_exec_t --> system_mail_t -- Dozwolone False [ httpd_can_sendmail=0 ]
httpd_t @ sendmail_exec_t --> system_mail_t -- Dozwolone False [ httpd_can_sendmail=0 ]
httpd_t @ postfix_postdrop_t --> system_mail_t -- Dozwolone False [ httpd_can_sendmail=0 ]
```

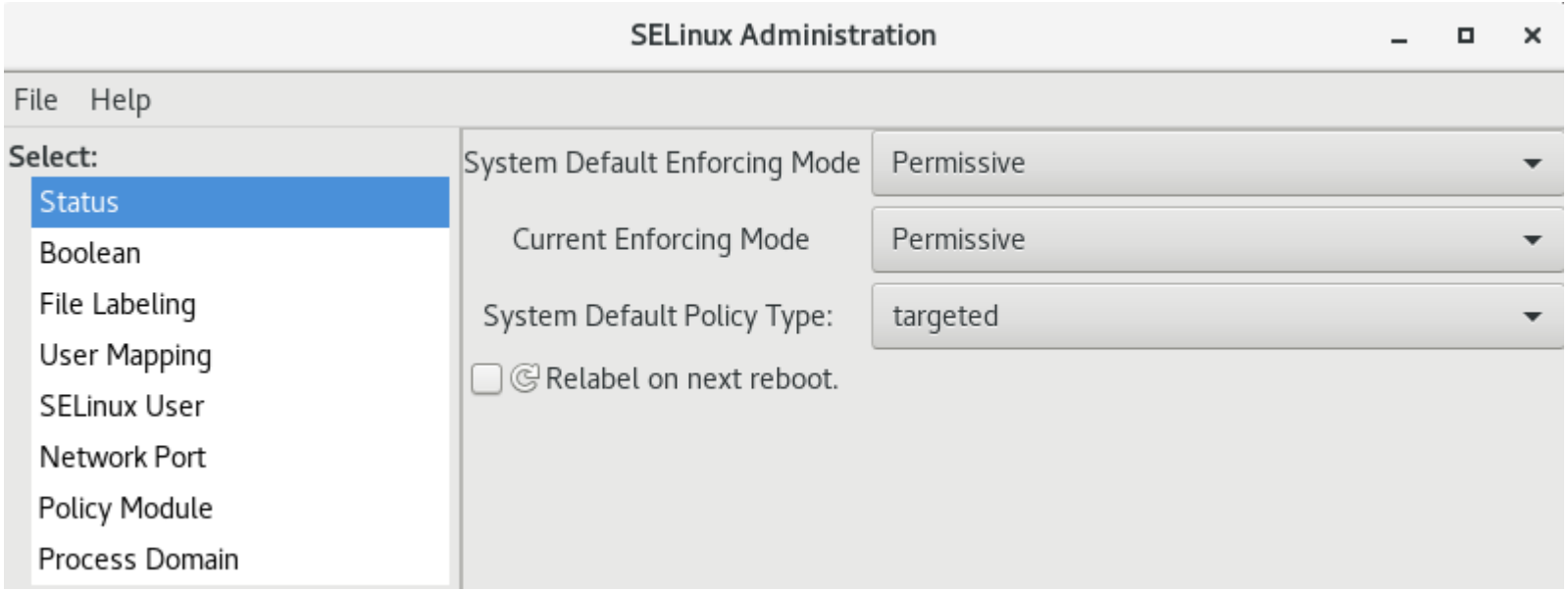
Dostosowywanie polityki bezpieczeństwa

Semanage arguments:

```
import                  Import local customizations
export                  Output local customizations
login                   Manage login mappings between linux users and SELinux
                        confined users
user                    Manage SELinux confined users (Roles and levels for an
                        SELinux user)
port                    Manage network port type definitions
ibpkey                  Manage infiniband ibpkey type definitions
ibendport               Manage infiniband end port type definitions
interface               Manage network interface type definitions
module                  Manage SELinux policy modules
node                    Manage network node type definitions
fcontext                Manage file context mapping definitions
boolean                 Manage booleans to selectively enable functionality
permissive              Manage process type enforcement mode
dontaudit               Disable/Enable dontaudit rules in policy
```

Dostosowywanie polityki bezpieczeństwa

Interfejs graficzny - `system-config-selinux`



Dostosowywanie polityki bezpieczeństwa

SELinux status:

DISABLE – całkowite wyłączenie SELinuksa wymaga restartu systemu Linux
Wymaga przed restartem zmiany w pliku konfiguracyjnym (`/etc/selinux/config`) lub wprowadzenia zmian w programie ładującym (grub)

Permissive – akcje zabronione przez AVC są jedynie logowane

```
# setenforce 0
```

Enforced – akcje zabronione nie są wykonywane i logowane!

```
# setenforce 1
```

UWAGA:
System Linux, który choćby przez krótki czas pracował z wyłączonym SELinuksem nie może być już nigdy traktowany jako zaufany (trusted)

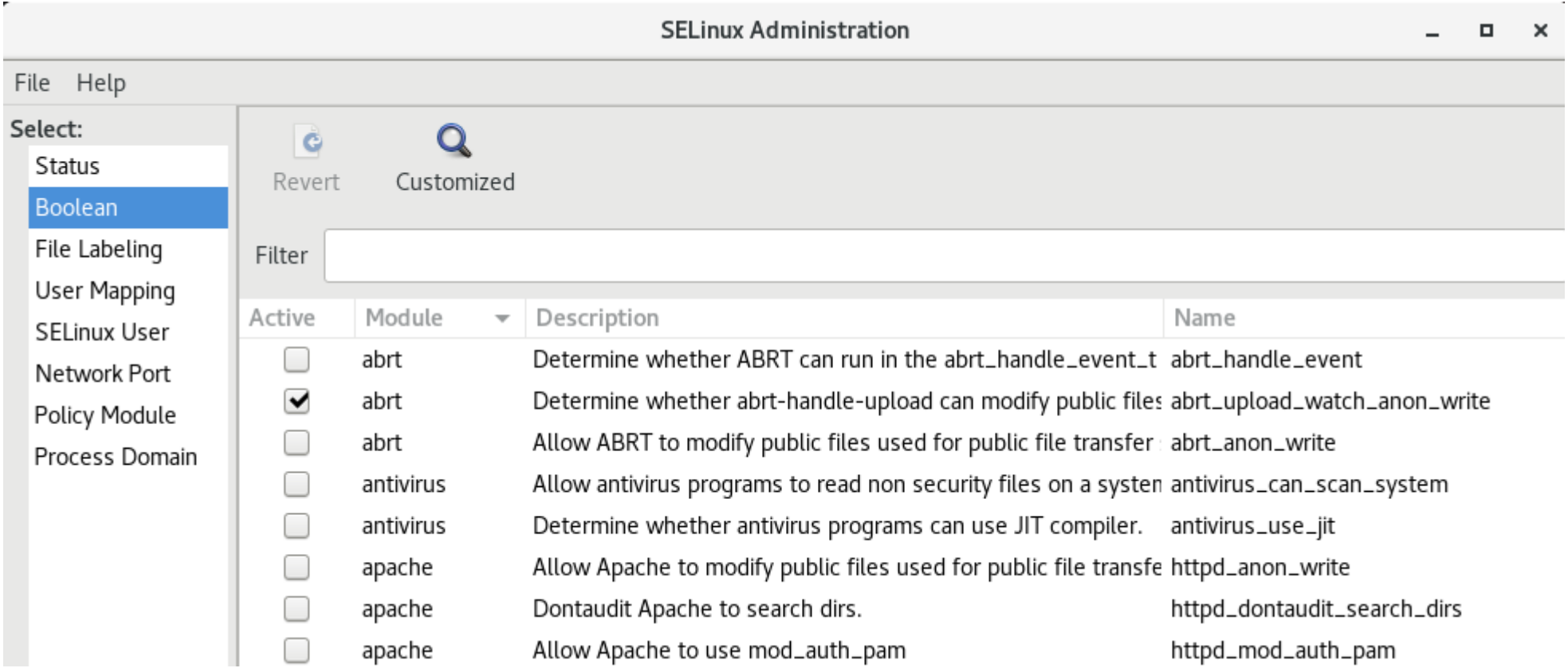
Dostosowywanie polityki bezpieczeństwa

Przełączniki (booleans) SELinux:

```
[root@localhost conf]# semanage boolean -l | grep httpd | less
```

```
SELinux boolean                 State,Default Description
...
httpd_anon_write                (off  ,  off)  Allow Apache to modify public files used for public...
httpd_builtin_scripting         (on   ,  on)  Allow httpd to use built in scripting (usually php)
httpd_can_check_spam            (off  ,  off)  Allow http daemon to check spam
httpd_can_connect_ftp           (off  ,  off)  Allow httpd to act as a FTP client connecting to the ftp
port...
httpd_can_connect_ldap          (off  ,  off)  Allow httpd to connect to the ldap port
httpd_can_connect_mythtv        (off  ,  off)  Allow http daemon to connect to mythtv
httpd_can_connect_zabbix        (off  ,  off)  Allow http daemon to connect to zabbix
httpd_can_network_connect       (off  ,  off)  Allow HTTPD scripts and modules to connect to the network...
httpd_can_network_connect_cobbler (off  ,  off)  Allow HTTPD scripts and modules to connect to cobbler...
httpd_can_network_connect_db    (off  ,  off)  Allow HTTPD scripts and modules to connect to databases…
...
```

Dostosowywanie polityki bezpieczeństwa



Dostosowywanie polityki bezpieczeństwa

Użytkownicy (semanage login) SELinux:

```
[root@localhost security]# semanage login -l

Login Name           SELinux User         MLS/MCS Range        Service

__default__          unconfined_u         s0-s0:c0.c1023          *
asia                 staff_u              s0:c1                   *
root                 unconfined_u         s0-s0:c0.c1023             *
tomekb               secoff_u             s0:c3                   *
```

Domyślnie nowo zakładani użytkownicy systemu Linux są nieograniczani (unconfined)

Dostosowywanie polityki bezpieczeństwa

Etykiety plików (fcontext) SELinux:

```
[root@localhost conf]# semanage fcontext -l | less
```

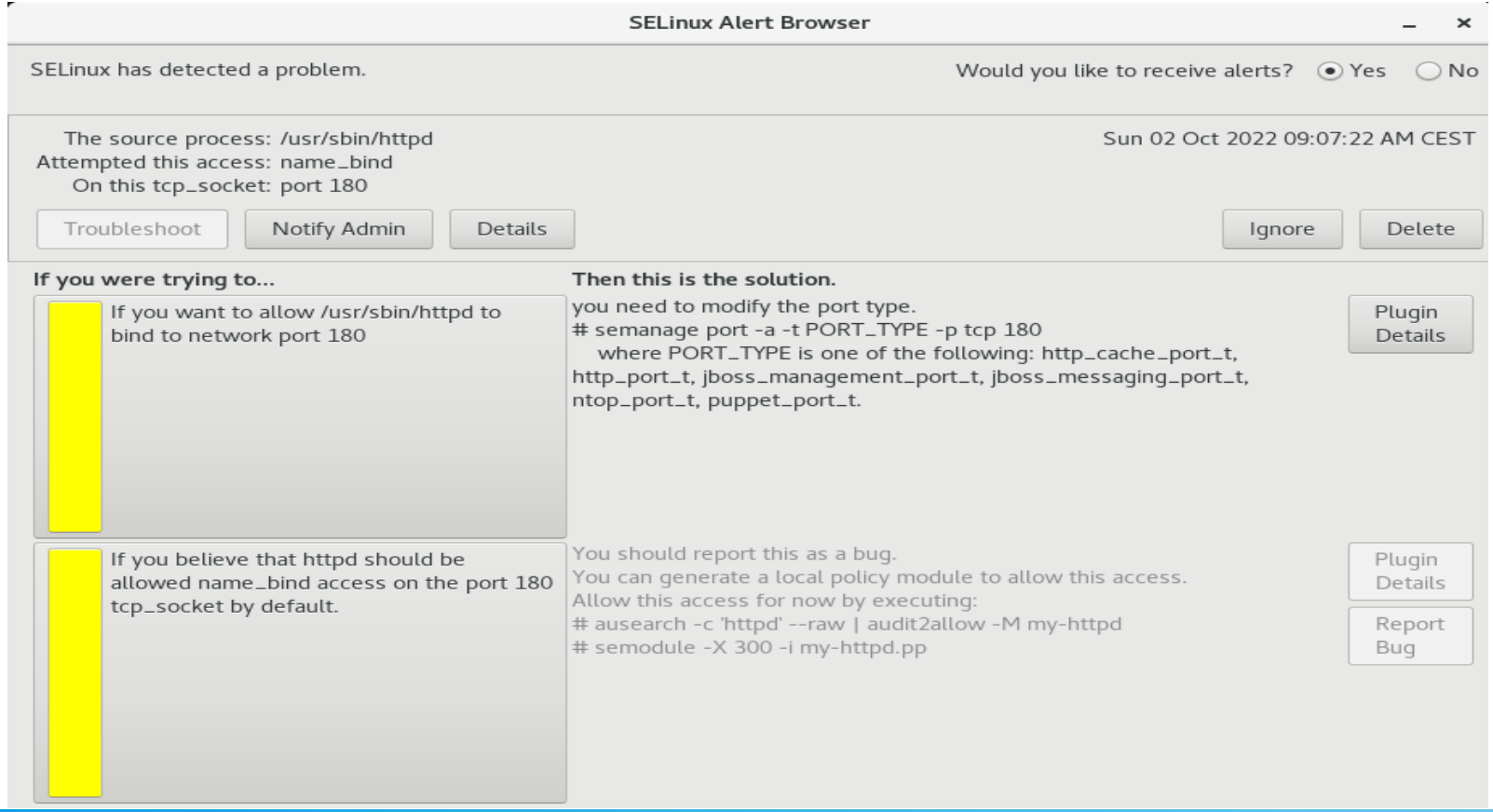
fcontext SELinuksa	typ	Kontekst
/	directory	system_u:object_r:root_t:s0
/.*	all files	system_u:object_r:default_t:s0
/[^/]+	regular file	system_u:object_r:etc_runtime_t:s0
/\.autofsck	regular file	system_u:object_r:etc_runtime_t:s0
/\.autorelabel	regular file	system_u:object_r:etc_runtime_t:s0
/\.ismount-test-file	regular file	system_u:object_r:sosreport_tmp_t:s0
/\.journal	all files	<<None>>
/\.snapshots(/.*)?	all files	system_u:object_r:snapperd_data_t:s0
/\.suspended	regular file	system_u:object_r:etc_runtime_t:s0
/a?quota\.(user|group)	regular file	system_u:object_r:quota_db_t:s0
/afs	directory	system_u:object_r:mnt_t:s0
/bacula(/.*)?	all files	system_u:object_r:bacula_store_t:s0
/bin	all files	system_u:object_r:bin_t:s0
/bin/.*	all files	system_u:object_r:bin_t:s0
/bin/alsaunmute	regular file	system_u:object_r:alsa_exec_t:s0
/bin/bash	regular file	
system_u:object_r:shell_exec_t:s0 ...		

Błędy polityki bezpieczeństwa

[root@localhost ~]# sealert -l "*"

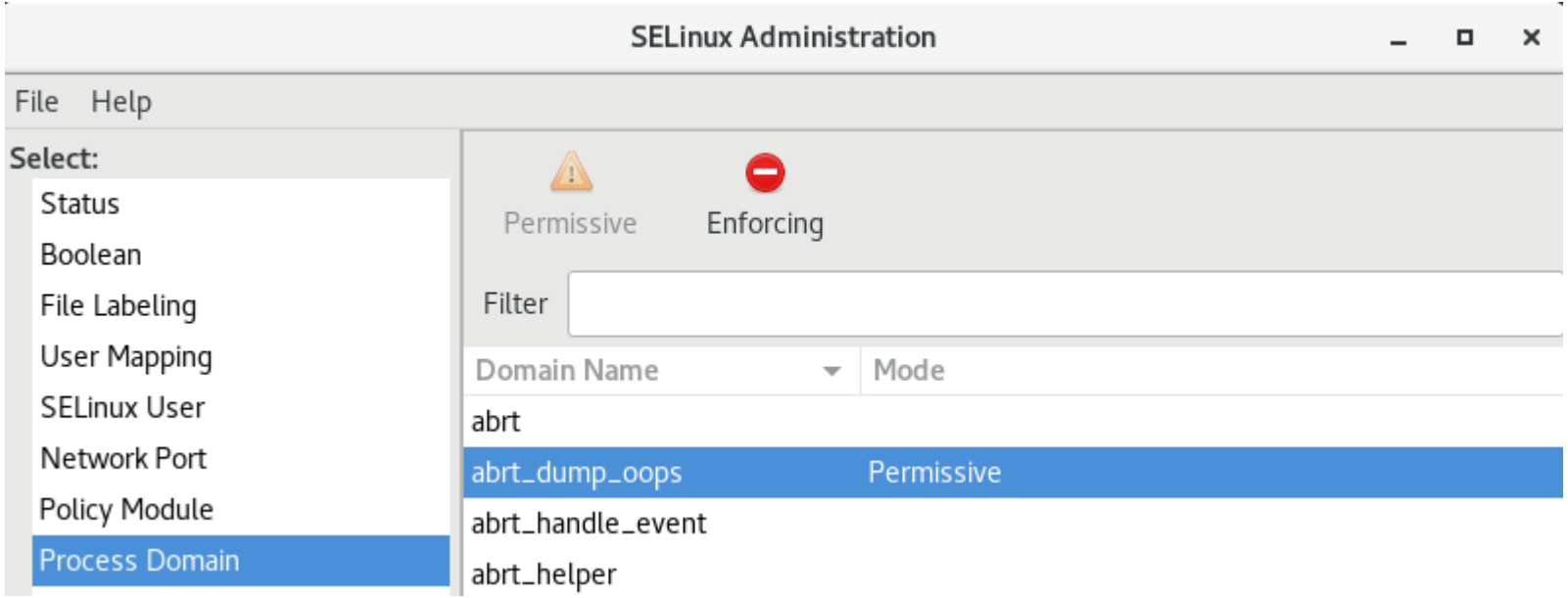
```
SELinux powstrzymuje /usr/sbin/httpd przed dostępem name_bind w tcp_socket port 180.

*****  Wtyczka bind_ports (99.5 zaufania) sugeruje   *************************

Aby zezwolić /usr/sbin/httpd na dowiązywanie do portu sieciowego 180
Wtedy you need to modify the port type.
Wykonać:
# semanage port -a -t TYP_PORTU -p tcp 180,
    gdzie TYP_PORTU jest jednym z: http_cache_port_t, http_port_t, jboss_management_port_t,
jboss_messaging_port_t, ntop_port_t, puppet_port_t.

*****  Wtyczka catchall (1.49 zaufania) sugeruje   ***************************

Aby httpd powinno mieć domyślnie name_bind dostęp do port 180 tcp_socket.
Wtedy proszę to zgłosić jako błąd.
Można utworzyć lokalny moduł polityki, aby umożliwić ten dostęp.
Wykonać:
Można tymczasowo zezwolić na ten dostęp wykonując polecenia:
# ausearch -c 'httpd' --raw | audit2allow -M my-httpd
# semodule -X 300 -i my-httpd.pp
```

Błędy polityki bezpieczeństwa - GUI



Zmiana statusu wybranych usług - GUI

**SELinux pozwala na zmianę statusu dla wybranych usług -
np. wyłączenie ich z nadzoru (nie z logowania) pomimo trybu enforcing:**



Polityki bezpieczeństwa

Dostępne polityki bezpieczeństwa w SELinux (implementacja RedHat)

- Targeted Policy
Domyślna polityka bezpieczeństwa, wykorzystująca odrębne domeny dla poszczególnych serwisów – np. httpd:

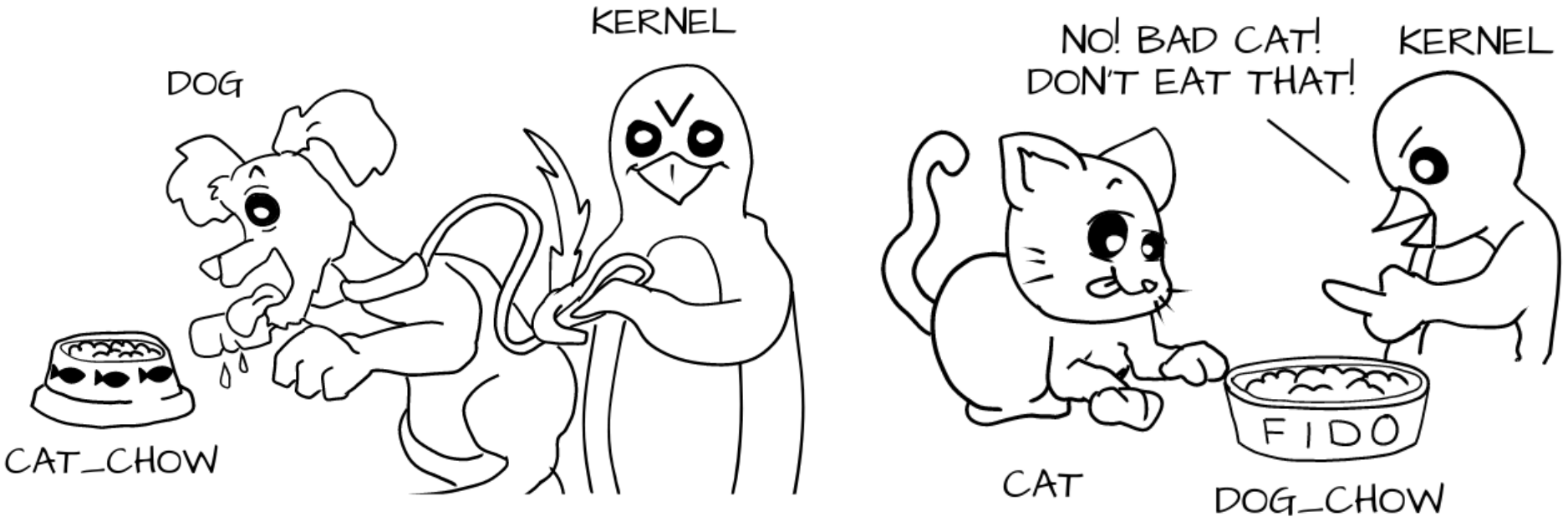
Typ obiektu (pliku *.html) musi odpowiadać domenie usługi httpd
wget http://localhost/index.html działa:

```
[root@localhost ~]# ls -lZ /var/www/html/index.html
-rw-r--r--. 1 root root system_u:object_r:httpd_sys_content_t:s0 24 Oct  1 11:48 /var/www/html/index.html
```

wget http://localhost/tmp/tmp.html nie działa – blokowane przez SELinux:

```
[root@localhost ~]# ls -lZ /tmp/tmp.html
-rw-r--r--. 1 root root system_u:object_r:tmp_t:s0 27 Oct  1 11:52 /tmp/tmp.html
```

Polityki bezpieczeństwa – targeted policy



https://people.redhat.com/duffy/selinux/selinux-coloring-book_A4-Stapled.pdf

Polityki bezpieczeństwa – targeted policy

Domyślna (i najczęściej stosowana) polityka SELinux:

- należy do rodziny “Labeled Security Protection” (dawny poziom B1)

- zadaniem jest kontrola dostępu usług (services) lub programów (subjects) do zasobów

- dostęp jest określony przez indywidualną etykietę zasobu (nie przez ścieżkę dostępu)

- zapobiega nieuprawnionej eskalacji uprawnień usługi lub programu

- w domyślnej wersji nie wpływa na systemowe uprawnienia użytkowników (także root!)

- możliwe jest selektywne ograniczanie lub poszerzanie zakresu kontroli

Polityki bezpieczeństwa -targeted policy

Targeted Policy -
Jej celem jest ograniczenie i monitorowanie dostępu procesu lub usługi jedynie do obiektów przez nią wykorzystywanych.

Ogranicza to skutki związane z nieprawidłowym działaniu usługi lub wykorzystywanych przez nią programów.

W razie potrzeby można podłączać dodatkowe usługi – np. pocztę elektroniczną:

```
~]$ sepolicy transition -s httpd_t -t system_mail_t
httpd_t @ exim_exec_t --> system_mail_t
httpd_t @ courier_exec_t --> system_mail_t
httpd_t @ sendmail_exec_t --> system_mail_t
httpd_t ... httpd_suexec_t @ sendmail_exec_t --> system_mail_t
...
```

Polityki bezpieczeństwa – moduły:

Zarządzanie modułami polityki bezpieczeństwa (policy profiles) – **semanage module -l :**

Module Name	Priority	Language
abrt	100	pp
accountsd	100	pp
acct	100	pp
afs	100	pp
aiccu	100	pp
aide	100	pp
Ajaxter	100	pp
alsa	100	pp
amanda	100	pp
amtu	100	pp
anaconda	100	pp
antivirus	100	pp
apache	100	pp
...		

Module Name	Priority	Language
my-Xorg	300	pp
my-dbusdaemon	300	pp
my-gnomeshell	300	pp
My-grub2setbootf	300	pp
my-load_policy	300	pp
my-loadpolicy	300	pp
my-polkitagenthe	300	pp
my-pulseaudio	300	pp
my-sefcontextcomp	300	pp
my-semanage	300	pp
my-systemconfigs	300	pp

Moduły (profile) polityki bezpieczeństwa (krok 1)

Profile polityki bezpieczeństwa tworzone są dla konkretnych usług (services) i są zazwyczaj dostarczane wraz z oprogramowaniem realizującym tą usługę.

Usługi systemowe są uruchamiane przez systemd i standardowo otrzymują etykietę:

```
system_u:system_r:unconfined_service_t:s0    3510   0   12:24   ?
```

sepolicy generate --init /usr/local/bin/myprogram
Generowane są następujące pliki:

```
myprogram.te              # Type enforcement file
myprogram.if              # Interface file
myprogram.fc              # File contexts file
myprogram_selinux.spec    # Spec file (for rmpbuild)
myprogram.sh              # setup script
```

Moduły (profile) polityki bezpieczeństwa (krok 1)

Polityka bezpieczeństwa (moduły *pp – Policy Profile) powinna być dostarczona wraz z określonym oprogramowaniem (np. Apache, MariaDB itp.).
Po przygotowaniu i sprawdzeniu polityki bezpieczeństwa (**`sepolicy generate`**) można wykorzystać plik **`<nazwa>_selinux.spec`** (wymagana jest instalacja **`rpmbuild`**)

Narzędzie **`sepolicy generate`** tworzy kilka plików. W pliku **`<nazwa>.te`** (type enforcing) zawarta jest domyślnie linia:

`permissive mydaemon_t;`
Jeżeli **`mydaemon`** *ma być wykorzystywany w trybie* **`enforcing`** *należy tą linię usunąć.*

Moduły (profile) polityki bezpieczeństwa (krok 2)

Uruchamiamy setup script przygotowany w poprzednim kroku:

 # /home/kazio/myprogram.sh

```
Building and Loading Policy
+ make -f  /usr/share/selinux/devel/Makefile   myprogram.pp
Compiling targeted myprogram module
Creating targeted myprogram.pp policy package
rm tmp/myprogram.mod.fc tmp/myprogram.mod
+ /usr/bin/semodule -i myprogram.pp
```

restorecon -v /usr/local/bin/myprogram /usr/lib/systemd/system
systemctl restart myprogram

```
ps -efZ | grep myprogram
system_u:system_r:myprogram_t:s0  root  5467 0  13:11 ?
```

Moduły (profile) polityki bezpieczeństwa (krok alt)

Sprawdzamy, co nam przeszkadza:

```
ausearch -c ‘httpd’ --raw
...
type=AVC msg=audit(1668156239.976:176): avc:  denied  { name_bind } for  pid=4604 comm="httpd" src=180
scontext=system_u:system_r:httpd_t:s0 tcontext=system_u:object_r:reserved_port_t:s0  tclass=tcp_socket permissive=0
...
```

Kompilujemy i instalujemy nową politykę bezpieczeństwa
nadając jej priorytet 300:

```
ausearch -c ‘httpd’ --raw | audit2allow -M my_httpd
```

```
semodule -X 300 -i my_httpd.pp
```

Multi Category Security (MCS)

Polityka MCS dla użytkowników nie jest domyślnie skofigurowana dla targeted policy, możliwe jest jednak jej stosowanie w połączeniu z polem TE:

```
# vim local_mcs_user.cil
(typeattributeset mcs_constrained_type (user_t))
...
# semodule -i local_mcs_user.cil
```

Po zdefiniowaniu nazw I powiązanych z nimi kategorii (w pliku `/etc/selinux/targeted/setrans.conf`) należy zrestratować usługę:

```
# systemctl restart mcstrans
```

Multi Category Security (MCS)

SELinux umożliwia wprowadzenie dodatkowych parametrów (kategorii) zarówno dla obiektów, jak oraz użytkowników i procesów:

```
[basia@localhost ~]$ id -Z
user_u:user_r:user_t:s0:c0.c9   # umożliwia dostęp do obiektów o kategoriach od c0 do c9
user_u:user_r:user_t:s0:c0,c1,c5   # umożliwia dostęp do obiektów o kategoriach od c0,c1,c5
```

```
[basia@localhost ~]$ ls -lZ file*
-rw-rw-r--. 1 basia basia user_u:object_r:user_home_t:s0:c50 4 11-04 10:33 file
-rw-rw-r--. 1 basia basia user_u:object_r:user_home_t:s0:c5  6 11-04 11:19 file1

[basia@localhost ~]cat file1
Basia

[basia@localhost ~]cat file
Cat: file: Permission denied
```

Multi Category Security (MCS)

SELinux umożliwia wprowadzenie dodatkowych parametrów (kategorii) dla użytkowników:

SELinux user:

```
[root@localhost ~]# semanage user -m -r s0:c0,c1-s0:c0.c9 user_u
```

Linux user: # *UWAGA – zakres nie może przekraczać zakresu SELinux user (np. user_u)*

```
[root@localhost ~]# semanage login -m -r s0:c0.c8 basia
[root@localhost ~]# chcat -L -l basia
basia: s0:c0.c8

[basia@localhost ~]$ id -Z
user_u:user_r:user_t:s0:c0.c8
```

Multi Category Security (MCS)

```
[root@localhost targeted]# more setrans.conf
#
# Multi-Category Security translation table for SELinux
#
# Uncomment the following to disable translation libary
# disable=1
#
# Objects can be categorized with 0-1023 categories defined by the admin.
# Objects can be in more than one category at a time.
# Categories are stored in the system as c0-c1023.  Users can use this
# table to translate the categories into a more meaningful output.
# Examples:
s0:c0=CompanyConfidential
s0:c1=PatientRecord
s0:c2=Unclassified
s0:c3=TopSecret
s0:c1,c3=CompanyConfidentialRedHat
s0=SystemLow
s0-s0:c0.c1023=SystemLow-SystemHigh
s0:c0.c1023=SystemHigh

[root@localhost targeted]# systemctl restart mcstrans.service
```

Multi Category Security (MCS)

SELinux umożliwia wprowadzenie dodatkowych parametrów (kategorii) dla obiektów (plików):

```
$ chcat -- +<category1>,+<category2> <path/to/file1>

$ chcat -- -<category1>,-<category2> <path/to/file1>
```

Po przyporządkowaniu kategoriom nazw (w pliku **`setrans.conf`**) możliwe jest ich wykorzystywanie:

```
-rw-rw-r--. 1 basia basia user_u:object_r:user_home_t:TopSecret 8 11-11 17:05 kasa
```

Użytkownik (subject) oraz plik (obiekt) mogą należeć równocześnie do wielu kategorii lub obejmować ich zakres (domyślnie **`c0.c1023`**)

Multi Category Security (MCS)

SELinux MCS może być szeroko wykorzystywany do separowania procesów i usług:

“Piaskownic” – sandbox zarówno w środowisku znakowym, jak i graficznym uruchamiany jest osobny X Server (w nowych wersjach Xwayland)

Maszyn wirtualnych (gości) – odseparowanie zasobów (łącznie z image) każdego gościa oraz ochrona systemu gospodarza przed atakiem za pośrednictwem maszyny wirtualnej

Kontenerów (LXC, docker, podman) – separacja usług, kontrola dostępu do zewnętrznych zasobów, wzajemna izolacja kontenerów

Orkiestratorów (OpenShift, Kubernetes…) - częściowa automatyzacja tworzenia oraz zarządzania usługami rozproszonymi.

Multi Category Security (MCS)

Dlaczego MCS, a nie MLS?

MCS (w przeciwieństwie do MLS) nie jest systemem hierarchicznym. Kategorie nie są od siebie uzależnione. MCS można wykorzystywać w trybie SELinux targeted policy.

Jednemu procesowi lub obiektowi można przyporządkować wiele kategorii

<user_r>:<role_r>:<type_t>:s0:c0.c1023 *# można definiować zakres lub pojedyncze kategorie np.:*

```
system_u:system_r:container_t:s0:c274,c305 tomus 4052 4040  0 10:18 ?
```

```
-rw-r--r--. 1 tomus tomus system_u:object_r:container_file_t\
:s0:c274,c305 0 Nov 14 10:21 file
```

Multi Category Security (MCS)

Dlaczego MCS, a nie MLS?

Kategorie mogą być przydzielane dynamicznie (ust. domyślne) lub definiowane statycznie jako “private labels” (opcje CLI, pliki konfiguracyjne itp.).

Kontener 1:

```
system_u:system_r:container_t:s0:c274,c305 tomus 4052 4040  0 10:18 ?
```

Kontener 2:

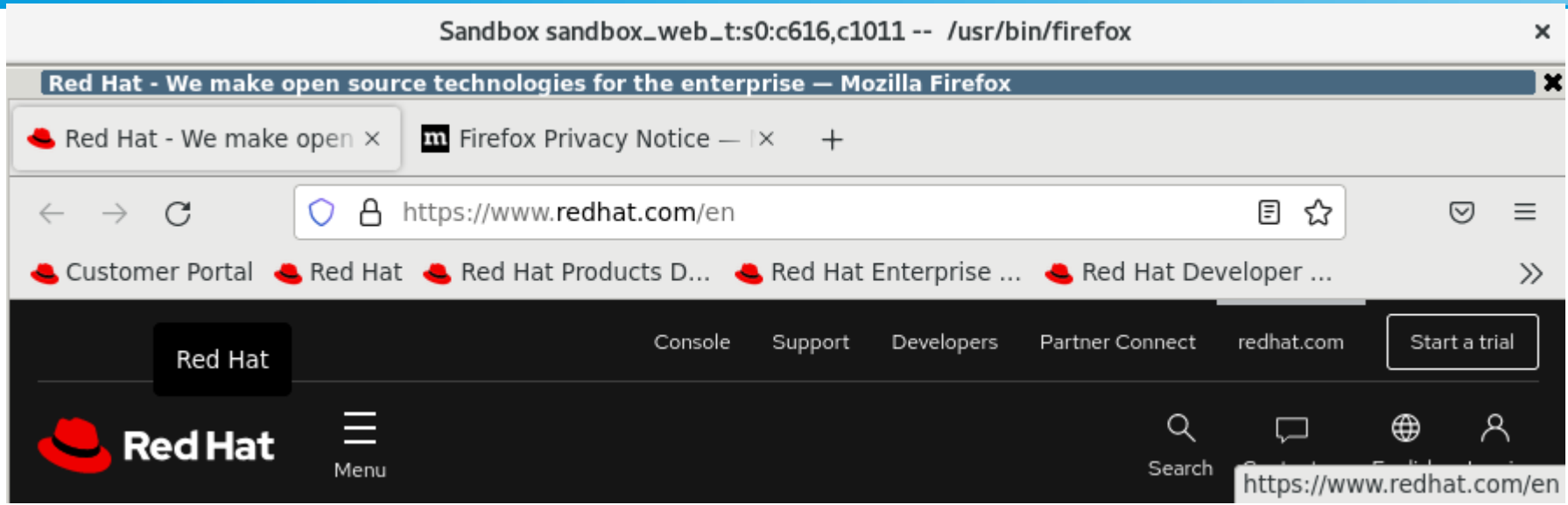
```
system_u:system_r:container_t:s0:c134,c227 tomus 4704 4510  0 10:32 ?
```

Kontener 3:

```
system_u:system_r:container_t:s0:c401,c556 tomus 4100 4778  0 11:08 ?
```

```
-rw-r--r--. 1 tomus tomus system_u:object_r:container_file_t\
:s0:c274,c305 (“s0:c134,c227” “s0:c401,c556”) file(K1, K2, K3)
```

Multi Category Security - sandbox

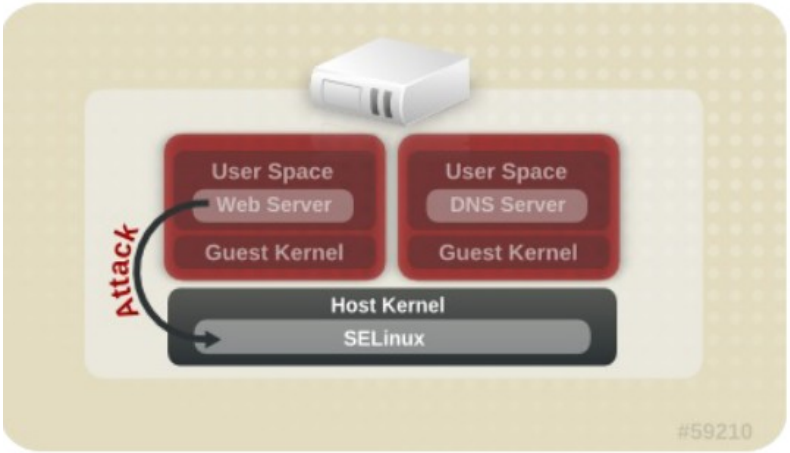


Jeśli nie wprowadzono opcji -l kategorie MCS będą przydzielane dynamicznie (jak w przykładzie).

Możliwe jest również statyczne przydzielenie kategorii MCS – np.:

```
sandbox -l s0:c100,c150 -X -H ~/sandbox/home -T ~/sandbox/tmp -t sandbox_web_t firefox
```

SELinux MCS i maszyny wirtualne



Zadaniem SELinuksa jest zapobieganie propagowania ataków z systemu operacyjnego maszyny wirtualnej na system operacyjny gospodarza -
i w konsekwencji na inne wykorzystywane maszyny wirtualne.

Do izolacji maszyn wirtualnych – gości wykorzystywane są etykiety MLS/MCS (w przypadku polityki targeted jedynie MCS):

```
[root@localhost ~]# ps -eZ | grep qemu-kvm
system_u:system_r:svirt_t:s0:c38,c305 6040 ?     00:07:11 qemu-kvm

[root@localhost images]# ls -lZ
-rw-------. 1 qemu qemu system_u:object_r:svirt_image_t:s0:c38,c305 5369757696 11-12 12:12 rhel8.qcow2
```

SELinux MCS i maszyny wirtualne

Etykiety (s – sensitivity i c – categories) są nadawane automatycznie (dynamic).
Ze względu na bardzo dużą liczbę kombinacji prawdopodobieństwo ich powtórzenia jest bardzo małe.

```
virsh dumpxml rhel8 > rhel8.xml

less rhel8.xml
...
<seclabel type='dynamic' model='selinux' relabel='yes'>
    <label>system_u:system_r:svirt_t:s0:c38,c305</label>

<imagelabel>system_u:object_r:svirt_image_t:s0:c38,c305</image
label>
  </seclabel>
...
```

SELinux MCS i maszyny wirtualne

Etykiety (s – sensitivity i c – categories) są nadawane automatycznie (dynamic).
Możliwa jest także konfiguracja statyczna:

```
...
<seclabel type='static' model='selinux' relabel='yes'>
    <label>system_u:system_r:svirt_t:s0:c10,c100</label>

<imagelabel>system_u:object_r:svirt_image_t:s0:c10,c100</image
label>
  </seclabel>
...
```

Można także wyłączyć opcję `relabel:`

```
<seclabel type='static' model='selinux' relabel='no'>
    <label>system_u:system_r:svirt_t:s0:c10,c100</label>
</seclabel>
```

SELinux i maszyny wirtualne - przełączniki

SELinux Boolean	Description
staff_use_svirt	Allow staff user to create and transition to SVirt domains.
unprivuser_use_svirt	Allow unprivileged user to create and transition to SVirt domains.
virt_sandbox_use_audit	Allow sandbox containers to send audit messages.
virt_sandbox_use_netlink	Allow sandbox containers to use netlink system calls.
virt_sandbox_use_sys_admin	Allow sandbox containers to use sys_admin system calls, e.g. mount.
virt_transition_userdomain	Allow virtual processes to run as userdomains.
virt_use_comm	Allow virt to use serial/parallel communication ports.
virt_use_execmem	Allow confined virtual guests to use executable memory and executable stack.
virt_use_fusefs	Allow virt to read FUSE mounted files.
virt_use_nfs	Allow virt to manage NFS mounted files.
virt_use_rawip	Allow virt to interact with rawip sockets.
virt_use_samba	Allow virt to manage CIFS mounted files.
virt_use_sanlock	Allow confined virtual guests to interact with the sanlock.
virt_use_usb	Allow virt to use USB devices.
virt_use_xserver	Allow virtual machine to interact with the X Window System.

SELinux i kontenery (podman)

$ podman run -dit --volume ~/src:/dest:z --name busybox busybox

$ system_u:object_r:container_file_t:s0 ./src/file

$ podman run -d --name busybox-top -v ./src:/dest:Z busybox top

$ ps -eZ | grep container_t |grep topsystem_u:system_r:container_t:s0:c260,c602 26474 pts/0 00:00:00 top

$ system_u:object_r:container_file_t:s0:c260,c602 src

$ podman run -d --name busybox-iostat -v ./src:/dest:Z busybox iostat 1

$ ps -eZ | grep container_t |grep iostatsystem_u:system_r:container_t:s0:c327,c995 26876 pts/0 00:00:00 iostat

$ system_u:object_r:container_file_t:s0:c327,c995 src

SELinux i kontenery (podman)

```
system_u:system_r:container_t:s0:c9,c135 root 19129 19119 0 12:22 pts/0 00:00:00
```

```
"MountLabel": "system_u:object_r:container_file_t:s0:c9,c135",
"ProcessLabel": "system_u:system_r:container_t:s0:c9,c135",
```

```
shm on /var/lib/containers/storage/overlay-containers/
ac54c2f3a0b648f659d777db466b7b263fd3f5b72e46b4e4808cf5d06e5884e7/userdata/shm
type tmpfs (rw,nosuid,nodev,noexec,relatime,context=
"system_u:object_r:container_file_t:s0:c9,c135",size=64000k)

overlay on /var/lib/containers/storage/overlay/
707a331596dbbf59629239ad1374a6e3c140047d7e0e3776cc1a8f5fbd5d209d/merged type overlay
(rw,nodev,relatime,context="system_u:object_r:container_file_t:s0:c9,c135"
```

Multi Level Security (MLS)

Polityka MLS jest bezpośrednią implementacją przeniesioną z praktyki ochrony informacji niejawnych.

Ochrona danych (w przeciwieństwie do MCS) ma charakter hierarchiczny:

		I
Niechronione	Unclassified	Informacje niechronione z mocy ustawy
Zastrzeżone	Company Classified	Podstawowy poziom ochrony
Poufne	Classified	Informacje istotne dla Państwa
Tajne	Secret	Istotne dla bezpieczeństwa
Ściśle tajne	Top Secret	Istotne dla obronności

Multi Level Security (MLS)

Polityka MLS jest bezpośrednią implementacją przeniesioną z praktyki ochrony informacji niejawnych. W Polsce stosowane są klauzule:

- *ściśle tajne*, jeśli ich nieuprawnione ujawnienie spowoduje wyjątkowo poważną szkodę dla Rzeczypospolitej Polskiej przez to, że: zagrozi niepodległości, suwerenności lub integralności terytorialnej Rzeczypospolitej Polskiej;
- *tajne*, jeśli ich nieuprawnione ujawnienie spowoduje poważną szkodę dla Rzeczypospolitej Polskiej przez to, że: uniemożliwi realizację zadań związanych z ochroną suwerenności, lub porządku konstytucyjnego Rzeczypospolitej Polskiej;
- *poufne*, jeśli ich nieuprawnione ujawnienie spowoduje szkodę dla Rzeczypospolitej Polskiej przez to, że: utrudni prowadzenie bieżącej polityki zagranicznej Rzeczypospolitej Polskiej;
- *zastrzeżone*, jeśli nie nadano im wyższej klauzuli tajności, a ich nieuprawnione ujawnienie może mieć szkodliwy wpływ na wykonywanie przez organy władzy publicznej lub inne jednostki organizacyjne zadań w zakresie obrony narodowej Rzeczypospolitej Polskiej.

Obiekty – są oznaczane klauzulami (classification) – dostępne są poziomy s0 – s15

Użytkownicy i programy otrzymują uprawnienia (clearances) - dostępne są poziomy s0 – s15

Ujawnianie informacji niejawnych jest zagrożone odpowiedzialnością karną!

Multi Level Security (MLS)

Zasada “no read up – not write down” + “write equality”:

Użytkownik, któremu nadano uprawnienia (clearances) w zakresie s1-s2 może:

- może:
- **czytać pliki o klauzulach s0 i s1,**
- **może modyfikować (zapisywać) pliki o klauzuli s1,**
- **zmienić swoje uprawnienia na poziom s2**

- nie może:
- **czytać plików o klauzuli s2 lub wyższej,**
- **modyfikować i zapisywać plików o klauzuli różnej od s1** **“write equality”**

Multi Level Security (MLS)

Polityka MLS jest bezpośrednią implementacją przeniesioną z praktyki ochrony informacji niejawnych – rózni się ona znacząco od domyślnej **targeted policy**:

- nie jest instalowana domyślnie – trzeba to zrobić “ręcznie”,

- nie zawiera modułu “unconfined” (unconfined_u, unconfined_r I unconfined_t) wszyscy Użytkownicy (łącznie z root) podlegają nadzorowi:

MLS:
[root@localhost ~] id -Z
staff_u:staff_r:staff_t:s0-s15:c0.c1023

Targeted:
[root@localhost ~] id -Z
unconfided_u:unconfined_r:unconfined_t:s0-s15:c0.c1023

Multi Level Security (MLS)

Polityka MLS jest bezpośrednią implementacją przeniesioną z praktyki ochrony informacji niejawnych – nowi użytkownicy są domyślnie kwalifikowani przez SELinux jako user_u (nie jako unconfided_u):

```
[root@localhost ~]# semanage login -l

Login Name           SELinux User         MLS/MCS Range        Service

__default__          user_u               s0-s0                *
asia                 staff_u              s1                   *
kasia                user_u               s1                   *
root                 staff_u              s0-s15:c0.c1023      *
tomekb               user_u               s3                   *
tomus                staff_u              s0-s15:c0.c1023      *
```

Użytkownicy o najwyższych uprawnieniach (clearances) w SELinux nie mogą “obchodzić” klasycznych praw dostępu DAC.
Uprawnienia użytkownika root wynikają z jego roli (staff_r) i są kontrolowane.

Multi Level Security (MLS)

Cenne dodatki:

Polityka MLS pozwala na efektywne rozdzielenie uprawnień (administratora od uprawnień Oficera_bezpieczeństwa).
Realizowane jest to przy wykorzystaniu RBAC – w efekcie możliwy jest podział zadań i tworzenie “protected subsystems”:

```
secofficer ALL=(ALL) TYPE=secadm_t ROLE=secadm_r ALL

standardowo obslugiwany jest moduł złożony sysadm_secadm i należy go usunąć:

semodule -d sysadm_secadm
```

Definiować “bezpieczne” terminale (np. umożliwiające wykorzystywanie ‘newrole’), zapis plików o niższym stopniu ochrony itp. Read, Please Friendly Manuals!