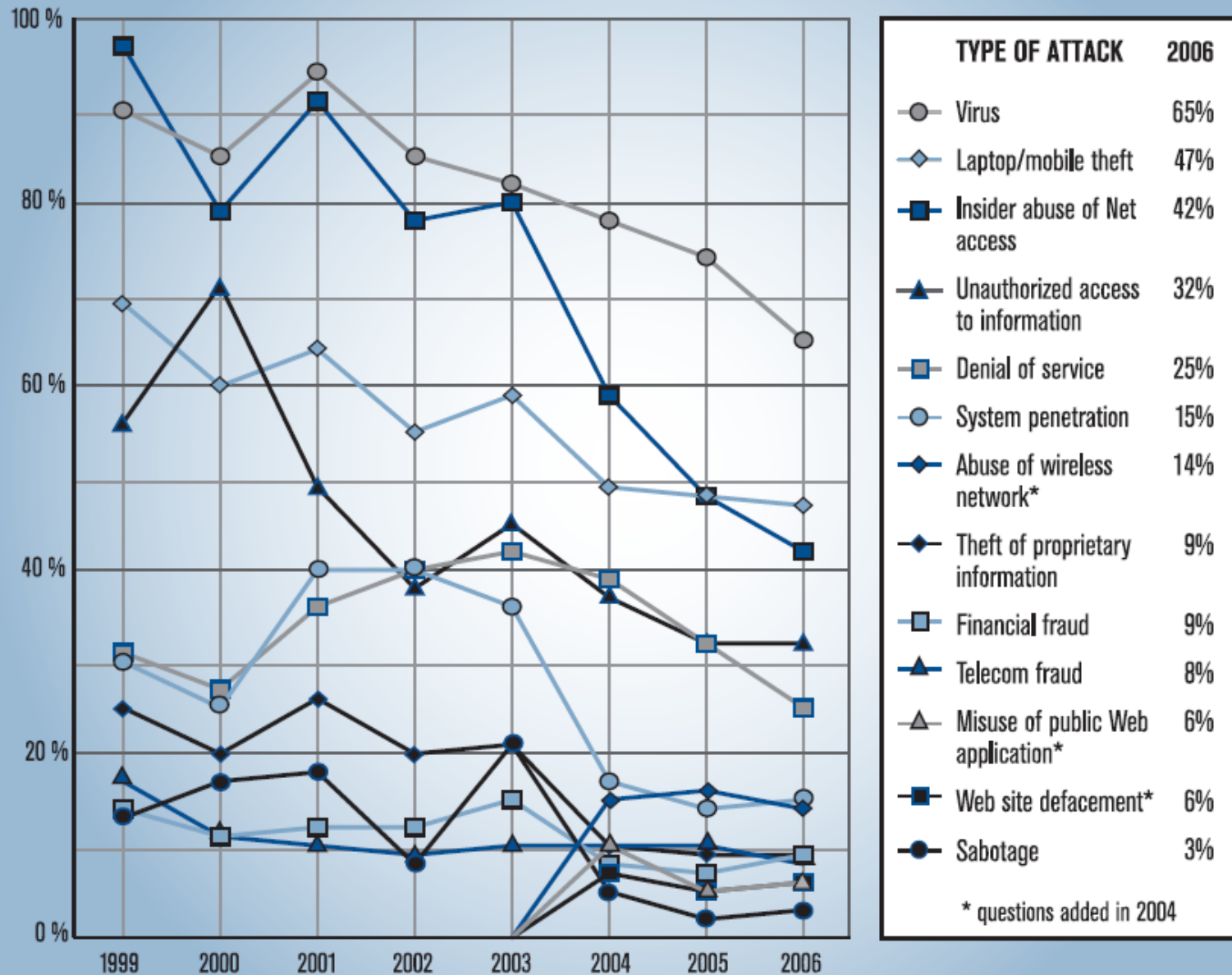


# SELinux – ~~Linux på MAC~~ MAC på Linux

Lars Strand  
7. mai 2007

# FBI/CSI rapport (2006)



02.11.0

2

- Oppdaterings-syklus:
  1. Sårbarhet oppdaget!
  2. Forfatter tetter hullet og publiserer en patch for sårbarheten.
  3. Bruker laster ned, tester og installerer patch.
- Tiden fra punkt 1 til 3 må gå raskest mulig.
- Største problem er nok punkt 3.
- Mer kjedelig problem: Tiden mellom 1 og 2.
- “Zero-day” - sårbarhet og angrepsmetode/verktøy publisert/tilgjengelig samme dag.
- 5% zero-day-angrep 2003. Hvordan beskytte?

Sikkerhetspolicy kan benytte to typer aksesskontroll (alene eller i kombo):

1. Aksesskontroll avgjøres av eier (bruker) av objektet (fil/prosessen).
2. Aksesskontroll avgjøres av OSet.

- Aksess bestemmes av *eier* av objektet.
  - Eks: Tilgang til dagbok.
  - Eks: Filer under \$HOME.

## Definisjon:

“If an individual user can set an access control mechanism to allow or deny access to an object, that mechanism is a *discretionary access control* (DAC), also called *identity-based access control* (IBAC).”

-- M. Bishop, computer security (2003).

- DAC = “Subjektbestemt aksess-/tilgangskontroll”.

- Aksess bestemmes av en *overordnet policy*.
- Policy kan ikke overstyres av subjekt (eier).
  - Eks: Sjekk av vandel.

## Definisjon:

“When a system mechanism controls access to and an individual user cannot alter that access, that control is a *mandatory access control* (MAC), occasionally called a *rule-based access control*.”

-- M. Bishop, computer security (2003).

- MAC = “Obligatorisk/overordnet aksesskontroll”.

- Linux = DAC.
  - Tradisjonell aksesskontroll.
  - Bruker X har kontroll over alle sine filer.
  - Eksekveres 'ls', kjøres den med samme rettigheter som eksekveringen av 'OpenOffice'.
  - Grov inndeling: Superbruker (uid=0) og andre brukere.
- MAC medfører gjerne en instramming.
- Ofte mer granularitet.

- Mutt – lese mine ssh-nøkler. Slette alle mine filer.
- Mutt = mailleser. Hva trenger den av rettigheter?
- Begrense rettigheter slik at en prosess fungerer normalt, og nekte alt annet.

## Definisjon:

"The principle of least privilege states that a subject should be given only those privileges that it need in order to complete its task."

-- M. Bishop, computer security (2003).

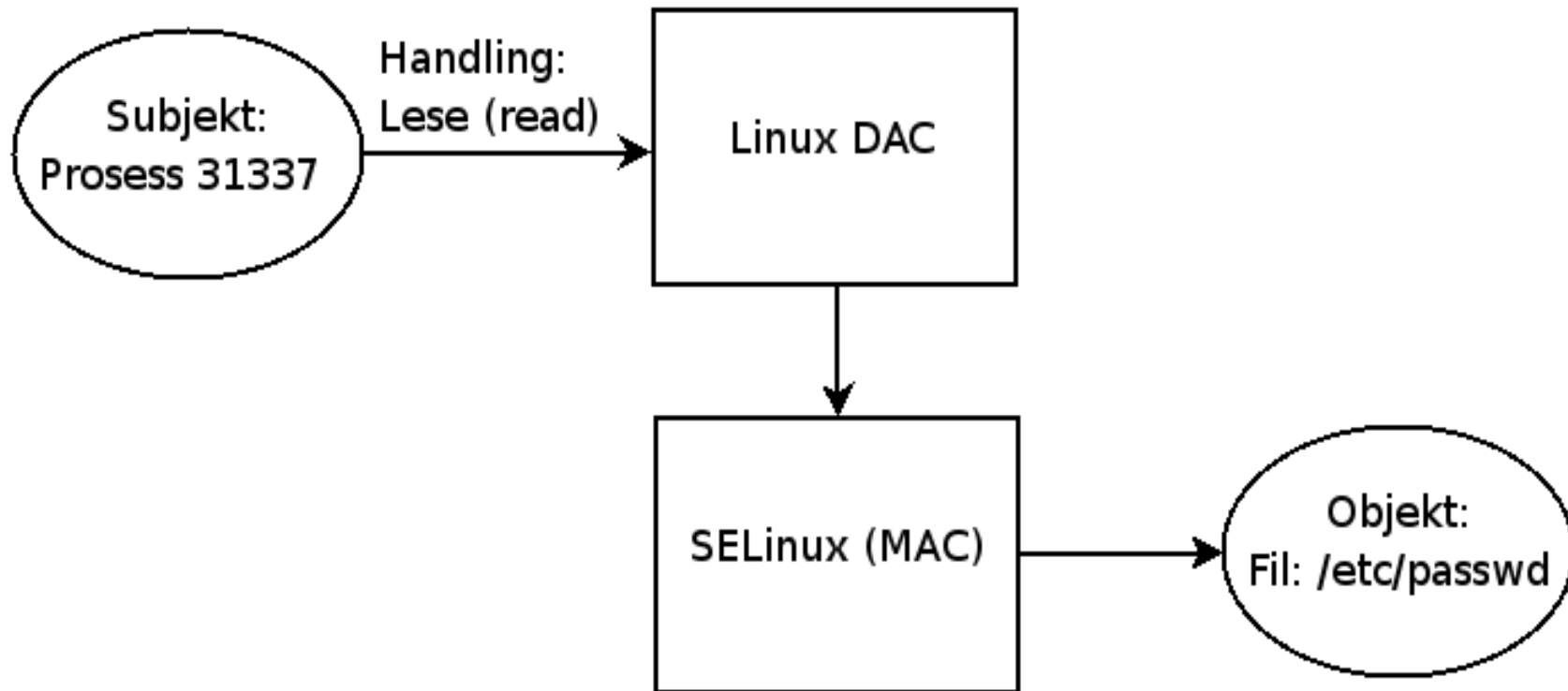


- *Security-Enhanced* Linux = SELinux.
- Opprinnlig utviklet av NSA.
- Overrasket alle ved å gjøre SELinux open source.
- I det offisielle kjernetreet fra 2.6.
- Benytter “Linux Security Modules” (LSM).
- I dag: Fedora og Redhat.
- Runner-up: Debian, Gentoo, (Ubuntu).

Må skille mellom:

- **Subjekt**
  - programmer
  - prosesser
- **Objekt (“security classes”)**
  - filer, hard- og softlinker, kataloger, sockets, ...
  - filsystemer, prosesser, ...
- **Handling**
  - Lese, skrive, legge til, eksekvere, låse, ...

- Eks: Prosess 31337 lov å lese /etc/passwd?



- Nøkkelegenskap = supplerer tradisjonell DAC med MAC:

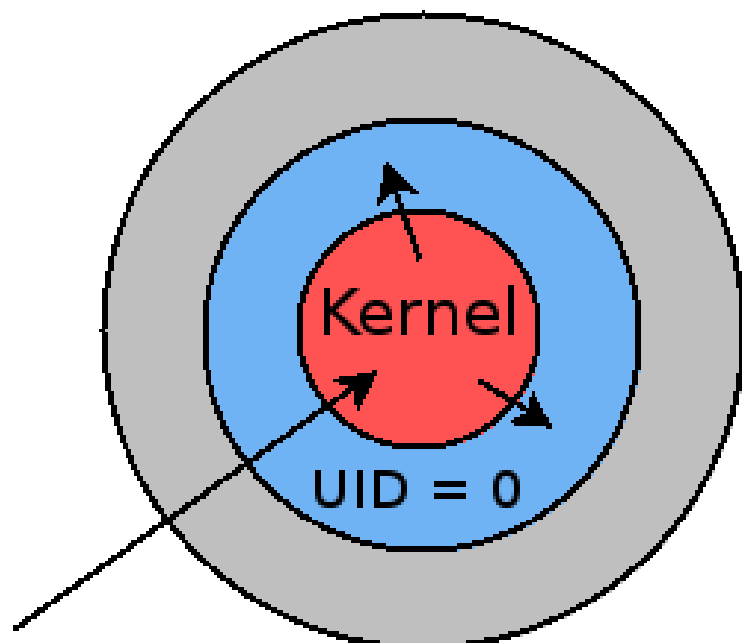
- Subjekt og objekter grupperes i ulike klasser.
- Eks: Alt som har med apache i en “klasse”/”gruppe”.
- Konstrueres vha. “security attributes”.
- Fire attributter:
  1. brukeridentitet
  2. rolle
  3. type / domain
  4. nivå og/eller kategori

- SELinux benytter **egne** brukeridenter på:
  - Subjekt (prosess): Brukeridenten som kjører prosessen.
  - Objekt (fil/program): Eier av filen.
- Separat fra Linux' DAC.
- Mapper “vanlige” brukere til SELinux'.
- Vanligvis bare en håndfull (SELinux-) brukere:
  - **user\_u**: Vanlige brukere.
  - **system\_u**: Prosesser startet (ved boot).
  - **root**: Administrator (fra konsollet).

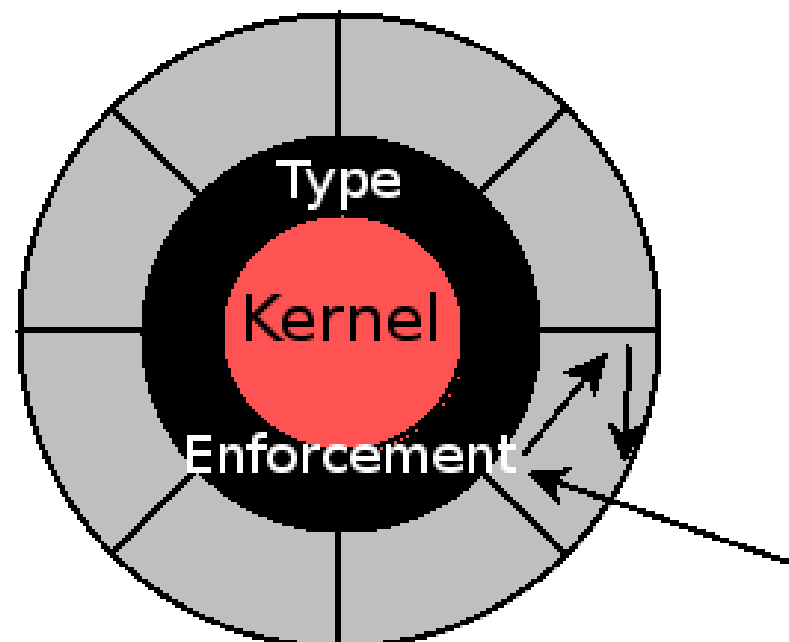
- Brukere kan tre inn i ulike roller.
- Rettigheter gis ikke til roller.
- Rollen gir “tilgang” til ulike typer / domains.
- Benyttes for prosesser, for filer er den alltid **object\_r**.

- Type, også kalt domain, er “primærattributten”.
  - 'type' for filobjekter.
  - 'domain' for prosesser.
- Få brukere/roller - flere titalls domains.
- Oppretter sandkasser som begrenser aksess.
- Type Enforcement (TE)
- “Domain transition” - regler for bytte av “sandkasser”:
  - Ny prosess. Arver den parent domain?
  - eks: `initrc_exec_t` til `httpd_t`
  - Ny fil. Arver den type?

### 3. Type / domain



Klassisk aksesskontroll (DAC).  
UID 0 har full aksess.



Domain/Type aksesskontroll (MAC).  
Kernel policy bestemmer aksess.  
Grupper med "sandkasser".



- Sette kategori og/eller nivå (horisontalt / vertikalt).
- Relativt “nytt” felt. Introdusert i RHEL5.
- Eks kategori: Driftsavdelingen, Administrasjonen.
- Eks nivå: Begrenset, hemmlig, strengt hemmlig.
- Eks nivå/kategori:
  - Hemmlig/Hæren er forskjellig fra Hemmlig/Marinen.

- Oppsummert:

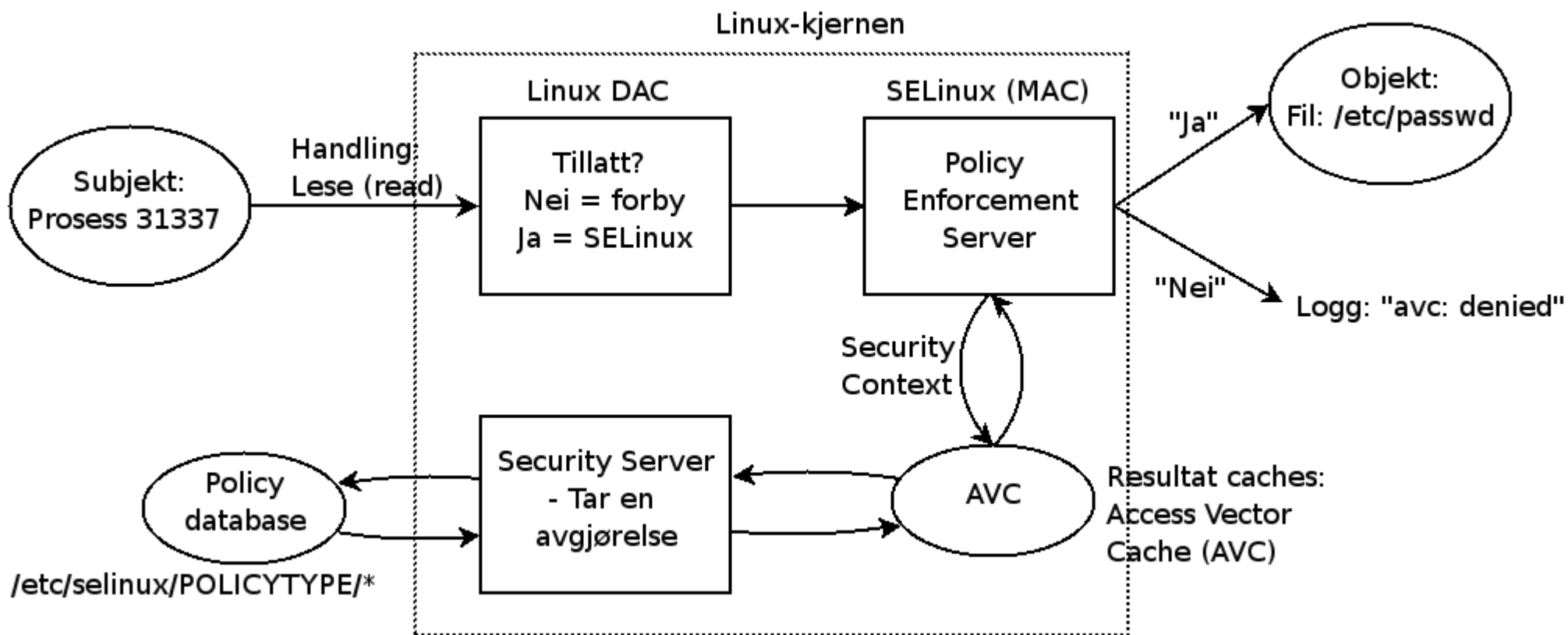
`<bruker>:<rolle>:<type>:<kategori/nivå>`

- Eks:

`system_u:system_r:unconfined_t:s0:c0`

- Disse attributtene danner en “security context”:

Security attribute	Navnkonvensjon	Eksempelnavn
Bruker	<code>_u</code>	<code>user_u</code>
Rolle	<code>_u</code>	<code>object_r</code>
Type	<code>_t</code>	<code>unconfined_t</code>
Kategori / nivå	(ingen)	<code>s:c</code>



1. Kernel
2. SELinux bibliotek 'libselinux' (ls, ps, id, ..)
3. SELinux administrasjonsverktøy
4. Policy

- /etc/selinux/POLICYTYPE/\*
- Skrevet med M4.
- Kompilert til binærformat.
- Lastes tidlig i boot-prosessen (init).
- Modifisert policy kan lastes i realtime:
  - `cd /etc/selinux/targeted/src/policy/ && make reload`
- Modulær:
  - `semodule -l`
  - `semodule -i modul.pp`

- Oppførsel (settes i /etc/selinux/config):
  - enforcing
  - permissive
  - disabled
  
- Ulike policy /etc/selinux/POLICYTYPE/policy/
  - targeted (1.1M)
  - strict (2.5M)
  - mls (2.2M)

- Et sett med daemoner omfavnet av policy (RHEL4):
  - dhcpd, apache, named, nscd, ntpd, portmap, squid,..
  - Dvs. disse har skreddersydd policy.
- Resten har “full aksess”.
  - Puttet i en “unconfined” domain.
  - 'unconfined\_t' eller 'initrc\_t'
  - Samme aksess som om SELinux var skrudd av.
- Policy enforcement kan skrur av for hver daemon.
- Ingen fokus på SELinux-bruker og -rolle i targeted.

Det er denne du vil bruke.



- **sestatus** – viser SELinux status.
- **/selinux** – egen selinux-statuskatalog (lik /proc)
- “SELinux-opsjonen” = “Z”
- Viser security context. Eks:

```
$ id -Z
```

```
user_u:system_r:unconfined_t
```

```
$ ps -Z
```

LABEL	PID	TTY	TIME	CMD
user_u:system_r:unconfined_t	11051	pts/0	00:00:00	bash
user_u:system_r:unconfined_t	11509	pts/0	00:00:00	ps

02.11.07

25

- Endre security context på filer:
  - `chcon` - for endring av security context (lik `chmod`)
  - `restorecon/fixfiles` – sjekk og fiks av security context i henhold til policy.
  - `touch /.autorelabel && reboot` (anbefalt!)
- '`tar --selinux`' – bevarer security context.
- Finne filer med bestemt security context:
  - `find / -context system_u:object_r:net_conf_t`

- Kan tune policy uten å endre, bygge og lastes på nytt.
- Kan “skru av SELinux” for utvalgte daemons.
  - Mer presist: Transition til spesifikk domain droppes.
  - Dvs. prosessen forblir i en unconfined domain.
- Definert: `/etc/selinux/targeted/booleans`
- Gjenspeilt: `/selinux/booleans/*`
- `'getsebool -a'` – for å liste.
- `'setsebool -P VARIABEL'` – for å sette.
- GUI = `system-config-selinux`

- NFS / Samba.
  - Løsning: Setter én security context ved mount:
  - `mount -o context=user_u:object_r:user_home_t ..`
- Brukervennlighet:
  - “Det fungerer ikke! Har satt riktig eierskap!”
  - \*Sjekk logger\*:
  - ```
May 4 17:54:24 valhall15 kernel: audit(1178466795.750:63):  
avc: denied { read } for pid=1038 comm="setfiles"  
name="2" dev=proc ino=131074  
scontext=system_u:system_r:setfiles_t:s0  
tcontext=system_u:system_r:kernel_t:s0 tclass=dir"
```
  - “WTF!?!?” (Skrur av SELinux)

- Økt omfavnelse.
  - dvs. mer skal beskyttes av policy.
- Økt fokus på brukervennlighet!
  - flere (grafiske) verktøy.
- RHEL legger grunnlaget for videre sikkerhetsmodellering.
  - MLS og MCS.
  - For å oppnå sertifiseringer (EAL4+, LSPP, ..)
- Lik teknologi (en konkurrent?): AppArmor.