Systems security

Carles Mateu

GREiA Research Group
Universitat de Lleida

Outline

System installation

Storage

Network

Software

Password

Running system

Filesystem

Access control and authentication

SELinux

Logging

Auditing

Outline

System installation

Storage

Network

Software

Password

Running system

Filesystem

Access control and authentication

SELinux

Logging

Auditing

System installation

When dealing with systems security, specially on servers and infraestructure security, the first step on security is a “right” installation.

A messed up installation can lead to problems hard to solve.

Distribution

Maintenance life.

Time during which we'll receive updates (specially security updates).

Desktop distributions:

- Ubuntu:
  - Release each 6 months. Back support: 2 distributions.
  - LTS Release each 2 years. Support for 5 years.
- Fedora:
  - Release each 6 months. Back support: 2 distributions.
  - Silverbullet: immutable desktop operating system.
- CentOS (RedHat):
  - Release: Major release variable. Minor release 1/year.
  - Support: 4 years, security 7 years.

Functionality

Generalistic

Good-for-all distributions, you can have servers, development workstations, etc.

- Ubuntu, Fedora, Debian, CentOS, etc.

Single purpose.

We can have different specialization degrees:

- Lower: Centos ("server")
- High.

Highly specific - Security

- Firewalls
- VPNs
- Antivirus

Usually: web interface, easy configuration, fast install, less hardware.

m0n0wall, smoothwall, Untangle, ClearOS, Zentyal, OPNsense

Virtualization

- Management consoles
- Hosts
- Easy virtual machine migrations
- Supporting multiple virtualization mechanisms

Proxmox VE, VMware ESXi, etc.

Infraestructure

- Servers:
  - Web, mail, VoIP, etc.
- Network infraestructure:
  - Routers, etc.

ClearOS, Zentyal, Untangle, FreePBX, openWRT, VitalPBX, etc

Outline

System installation
Storage
Network
Software
Password
Running system
Filesystem
Access control and authentication
SELinux
Logging
Auditing

Partitioning

Default installation:

- LVM (Logical Volume Management)
- Partitions
  - /boot : boot files
  - swap: virtual memory
  - / : root directory (rest of the space)

Partitioning

/tmp : Temporary files. Enough space (depends on use).

Must be on separate partition, all users can write here, so overflowing other spaces.

/var: Used by system daemons(servers) to store volatile or changeable data.

Create partition to avoid a daemon to overflow other services.

Partitioning

/var/log : System log files. Usually 10GB is enough, but depends on how much backwards in time we could need logs.

Needs partition to avoid any problem (thus more logging) filling up main partitions.

/var/log/audit : System audit log files. If we use audits we will need a separate partition.

Partitioning

/home : If we have user home dirs, we will need partitions.

- Avoids users from filling partitions.
- Eases migration of operating system/version (WITH CARE!).

/boot : Boot strap files.

- Required only if boot manager doesn't handles main FS.
- Useful for upgrades.
- When using *preupgrade* careful with space (>500 Mbytes).

Particionament

Cyphered partitions:

- Perfect for laptops: in case of losing the unit, data is safe.
- Not suitable for servers:
- Less performant
- Harder to repair
- (not really needed, won't loose a server)

Boot manager

- Password protected for modifications
- SINGLE BOOT!!!

Outline

System installation
Storage
Network
Software
Password
Running system
Filesystem
Access control and authentication
SELinux
Logging
Auditing

Dispositius de xarxa

Servers:

- Do not use DHCP if not required.
- If required: do not use DHCP.
- Don't.

- Server boot sequence not defined.
- Adds another weak spot for attackers (and profitable)
- Only in case of **huge** deployments with a well planned deployment.

Dispositius de xarxa

Only what is needed.

- IPv4: Usually needed.
- IPv6: As of now not needed much.
- DNS is required, use 2 or better 3.

Outline

System installation
Storage
Network
Software
Password
Running system
Filesystem
Access control and authentication
SELinux
Logging
Auditing

Software

Install only what is needed: minimum software required to do the job.

Best “trick”: remove all software selections and add just the required packet for the job (web server, database, etc.), dependency management will take care of what’s needed.

Outline

System installation
Storage
Network
Software
Password
Running system
Filesystem
Access control and authentication
SELinux
Logging
Auditing

Password

Use hard to break passwords, different for each server.

- Minimum 12 chars.
- Captitals, lower case, special chars, numbers.
- No dictionary words. Use mnemonic passwords.

Outline

System installation
Storage
Network
Software
Password
Running system
Filesystem
Access control and authentication
SELinux
Logging
Auditing

Armoring server

- **Firewall:** Activated, hard to justify otherwise on servers.
- **SELinux(similar):** Enforcing (unless we cannot get our core software to run).

Updates

- Update **immediately** after installation, checking we won't break anything we need.
- Keep the machine updated, but keep an eye on updates.
- Learn how to downgrade, check crypto signatures, protect packets, etc.
- Careful with automatic updates.

CAREFUL with 3rd party repositories.

- Security not guaranteed by vendor.
- Packet conflicts -> automated updates fail.

Outline

System installation

Storage

Network

Software

Password

Running system

Filesystem

Access control and authentication

SELinux

Logging

Auditing

File modifications

Use a local IDS: they create a database of cryptographic signatures of disk files (specially executables) to check for modifications to them.

Use it for stable systems and take into considerations updates.

AIDE

Install AIDE, create the initial database, and check periodically it. Besides, copy it to external medium, so in case of a security incident we can check what's been altered.

Devices

Device nodes (files) should only be on root partitions. Avoid any other mount point to be able to host device nodes:

- All non-root partitions should be mounted nodev

The same for removable media (USBs, etc.), and add nosuid,noexec (no SUID bits and no executables).

Filesystem

- Restrict device access only to console (difficults maintenance, make attacks hard).
- Remove USB auto mount, specially if server can be physically accessed.
- Remove auto mount of devices and non needed filesystem drivers.

File Permissions

```
- r w [xs] r w s r w t      Fitxer
1 2 3  4   5 6 7 8 9 10
```

1. Type: - file, d directory, l link, c char device, b block device, s socket, p pipe.
2. Read permission owner
3. Write permission owner
4. Execute permission owner (x) / Set UID (s)
5. Read permission group
6. Write permission group
7. Execute permission group (x) / Set GID (s)
8. Read permission other
9. Write permission other
10. Execute permission other (x) / Sticky (t)

Usually written in octal: [0-7][0-7][0-7][0-7] each one a binary representation of rwx bits.

File Permissions

- Ensure system files (/etc/passwd, /etc/shadow, etc.) have just enough permissions.
- Check for spurious SUID/SGID bits.
- Check only required users can write in special directories.

Executable Files

- If you BIOS allows you to use NX: activate it.
- Ensure you have ExecShield activated:

/etc/sysctl.conf:

- kernel.exec-shield = 1
- kernel.randomize_va_space = 1

Check it on run-time:

- sysctl kernel.exec-shield
- sysctl kernel.randomize_va_space

Root access

- If possible: restrict root to console.
- Restrict the ability to sudo to controlled users.
- Use sudo if possible.
  - Use visudo to edit /etc/sudoers.

Login restrictions

- Restrict login to system accounts.
- Lock non-root system accounts (usermod -L)
- Remove shell of non required accounts.

Outline

System installation
Storage
Network
Software
Password
Running system
Filesystem
Access control and authentication
SELinux
Logging
Auditing

Access control

- Revise password expiration policy.
  In /etc/login.defs for example:
  PASS_MAX_DAYS 60
  PASS_MIN_DAYS 7
  PASS_MIN_LEN 8
  PASS_WARN_AGE 7

Access control

- Keep each user in a separate group.
- Keep a group of human users.

- Configure single user boot to ask for authentication.
- Avoid the ability to boot in single user mode, as it give user on console root power.

PAM

- Modular authentication system.
- Dynamic objects we can load/unload for each authentication.
- Allows to configure multiple security combinations.
- Permits using new authentication systems: biometrics, smartcard, NFC keys, etc.
- In /etc/pam.d/ configuration per-server of authentication.

PAM

- We can add stuff like:
  - pam_cracklib/pam_passwdqc to check for password quality.
  - pam_tally2: Block users with failed logins.
  - pam_ldap: use LDAP to manage users.
  - pam_smb: auth against SMB server.
  - pam_krb5: use Kerberos 5 for auth.
  - pam_ccreds: cache authentication credentials.
  - pam_pkcs11: PKCS #11/NSS based authentication.

Outline

System installation
Storage
Network
Software
Password
Running system
Filesystem
Access control and authentication
SELinux
Logging
Auditing

SELinux

SELinux is an security armor mechanism that controls and mitigates the effects, of:

- Bad configurations of servers/daemons
- Access to files by part of software.
- User rights to do actions and access objects.

SELinux: Unix DAC

The UNIX DAC Model is:

- Processes belong to a user+groups.
- Files belong to user+group.
- Permissions on files (objects) to owner/group/others.
- Final permissions: function of user/group running process+file permissions

SELinux: MAC

SELinux provides MAC control (Mandatory Access Control):

- Each process/object as a security context.
- That's a label containing detailed object information.
- For a process to access an object: there must be a rule, context related, allowing that.

SELinux

- SELinux defines also context transitions (change on context driven by rules)
  - A user initiated process can be on a different context of the user's one (a more restricted one, for example).
- Doesn't substitute DAC. DAC+MAC **must** be met.
- So: SELinux only cap uppen security, not weaken it anyway.

SELinux

- It's hard to define a policy. There are some available, some installed by default.
  - targeted: TE (Type Enforcing) type rules and some RBAC (Role Based Access Control). Restricts most servers and gives lots of freedoms to *human* users.
  - strict: TE (Type Enforcing) type rules and several RBAC (Role Based Access Control). Controls most of the system software.

SELinux

- Check it in your servers:

```
[root@server ~]# sestatus
SELinux status: enabled
SELinuxfs mount: /selinux
Current mode: enforcing
Mode from config file: enforcing
Policy version: 21
Policy from config file: targeted
```

SELinux

- Existing policies are, usually either enough or a good base to build on:
  - yum search selinux-policy-*
- To Define policies use selinux-polgengui policycoreutils-gui (yum install policycoreutils-gui)
  Can add: udica setools-gui setools-console sepolicy_analysis
- **DO NOT DO THIS ON THE SECURED MACHINE. Use your desktop and move files back/forth**

SELinux

- SELinux targeted are rules letting a subject do some operations on specific objects.
- Information on those security decisions is stored on the AVC (Access Vector Cache).
- Denied decisions are logged by the audit daemon (/var/log/audit/audit.log or /var/log/messages) and setroubleshoot.

SELinux

- If we have problems (and are suspicious of SELinux), or want to check what happened agaist our SELinux:

```
ausearch -m AVC, USER_AVC -sv no
```

- Will list all negative messages from events of type (AVC and USER_AVC).

```
ps -p <PID> -Z
```

- Show the context a process is running (PID).

```
ls -Z <fitxer>
```

- Show the context a file belongs to.

SELinux

We can have denials for:

- Process not in the right context -> exec file on wrong context.
- Used object has not the right context -> reconfigure file (restorecon).
- Bad policy definition (use semanage)
- Process and objecte are right but have no policy linked them.
  - If we have already policy, fix it: getsebool/setsebool
  - Create one:

```
ausearch -m AVC -sv no -ts recent | audit2allow
```

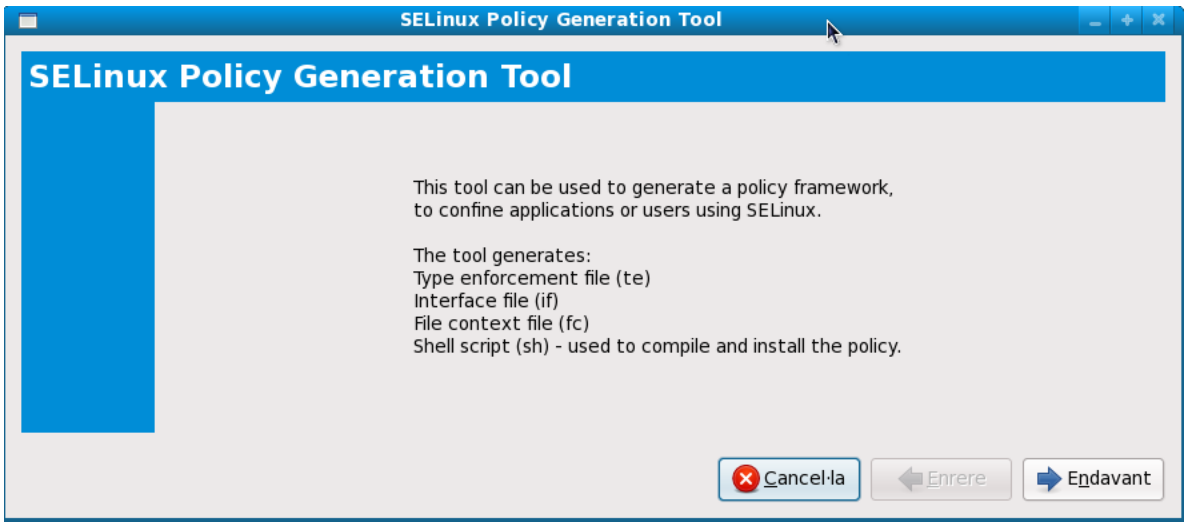
We can set individual booleans:

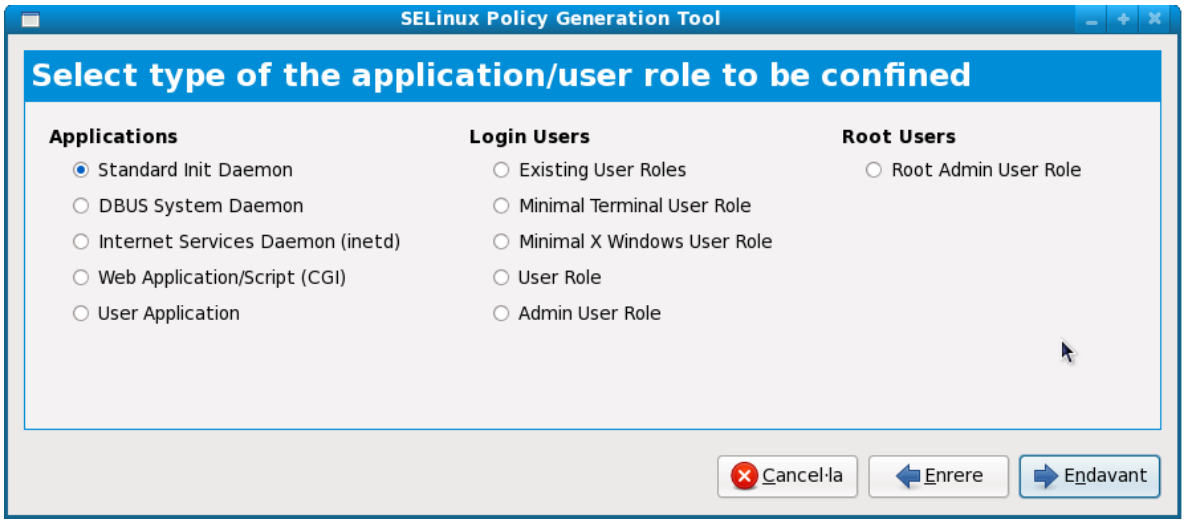
- A boolean is a variable indicating is something is allowed (globally or object-wise):

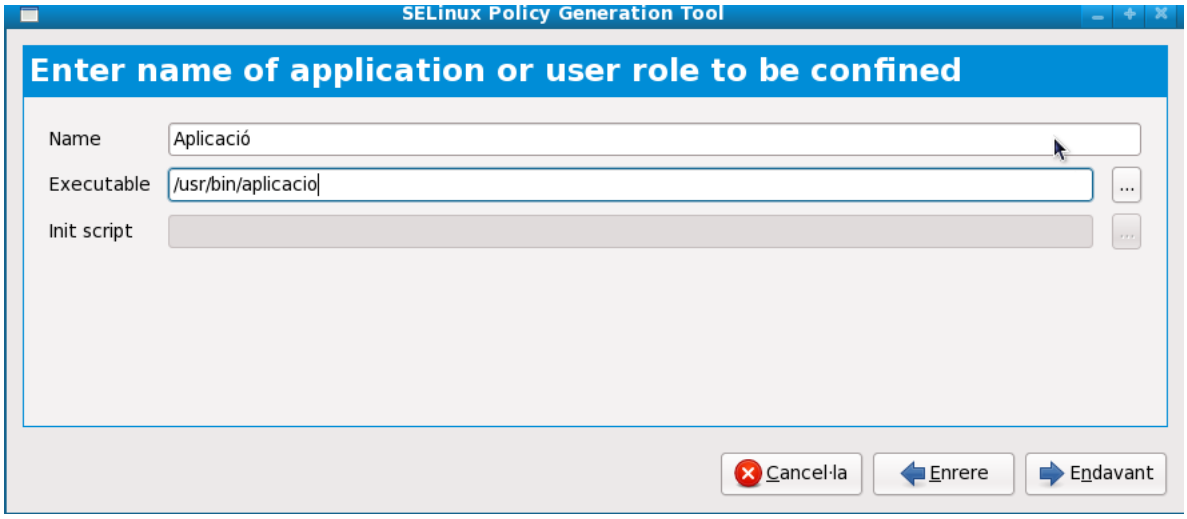
  ```
  getsebool -a
  ```

- In **/usr/share/doc/selinux-policy-.../html/** there's detailed configuration on all them.

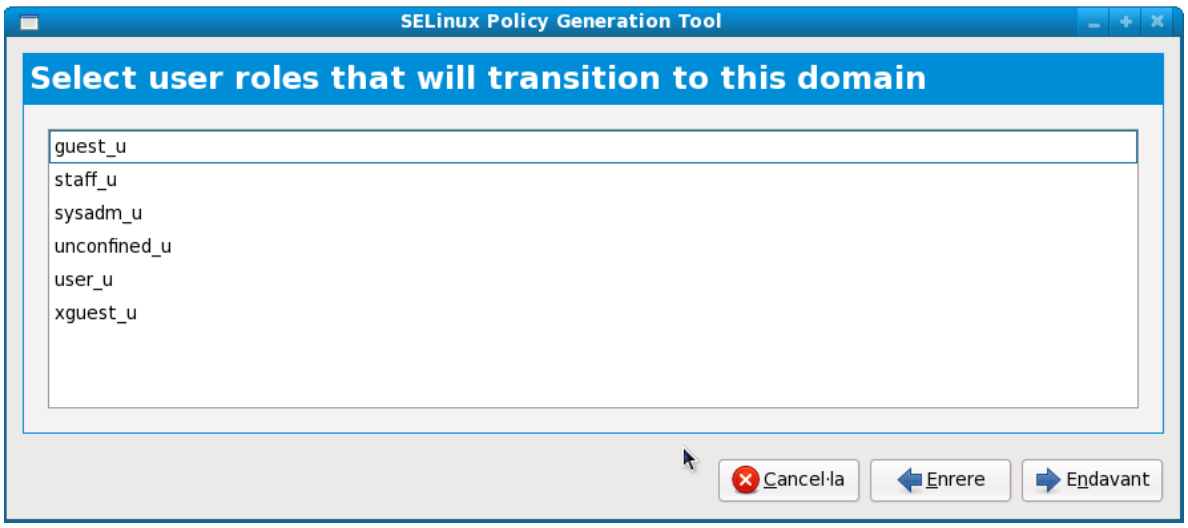
SELinux: selinux-polengui



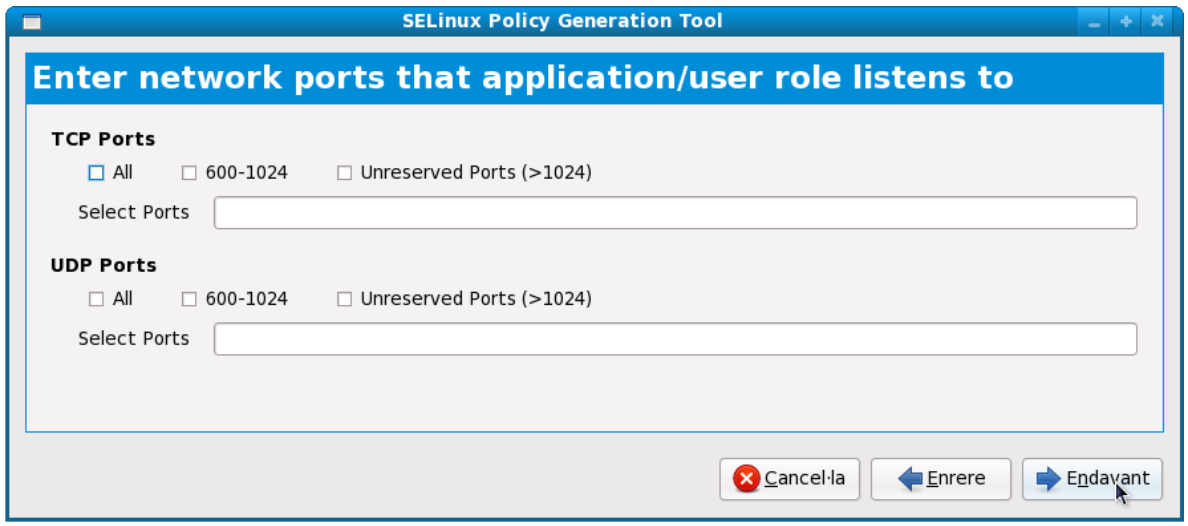

SELinux Policy Generation Tool
Select type of the application/user role to be confined
Applications
Standard Init Daemon
DBUS System Daemon
Internet Services Daemon (inetd)
Web Application/Script (CGI)
User Application
Login Users
Existing User Roles
Minimal Terminal User Role
Minimal X Windows User Role
User Role
Admin User Role
Root Users
Root Admin User Role
Cancel·la
Enrere
Endavant


SELinux Policy Generation Tool
Enter name of application or user role to be confined
Name
Aplicació
Executable
/usr/bin/aplicacio
Init script
Cancel·la
Enrere
Endavant

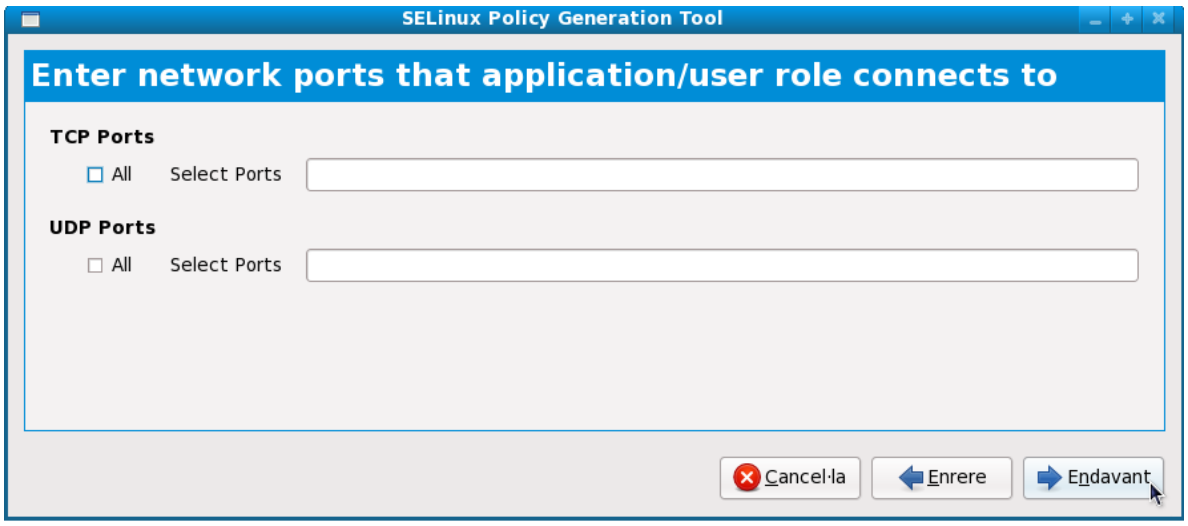
SELinux: selinux-polengui



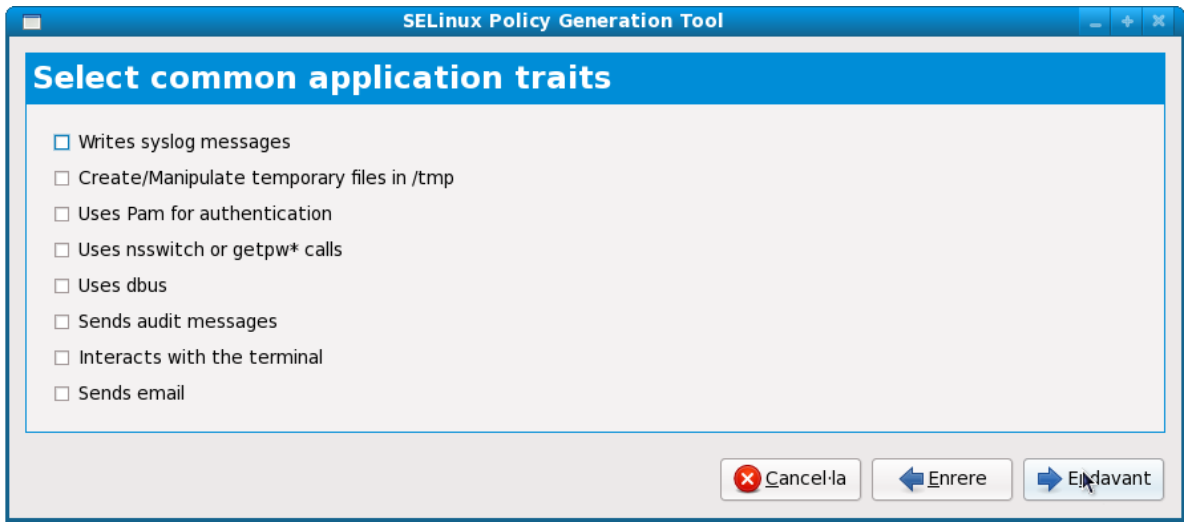
SELinux: selinux-polengui



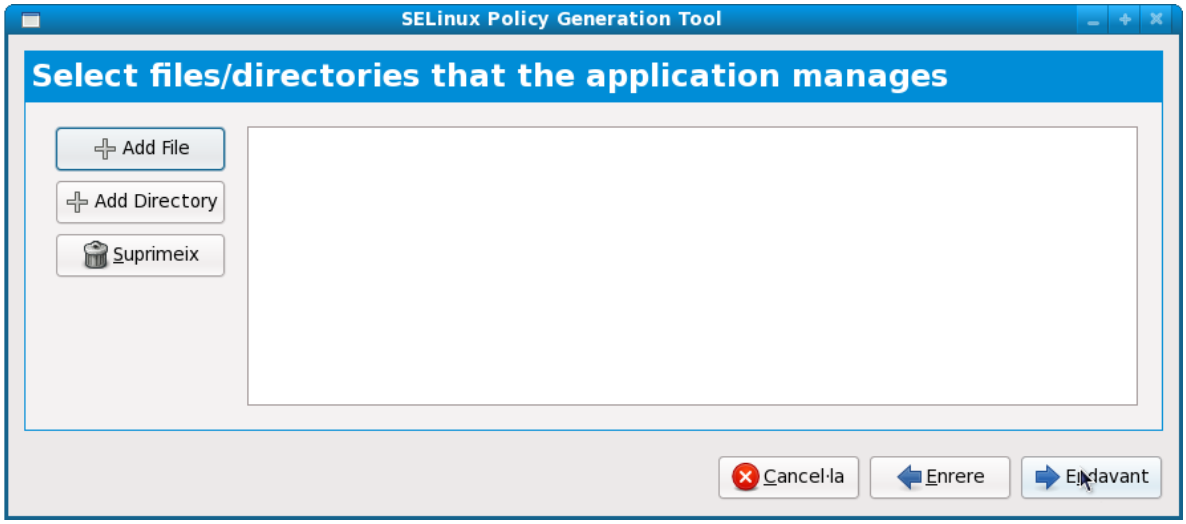
SELinux: selinux-polengui



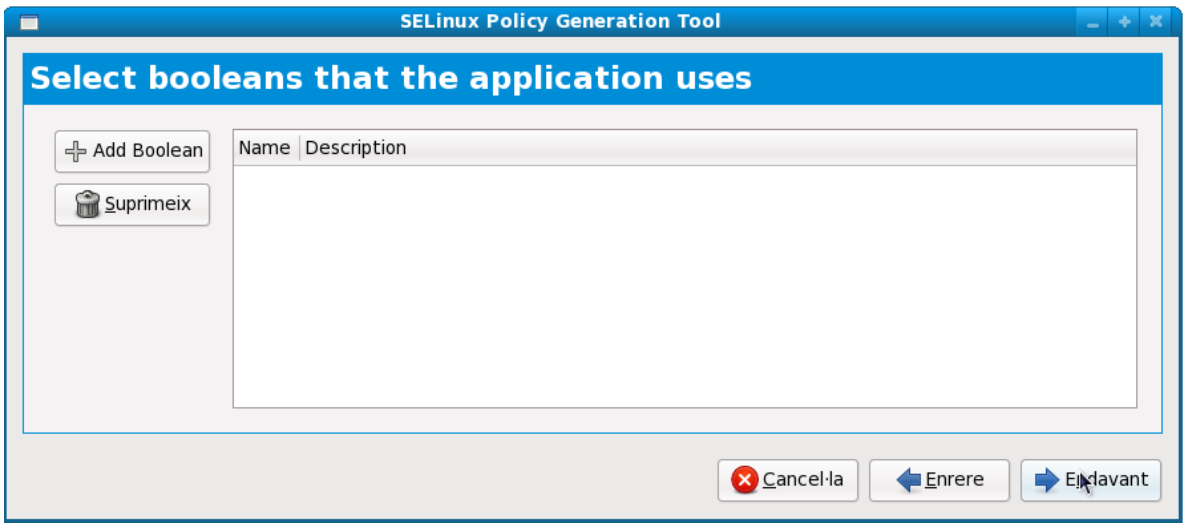
SELinux: selinux-polengui



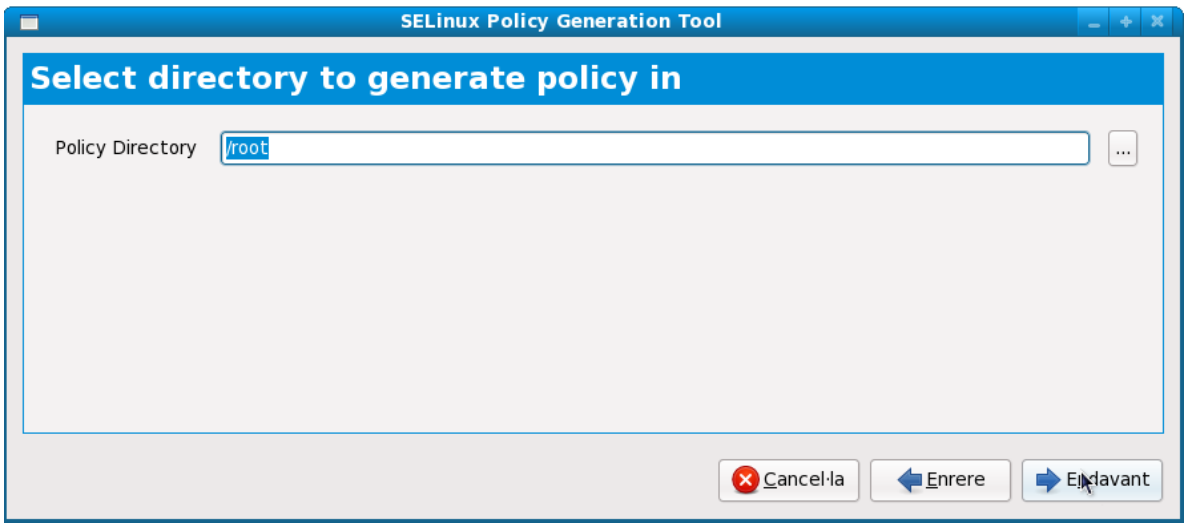
SELinux: selinux-polengui



SELinux: selinux-polengui



SELinux: selinux-polengui



SELinux: other policies

If we change the policy:

- minimum
- targeted
- strict
- MLS

(In increase security level order). targeted is recommended, strict if we can afford it.

If we change policy, we have to relabel disk:

- touch /.autorelabel
- autorelabel (as grub parameter)

Outline

System installation
Storage
Network
Software
Password
Running system
Filesystem
Access control and authentication
SELinux
Logging
Auditing

Logging

Ensure logging configuration is correct (/etc/syslog.conf)

```
kern.*                                          /var/log/kernel.log
auth,user*                                      /var/log/messages.log
daemon.*                                        /var/log/daemon.log
syslog.*                                        /var/log/syslog.log
*.info;mail.none;authpriv.none;cron.none        /var/log/messages
authpriv.*                                      /var/log/secure
mail.*                                          -/var/log/maillog
cron.*                                          /var/log/cron
*.emerg                                         *
uucp,news.crit                                  /var/log/spooler
local0,local1,local2                            /var/log/unused.log
local3,local4,local5                            /var/log/unused.log
local6.*                                        /var/log/unused.log
local7.*                                        /var/log/boot.log
```

Logging

- Check log file permissions (must exist a priori and be: root:root, 0600).
- Check that log rotation is working:
  - Review /etc/logrotate.conf and /etc/logrotate.d/*
  - Have enought disk for storing them

Logging

On big or critical deployments: send logs to log processing servers.

- On /etc/syslog.conf

  ```
  *.* @serverlog.udl.cat
  ```

- On the log server, on /etc/sysconfig/syslog:
  `console SYSLOGD_OPTIONS="-m 0 -r -s udl.cat"`

On the log server provisioning enough disk space, and having correct log rotation configuration is critical.

Logging

On the log server use log monitoring tools (or if we don't have central log server on as many servers as possible).

Logwatch

One simple tool we can use for that. Minimal configuration, on /etc/logwatch/logwatch.conf:

```
HostLimit = no
SplitHosts = yes
MultiEmail = no
Service = -zz-disk_space
```


Outline

System installation
Storage
Network
Software
Password
Running system
Filesystem
Access control and authentication
SELinux
Logging
Auditing

Auditing

We will have auditing tools enabled.

- On Linux we have auditd (auditd complies with DoD requirements).
- Activate: **chkconfig auditd on**
- Data is kept on: /var/log/audit/audit.log

Auditing: auditd

- Configured on /etc/audit/audit.conf
- Ensure enought space.
- On critical systems it's recommended that in case of unable to keep auditd: system should halt.
- Auditd keeps usually 4 files of *SIZE* size by default.
  - max_log_file = *SIZE*

Auditing: auditd

- Recommended: one partition for audit data.
- Configure for halt on space issues:

```
space_left_action = mail
action_mail_acct = root
admin_space_left_action = halt
```

- Auditd rotates logs to bypass space constraints. To disable rotations:

```
max_log_file_action = keep_logs
```


Auditing: auditd

- Make init processes use auditd:

```
audit=1     ## on grub2.cfg
```

- We have prepared rules on (/usr/share/doc/audit*/*rules)
  - capp.rules lspp.rules nispom.rules stig.rules
- Corresponding to DoD/NSA recommendations.
- STIG probably most insteresting (enough).
- cp /usr/share/doc/audit*/stig.rules /etc/audit/audit.rules

Auditing: auditd

- aureport will show us events.
- Login tries today and yesterday:

```
aureport -l -i -ts yesterday -te today
```

- To list anomalies:

```
aureport --anomaly
```


Auditing

```
Summary Report
======================
Range of time in logs: 02/10/20 22:37:31.913 - 04/10/20 17:26:15.754
Selected time for report: 02/10/20 22:37:31 - 04/10/20 17:26:15.754
Number of changes in configuration: 10
Number of changes to accounts, groups, or roles: 0
Number of logins: 1
Number of failed logins: 7898
Number of authentications: 3
Number of failed authentications: 22801
Number of users: 2
Number of terminals: 4
Number of host names: 527
Number of executables: 4
Number of commands: 2
Number of files: 8
Number of AVC's: 30
Number of MAC events: 1
Number of failed syscalls: 0
Number of anomaly events: 0
Number of responses to anomaly events: 0
Number of crypto events: 79266
Number of integrity events: 0
Number of virt events: 26
```