SELinux

Red Hat

Michal Marciniszyn

10th November 2010

Agenda

1 History

2 MAC vs DAC

3 Implementation

4 Benefits, Drawbacks & Myths

5 Example

Section 1

History

SELinux History

- Originally developed by NSA as an implementation of Flask OS security architecture.
- First introduced in Linux by Fedora Core 2.
- FC 2 SELinux in strict mode was not accepted by community.
- Targeted policy was developed to be useable by broad community and is nowadays part of several distros.

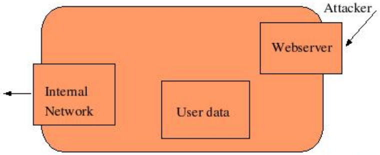
Section 2

MAC vs DAC

What is DAC?

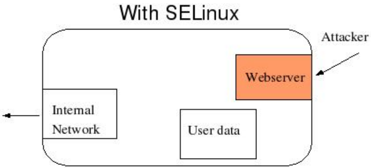
- DAC stands for Discreet Access Control.
- Processes are given rights based on UID and GIDs.
- In other words, webserver run by root can perform any action root can perform.

Without SELinux



What is MAC?

- MAC stands for Mandatory Access Control.
- Processes are given rights based on what they need to perform.
- Webserver cannot access /home/ directory.



Section 3

Implementation

What is Security Policy

- Defines Security Context:
  - User identity - defines which roles user can act as.
  - Role - defines which types user can access.
  - Type/Domain - main security distinction.
  - Sensitivity - MLS policy, used solely by military.
  - Category - security 'groups'.
- Defines how each domain may access each type.
- Defines transitions and other access.

Targeted policy

- Targeted policy uses type enforcement - each object and subject has security context.
- Only type is considered in targeted policy.
- Context is stored in xattrs for files, in kernel for ports, NIS, etc.
- Format: *user:role:type:sensitivity:category*.
- Users run in special domain - *unconfined_t* not restricted by SELinux.
- Policy is tuneable using booleans that may allow certain functionality on demand (httpd can read /public_html)

Tools

- Whole toolset has been adjusted to work with SELinux, usually by adding -Z option.
- ls -Z, ps -Z, id -Z, install -Z, mkdir -Z, find -context
- We have to be aware of different behaviour of cp vs. mv.
- tar & star can archive SELinux context and are still backwards compatible.
- getfattr -d -m security.selinux -R /etc/
- setfattr -h –restore=/tmp/backup.xattrs
- getenforce, setenforce, getsebool, setsebool

How does it work?

- Files get context from the parent directory.
- Contexts are defined in the policy.
- When a daemon is started, transition rule states which domain it will run in.
- Policy states which access is allowed. Everything is disabled by default.
- Violation is logged.
- Policy runs in kernel, to tackle SELinux, we have to exploit kernel.

redhat.

Section 4

Benefits, Drawbacks & Myths

Performance

- Performance hit is few percent.
- Few space on disk is needed to store context.
- Usually one running daemon called auditd to log denials.

Benefits

- We can restrict allmighty root.
- Programs get only privileges they need.
- Protects against exploits.
- Divides security administrator vs. application administrator.

Myths

- SELinux cannot perform code audits.
- No encryption of data.
- Services need to be updated.

Section 5

Example

Dummy web page

We show how SELinux can prevent our box from very bad web page.

Question and answers...

The end.

Thanks for listening.