A Re-Introduction to SELinux
Why a Re-Introduction?

- SELinux applies virtualization, container isolation and Android recently.

- However, many server engineers have disabled SELinux up to now.

It means “disable”
Agenda

- SELinux Overview

- SELinux Access Control Model
  - Type Enforcement (TE)
  - Role-based Access Control (RBAC)
  - Multi-level Security (MLS) / Multi-category Security (MCS)

- SELinux Security Policy
  - Strict Policy (deprecated)
  - Targeted Policy
  - Minimum Policy
  - MLS/MCS Policy
A Re-Introduction to SELinux

SELinux Overview
What is SELinux?

- SELinux is a security extension for Linux kernel
  - Developed by NSA
  - Policy processing based on FLASK architecture
  - One of Linux security module implementation

- SELinux aims military level security

- Reference monitor (=linux kernel) forces restriction of application behavior

- Demolish *root privilege* for the principle of least privilege
MAC: Mandatory Access Control

- Giving SELinux context to all resources

- Linux security module hooks can mediate system calls completely

- SELinux cannot force application-internal access control
  - However, several application embedded SELinux hook into itself
    - X Window System (XACE)
    - PostgreSQL (SE-PostgreSQL)
    - Systemd
    - D-Bus
Benefits

- Fine-grained (Process-level) access control
- Strong isolation
- Abolish root privilege
SELinux is not

- Antivirus software
- Intrusion detection
- Memory protection
A Re-Introduction to SELinux

SELinux Access Control Models
SELinux Access Control Models

- TE: Type Enforcement
- RBAC: Role-based Access Control
- MLS/MCS: Multi-level Security/Multi-category Security
Type Enforcement

- The type defines a type for files
- The domain defines a type for processes
- The rule defines permission between domain and type
Domain transition

- Child process inherits SELinux context from parent process normally when process forked
Domain transition

- SELinux supports process domain transition when process execution using type of execution file.
Example 1: myapp.te

# Type definition
type myapp_t;
type myapp_exec_t;

# Declaring myapp_t
domain_type(myapp_t)

# Assigning myapp_t when process executes from myapp_exec_t file
domain_entry_file(myapp_t, myapp_exec_t)

# Type definition
type myapp_log_t

# Declaring interface
logging_log_file(myapp_log_t)

# myapp_t domain can read and append for myapp_log_t files
allow myapp_t myapp_log_t:file { read_file_perms append_file_perms };
Describing Rule of Type Enforcement

- SELinux TE rule declaration is primitive
  - SELinux prepare some expressive macros, but…

- Strictness policy description is composed by huge TE rules
  - It is difficult to understand for user (incl. security administrator)
SELinux Policy Module

- SELinux TE rule modularization system
- Can load each policy modules
  - Can defines interfaces of the module
  - Reusable utilities for security administrator

See also:
“Getting Started with Reference Policy”
http://oss.tresys.com/projects/refpolicy/wiki/GettingStarted
RBAC (Role-based Access Control)

- A user is assigned to some roles
- Role is granted some permissions
SELinux Users and roles

- SELinux maps Linux users on SELinux users
  - Moreover, maps SELinux users on SELinux roles
  - Because Linux user controls under discretionary access control its not mandatory

Major Linux users:
- user1
- root

SELinux users:
- user_u
- root
- staff_u
- system_u
- sysadm_u
- unconfined_u

Major roles:
- user_r
- staff_r
- system_r
- sysadm_r
- unconfined_r
MLS/MCS (Multi-Level/Category Security)

- Multi level security is an access control using security level and category
  - Security level based on a rank of organization (like army)
  - Isolating access from other category

\[ MLS = \text{Security level} \times \text{Category} \]
Bell-LaPadula Model

- Mathematical model of Multi level security
  - Main interests is information leakage prevention
- Users cannot read information from upper rank document
  - Users can write report to upper rank document
- User can read information from lower rank document
  - Users cannot write report to lower rank document

<table>
<thead>
<tr>
<th>Security level</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>s0</td>
<td>Users can read from lower (incl. same) rank document only</td>
</tr>
<tr>
<td>s1</td>
<td>Users can write to upper (incl. same) rank document only</td>
</tr>
<tr>
<td>s2</td>
<td></td>
</tr>
<tr>
<td>s3</td>
<td></td>
</tr>
</tbody>
</table>
Access Control using MLS/MCS

- Assigning default security level and category for all resources
  - In addition, users are assigned clearance

- MLS permissions follows in Bell-LaPadula Models

- Different category access requires clearance

Default security level & category  clearance

\[ S0 - s0:c0.c1023 \]

MLS Contexts
File contexts (SELinux Contexts on Files)

- A file context enumerates the mapping between resource and SELinux context

- File contexts are stored xattr(or xattr like file system extension) by labeling scripts
  - SELinux requires labeling before enforcing access control

Target
/bin/systemd - system_u:object_r:init_exec_t:s0

SELinux User
RBAC: Role
TE: Type
MLS/MCS Security level
SELinux Modes

• SELinux has three modes
  – Enforcing: Enabling SELinux access control
  – Permissive: Policy check only
  – Disabled: SELinux disabled

• Checking SELinux status on your host:
  – Run sestatus command

• If you want change the SELinux mode on boot time:
  – Edit SELINUX=enforcing/permissive in /etc/selinux/config
sestatus

- Show current SELinux configurations

```
# sestatus
SELinux status: enabled
SELinuxfs mount: /sys/fs/selinux
SELinux root directory: /etc/selinux
Loaded policy name: targeted
Current mode: enforcing
Mode from config file: enforcing
Policy MLS status: enabled
Policy deny_unknown status: allowed
Max kernel policy version: 29
```
See Also: stopdisablingselinux.com

Seriously, stop disabling SELinux.

Learn how to use it before you blindly shut it off.

Every time you run setenforce 0, you make Dan Walsh weep.

Dan is a nice guy and he certainly doesn't deserve that.

Fan of SELinux? Get the t-shirt!

A public service from Major Hayden
A Re-Introduction to SELinux

SELinux Security Policy
Current SELinux Security Policies

- **Type Enforcement(+RBAC)**
  - Strict (deprecated)
  - Targeted
  - Minimum

- **Multi Level/Category Security**
  - MLS
  - MCS

Default policy is **Targeted + MCS** on Fedora, RHEL and RHEL-based distributions
Strict policy (deprecated)

“SELinux designed to be a strict policy.” – Dan Walsh (2005)

- Restrict all processes completely
  - Using both TE and RBAC policies

- NSA and Red Hat encouraged strongly
  - However, SELinux operation cost is unsuitable for the security

- Everybody thought that SELinux operation using strict policy is impracticable
  - Red Hat integrates a part of strict policy into targeted policy
Targeted Policy

• Restrict some process
  – Only Internet server(httpd, named…) and network service
  – Preparing against privilege escalation

• Permit to run `unconfined_t` type/domain with no limitation
  – Ex. Login shell execute with unconfined_t domain

• In Fedora20, Red Hat prepared a lot of predefined policy modules
  – If you want to enforce with strict like policy, remove unconfined_t, unconfined_r
Predefined Policy Modules in Targeted Policy on Fedora20

Minimum Policy

• New policy type since Fedora10
  – Minimum policy focus to reduce memory and storage usage consumption

• Suitable size for embedded systems, container, cloud components
  – Policy binary file size reduced to 2.0MB(minimum) from 3.5MB(targeted)
Multi Level Security Policy (MLS)

- MLS policy enforces MLS access control
  - Default security level is s0, category is nothing

- Practically, military use only

- Cannot run on X window system
  - MLS policy is unsuitable for X

```
# ls -laZ
--snip--
lrwxrwxrwx. root root system_u:object_r:bin_t:s0 sbin -> usr/sbin
drwxr-xr-x. root root system_u:object_r:var_t:s0 srv
dr-xr-xr-x. root root system_u:object_r:sysfs_t:s0 sys
drwxrwxrwt. root root system_u:object_r:tmp_t:s0 tmp
drwxr-xr-x. root root system_u:object_r:usr_t:s0 usr
drwxr-xr-x. root root system_u:object_r:var_t:s0 var
```
Multi Category Security Policy (MCS)

- MLS enables casual resource isolation
  - Red Hat applied MLS to sandbox, container, virtualization

- Enabling since FedoraCore6
  - However, every user has the right of access to category 0-1023
  - All resources have no assigned category

```bash
# id -Z
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```
To continue enforcing SELinux

- Take care with file context
  - Verify existing SELinux context using `semanage`
  - `sealert` command recommends good labeling to you when access violation has occurred

- You cannot change SELinux context, generate policy module using access violation log
  - Using `audit2allow` command
    - You *MUST* confirm generated policy source
Quick Reference

- Sestatus
  - Show SELinux status

- Semanage module/user/login/fcontext -l
  - Show SELinux configuration on your system

- sealert -a /var/log/audit/audit.log
  - Recommend good labeling to you when access violation has occurred
Quick Reference (2)

- Chcon –t hogehoge_t /var/www/hogehoge/index.html
  - Changing SELinux context (temporary)

- Semanage fcontext –a –t hogehoge_t “/var/www/hogehoge(/.*?)”
  - Changing SELinux context (permanently)

- Restorecon –rv /var/www
  - Applying file context rule to directory
Conclusion

- Each SELinux access control model is simple, but actually access control is more complex

- Red Hat puts a lot of effort into SELinux, policy and utils for SELinux usability
  - Enlarging default policy modules
  - Encouraging Policy module system
  - Analyzing and generating policies from access violation log
Appendix: SELinux history

- 2003
  - Merged Linux kernel 2.6
- 2004
  - Enabling SELinux default on FC2 (targeted)
- 2006
  - Policy reconstruction with reference policy (semanage, policy module)
  - Full labeled networking support
  - setroubleshot developed by Red Hat
  - MCS debut on FC5
- 2007
  - Xguest developed by Dan Walsh
  - SE-PostgreSQL developed by Kohei Kaigai
  - Strict policy sunked on Fedora 8 (merged targeted policy)
- 2009
  - sVirt presented by Red Hat
  - SELinux Sandbox developed by Dan Walsh
- 2012
  - SE-Android developed by NSA
- 2014
  - Enforcing SELinux on Android 4.4
References

- “SELinux Targeted vs Strict policy History and Strategy”
- SELinux/Tutorials/How is the policy provided and loaded
  http://wiki.gentoo.org/wiki/SELinux/Tutorials/How_is_the_policy_provided_and_loaded
- NB PolicyType
  http://selinuxproject.org/page/NB_PolicyType#PolicyVersionsMonolithic
- Getting Started with Reference Policy
  http://oss.tresys.com/projects/refpolicy/wiki/GettingStarted
- Using SELinux on RHEL 6
- Introducing the SELinux Sandbox
  http://danwalsh.livejournal.com/28545.html
- Fedora19 Security Guide - Fedora Documentation
- MCS (Multi Category Security), New feature of Fedora Core 5
Contact Information

E-Mail : research—feedback@ffri.jp
Twitter: @FFRI_Research