Demystifying SELinux:
What is it trying to tell me?

David Quigley
dpquigl@davequigley.com

What is Access Control?

A system for restricting who or what is allowed to access specific resources and how

Discretionary vs Mandatory Access Control

- Traditional form of access control in operating systems.
- Decisions based on user identity/ownership.
- Users and their programs are free to change access rules (e.g. file modes, ACLs).
- No protection against malicious and flawed software.
- Coarse-grained privilege, prone to escalation.

- Historically limited to separate “trusted” operating systems.
- Decisions based on security labels.
- Access rules defined by admin/organization.
- Control over all processes and objects.
- Can confine malicious and flawed software.
- Can enforce system-wide security requirements.

What is SELinux?

- SELinux is a security labeling system
- Every process has a label, every object on the system has a label
  - Files, Directories, network ports …
- The SELinux policy controls how process labels interact with other labels on the system
- The kernel enforces the policy rules

What is a Label?

- All information needed for SELinux to make an access control decision
  - User, Role, Type, MLS

-rw-r--r-- root root system_u:object_r:**etc_t**:s0 /etc/passwd

DAC Components Security Label

system_u:object_r:**etc_t**:s0

User Role **Type** MLS

How do I see Labels?

- Files
  - ls -Z
- Processes
  - ps -Z, pstree -Z
- Ports
  - netstat -Z, semanage ports -l

How to tell if something is wrong?

- Logged to /var/log/messages if no auditd or during early boot before auditd.
  - grep avc /var/log/messages
  - grep compute_sid /var/log/messages
- Logged to /var/log/audit/audit.log if running auditd.
  - /sbin/ausearch -mAVC,SELINUX_ERR -i
- Notification via setroubleshoot if running.
  - /var/log/messages, desktop pop-up

Example: AVC Denial

- type=AVC msg=audit(09/07/2010 14:06:38.240:54981) : avc: denied { read } for pid=4866 comm=bash name=.bash_history dev=dm-0 ino=138 scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:admin_home_t:s0 tclass=file
- An attempt by a bash process to read a .bash_history file was denied, where the bash process was running in the httpd_t domain and the .bash_history file was labeled with admin_home_t (i.e. under /root).

Silent Denials

- Permission denials may be silenced by dontaudit rules in the policy.
- Used to avoid filling audit logs with noise from harmless application probing.
- May hide the cause of a denial when developing policy.
- Use semodule -DB to strip dontaudit rules.
- Use semodule -B to restore them.

4 Common SELinux Reasons of Errors

- Labeling Problems
- A confined process is configured in a way different then the default SELinux expected
- Bug in Policy or an Application
- Your machine has been compromised

Labeling Problems

- Every process and object on the system is labeled
- If labels are not correct access may be denied
- Causes
  - Alternative paths (semanage fcontext)
  - Files created in wrong context (restorecon)
  - Processes started in wrong context

LAB: Fix improper label

- Check security context of /var/www/index.html
  - What is it?
- Create ~/test.txt & move to /var/www
- Try accessing http://localhost/test.txt
- Either restore just that one file or the entire public _html directory.
  - chcon -t httpd_sys_content_t /var/www/test.txt
  - restorecon -vvr /var/www

Non-Default Configuration

- SELinux needs to know how a confined daemon is configured
- Booleans
  - Allow option functionality to be enabled
- Non-default directories
  - Need to ensure files are labeled properly
- Non-default ports
  - Need to ensure ports labeled properly

LAB: Non-Default Locations

- Edit /etc/httpd/conf/httpd.conf
  - Change webroot to /opt/www
- Copy old webroot to new webroot
  - cp -R /var/www /opt/www
- Open http://localhost
- Why didn't it work?
  - semanage fcontext -a -e /var/www /opt/www

LAB: Booleans

- Create a file test.txt under ~/public_html
  - What is it's security context?
- Try to go to http://localhost/~sedemo/test.txt
  - Does it work?
- Why didn't it work?

Fixing Booleans

- List all policy booleans
  - getsebool -a
- Look for the right boolean
  - httpd + home directories?
- Set the boolean
  - setsebool <boolean> true
- Set the boolean permenantly
  - setsebool -P <boolean> true

Lab: Non-Default Ports

- Pick a tcp port to use
  - 8082 is free
- Edit /etc/httpd/conf/httpd.conf
  - Change listen to 8082
- Restart Apache
  - service httpd restart
- What happens?

Fixing: Non-Default Ports

- View Listing of all ports and find http port type
  - semanage port -l
- Add new port mapping
  - semanage port -a -t http_port_t -p tcp 8082
- Restart Apache
  - service httpd restart

Bugs in Policy/Apps

- SELinux policy bugs
  - Incomplete policy (unusual code path)
  - Unknown application configuration
- Application bugs
  - Leaked File Descriptors
  - Executable Memory (execmem)
  - Badly built libraries (execmem and others)

Bugs in Policy/Apps (2)

- Options
  - Report bugs in bugzilla (Best long term solution)
  - Create a policy module (Temporary fix)
- Labeling is correct? No appropriate booleans?
  - Use audit2allow to create a policy module
- Examing resulting policy
  - Make sure it's safe
  - Ask for help (#fedora-selinux and mailing lists)

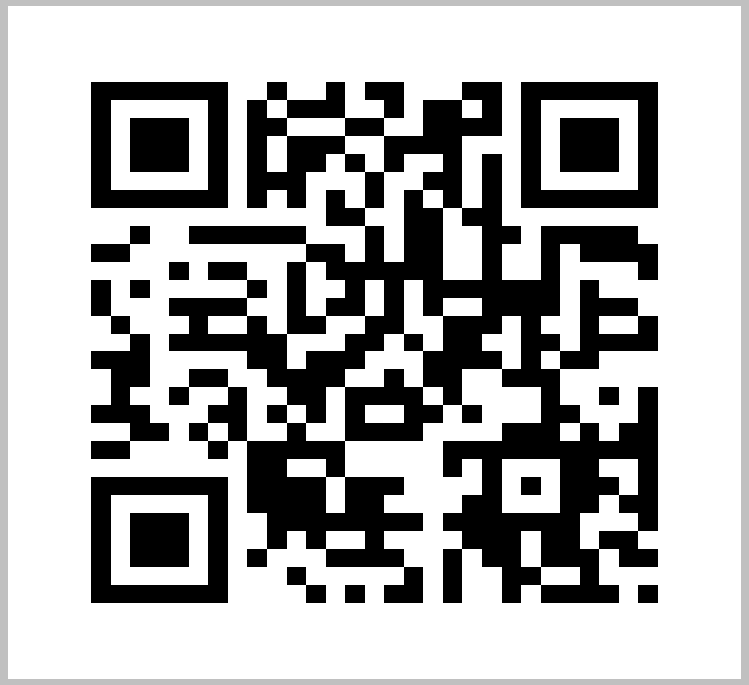
Your machine may have been compromised

- Current tools not good at differentiating
  - Warning signs: a confined domain tries to:
    - Load a kernel module
    - Turn off SELinux enforcing mode
    - Write to etc_t or shadow_t
    - Modify iptables rules
    - Sendmail
    - others
  - You might be compromised

Questions?

Survey

Thank you for listening to me talk. Please help improve the talk by filling out a quick survey at http://goo.gl/KJDfF



Tools

- Auditing
  - ausearch, aureport, auditctl, audit2why
- Policy Management tools
  - semodule, semanage, {get,set}sebool
- Policy Querying Tools
  - sesearch, sediff, apol
- Policy Generation Tools
  - audit2allow, sepolgen
- GUI Tools
  - setroubleshoot, system-config-selinux, apol, SLIDE

Audit2allow

- If the prior cases don't apply, you may need to create local policy to allow the access.
- audit2allow is a tool for generating policy from audit messages.
- Use with caution!

Audit2allow Examples

- Create and insert a local policy module that allows all logged denials since the last reload.
  - audit2allow -l -a -M mypolicy
  - semodule -i mypolicy.pp
- Create and insert a local policy module that allows all denials logged on the httpd program.
  - ausearch -m avc -c httpd | audit2allow -M myhttpd
  - semodule -i myhttpd

Audit2allow -R

- By default, audit2allow emits raw policy rules.
- Existing policy is written using macros (interfaces).
- Audit2allow -R will try to find the right interface and use it.
  - Audit2allow -l -a -R -M mypolicy
- Imperfect, but can be helpful.