Fun with SELinux

Writing SELinux Policy | Permissive Domains | Real bugs

Presented by
Miroslav Grepl
mgrepl@redhat.com

Today's Topics

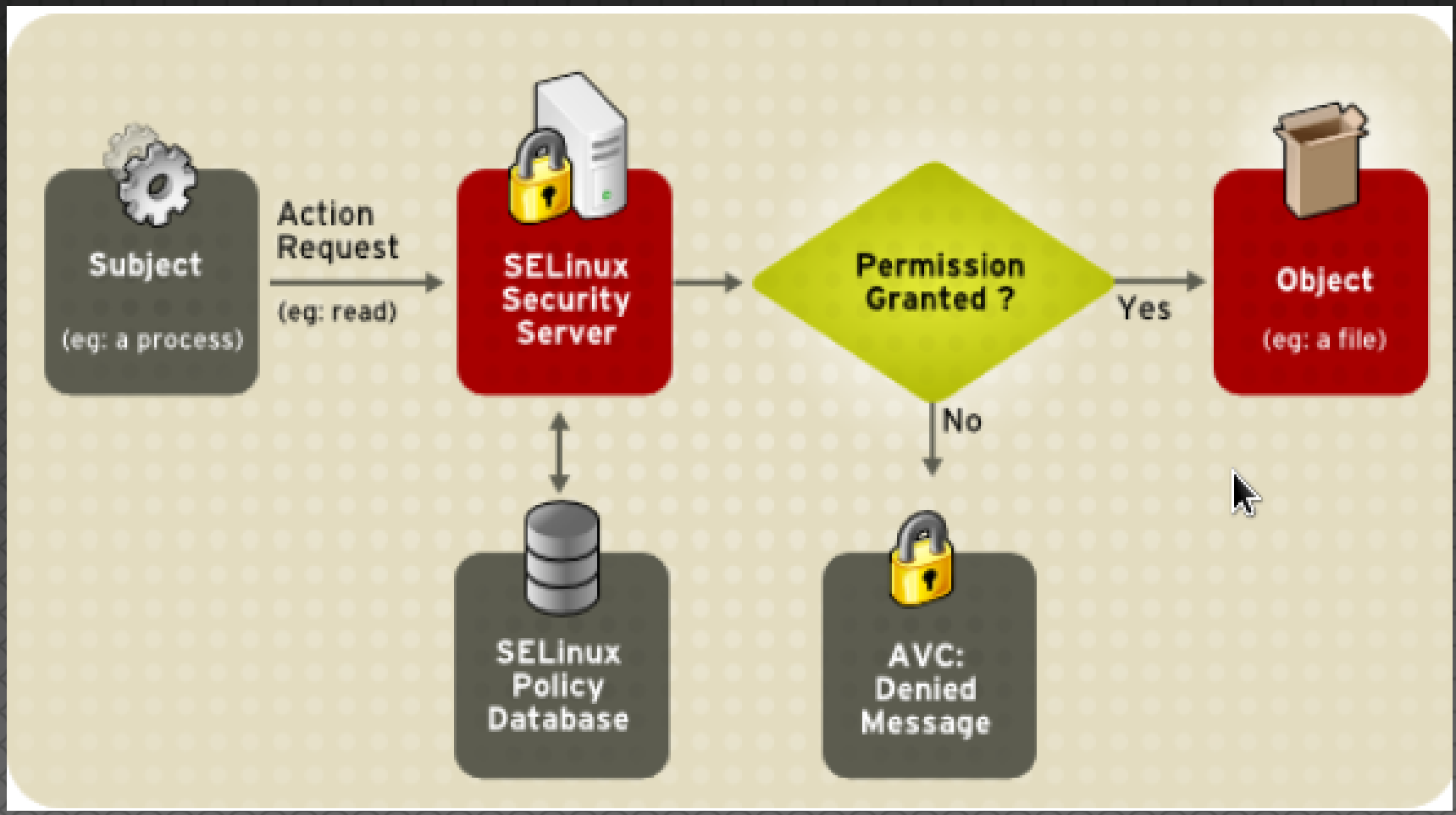
1. **Show process of writing a policy**
   - understanding basics of SELinux == **labels**
   - using SELinux tools
   - Permissive Domains
2. **Real examples**
   - creating & testing portreserve policy
   - how to solve real bug (Bip – IRC proxy)
   - creating a new policy???

Known basics?

- the most important part of SELinux
  - type enforcement language
  - everything on a SELinux system has a type - processes, files
- security context
  - system_u:object_r:**etc_t**:s0
- policy decision
  - security context labels are used to make access control decisions between processes and objects

Known basics?

- SELinux decision



Known basics?

- using security context

  # id -Z

  root:staff_r:staff_t

  # cat /etc/shadow

  cat: /etc/shadow: Permission denied

  # sesearch -A -s staff_t -t shadow_t -c file -p read

  ... what does it return??

- audit log records the following:

  *avc: denied { **read** } for pid=13653 exe=/bin/cat name=shadow dev=hda6 ino=1361441 scontext=root:staff_r:**staff_t** tcontext=system_u:object_r:**shadow_t** tclass=**file***

Policy rules

- type field
  - each subject (process), object (file) has a type
  - ***declaration***

    type portreserve_t; # Process Type (Domain)

    type portreserve_exec_t; # File Type

Policy rules

- policy rules statement

  COMMAND SOURCETYPE TARGETTYPE:CLASS PERMS;

  - **COMMAND**

    allow, dontaudit, audit2allow, neverallow

    allow staff_t etc_t:file { open read getattr ioctl lock};

    dontaudit staff_t shadow_t:file { open read getattr ioctl lock};

Policy rules

- policy rules statement

  COMMAND SOURCETYPE TARGETTYPE:CLASS PERMS;

  - **CLASS**

    file, dir, sock_file, tcp_socket, process

  - **PERMS**

    read, open, write

    - macros can be used

      define(`**r_file_perms**', `{ open read getattr lock ioctl }

/usr/share/selinux/devel/include/support/obj_perm_sets.spt

Policy rules

- attribute
  - group types

```
attribute file_type
type etc_t, file_type
typeattribute etc_t, file_type
allow rpm_t file_type:file manage_file_perms
```

Policy rules

- Attributes
  - decrease size of policy
  - on a Fedora 15

```
$ seinfo
Allow:        282 444
Dontaudit:    184 516
```

  - on Fedora 16

```
$ seinfo
Allow:        88 242
Dontaudit:    11 302
```

Policy module

- place where all policy statements are located
- allows users to easily customize policy
- allows third parties to ship policy with their rpms
- similar to kernel modules
  - recompile and reload

Policy module

- Three Components
  - **Type Enforcement** (TE) File
    - Contains all the rules used to confine your application
  - **File Context** (FC) File
    - Contains the regular expression mappings for on disk file contexts
  - **Interface** (IF) Files
    - Contains the interfaces defined for other confined applications, to interact with your confined application
- Policy Package (pp)
  - Compiler/packager roles generates policy package to be installed on systems.

LET'S START GENERATING POLICY

Setup environment

- Disable portreserve policy

  # semodule -d portreserve.pp

- Fix labels

  # for i in `rpm -ql portreserve`;do restorecon -R -v $i;done

  # systemctl restart portreserve.service

- Default initrc_t domain
  - unconfined domain
  - for process started by init system
  - process without policy

SELinux transition, labels

- Transitions
  - without transition using service script
    - initrc_t @bin_t -> initrc_t
  - with transition using service script
    - initrc_t @portreserve_exec_t -> portreserve_t
  - run directly
    - unconfined_t @portreserve_exec_t-> unconfined_t

=> **SELINUX IS ALL ABOUT LABELS**

Generating initial policy

- Using sepolgen or sepolgen-gui
  - give you policy files

 # sepolgen -n myportreserve -t 0 `which portreserve`

Created the following files in:

./

portreserve.te # Type Enforcement file

- Contains all the rules used to confine your application

portreserve.fc # Interface file

- Contains the regular expression mappings for on disk file contexts

portreserve.if # File Contexts file

- Contains the interfaces defined for other confined applications, to interact with your confined application

Generating initial policy

- Install policy
  - using setup script

    # sh myportreserve.sh
  - using Makefile

    # make -f /usr/share/selinux/deve/Makefile

    # semodule -i myportreserve.pp

    # for i in `rpm -ql portreserve`;do restorecon -R -v $i;done
- Do some checks

  # semodule -l | grep portreserve

  # ps -eZ | grep portre

  # ausearch -m avc -ts recent

Permissive Domains

- initial policies are running as permissive domains

  # permissive myportreserve_t

- checks are performed but not enforced
- users don't have to switch to permissive mode globally
- we can catch AVC messages

  # ausearch -m avc -ts recent | grep portreserve

- make domain permissive

  # semanage permissive -a httpd_t

Building policy

- loop until good policy
  - test application
  - generate avc messages
- audit2allow
  - examines /var/log/audit/audit.log and /var/log/messages for AVC messages
  - searches interface files for correct interface
  - if no interface found generates allow rules

Building policy

- audit2allow in practise

type=AVC msg=audit(04/22/2011 11:53:51.194:49) : avc: denied { **read** } for pid=7695 comm=dictd scontext=unconfined_u:system_r:**dictd_t**:s0 tcontext=system_u:object_r:**sysctl_kernel_t**:s0 tclass=**fil**

- audit2allow -R

require {

type dictd_t;

}

#============= dictd_t ==============

kernel_read_kernel_sysctls(dictd_t)

Complete our policy

- ausearch, audit2allow tools
  - # ausearch -m avc -ts today | grep portreserve | audit2allow -R
- compile and load rules
  - # ausearch -m avc -ts today | grep portreserve | audit2allow -R >> myportreserve.te
  - # make -f /usr/share/selinux/devel/Makefile
  - # semodule -i myportreserve.pp
- test it without permissive domain
  - sed -i s/^permissive/#permissive/ portreserve.te

Complete our policy

MOST IMPORTANT THING TO LEARN TODAY

audit2allow – Just MAKE IT WORK?????

Real bug – bip issue

- new policies for new unconfined services/apps?
  - are not always necessary
    - spamc_t domain type treat a lot of spam apps
    - does not make sense creating new policy for each spam apps?
  - policy has many types to use
    - for example bip IRC proxy
    - there was the following bug

Real bug – bip issue

https://bugzilla.redhat.com/show_bug.cgi?id=783693

avc: denied { **name_bind** } for pid=2897 comm="bip" src=6667 scontext=system_u:system_r:**initrc_t**:s0 tcontext=system_u:object_r:**ircd_port_t**:s0 tclass=**tcp_socket**

- runnig as initrc_t -> causes issues
  - add a custom module using audit2allow
  - create a new policy
  - use a current policy
    => which one ???

Real bug – bip issue

- use a current policy
  - which one?

    => we know bitlbee is similar

    => does bitlbee policy exist?

    ```
    # seinfo  -t |grep bitlbee
    ```

  - which type will we use for bip binary?

    ```
    # chcon -t ???_t `which bip`
    # service bip restart
    ```

Real bug – unconfined services

- There are services without SELinux confinement

  => running as initrc_t

  - bcfg2-server, glusterd
  - rpc.rstatd, rpc.rusersd
  - pacemaker, pkcsslotd, fence_virtd, **openhpid**, isnsd, sfcbd, svnserve, stap-serverd

Backup your environment

- load the default policy using semodule

  # semodule -r myportreserve -e portreserve

- fix labels using restorecon

  # for i in ..

  # systemctl restart portreserve.service

- remove permissive domain using semanage
  - # semanage permissive -d httpd

Links

- http://danwalsh.livejournal.com/
- http://dwalsh.fedorapeople.org/
- http://mgrepl.wordpress.com/
- http://mgrepl.fedorapeople.org/

Questions?

Contact:
mgrepl@redhat.com