# Fun with SELinux

## Writing SELinux Policy

Presented by
Miroslav Grepl
mgrepl@redhat.com

cc BY NC SA

# Today's Topics

## 1. Show process of writing a policy

- understanding basics of SELinux == **labels**

  => SELinux is not difficult and is your friend

- using SELinux tools (audit2allow, ausearch, sepolicy)

## 2. Real examples

- **re-creating & testing hddtemp policy**
- how to solve real bug (Bip – IRC proxy)
- **creating a new policy for pesignd service**

# Today's Topics

**Before we start, please prepare your system.**

1. Download scripts from

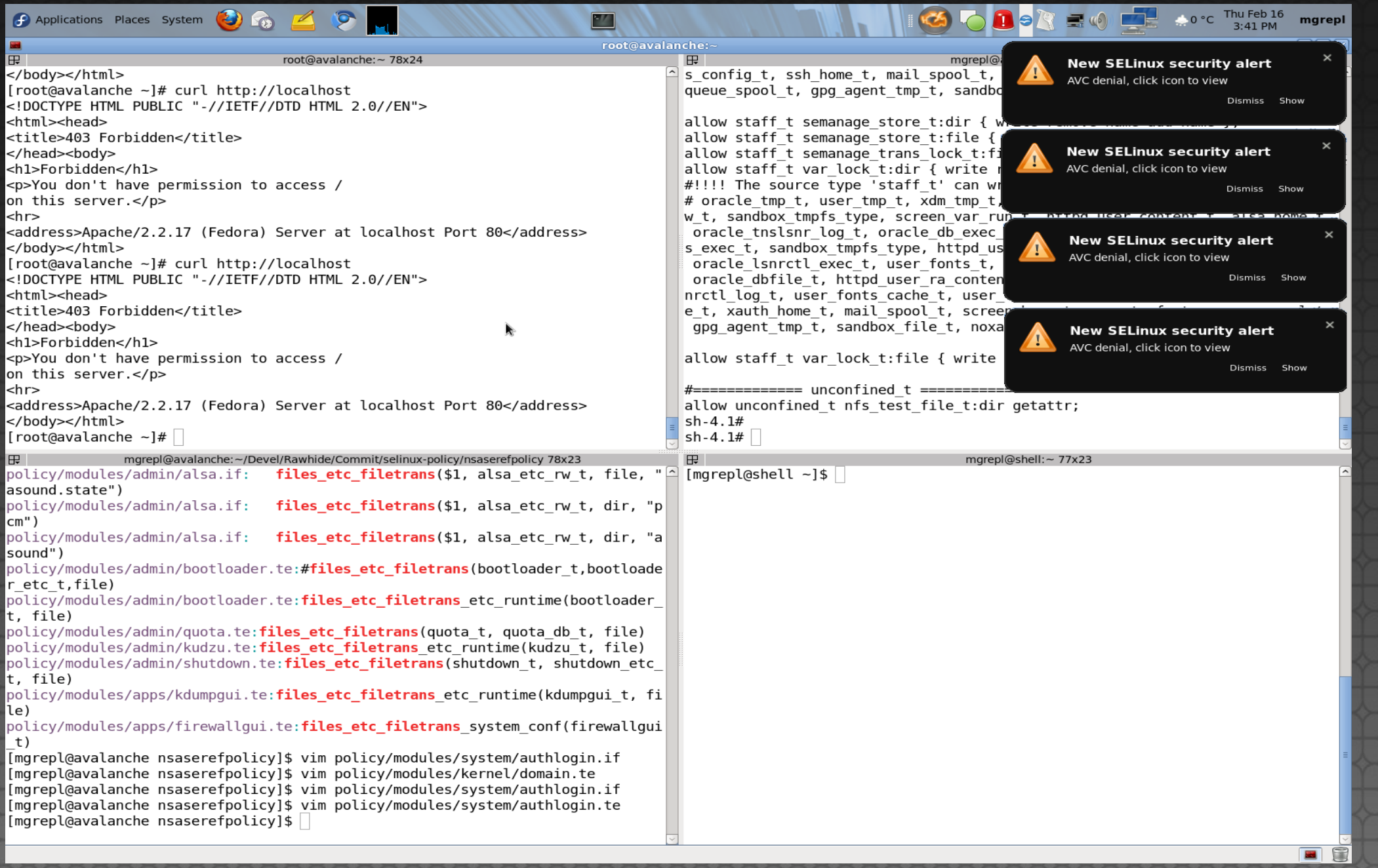
   **http://mgrepl.fedorapeople.org/PolicyCourse/**

   to /root directory.

2. Execute

   *./setup.sh*

# WHAT IS SELINUX?


Applications Places System
0 °C Thu Feb 16 3:41 PM mgrepl
root@avalanche:~
root@avalanche:~ 78x24
</body></html>
[root@avalanche ~]# curl http://localhost
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
You don't have permission to access /
on this server.
<hr>
<address>Apache/2.2.17 (Fedora) Server at localhost Port 80</address>
</body></html>
[root@avalanche ~]# curl http://localhost
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
You don't have permission to access /
on this server.
<hr>
<address>Apache/2.2.17 (Fedora) Server at localhost Port 80</address>
</body></html>
[root@avalanche ~]#
s_config_t, ssh_home_t, mail_spool_t,
queue_spool_t, gpg_agent_tmp_t, sandbo
allow staff_t semanage_store_t:dir { w
allow staff_t semanage_store_t:file {
allow staff_t semanage_trans_lock_t:fi
allow staff_t var_lock_t:dir { write
#!!!! The source type 'staff_t' can wr
# oracle_tmp_t, user_tmp_t, xdm_tmp_t,
w_t, sandbox_tmpfs_type, screen_var_ru
oracle_tnslsnr_log_t, oracle_db_exec
s_exec_t, sandbox_tmpfs_type, httpd_us
oracle_lsnrctl_exec_t, user_fonts_t,
oracle_dbfile_t, httpd_user_ra_conten
nrctl_log_t, user_fonts_cache_t, user_
e_t, xauth_home_t, mail_spool_t, screen
gpg_agent_tmp_t, sandbox_file_t, noxa
allow staff_t var_lock_t:file { write
#============= unconfined_t ==========
allow unconfined_t nfs_test_file_t:dir getattr;
sh-4.1#
sh-4.1#
New SELinux security alert
AVC denial, click icon to view
Dismiss Show
New SELinux security alert
AVC denial, click icon to view
Dismiss Show
New SELinux security alert
AVC denial, click icon to view
Dismiss Show
New SELinux security alert
AVC denial, click icon to view
Dismiss Show
mgrepl@avalanche:~/Devel/Rawhide/Commit/selinux-policy/nsaserefpolicy 78x23
policy/modules/admin/alsa.if: files_etc_filetrans($1, alsa_etc_rw_t, file, "asound.state")
policy/modules/admin/alsa.if: files_etc_filetrans($1, alsa_etc_rw_t, dir, "pcm")
policy/modules/admin/alsa.if: files_etc_filetrans($1, alsa_etc_rw_t, dir, "asound")
policy/modules/admin/bootloader.te:#files_etc_filetrans(bootloader_t,bootloader_etc_t,file)
policy/modules/admin/bootloader.te:files_etc_filetrans_etc_runtime(bootloader_t, file)
policy/modules/admin/quota.te:files_etc_filetrans(quota_t, quota_db_t, file)
policy/modules/admin/kudzu.te:files_etc_filetrans_etc_runtime(kudzu_t, file)
policy/modules/admin/shutdown.te:files_etc_filetrans(shutdown_t, shutdown_etc_t, file)
policy/modules/apps/kdumpgui.te:files_etc_filetrans_etc_runtime(kdumpgui_t, file)
policy/modules/apps/firewallgui.te:files_etc_filetrans_system_conf(firewallgui_t)
[mgrepl@avalanche nsaserefpolicy]$ vim policy/modules/system/authlogin.if
[mgrepl@avalanche nsaserefpolicy]$ vim policy/modules/kernel/domain.te
[mgrepl@avalanche nsaserefpolicy]$ vim policy/modules/system/authlogin.if
[mgrepl@avalanche nsaserefpolicy]$ vim policy/modules/system/authlogin.te
[mgrepl@avalanche nsaserefpolicy]$
mgrepl@shell:~ 77x23
[mgrepl@shell ~]$

:-)

.. is a history

.. was on my F16 laptop

# Now seriously ..

# WHAT IS SELINUX?

# WHAT IS SELINUX?

**SELINUX IS A LABELING SYSTEM.**

# WHAT IS SELINUX?

Every **subject (process)** has a **label**.

# WHAT IS SELINUX?

Every **object** on the system has a **label**.

.. files, directories, network ports .

# WHAT IS SELINUX?

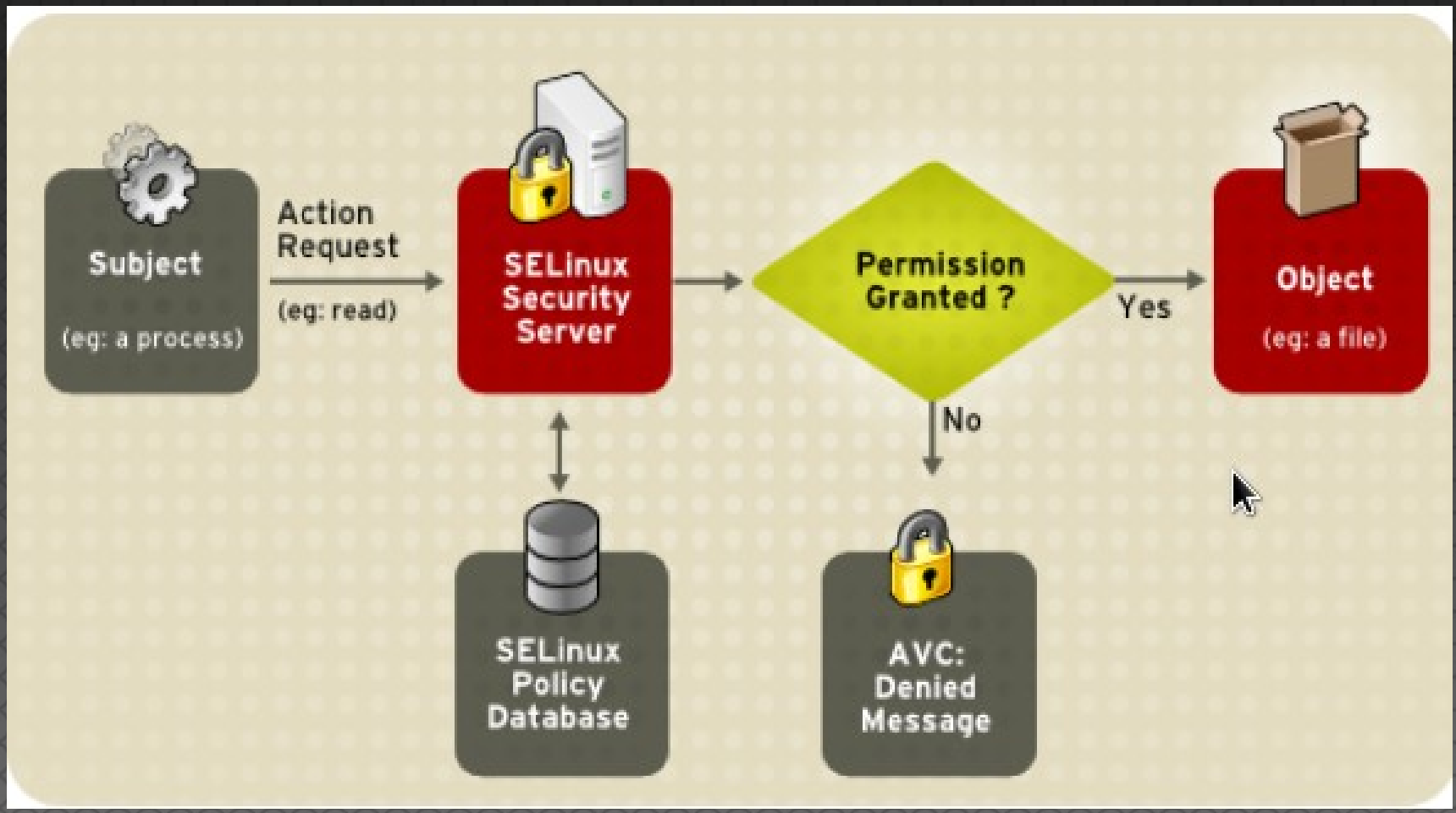
The **SELinux policy** controls how **process labels** *interact* with **other labels** on the system.

# WHAT IS SELINUX?

The kernel **enforces** the **policy rules**.

# WHAT IS SELINUX?

- SELinux decision



# SELINUX BASICS

- how do we call labels? - **security context**
  - examples

    system_u:object_r:**etc_t**:s0

    unconfined_u:unconfined_r:**unconfined_t**:s0

- the most important part of labels = **type field**
  - **all subjects and objects have a label => have a type**
  - decisions are made according these **types**

    => we talk about **TYPE ENFORCEMENT (TE)**

    => is a way how SELinux enforce MAC

# SELINUX BASICS

- security context (labels) in the game

*# ps -eZ |grep sshd*

*system_u:system_r:**sshd_t**:s0-s0:c0.c1023 **...** process label*

*# ls -Z /etc/shadow*

*system_u:object_r:**shadow_t:**s0 ... file label*

*# id -Z*

*staff_u:staff_r:**staff_t**:s0-s0:c0.c1023*

# SELINUX BASICS

- security context (labels) in the game

*# ls -Z  /root/my_secrets*

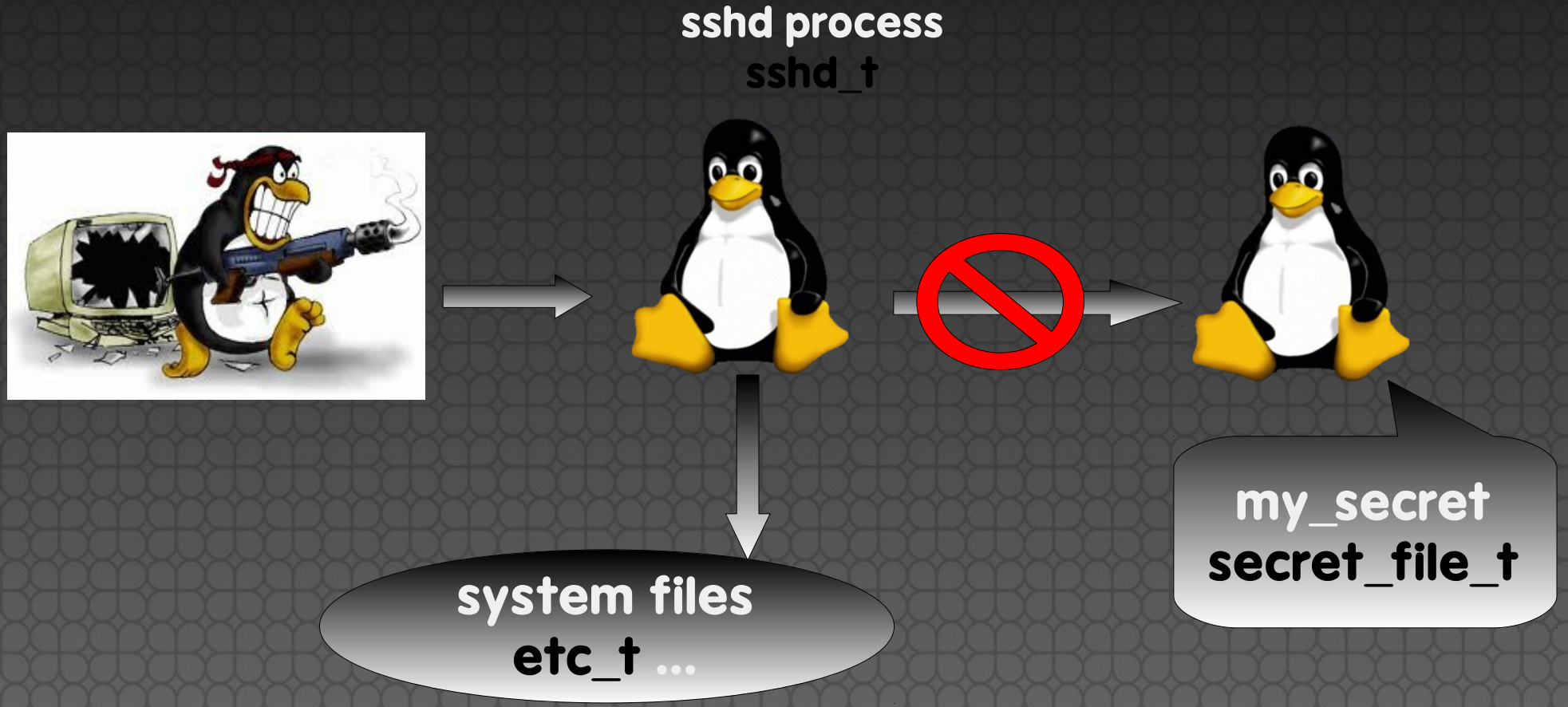
*#  selinuxrun sshd /etc/hostname*

*#  selinuxrun sshd ls /root/my_secrets*

*# **ls: cannot access /root/my_secret: Permission denied***

**WHAT HAPPENED ???**

# SELinux in the game



# SELINUX BASICS

- Where could I find more SELinux info about the operation?
  - /var/log/audit/audit.log file

type=AVC msg=audit(1366360758.832:776): avc:  denied { **read** } for  pid=6604 comm="cat" name="my_secret" dev="dm-1" ino=266659 **scontext**=system_u:system_r:**sshd_t**:s0-s0:c0.c1023 **tcontext**=staff_u:object_r:**secret_file_t**:s0 tclass=**file**

# POLICY RULES

LET'S START WITH POLICY RULES

# Policy rules

- type field
  - each subject (process), object (file) has a type
  - type is a part of security context ... as you know

  - ***declaration***

    *type hddtemp_t;  # Process Type (Domain)*

    *type hddtemp_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

# Policy rules

- **m4 macro language**
  - policy macros are 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 19

    $ seinfo

    Allow: **89771**

    Dontaudit: 7264

# 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 hddtemp*

- Fix labels

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

  *# systemctl restart hddtemp*

  *# ps -eZ |grep hddtemp*

- What are you getting?

# Setup environment

- What are you getting?
  - => you should see **initrc_t** type
  - **default type for process without SELinux policy started by init system**
  - unconfined domain
  - we don't want to have initrc_t on a system => we need to create a policy

# Generating initial policy

- Using new **sepolicy** tool
  - gives you policy files + other files

    *# sepolicy generate --help*

    *# sepolicy generate -n myhddtemp- -init `which hddtemp `*

    Created the following files in:

    *./*

    ***myhddtemp.te*** # Type Enforcement file
    - Contains all the rules used to confine your application

    ***myhddtemp.fc*** # Interface file
    - Contains the regular expression mappings for on disk file contexts

    ***myhddtemp.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 myhddtemp.sh*

  - using Makefile

    *# make -f /usr/share/selinux/deve/Makefile myhddtemp.pp*

    *# systemctl hddtemp stop*

    *# semodule -i myhddtemp.pp*

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

# Generating initial policy

- Do some checks

  *# semodule -l | grep hddtemp*

  *# ls -Z `which hddtemp`*

  *# systemctl start hddtemp*

  *# ps -eZ | grep hddtemp*

  *# ausearch -m avc -ts recent*

  => probably you see AVC msgs

- Does the service work correctly?
- Does it work without permissive statement?

# Permissive Domains

- initial policies are running as permissive domains

  *# permissive myhddtemp_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 hddtemp*
- make domain permissive

  *# semanage permissive -a hddtemp_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=**file**

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

# Complete our policy

MOST IMPORTANT THING TO LEARN TODAY

audit2allow – Just MAKE IT WORK?????

# SELinux is all about labels!!!

- Confined vs unconfined daemon
  - without myhddtemp policy
    - *ls -Z /sbin/hddtemp* -> **bin_t** type
    - init_t @bin_t -> initrc_t
  - with the myhddtemp policy
    - *ls -Z /sbin/hddtemp* -> **myhddtemp_exec_t** type
    - init_t @hddtemp_exec_t -> hddtemp_t
  - run directly
    - unconfined_t @hddtemp_exec_t-> hddtemp_t

# 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 bugs – unconfined services

- There are services without SELinux confinement

  => running as initrc_t

  - openhpid, **pesignd**, ldirectord, rtas_errd

# Backup your environment

- load the default policy using semodule

  # semodule -r myhddtemp -e hddtemp

  # cd /root

  # make -f /usr/share/selinux/devel/Makefile clean

  # rm -rf myhddtemp*

- fix labels using restorecon

  # for i in ..

  # systemctl restart hddtemp

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

# Links

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

# Questions?

Contact:
mgrepl@redhat.com

CC BY NC SA