

# Red Hat Deep Dive Sessions

## SELinux

**Shawn D. Wells, RHCE**

**Email: [swells@redhat.com](mailto:swells@redhat.com)**

**Solutions Architect @ Red Hat**

# Agenda

1) **Why do we need SELinux? What are the principal concepts?**

2) **SELinux Details**

- **Type Enforcement**
- **What are the available policies?**
- **What's a policy actually made of?**
- **How do I {add, change} a policy?**
- **What's the associated overhead?**

3) **Usage**

- **User Perspective**
- **Admin Perspective**

4) **Scenarios**

- **Fixing the RHT Corporate VPN “update”**

# Why do we need SELinux?

# Linux Access Control Problems

## 1) Access is based off users' access

**Example:** Firefox can read SSH keys

```
# ps -x | grep firefox  
shawn 21375 1 35 11:38 ? 00:00:01 firefox-bin
```

```
# ls -l id_rsa  
-rw----- 1 shawn shawn 1743 2008-08-10 id_rsa
```

**Fundamental Problem:** Security properties not specific enough. Kernel can't distinguish applications from users.

# Linux Access Control Problems

## 2) Processes can change security properties

**Example:** Mail files are readable only by me..... but  
Thunderbird could make them world readable

### Fundamental Problems:

- Standard access control is discretionary
- Includes concept of “resource ownership”
- Processes can escape security policy

# Linux Access Control Problems

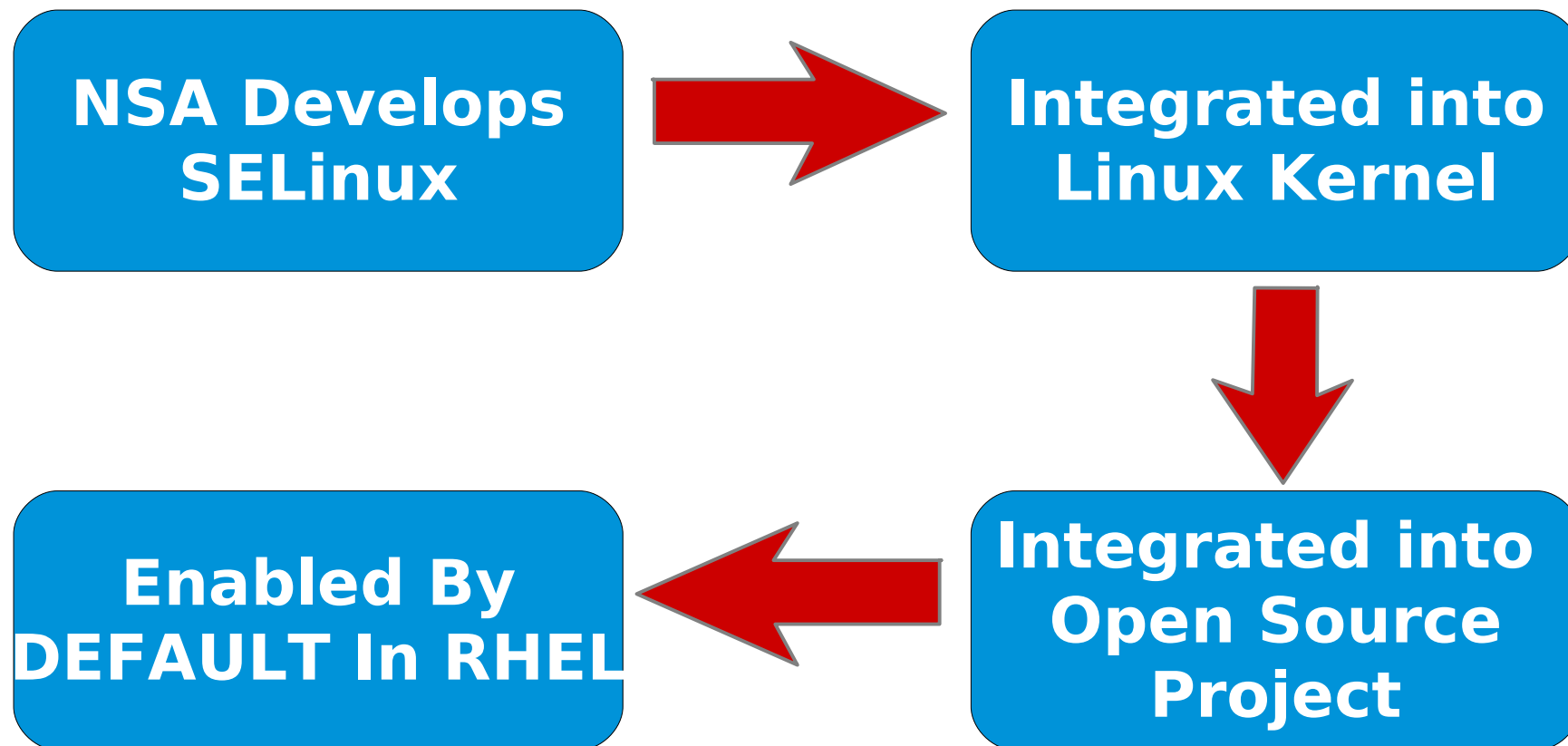
## 3) Only two privilege levels: User & root

**Example:** Apache gets hacked, allowing remote access to root. Entire system is compromised.

### Fundamental Problems:

- Simplistic security policy
- No way to enforce least-privilege

# SELinux: Building Security Openly



**Customers, NSA, Community, and Red Hat continue evolution**



# Red Hat Security Certifications

## **NIAP/Common Criteria: The most evaluated operating system platform**

- Red Hat Enterprise Linux 2.1 – EAL 2 (Completed: February 2004)
- Red Hat Enterprise Linux 3 EAL 3+/CAPP (Completed: August 2004)
- Red Hat Enterprise Linux 4 EAL 4+/CAPP (Completed: February 2006)
- Red Hat Enterprise Linux 5 EAL4+/CAPP/LSPP/RBAC (Completed: June 2007)

## **DII-COE**

- Red Hat Enterprise Linux 3 (Self-Certification Completed: October 2004)
- Red Hat Enterprise Linux: First Linux platform certified by DISA

## **DCID 6/3**

- Currently PL3/PL4: ask about kickstarts.
- Often a component in PL5 systems

## **DISA SRRs / STIGs**

- Ask about kickstarts.

## **FIPS 140-2**

- Red Hat / NSS Cryptography Libraries certified Level 2



# Security Standards Work

## **Extensible Configuration Checklist Description Format (XCCDF)**

- Enumeration for configuration requirements
- DISA FSO committed to deploying STIG as XCCDF
- Others working with NIST
- Security policy becomes one file

## **Open Vulnerability & Assessment Language (OVAL)**

- Machine-readable versions of security advisories

## **Common Vulnerability and Exposures (CVE) Compatibility**

- Trace a vulnerability through multiple vendors



# How's it work?

# Linux Access Control Introduction

Linux access control involves the kernel controlling

- **Processes** (running programs), which try to access...
- **Resources** (files, directories, sockets, etc)

For example:

- Apache (process) can read web files
- But **not** the /etc/shadow file (resource)

Traditional methods do not clearly separate the privileges of users and applications acting on the users behalf, increasing the damage that can be caused by application exploits.

**So, how should these decisions be made?**

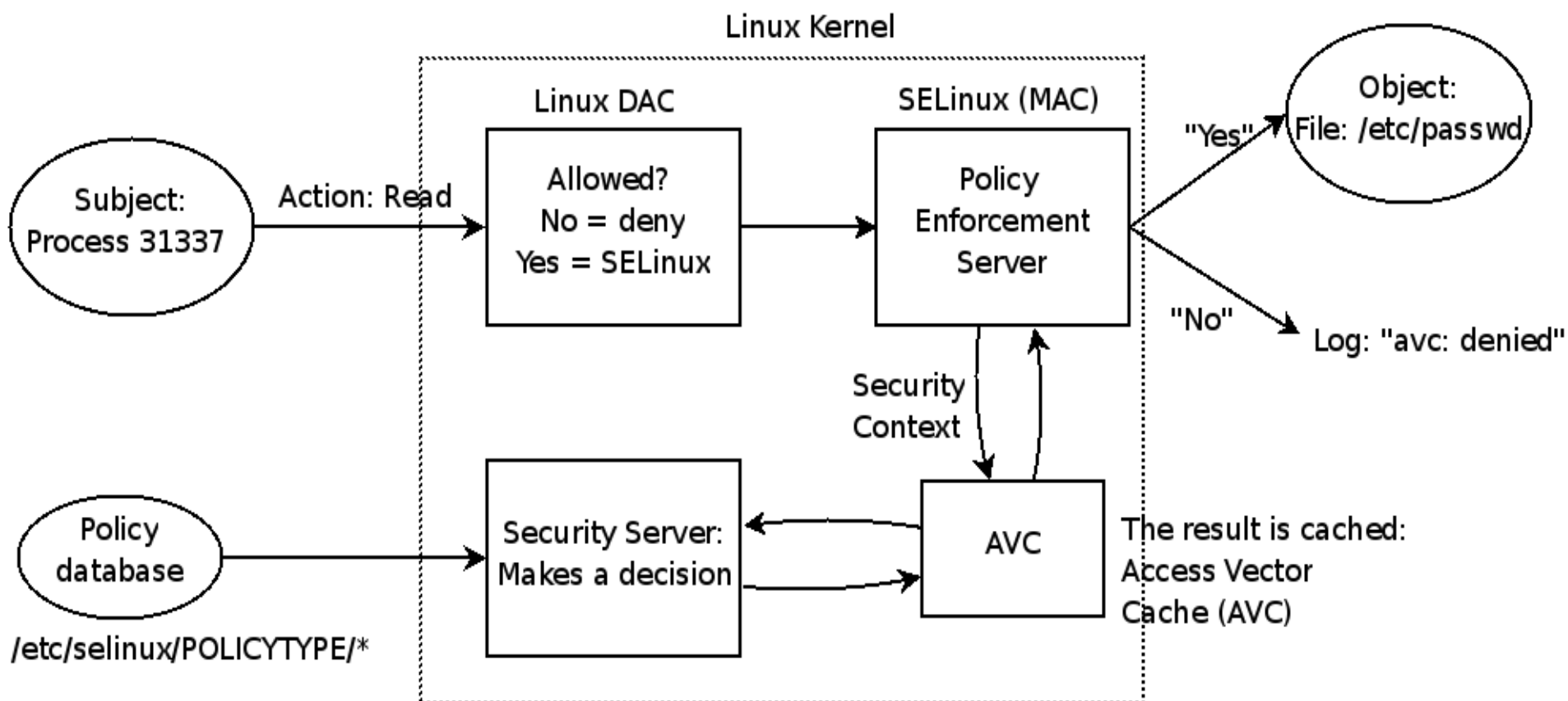
# Security Architecture

Every subject (i.e process) and object (i.e. data files) are assigned collections of security attributes, called a **security context**

- 1) Security context of subject & object passed to SELinux
- 2) Kernel/SELinux check, verify access
  - 2a) Grant access. Record allowance in AVC (Access Vector Cache)
  - 2b) Deny access, log error

# Security Architecture

Or in picture view...



# Role Based Access Control (RBAC)

“root” really isn't “root”

i.e:

root\_u:**WebServerAdmin\_r**:SysAdmin\_t

root\_u:**OracleDBAdmin\_r**:SysAdmin\_t



# SELinux Details



# Type Enforcement

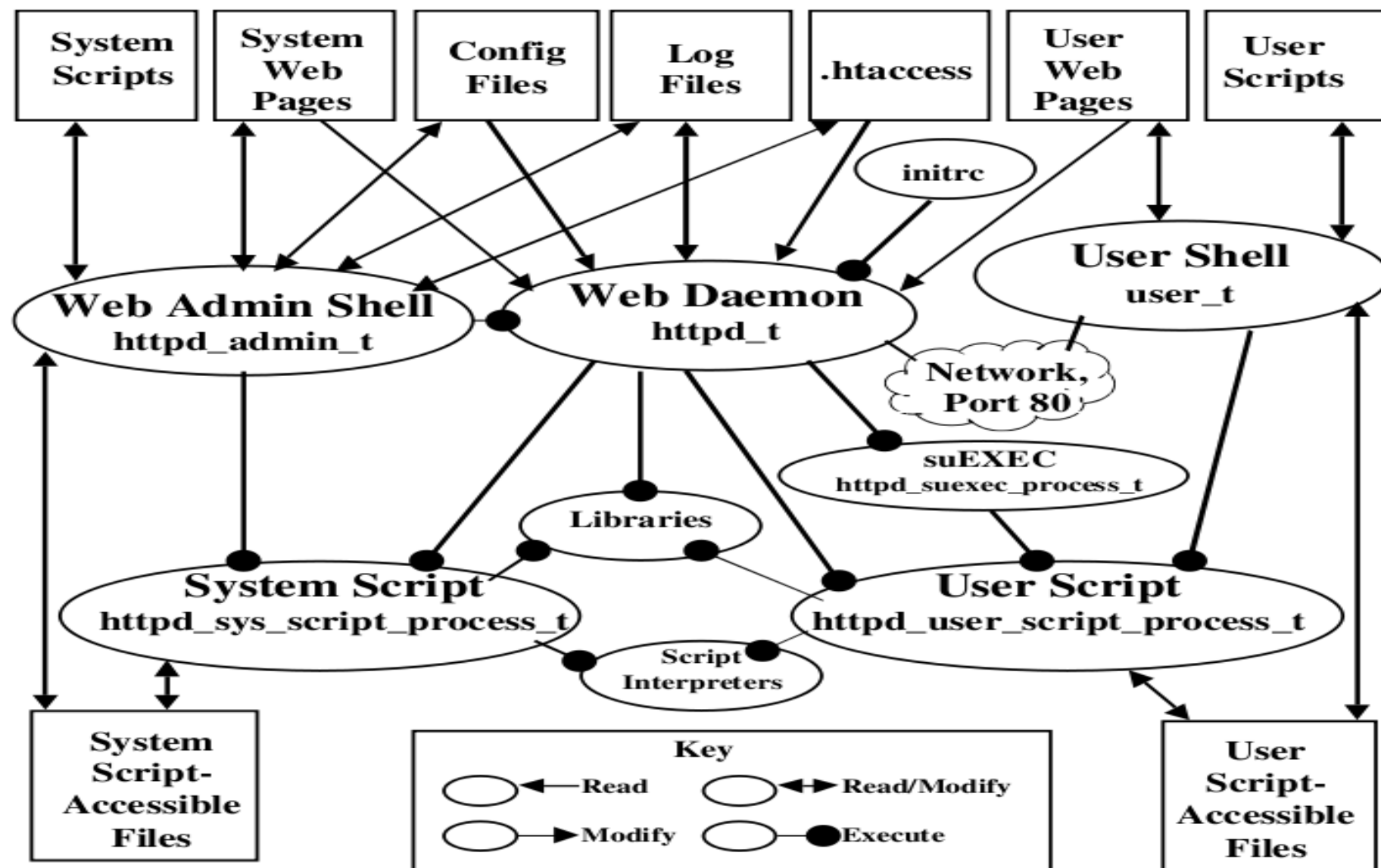
- SELinux implements the MAC model through type enforcement.
- In RHEL5, SELinux also provides RBAC and Bell-LaPadula (MLS), but it uses type enforcement to implement them.
- Type Enforcement involves defining a type for every subject, that is, process, and object on the system.
- Permissions are checked between the source type and the target type for each access.
- Objects include (but are not limited to):
  - Network Sockets
  - Shared Memory Segments
  - Files
  - Processes
  - etc.

# SELinux Contexts

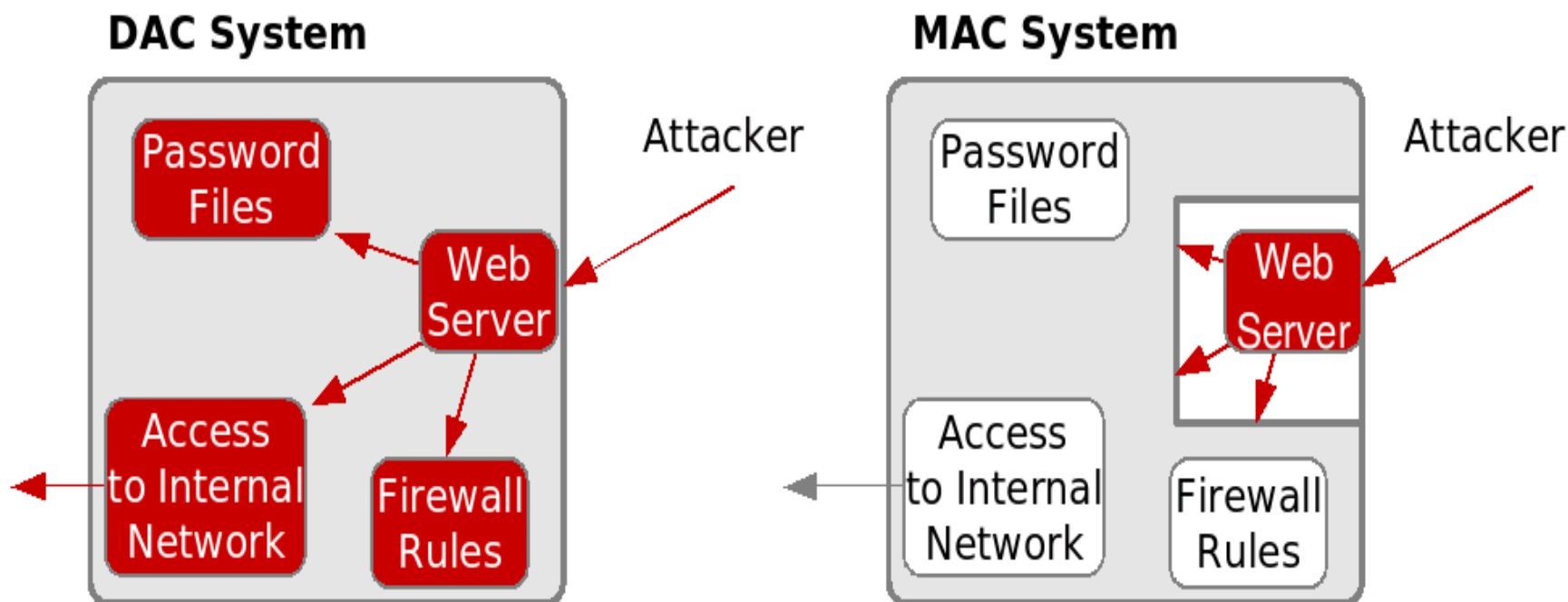
**root:object r:sysadm home t:s0:c0**

- The above is an SELinux context
- user\_t
- role\_t
- file\_t
- Sensitivity
- category

# SELinux Contexts



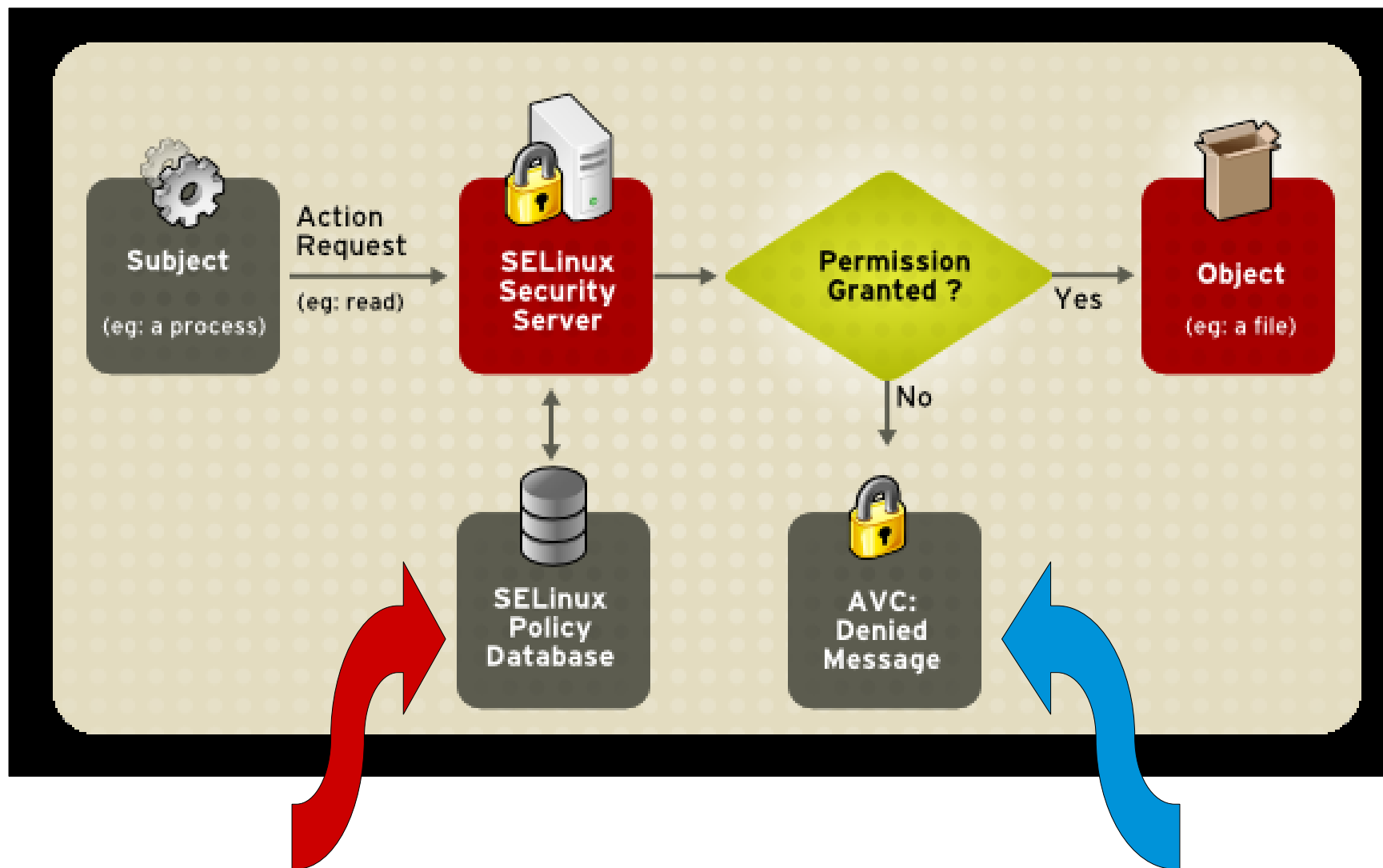
# DAC vs MAC



- Application can change attributes
  - User privileges
- =
- Process privileges

- Orthogonal to DAC
- Roles, Contexts, Types

# How does SELinux Work?



`/etc/selinux/{targeted,strict}/policy`

`/var/log/messages` 20

# SELinux Policy

- Policies are matrices of statements which tell SELinux if certain actions are allowed based on the context of the objects attempting those actions.
- There are three SELinux Policy Types

# The Three SELinux Policy Types

## 1) Targeted Policy

- *Default policy in RHEL5. Supported by HelpDesk.*
- Targets specific applications to lock down.
- Allows all other applications to run in the unconfined domain (`unconfined_t`)
- Applications running in the unconfined domain run as if SELinux were disabled



# The Three SELinux Policy Types

## 2) Strict Policy

- Denies access to everything by default
- Complete protection for all processes on the system
- Requires that policies be written for **all** applications, often requires customization
- Strict is type enforcement with added types for users (e.g. `user_t` and `user_firefox_t`).
- Not enabled by Red Hat as default

# The Three SELinux Policy Types

## 3) Multi-Level Security (MLS)

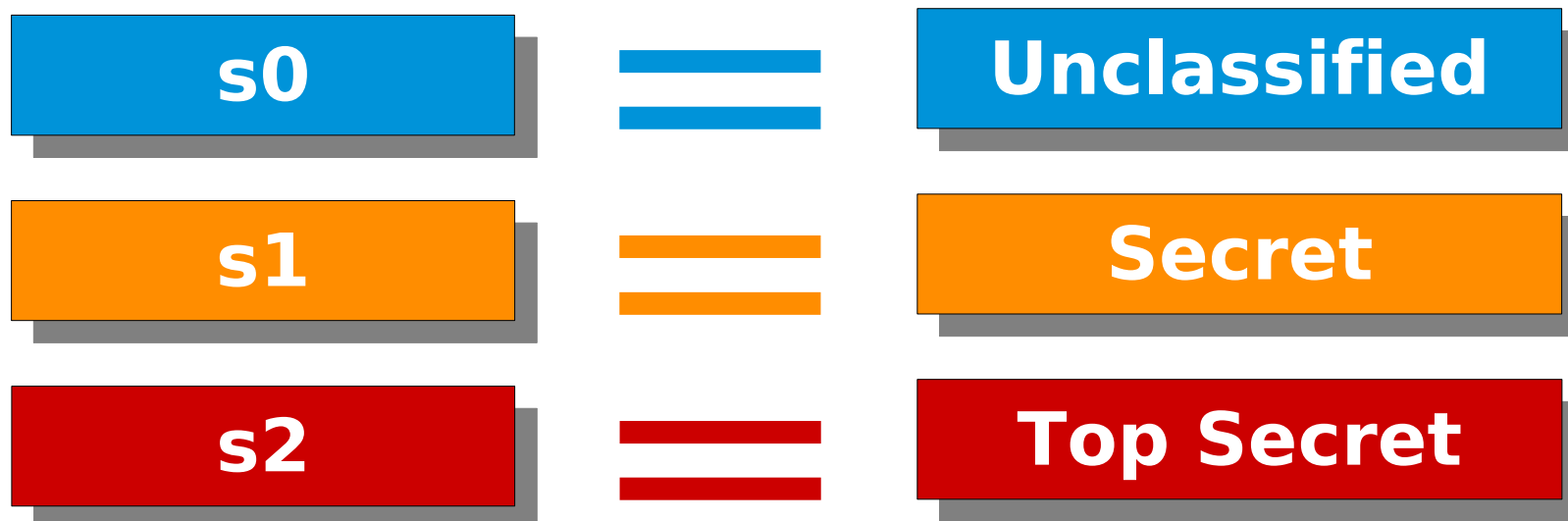
- Focuses on confidentiality (i.e. separation of multiple classifications of data)
- Ability to manage {processes, users} with varying levels of access. (i.e. “*the need to know*”)
- Uses category & sensitivity levels

# The Three SELinux Policy Types

## 3) Multi-Level Security (MLS)

### (a) Sensitivity Labels

- Mostly used by the government – Top Secret, Secret, Unclassified, etc



# The Three SELinux Policy Types

## 3) Multi-Level Security (MLS)

### (b) Category Labels

- Separation of data types, compartments, projects, etc

s0

**Unclassified**

s1

**Secret**

c0

**Project A**

c1

**Project B**

s1

**Top Secret**

c0

**Alpha**

c1

**Bravo**

c2

**Charlie**

c3

**Delta**

# The Three SELinux Policy Types

## 3) Multi-Level Security (MLS)

### (b) Polyinstantiation & pam\_namespace

- The pam\_namespace PAM module sets up a private namespace for a session with polyinstantiated directories
- A polyinstantiated directory provides a different instance of itself based on user name, or when using SELinux, user name, security context or both

# The Three SELinux Policy Types

## 3) Multi-Level Security (MLS)

### (b) Polyinstantiation & pam\_namespace

```
# id -Z
```

```
staff_u:WebServer_Admin_r:WebServer_Admin_t:s0:c0
```

```
# ls -l /data
```

```
secret-file-1
```

```
secret-file 2
```

```
# id -Z
```

```
staff_u:WebServer_Admin_r:WebServer_Admin_t:s1:c0
```

```
# ls -l /data
```

```
secret-file-1
```

```
secret-file 2
```

```
top-secret-file-1
```

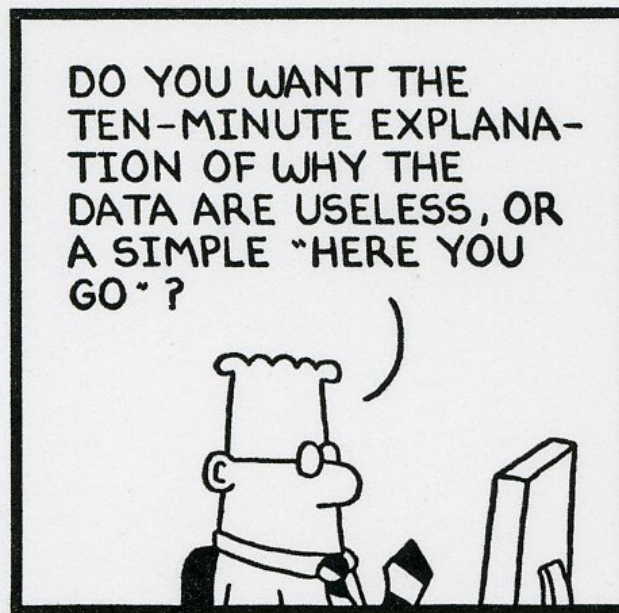
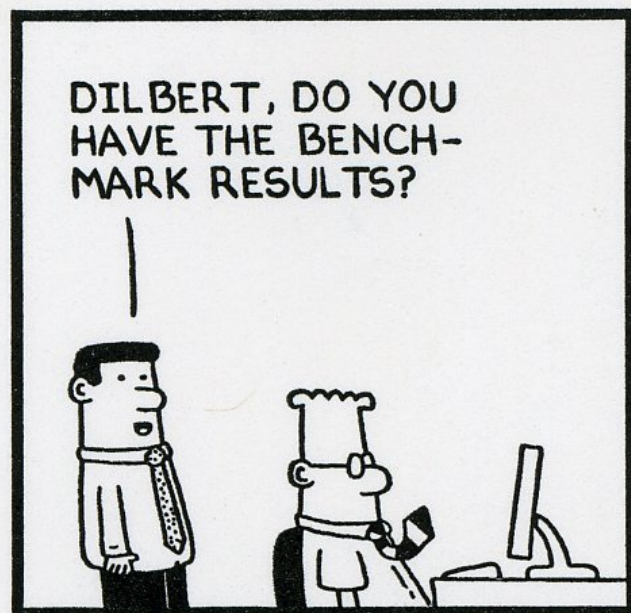
# The Three SELinux Policy Types

## Multi-Level Security (MLS) & Common Criteria

- The Common Criteria (CC) is an international security standard against which systems are evaluated. Many government customers require CC evaluated systems.
- Red Hat Enterprise Linux 5 meets EAL4+ with RBAC/LSPP/CAPP endorcements



# What's the Performance Overhead?



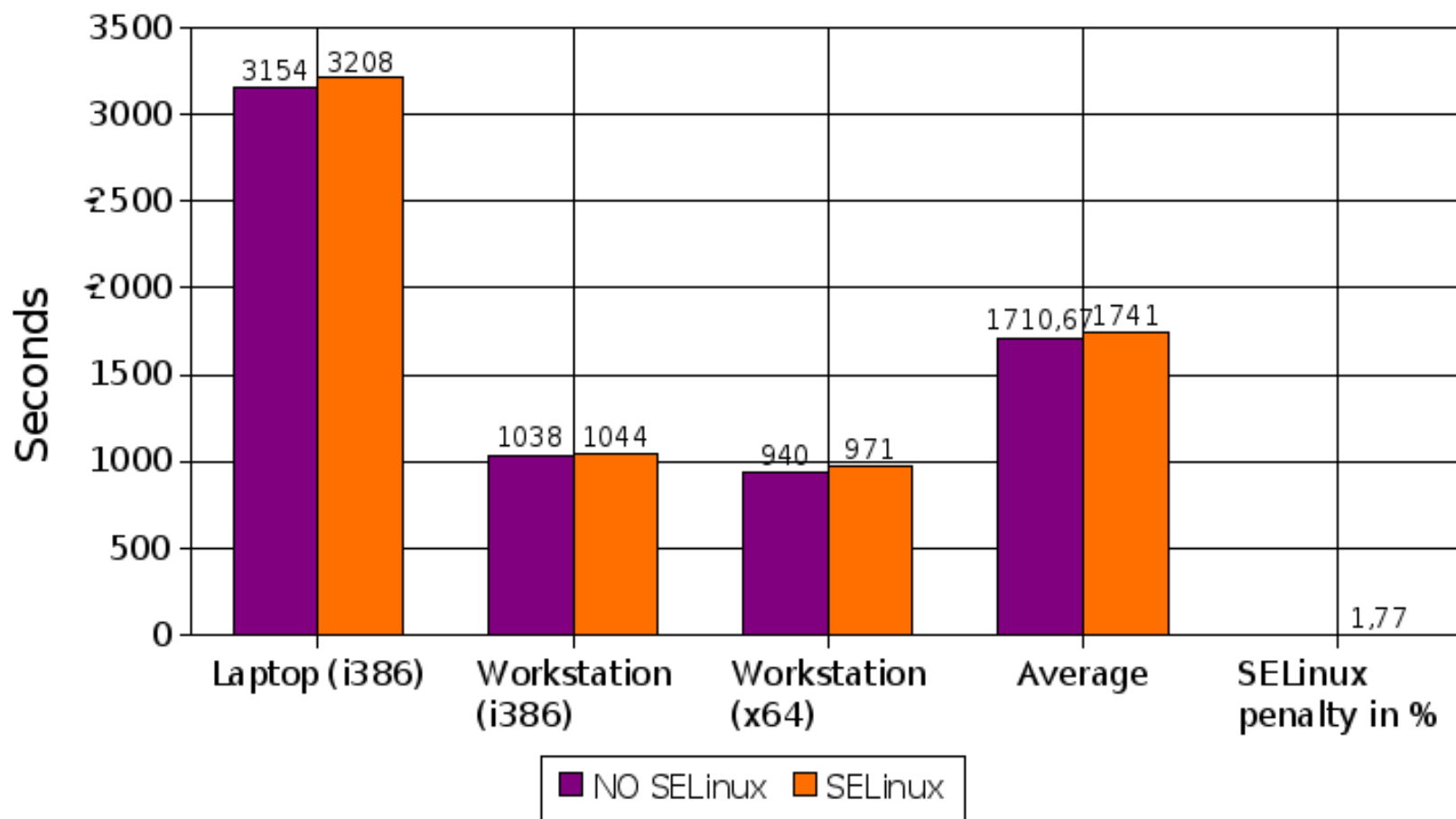
12-15-04 © 2004 Scott Adams, Inc./Dist. by UFS, Inc.



# What's the Performance Overhead?

RHEL5 SELinux: MySQL 5.0.22

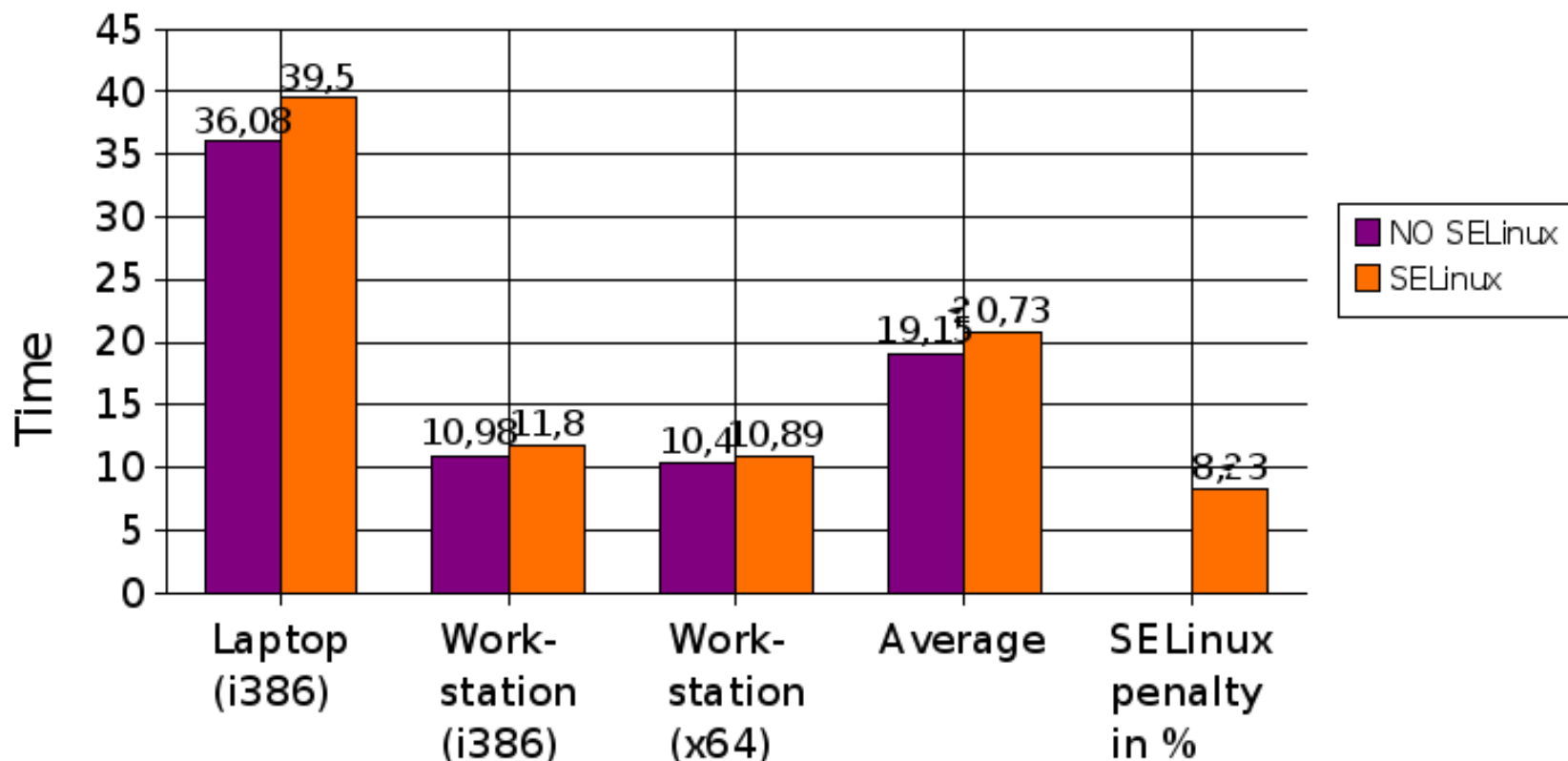
MySQL Benchmark suite: run-all-tests. Lower is better.



# What's the Performance Overhead?

RHEL5 SELinux: Apache 2.2.3 (worker)

11 tests: 100000 requests with 1-255 concurrent connections. Lower is better.



# What's the Performance Overhead?

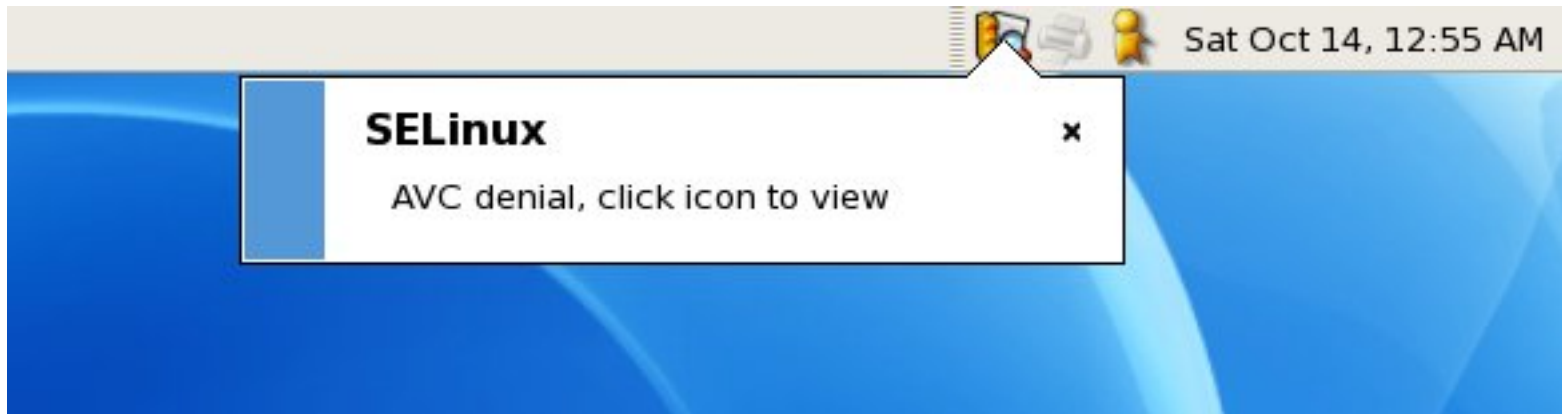
- Not official statistics
- Laptop = 2GHz, 2x 1GB RAM
- Workstation = 2.13GHz, 4x 1GB RAM
- Apache = Lots of threads
- MySQL = Lots of disk I/O

# **SELinux Usage**

## **(GUI & console)**

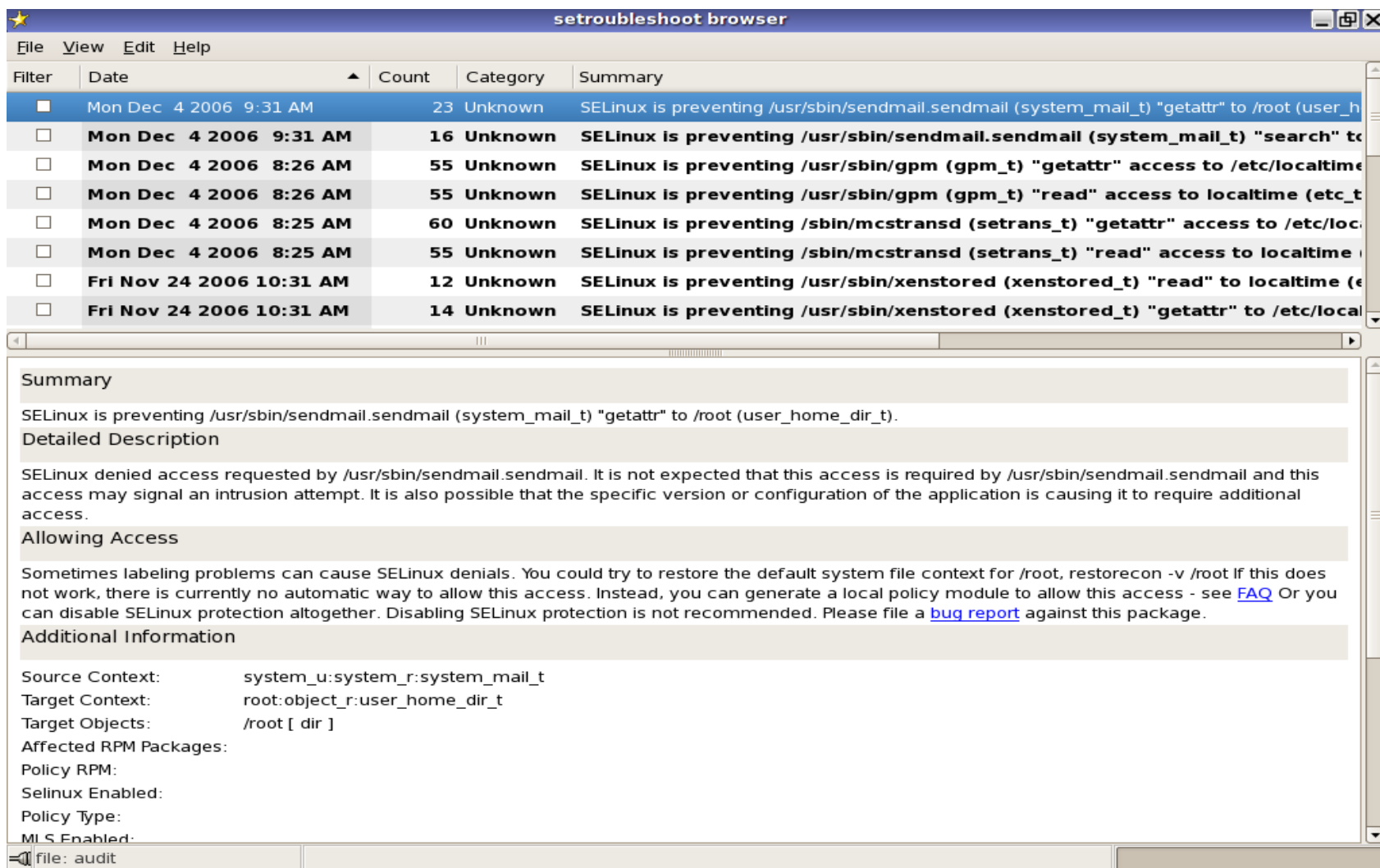
# End-User Perspective

- **sealert Notifications**



# End-User Perspective

- sealert Browser



The screenshot shows the 'setroubleshoot browser' window. It has a menu bar with 'File', 'View', 'Edit', and 'Help'. Below the menu is a table with columns: Filter, Date, Count, Category, and Summary. The table lists several SELinux denials. The first row is selected, and its details are shown in the 'Summary' pane below the table.

Filter	Date	Count	Category	Summary
<input checked="" type="checkbox"/>	Mon Dec 4 2006 9:31 AM	23	Unknown	SELinux is preventing /usr/sbin/sendmail.sendmail (system_mail_t) "getattr" to /root (user_h...
<input type="checkbox"/>	Mon Dec 4 2006 9:31 AM	16	Unknown	SELinux is preventing /usr/sbin/sendmail.sendmail (system_mail_t) "search" to
<input type="checkbox"/>	Mon Dec 4 2006 8:26 AM	55	Unknown	SELinux is preventing /usr/sbin/gpm (gpm_t) "getattr" access to /etc/localtime
<input type="checkbox"/>	Mon Dec 4 2006 8:26 AM	55	Unknown	SELinux is preventing /usr/sbin/gpm (gpm_t) "read" access to localtime (etc_t
<input type="checkbox"/>	Mon Dec 4 2006 8:25 AM	60	Unknown	SELinux is preventing /sbin/mcstransd (setrans_t) "getattr" access to /etc/loc
<input type="checkbox"/>	Mon Dec 4 2006 8:25 AM	55	Unknown	SELinux is preventing /sbin/mcstransd (setrans_t) "read" access to localtime
<input type="checkbox"/>	Fri Nov 24 2006 10:31 AM	12	Unknown	SELinux is preventing /usr/sbin/xenstored (xenstored_t) "read" to localtime (e
<input type="checkbox"/>	Fri Nov 24 2006 10:31 AM	14	Unknown	SELinux is preventing /usr/sbin/xenstored (xenstored_t) "getattr" to /etc/loc

**Summary**

SELinux is preventing /usr/sbin/sendmail.sendmail (system\_mail\_t) "getattr" to /root (user\_home\_dir\_t).

**Detailed Description**

SELinux denied access requested by /usr/sbin/sendmail.sendmail. It is not expected that this access is required by /usr/sbin/sendmail.sendmail and this access may signal an intrusion attempt. It is also possible that the specific version or configuration of the application is causing it to require additional access.

**Allowing Access**

Sometimes labeling problems can cause SELinux denials. You could try to restore the default system file context for /root, restorecon -v /root. If this does not work, there is currently no automatic way to allow this access. Instead, you can generate a local policy module to allow this access - see [FAQ](#). Or you can disable SELinux protection altogether. Disabling SELinux protection is not recommended. Please file a [bug report](#) against this package.

**Additional Information**

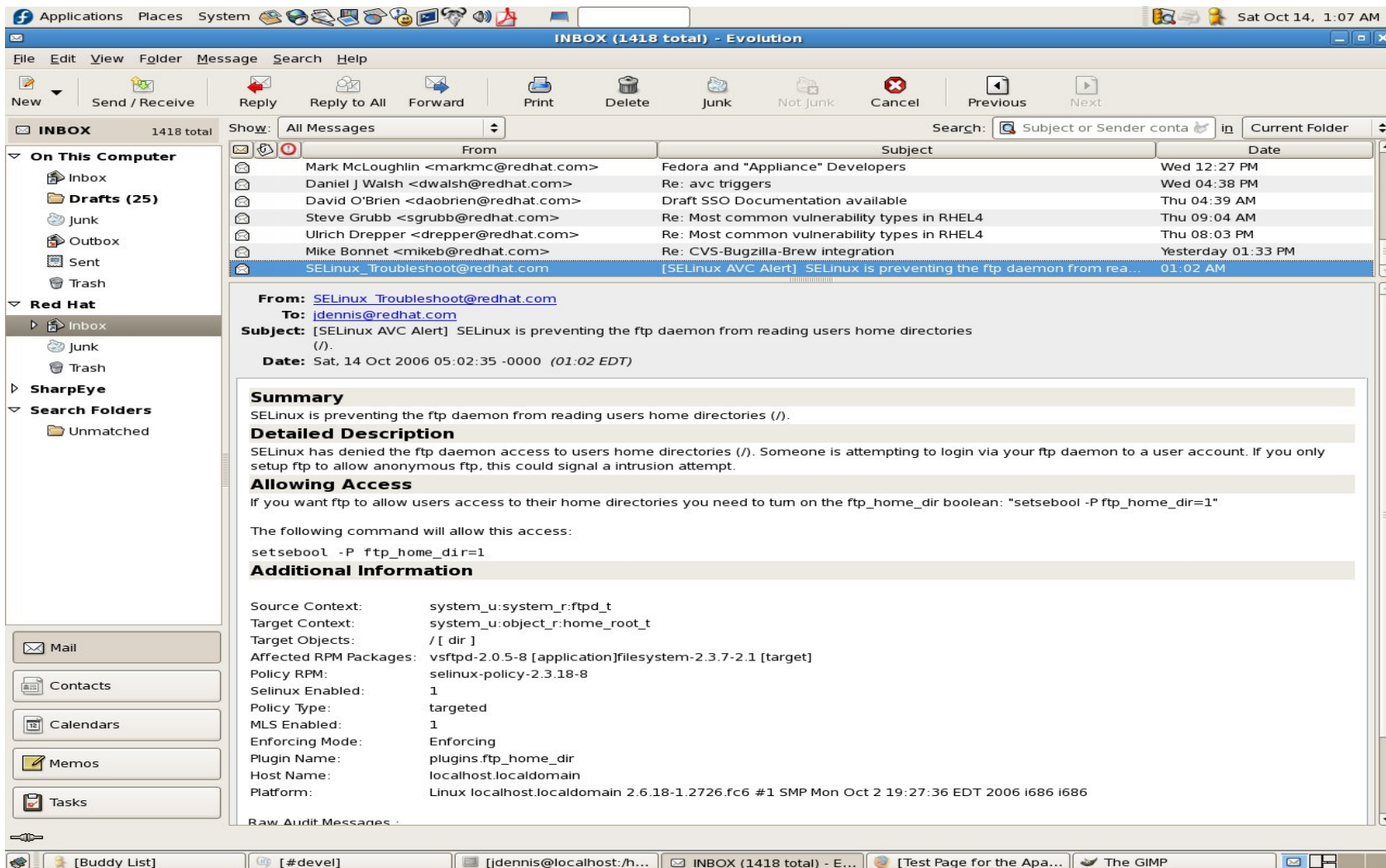
Source Context: system\_u:system\_r:system\_mail\_t  
 Target Context: root:object\_r:user\_home\_dir\_t  
 Target Objects: /root [ dir ]  
 Affected RPM Packages:  
 Policy RPM:  
 Selinux Enabled:  
 Policy Type:  
 MLS Enabled:

file: audit



# System Administrator Perspective

- **sealert + EMail Notifications**



The screenshot shows the Evolution email client interface. The left sidebar displays the folder structure: On This Computer (Inbox, Drafts (25), Junk, Outbox, Sent, Trash), Red Hat (Inbox, Junk, Trash), SharpEye, and Search Folders (Unmatched). The main pane shows the 'INBOX (1418 total) - Evolution' window. A list of messages is displayed, with the selected message being an SELinux AVC Alert from SELinux\_Troubleshoot@redhat.com to jdennis@redhat.com. The email content is as follows:

**From:** SELinux\_Troubleshoot@redhat.com  
**To:** jdennis@redhat.com  
**Subject:** [SELinux AVC Alert] SELinux is preventing the ftp daemon from reading users home directories (/).  
**Date:** Sat, 14 Oct 2006 05:02:35 -0000 (01:02 EDT)

**Summary**  
 SELinux is preventing the ftp daemon from reading users home directories (/).

**Detailed Description**  
 SELinux has denied the ftp daemon access to users home directories (/). Someone is attempting to login via your ftp daemon to a user account. If you only setup ftp to allow anonymous ftp, this could signal a intrusion attempt.

**Allowing Access**  
 If you want ftp to allow users access to their home directories you need to turn on the ftp\_home\_dir boolean: "setsebool -P ftp\_home\_dir=1"

The following command will allow this access:  
 setsebool -P ftp\_home\_dir=1

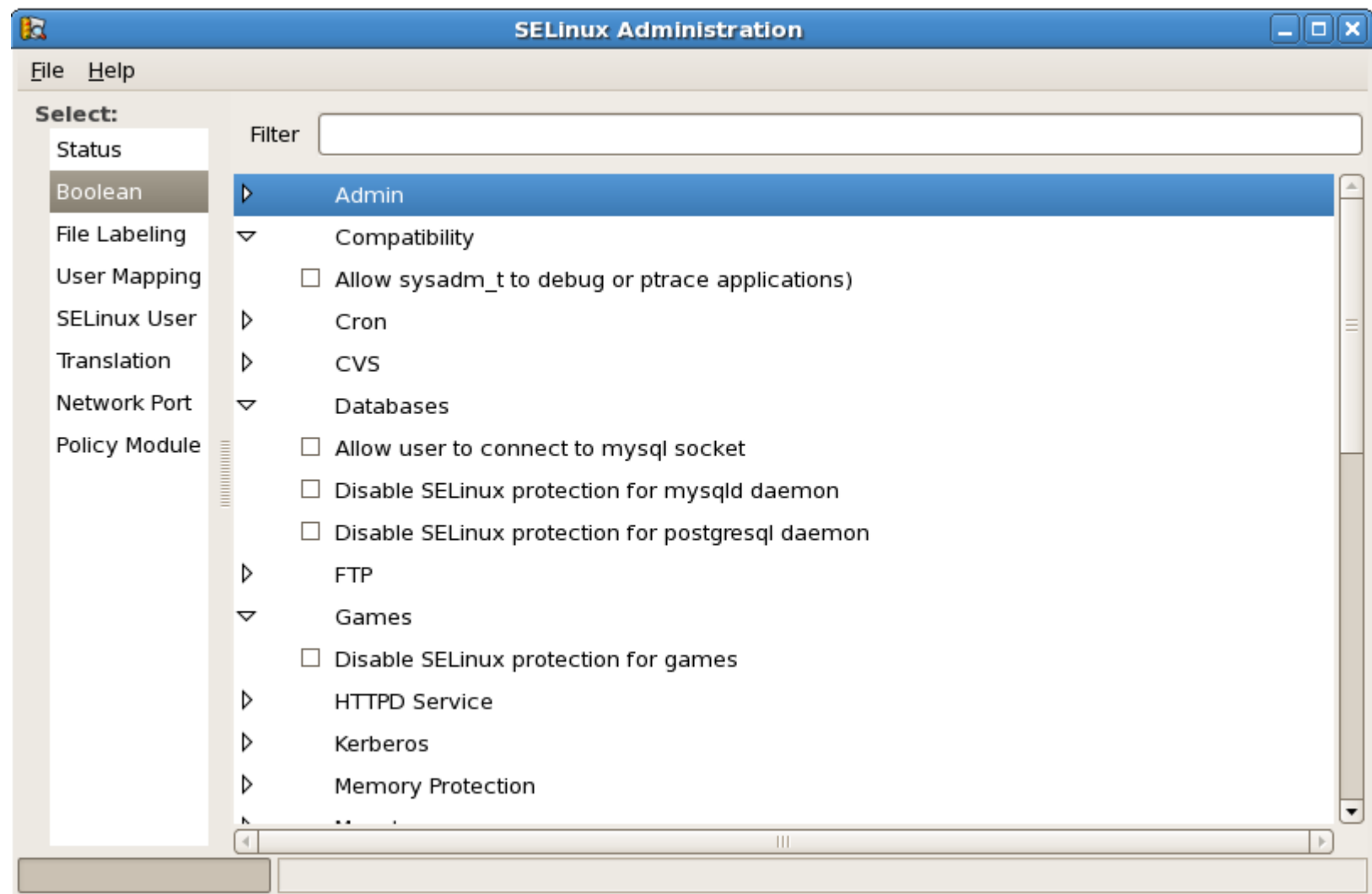
**Additional Information**

Source Context: system\_u:system\_r:ftpd\_t  
 Target Context: system\_u:object\_r:home\_root\_t  
 Target Objects: / [ dir ]  
 Affected RPM Packages: vsftpd-2.0.5-8 [application]filesystem-2.3.7-2.1 [target]  
 Policy RPM: selinux-policy-2.3.18-8  
 Selinux Enabled: 1  
 Policy Type: targeted  
 MLS Enabled: 1  
 Enforcing Mode: Enforcing  
 Plugin Name: plugins.ftp\_home\_dir  
 Host Name: localhost.localdomain  
 Platform: Linux localhost.localdomain 2.6.18-1.2726.fc6 #1 SMP Mon Oct 2 19:27:36 EDT 2006 i686 i686

Raw Audit Messages:

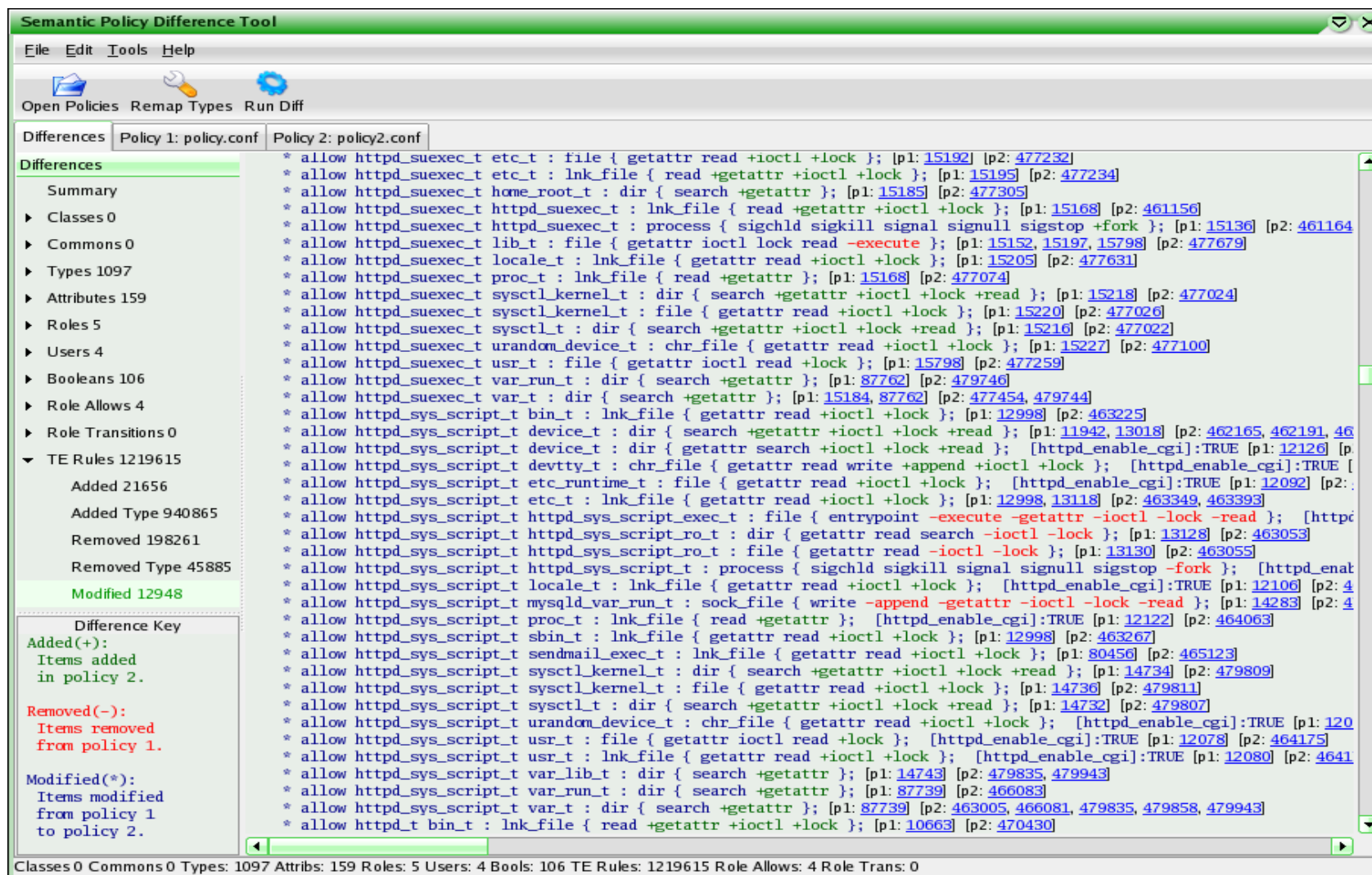
# System Administrator Perspective

- **system-config-selinux**



# System Administrator Perspective

- sediffx



**Semantic Policy Difference Tool**

File Edit Tools Help

Open Policies Remap Types Run Diff

Differences Policy 1: policy.conf Policy 2: policy2.conf

**Differences**

- Summary
- Classes 0
- Commons 0
- Types 1097
- Attributes 159
- Roles 5
- Users 4
- Booleans 106
- Role Allows 4
- Role Transitions 0
- TE Rules 1219615
  - Added 21656
  - Added Type 940865
  - Removed 198261
  - Removed Type 45885
  - Modified 12948

**Difference Key**

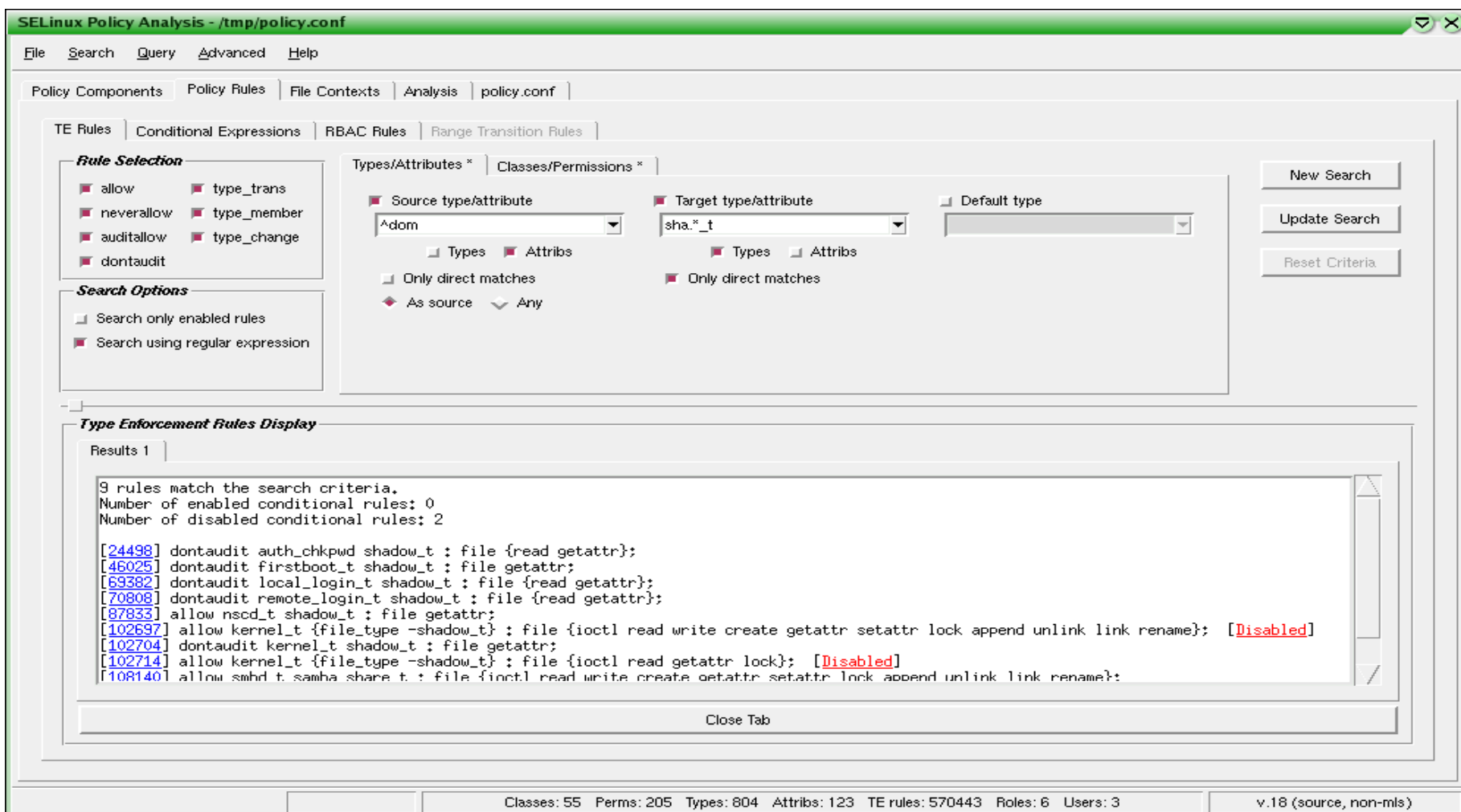
- Added(+): Items added in policy 2.
- Removed(-): Items removed from policy 1.
- Modified(\*): Items modified from policy 1 to policy 2.

\* allow httpd\_suexec\_t etc\_t : file { getattr read +ioctl +lock }; [p1: 15192] [p2: 477232]  
 \* allow httpd\_suexec\_t etc\_t : lnk\_file { read +getattr +ioctl +lock }; [p1: 15195] [p2: 477234]  
 \* allow httpd\_suexec\_t home\_root\_t : dir { search +getattr }; [p1: 15185] [p2: 477305]  
 \* allow httpd\_suexec\_t httpd\_suexec\_t : lnk\_file { read +getattr +ioctl +lock }; [p1: 15168] [p2: 461156]  
 \* allow httpd\_suexec\_t httpd\_suexec\_t : process { sigchld sigkill signal signull sigstop +fork }; [p1: 15136] [p2: 461164]  
 \* allow httpd\_suexec\_t lib\_t : file { getattr ioctl lock read -execute }; [p1: 15152, 15197, 15798] [p2: 477679]  
 \* allow httpd\_suexec\_t locale\_t : lnk\_file { getattr read +ioctl +lock }; [p1: 15205] [p2: 477631]  
 \* allow httpd\_suexec\_t proc\_t : lnk\_file { read +getattr }; [p1: 15168] [p2: 477074]  
 \* allow httpd\_suexec\_t sysctl\_kernel\_t : dir { search +getattr +ioctl +lock +read }; [p1: 15218] [p2: 477024]  
 \* allow httpd\_suexec\_t sysctl\_kernel\_t : file { getattr read +ioctl +lock }; [p1: 15220] [p2: 477026]  
 \* allow httpd\_suexec\_t sysctl\_t : dir { search +getattr +ioctl +lock +read }; [p1: 15216] [p2: 477022]  
 \* allow httpd\_suexec\_t urandom\_device\_t : chr\_file { getattr read +ioctl +lock }; [p1: 15227] [p2: 477100]  
 \* allow httpd\_suexec\_t usr\_t : file { getattr ioctl read +lock }; [p1: 15798] [p2: 477259]  
 \* allow httpd\_suexec\_t var\_run\_t : dir { search +getattr }; [p1: 87762] [p2: 479746]  
 \* allow httpd\_suexec\_t var\_t : dir { search +getattr }; [p1: 15184, 87762] [p2: 477454, 479744]  
 \* allow httpd\_sys\_script\_t bin\_t : lnk\_file { getattr read +ioctl +lock }; [p1: 12998] [p2: 463225]  
 \* allow httpd\_sys\_script\_t device\_t : dir { search +getattr +ioctl +lock +read }; [p1: 11942, 13018] [p2: 462165, 462191, 462192]  
 \* allow httpd\_sys\_script\_t device\_t : dir { getattr search +ioctl +lock +read }; [httpd\_enable\_cgi]:TRUE [p1: 12126] [p2: 463053]  
 \* allow httpd\_sys\_script\_t devtty\_t : chr\_file { getattr read write +append +ioctl +lock }; [httpd\_enable\_cgi]:TRUE [p1: 12092] [p2: 463053]  
 \* allow httpd\_sys\_script\_t etc\_runtime\_t : file { getattr read +ioctl +lock }; [httpd\_enable\_cgi]:TRUE [p1: 12092] [p2: 463053]  
 \* allow httpd\_sys\_script\_t etc\_t : lnk\_file { getattr read +ioctl +lock }; [p1: 12998, 13118] [p2: 463349, 463393]  
 \* allow httpd\_sys\_script\_t httpd\_sys\_script\_exec\_t : file { entrypoint -execute -getattr -ioctl -lock -read }; [httpd\_enable\_cgi]:TRUE [p1: 12106] [p2: 463053]  
 \* allow httpd\_sys\_script\_t httpd\_sys\_script\_ro\_t : dir { getattr read search -ioctl -lock }; [p1: 13128] [p2: 463053]  
 \* allow httpd\_sys\_script\_t httpd\_sys\_script\_ro\_t : file { getattr read -ioctl -lock }; [p1: 13130] [p2: 463055]  
 \* allow httpd\_sys\_script\_t httpd\_sys\_script\_t : process { sigchld sigkill signal signull sigstop -fork }; [httpd\_enable\_cgi]:TRUE [p1: 12122] [p2: 464063]  
 \* allow httpd\_sys\_script\_t locale\_t : lnk\_file { getattr read +ioctl +lock }; [httpd\_enable\_cgi]:TRUE [p1: 12106] [p2: 463053]  
 \* allow httpd\_sys\_script\_t mysql\_var\_run\_t : sock\_file { write -append -getattr -ioctl -lock -read }; [p1: 14283] [p2: 464063]  
 \* allow httpd\_sys\_script\_t proc\_t : lnk\_file { read +getattr }; [httpd\_enable\_cgi]:TRUE [p1: 12122] [p2: 464063]  
 \* allow httpd\_sys\_script\_t sbin\_t : lnk\_file { getattr read +ioctl +lock }; [p1: 12998] [p2: 463267]  
 \* allow httpd\_sys\_script\_t sendmail\_exec\_t : lnk\_file { getattr read +ioctl +lock }; [p1: 80456] [p2: 465123]  
 \* allow httpd\_sys\_script\_t sysctl\_kernel\_t : dir { search +getattr +ioctl +lock +read }; [p1: 14734] [p2: 479809]  
 \* allow httpd\_sys\_script\_t sysctl\_kernel\_t : file { getattr read +ioctl +lock }; [p1: 14736] [p2: 479811]  
 \* allow httpd\_sys\_script\_t sysctl\_t : dir { search +getattr +ioctl +lock +read }; [p1: 14732] [p2: 479807]  
 \* allow httpd\_sys\_script\_t urandom\_device\_t : chr\_file { getattr read +ioctl +lock }; [httpd\_enable\_cgi]:TRUE [p1: 12078] [p2: 464175]  
 \* allow httpd\_sys\_script\_t usr\_t : lnk\_file { getattr read +ioctl +lock }; [httpd\_enable\_cgi]:TRUE [p1: 12080] [p2: 464175]  
 \* allow httpd\_sys\_script\_t var\_lib\_t : dir { search +getattr }; [p1: 14743] [p2: 479835, 479943]  
 \* allow httpd\_sys\_script\_t var\_run\_t : dir { search +getattr }; [p1: 87739] [p2: 466083]  
 \* allow httpd\_sys\_script\_t var\_t : dir { search +getattr }; [p1: 87739] [p2: 463005, 466081, 479835, 479858, 479943]  
 \* allow httpd\_t bin\_t : lnk\_file { read +getattr +ioctl +lock }; [p1: 10663] [p2: 470430]

Classes 0 Commons 0 Types: 1097 Attrs: 159 Roles: 5 Users: 4 Bools: 106 TE Rules: 1219615 Role Allows: 4 Role Trans: 0

# System Administrator Perspective

- apol



The screenshot shows the SELinux Policy Analysis tool interface. The title bar is "SELinux Policy Analysis - /tmp/policy.conf". The menu bar includes File, Search, Query, Advanced, and Help. The main window has tabs for Policy Components, Policy Rules, File Contexts, Analysis, and policy.conf. Under the Analysis tab, there are sub-tabs for TE Rules, Conditional Expressions, RBAC Rules, and Range Transition Rules. The "Rule Selection" section on the left lists various rule types: allow, neverallow, auditallow, dontaudit, type\_trans, type\_member, and type\_change. The "Search Options" section includes checkboxes for "Search only enabled rules" and "Search using regular expression". The search criteria are defined in the "Types/Attributes \*" and "Classes/Permissions \*" sections. The "Source type/attribute" is set to "dom" and the "Target type/attribute" is set to "sha.\*\_t". The "Default type" is set to "file". The "Only direct matches" checkbox is checked. The "As source" checkbox is checked. The "Only direct matches" checkbox is checked. The "New Search", "Update Search", and "Reset Criteria" buttons are on the right. The "Type Enforcement Rules Display" section shows the results of the search. It indicates that 9 rules match the search criteria. The results are listed in a table with columns for ID, rule name, and action. The rules are: [24498] dontaudit auth\_chkpwd shadow\_t : file {read getattr}; [46025] dontaudit firstboot\_t shadow\_t : file getattr; [69382] dontaudit local\_login\_t shadow\_t : file {read getattr}; [70808] dontaudit remote\_login\_t shadow\_t : file {read getattr}; [87833] allow nsd\_t shadow\_t : file getattr; [102697] allow kernel\_t {file\_type -shadow\_t} : file {ioctl read write create getattr setattr lock append unlink link rename}; [Disabled] [102704] dontaudit kernel\_t shadow\_t : file getattr; [102714] allow kernel\_t {file\_type -shadow\_t} : file {ioctl read getattr lock}; [Disabled] [108140] allow smbd\_t.samba.share\_t : file {ioctl read write create getattr setattr lock append unlink link rename};. The "Close Tab" button is at the bottom. The status bar at the bottom shows: Classes: 55 Perms: 205 Types: 804 Attribs: 123 TE rules: 570443 Roles: 6 Users: 3 v.18 (source, non-mls).

# SELinux Usage

## (Hints & Tips)

# System Administrator Perspective

- **semanage**

Configure elements of SELinux policy without  
modification/recompilation of policy sources  
. . . . aka on the fly

**Example:** Dynamically Allowing Apache to listen on  
port 1234

```
# semanage port -a -t httpd_port_t -p tcp 1234
```

# System Administrator Perspective

- **semanage** (more examples)

**Example:** Allow shawn to join “webadmin\_u” group

```
# semanage login -a -s webadmin_u shawn
```

**Example:** Relabel files for access by Apache

```
# semanage fcontext -a -t \  
httpd_sys_content_t "/data/webpages(/.*)?"
```

# System Administrator Perspective

- **semanage** (most important example)

**You don't need to disable SELinux to fix a single error!**

```
type=SYSCALL msg=audit(1204719775.306:738): arch=400000003 syscall=54
success=no exit=-19 a0=4 a1=8933 a2=bfceclbc a3=bfceclbc items=0
ppid=3900 pid=5003 auid=501 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0
sgid=0 fsgid=0 tty=(none) comm="ip" exe="/sbin/ip"
subj=user_u:system_r:ifconfig_t:s0 key=(null)
```

## The Fix:

```
# semanage permissive -a ifconfig_t
```



# System Administrator Perspective

- **audit2allow**

Allows generation of SELinux policy rules from logs of denied operations

**Example:** Fix all the errors on the system (completely not a good idea on a real system)

```
# cat /var/log/audit/audit.log | audit2allow -M FixAll
```

```
Generating type enforcement file: FixAll.te
```

```
Compiling policy: checkmodule -M -m -o FixAll.mod FixAll.te
```

```
Building package: semodule_package -o FixAll.pp -m FixAll.mod
```

```
# semodule -i FixAll.pp
```



# Scenarios

# Scenario: Fixing the RHT corporate VPN “update”

- Red Hat has a Corporate Standard Build (CSB) for desktop environments
- Red Hat pushes updates to said CSB
- I “tweak” my configuration files
- When RHT pushed a CSB update, it broke my VPN settings

# Scenario: Fixing the RHT corporate VPN “update”

## **/var/log/messages:**

```
type=SYSCALL msg=audit(1204719775.306:738): arch=400000003 syscall=54
success=no exit=-19 a0=4 a1=8933 a2=bfceclbc a3=bfceclbc items=0
ppid=3900 pid=5003 auid=501 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0
sgid=0 fsgid=0 tty=(none) comm="ip" exe="/sbin/ip"
subj=user_u:system_r:ifconfig_t:s0 key=(null)
```

**Now what?**

# Scenario: Fixing the RHT corporate VPN “update”

```
type=SYSCALL msg=audit(1204719775.306:738): arch=400000003 syscall=54
success=no exit=-19 a0=4 a1=8933 a2=bfceclbc a3=bfceclbc items=0
ppid=3900 pid=5003 auid=501 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0
sgid=0 fsgid=0 tty=(none) comm="ip" exe="/sbin/ip"
subj=user_u:system_r:ifconfig_t:s0 key=(null)
```

## What I Know:

- 1) AVC Event ID 738
- 2) syscall=54 (I'd have to google this)
- 3) root (or an application on its behalf) was running /sbin/ip
- 4) context = user\_u:system\_r:ifconfig\_t:s0

# Scenario: Fixing the RHT corporate VPN “update”

```
type=SYSCALL msg=audit(1204719775.306:738): arch=400000003 syscall=54
success=no exit=-19 a0=4 a1=8933 a2=bfceclbc a3=bfceclbc items=0
ppid=3900 pid=5003 auid=501 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0
sgid=0 fsgid=0 tty=(none) comm="ip" exe="/sbin/ip"
subj=user_u:system_r:ifconfig_t:s0 key=(null)
```

## My Options:

1) Create a SELinux Policy Module

```
# ausearch -x "/sbin/ip" | audit2allow -M MyVPNFix
```

# Scenario: Fixing the RHT corporate VPN “update”

```
type=SYSCALL msg=audit(1204719775.306:738): arch=400000003 syscall=54
success=no exit=-19 a0=4 a1=8933 a2=bfceclbc a3=bfceclbc items=0
ppid=3900 pid=5003 auid=501 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0
sgid=0 fsgid=0 tty=(none) comm="ip" exe="/sbin/ip"
subj=user_u:system_r:ifconfig_t:s0 key=(null)
```

## My Options:

1) Create a SELinux Policy Module

```
# ausearch -x "/sbin/ip" | audit2allow -M MyVPNFix
# semodule -i MyVPNFix.pp
```

# Scenario: Fixing the RHT corporate VPN “update”

```
type=SYSCALL msg=audit(1204719775.306:738): arch=400000003 syscall=54
success=no exit=-19 a0=4 a1=8933 a2=bfceclbc a3=bfceclbc items=0
ppid=3900 pid=5003 auid=501 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0
sgid=0 fsgid=0 tty=(none) comm="ip" exe="/sbin/ip"
subj=user_u:system_r:ifconfig_t:s0 key=(null)
```

## My Options:

2) Disable enforcement of ifconfig\_t (there is no need to turn SELinux completely off!)

```
# semanage permissive -a ifconfig_t
```



# Questions