

RED HAT :: CHICAGO :: 2009

**SUMMIT**

**FOLLOW US:**

[TWITTER.COM/REDHATSUMMIT](http://TWITTER.COM/REDHATSUMMIT)

**TWEET ABOUT US:**

ADD #SUMMIT AND/OR #JBOSSWORLD TO THE END  
OF YOUR EVENT-RELATED TWEET

presented by



RED HAT :: CHICAGO :: 2009

SUMMIT

# SELinux for Mere Mortals

Thomas Cameron  
Solutions Architect Team Lead, Red Hat  
September 2nd, 2009

presented by



# Red Hat

- Leading SELinux vendor.
- John Dennis and Dan Walsh, both Red Hat engineers, acknowledged by the NSA for their contributions to SELinux at:  
<http://www.nsa.gov/research/selinux/contrib.shtml>
- Red Hat acknowledged by the NSA as a corporate contributor as well.

# Agenda

- What is SELinux
  - A brief History
  - Type Enforcement
  - Role-Based Access Control (RBAC)
  - Multi-Level Security (MLS)
  - Mandatory vs. Discretionary Access Control

# Agenda

- What Can I Do with SELinux?
  - Confine Programs' Privileges
  - Protect from Exploits
  - Prevent System Access to Users' Private Details

# Agenda

- SELinux Architecture
  - Security Context
    - Users
    - Roles
    - Domains/Types
    - Sensitivity
    - Category
  - Security Policy

# What is SELinux?

- A brief History
  - Created by the United States National Security Agency (NSA) as set of patches to the Linux kernel using Linux Security Modules (LSM)
  - Released by the NSA under the GNU General Public License (GPL) in 2000
  - Adopted by the upstream Linux kernel in 2003

# What is SELinux?

- Type Enforcement (TE)
  - Enforces Mandatory Access Control (MAC) over Discretionary Access Control (DAC)
  - Provides control over process execution as well as domain transition. For example, the init process kicks off various scripts in the /etc/rc.d directory and those scripts call executable binaries. TE manages the transition of those executables from their parent domain (init) to their own domain (like httpd).

# What is SELinux?

- Role-Based Access Control (RBAC)
  - Roles defined for various processes
  - Permissions assigned to roles rather than individual users

# What is SELinux?

- Multi-Level Security (MLS)
  - Allows different access levels to different data based on security levels
    - Top level can access top, middle and low
    - Mid level can access mid and low but not top
    - Low level can access low but not mid or top
  - Can even have different windows in the GUI with different levels of security and no way to copy/paste from high to mid

# What is SELinux?

- Mandatory vs. Discretionary Access Control
  - DAC – standard mechanism for Linux.
    - All processes run with a user and group. If that user/group has access to files, so does the process
    - Root and users have ability (discretion) to override or change security with chmod, chown and related utilities
    - Processes which run as root (think application services) can access **\*everything\***

# What is SELinux?

- Mandatory vs. Discretionary Access Control
  - MAC – SELinux is a MAC implementation
    - Fine grained permissions on all processes, files, devices, sockets, ports, etc.
    - Administratively defined policy.
    - Security decisions made based on all information, not just identity
    - Processes running as root can still only access those areas of the system which policy allows.

# What Can I Do With It?

- Confine Programs' Privileges
  - Programs confined to their own context, even programs running as root can still not access information outside of their own security context

# What Can I Do With It?

- Protects against exploits
  - RBAC and TE (sandboxing) means that even if a “bad guy” exploits something like a buffer overflow, since that process runs in a specific context it can't access anything else on the system.

# What Can I Do With It?

- Prevent System Access to Users' Private Details
  - Rogue/compromised processes can still not access home directories or mail files

# SELinux Architecture

- Security Context (example)

```
File Edit View Terminal Tabs Help
[root@host175 ~]# ls -lZ /etc/passwd /etc/httpd/conf/httpd.conf
-rw-r--r--  root root system_u:object_r:etc_t          /etc/passwd
-rw-r--r--  root root system_u:object_r:httpd_config_t /etc/httpd/conf/httpd.conf
[root@host175 ~]#
```

# User Identity and Role

- SELinux user account correlated with object, typically ends in `_u`, as in the `system_u` identity in the previous slide
- Role defines which SELinux user identities are permitted in which domains, typically ends in `_r` as in `object_r` in the previous slide
- Role can be changed using `newrole`
- Role has access to types or domains

# User identity and role

- Security Context
  - Users
  - Roles
  - Domains/Types
  - Sensitivity (optional)
  - Category (optional)

# Domain type and role

- Processes (subjects) execute in domains
- Resources (objects) are in protected domains called types
- Subjects and objects have a domain or a type, ending in `_t` as in the `httpd_config_t` and `etc_t` types in the earlier slide

# ID, Role, Domain and Type

- Example:

```
File Edit View Terminal Tabs Help
[tcameron@case ~]$ ps axZ | grep httpd
user_u:system_r:httpd_t      8487 ?      Ss      0:00 /usr/sbin/httpd
user_u:system_r:httpd_t      8489 ?      S      0:00 /usr/sbin/httpd
user_u:system_r:httpd_t      8490 ?      S      0:00 /usr/sbin/httpd
user_u:system_r:httpd_t      8491 ?      S      0:00 /usr/sbin/httpd
user_u:system_r:httpd_t      8492 ?      S      0:00 /usr/sbin/httpd
user_u:system_r:httpd_t      8493 ?      S      0:00 /usr/sbin/httpd
user_u:system_r:httpd_t      8494 ?      S      0:00 /usr/sbin/httpd
user_u:system_r:httpd_t      8495 ?      S      0:00 /usr/sbin/httpd
user_u:system_r:httpd_t      8496 ?      S      0:00 /usr/sbin/httpd
user_u:system_r:unconfined_t  8499 pts/2    S+     0:00 grep httpd
[tcameron@case ~]$ 
```

# Domain type and role

- For example, the web server binary (httpd) has a type of httpd\_exec\_t.
- The running process (httpd) belongs to the httpd\_t domain
- Web server data is of the type httpd\_sys\_content\_t.
- The targeted policy allows subjects in httpd\_t to access files with httpd\_sys\_content\_t

# Sensitivity and Categories

- Only used in strict and mls policies (think government)
- Hierarchical
- Category is optional
- No read up, no write down

# Security Context

- Every object and subject has a context
- Stored in extended attributes (xattrs) on ext2/3 filesystems
- Stored in running kernel for port, network interfaces and so on.

# Security Context

- Format:
  - user:role:type:sensitivity:category
- Can be changed with chcon, restorecon, fixfiles
- Defined in:
  - /etc/selinux/targeted/contexts/files/file\_contexts
  - /etc/selinux/targeted/contexts/files/file\_contexts.homedirs

# SELinux Policy

- Set of rules used by SELinux
  - Defines the security context
    - User identity
    - Role
    - Type/domain
    - Sensitivity
    - Category
  - Defines how each domain accesses each type
  - Defines transitions and other access

# Targeted Policy

- Default policy for RHEL, Fedora
- Targets only specific services
- Available in source or binary
- Source policy written with m4

# Targeted Policy

- Every subject and object runs in unconfined\_t domain except for those with defined policy

# Strict Policy

- EVERY subject is in a confined domain
- Much more complex
  - For instance, root has low privileges, must enter password for system configuration just like a regular user

# MLS Policy

- Multi-layer security
- Different security levels for different processes and objects
- Allows for different apps to run in different contexts on the same system
  - One window might be higher or lower security than another, no read up, no write down.

# Protected Services

- There are over 100, including all of the standard ones
  - dhcpcd
  - httpd
  - mysqld
  - named
  - nscd
  - ntpd
  - portmap
  - postmaster
  - snmpd
  - squid
  - syslogd
  - winbindd
  - bind
  - ypbind

# SELinux Tools

- system-config-selinux
- chcon
- restorecon
- setfiles
- fixfiles
- setenforce
- getenforce
- newrole
- getsebool
- setsebool

# Policy Location

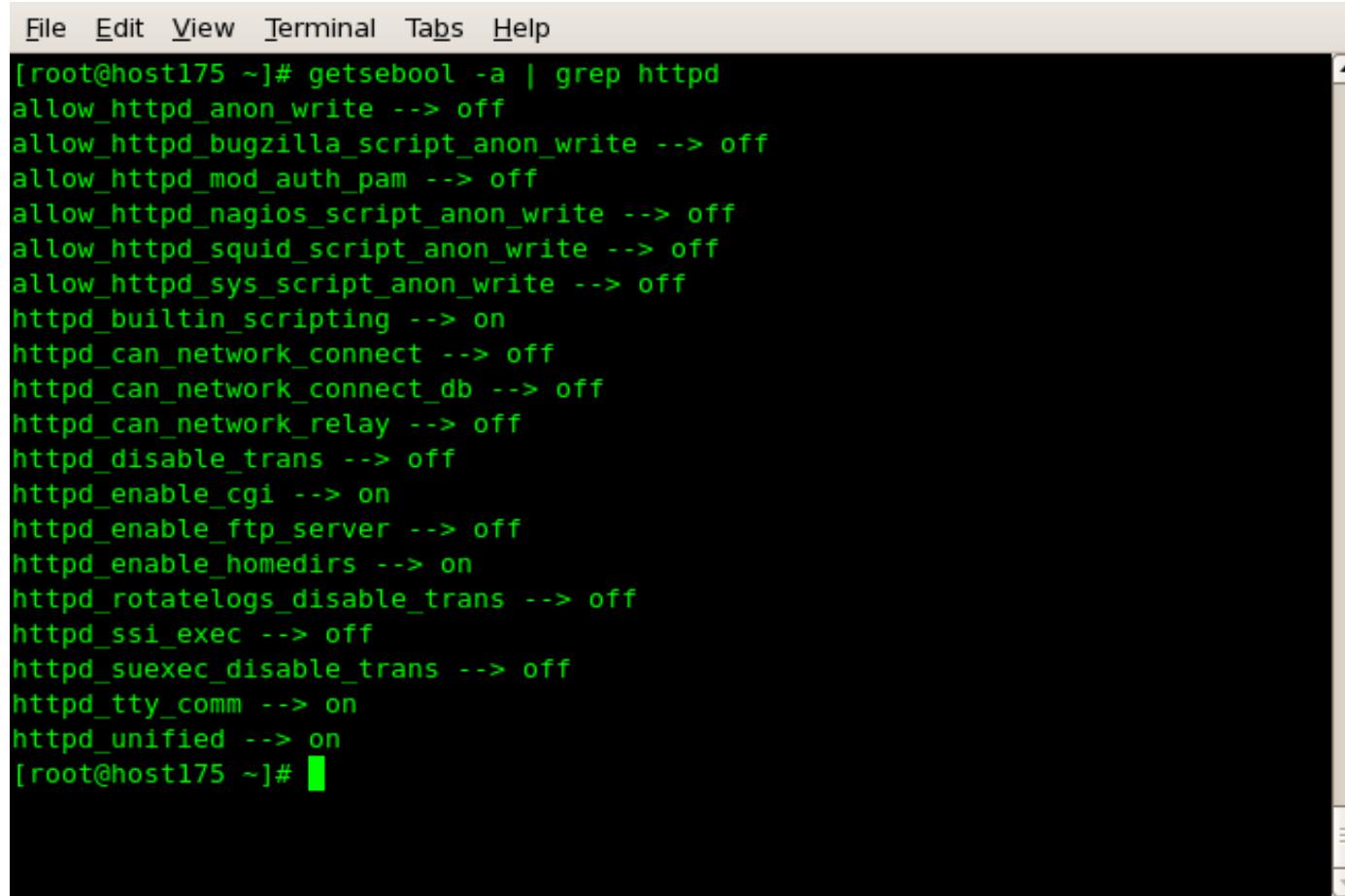
- Look in /etc/selinux
  - /etc/selinux/targeted
  - /etc/selinux/targeted/policy
  - /etc/selinux/targeted/contexts

# Policy Booleans

- Allow runtime policy modification
- Each has a default, usually false
- `getsebool` and `setsebool` to manage
- `setsebool -P` recompiles with change
- Writes changes to:
  - `/etc/selinux/targeted/modules/active/booleans.local`

# Policy Booleans

- Can display all booleans using getsebool -a (there are hundreds of them)



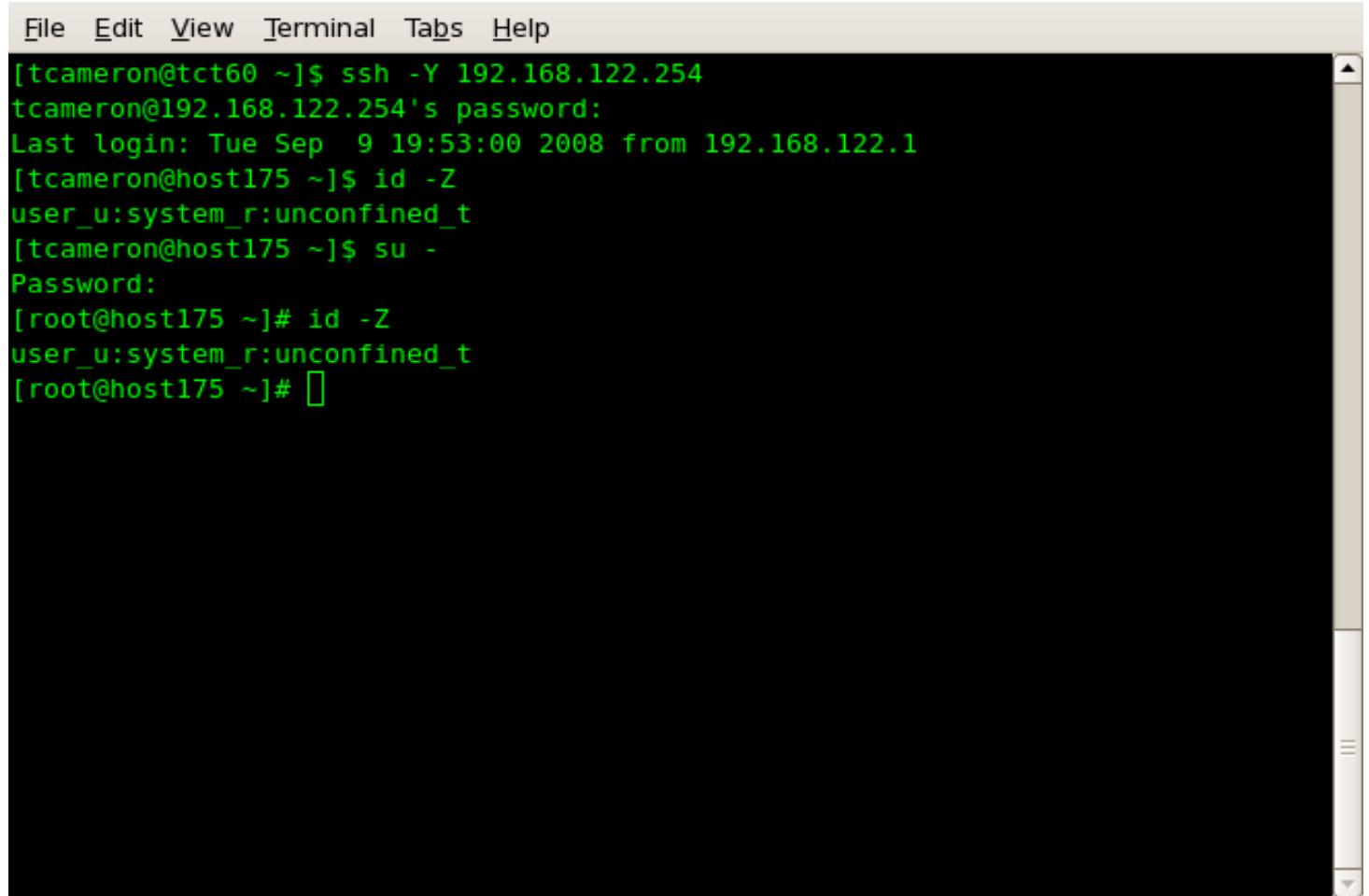
```
File Edit View Terminal Tabs Help
[root@host175 ~]# getsebool -a | grep httpd
allow_httpd_anon_write --> off
allow_httpd_bugzilla_script_anon_write --> off
allow_httpd_mod_auth_pam --> off
allow_httpd_nagios_script_anon_write --> off
allow_httpd_squid_script_anon_write --> off
allow_httpd_sys_script_anon_write --> off
httpd_builtin_scripting --> on
httpd_can_network_connect --> off
httpd_can_network_connect_db --> off
httpd_can_network_relay --> off
httpd_disable_trans --> off
httpd_enable_cgi --> on
httpd_enable_ftp_server --> off
httpd_enable_homedirs --> on
httpd_rotatelogs_disable_trans --> off
httpd_ssi_exec --> off
httpd_suexec_disable_trans --> off
httpd_tty_comm --> on
httpd_unified --> on
[root@host175 ~]#
```

# Security Context Info

- Almost every command takes the -Z argument
  - ls -Z
  - id -Z or secon
  - ps -Z
  - mkdir -Z
  - install -Z
  - cp -Z
  - find -context

# Security Context Info

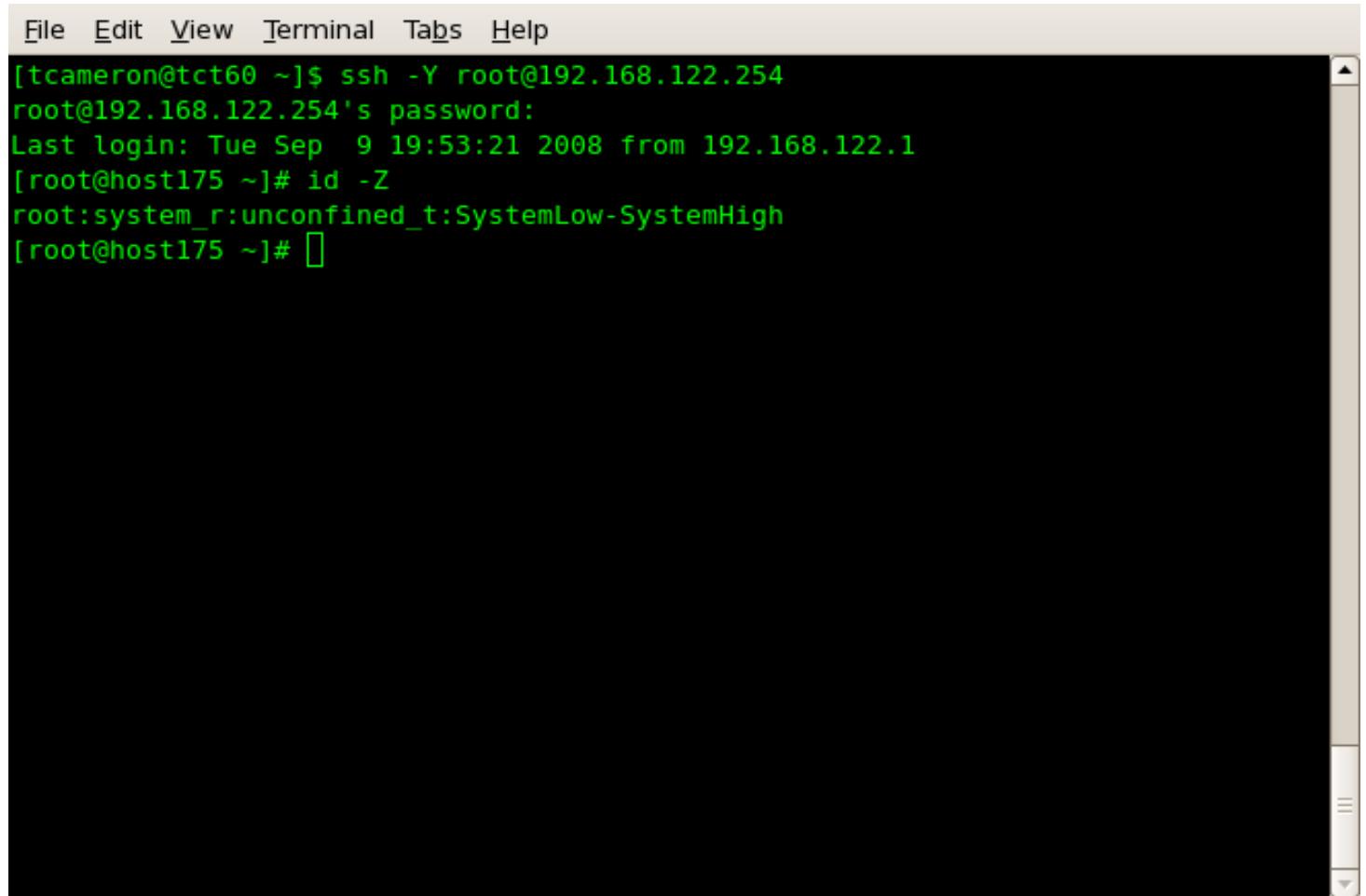
- Note that using su does NOT set context correctly!
- Example:



```
File Edit View Terminal Tabs Help
[tcameron@tct60 ~]$ ssh -Y 192.168.122.254
tcameron@192.168.122.254's password:
Last login: Tue Sep  9 19:53:00 2008 from 192.168.122.1
[tcameron@host175 ~]$ id -Z
user_u:system_r:unconfined_t
[tcameron@host175 ~]$ su -
Password:
[root@host175 ~]# id -Z
user_u:system_r:unconfined_t
[root@host175 ~]# [ ]
```

# Security Context Info

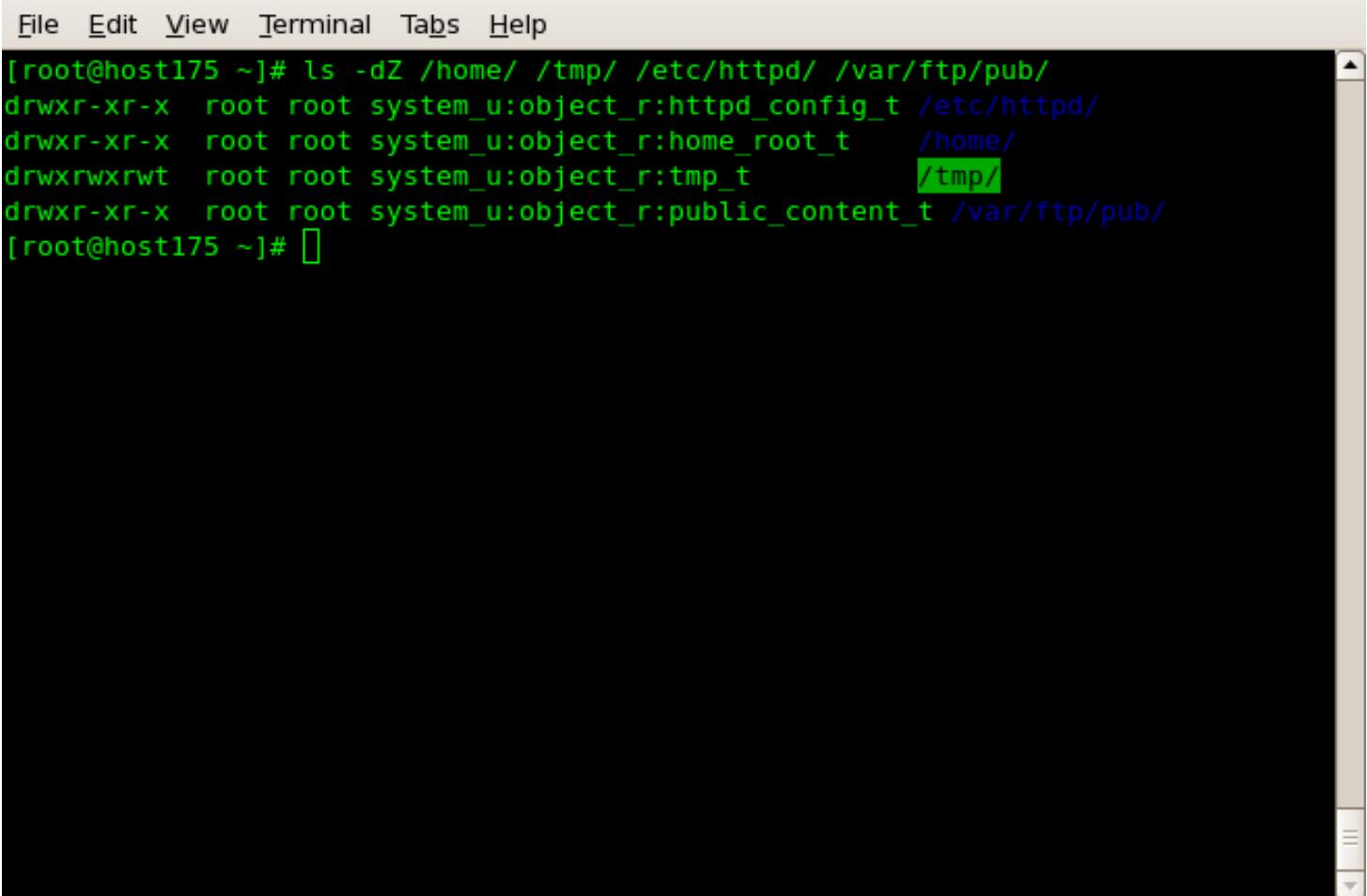
- In this example, root logs in via ssh:
- Example:



```
File Edit View Terminal Tabs Help
[tcameron@tct60 ~]$ ssh -Y root@192.168.122.254
root@192.168.122.254's password:
Last login: Tue Sep  9 19:53:21 2008 from 192.168.122.1
[root@host175 ~]# id -Z
root:system_r:unconfined_t:SystemLow-SystemHigh
[root@host175 ~]# 
```

# SELinux Examples

- Checking contexts of various files
- /home
- /tmp
- /etc/httpd
- /var/ftp/pub

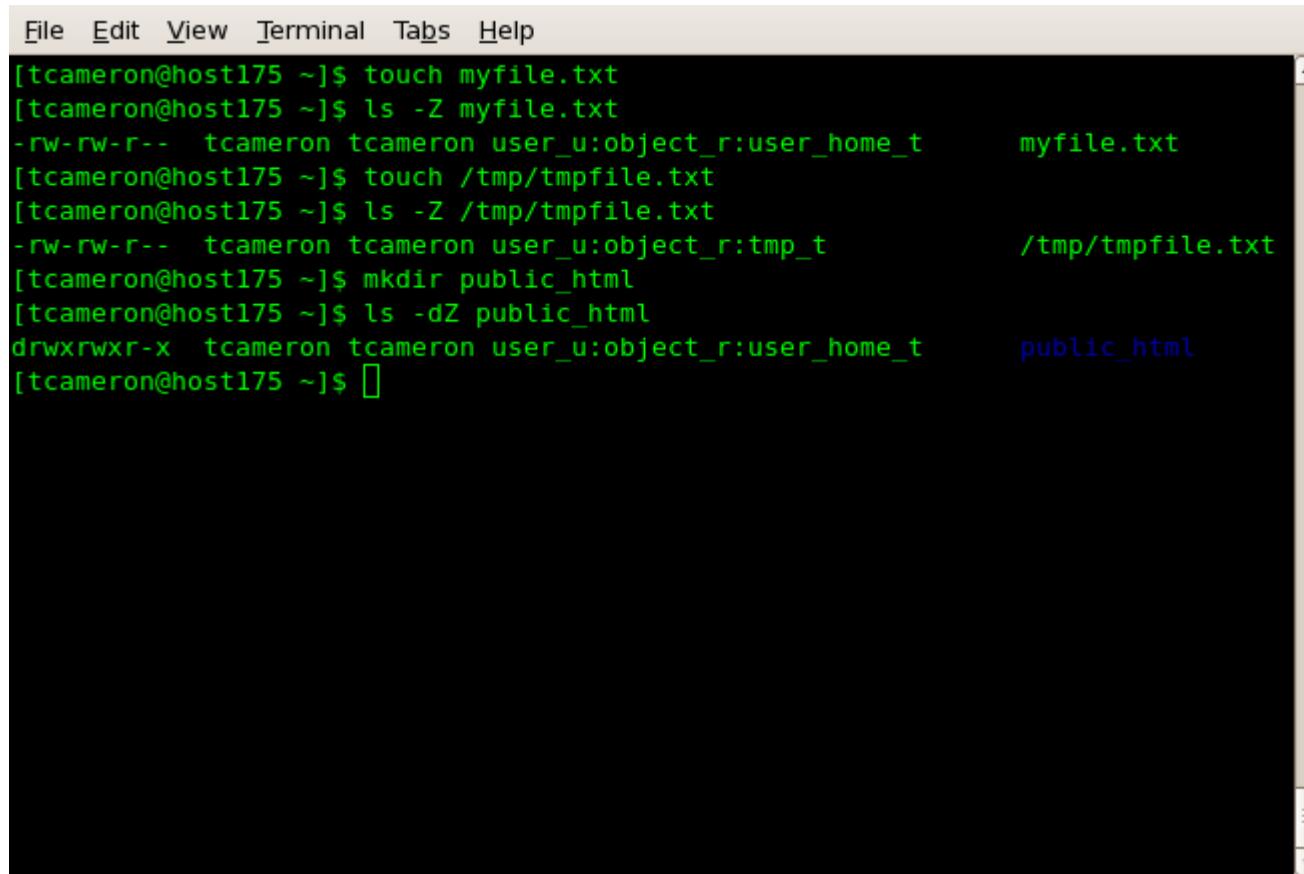


File Edit View Terminal Tabs Help

```
[root@host175 ~]# ls -dZ /home/ /tmp/ /etc/httpd/ /var/ftp/pub/
drwxr-xr-x  root root system_u:object_r:httdp_config_t /etc/httpd/
drwxr-xr-x  root root system_u:object_r:home_root_t    /home/
drwxrwxrwt root root system_u:object_r:tmp_t          /tmp/
drwxr-xr-x  root root system_u:object_r:public_content_t /var/ftp/pub/
[root@host175 ~]# []
```

# SELinux Examples

- Creating files and noting contexts



A screenshot of a terminal window with a light gray header bar containing the menu: File, Edit, View, Terminal, Tabs, Help. The main area is black with white text. It shows the following command-line session:

```
[tcameron@host175 ~]$ touch myfile.txt
[tcameron@host175 ~]$ ls -Z myfile.txt
-rw-rw-r-- tcameron tcameron user_u:object_r:user_home_t myfile.txt
[tcameron@host175 ~]$ touch /tmp/tmpfile.txt
[tcameron@host175 ~]$ ls -Z /tmp/tmpfile.txt
-rw-rw-r-- tcameron tcameron user_u:object_r:tmp_t /tmp/tmpfile.txt
[tcameron@host175 ~]$ mkdir public_html
[tcameron@host175 ~]$ ls -dZ public_html
drwxrwxr-x tcameron tcameron user_u:object_r:user_home_t public_html
[tcameron@host175 ~]$ []
```

# SELinux Examples

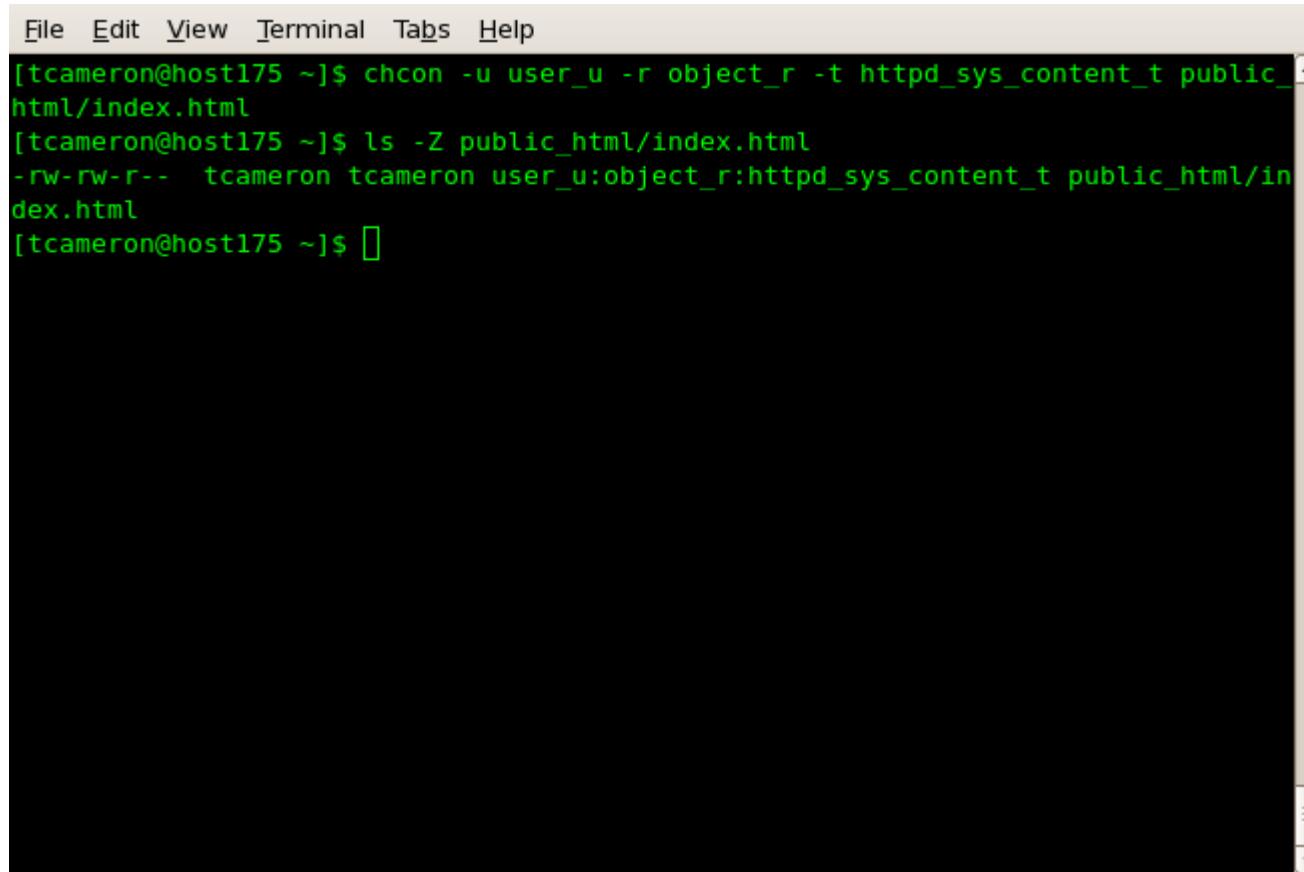
- Changing file contexts
  - Example - web page created in wrong context and moved

# SELinux Examples

```
File Edit View Terminal Tabs Help
[tcameron@host175 ~]$ echo "This is my page" > index.html
[tcameron@host175 ~]$ ls -Z index.html
-rw-rw-r-- tcameron tcameron user_u:object_r:user_home_t      index.html
[tcameron@host175 ~]$ mv index.html public_html/
[tcameron@host175 ~]$ ls -Z public_html/index.html
-rw-rw-r-- tcameron tcameron user_u:object_r:user_home_t      public_html/index
.html
[tcameron@host175 ~]$ 
```

# SELinux Examples

- Use chcon to manually change context

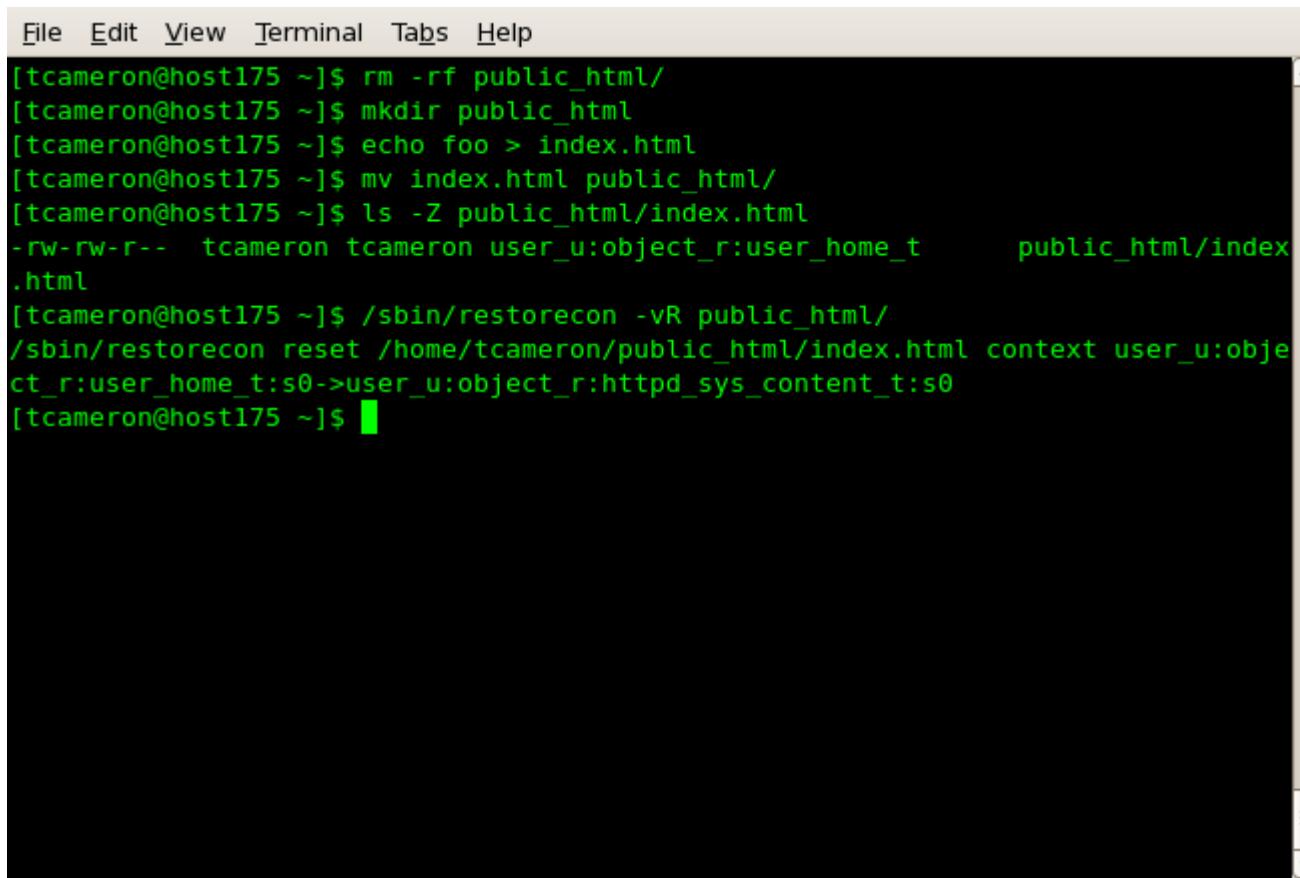


The image shows a terminal window with a dark background and light-colored text. At the top, a menu bar is visible with options: File, Edit, View, Terminal, Tabs, and Help. The main area of the terminal contains the following command-line session:

```
File Edit View Terminal Tabs Help
[tcameron@host175 ~]$ chcon -u user_u -r object_r -t httpd_sys_content_t public_html/index.html
[tcameron@host175 ~]$ ls -Z public_html/index.html
-rw-rw-r-- tcameron tcameron user_u:object_r:httpd_sys_content_t public_html/index.html
[tcameron@host175 ~]$
```

# SELinux Examples

- Or do it the easy way with restorecon:



The image shows a terminal window with a light gray header bar containing the menu items: File, Edit, View, Terminal, Tabs, and Help. The main area of the terminal is black with white text. It displays the following command-line session:

```
[tcameron@host175 ~]$ rm -rf public_html/
[tcameron@host175 ~]$ mkdir public_html
[tcameron@host175 ~]$ echo foo > index.html
[tcameron@host175 ~]$ mv index.html public_html/
[tcameron@host175 ~]$ ls -Z public_html/index.html
-rw-rw-r-- tcameron user_u:object_r:user_home_t      public_html/index
.html
[tcameron@host175 ~]$ /sbin/restorecon -vR public_html/
/sbin/restorecon reset /home/tcameron/public_html/index.html context user_u:obj
ct_r:user_home_t:s0->user_u:object_r:httpd_sys_content_t:s0
[tcameron@host175 ~]$ █
```

# SELinux Examples

- How to troubleshoot Apache vs. SELinux issues
  - For instance, allowing access to ~/public\_html

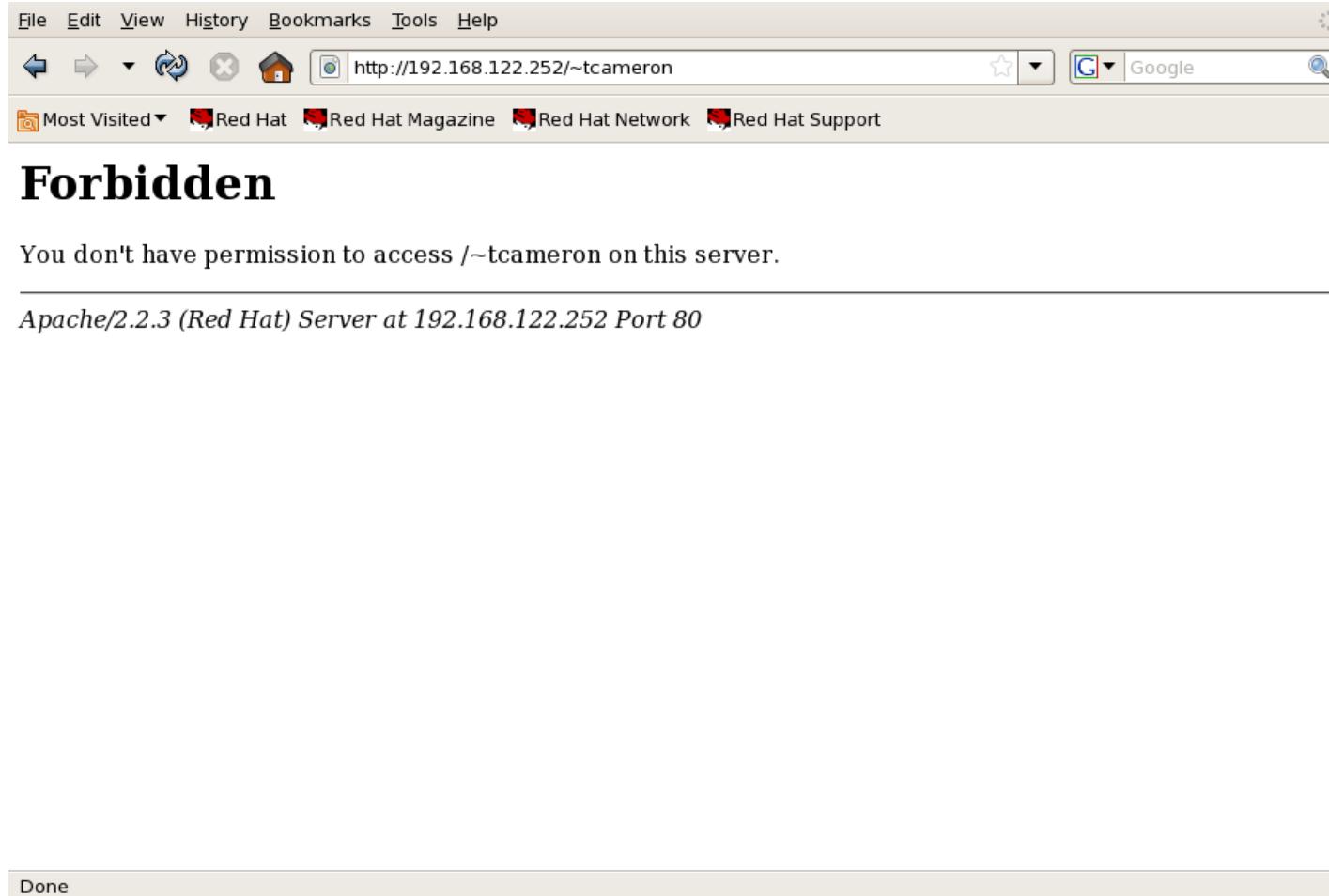
# SELinux Examples

- Set up Apache to allow access to ~/public\_html
- Add a user and have that user set up a ~/public\_html directory with all the right permissions (711 /home/user, 755 /home/user/public\_html)
- In this example, a simple error is made:

# SELinux Examples

```
File Edit View Terminal Tabs Help
[root@host175 ~]# useradd tcameron
[root@host175 ~]# passwd tcameron
Changing password for user tcameron.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[root@host175 ~]# ssh tcameron@localhost
tcameron@localhost's password:
[tcameron@host175 ~]$ mkdir public_html
[tcameron@host175 ~]$ echo This is my page > index.html
[tcameron@host175 ~]$ mv index.html public_html/
[tcameron@host175 ~]$ 
```

# SELinux Examples



# SELinux Examples

- Use sealert to see what happened.
  - `sealert -a /var/log/audit/audit.log`

# SELinux Examples

```
File Edit View Terminal Tabs Help
[root@host175 ~]# sealert -a /var/log/audit/audit.log
100% donefound 1 alerts in /var/log/audit/audit.log
-----
Summary:
SELinux is preventing the httpd from using potentially mislabeled files
(/home/tcameron/public_html/index.html).

Detailed Description:
SELinux has denied httpd access to potentially mislabeled file(s)
(/home/tcameron/public_html/index.html). This means that SELinux will not allow
httpd to use these files. It is common for users to edit files in their home
directory or tmp directories and then move (mv) them to system directories. The
problem is that the files end up with the wrong file context which confined
applications are not allowed to access.

Allowing Access:
If you want httpd to access this files, you need to relabel them using
restorecon -v '/home/tcameron/public_html/index.html'. You might want to relabel
the entire directory using restorecon -R -v '/home/tcameron/public_html'.
```

# SELinux Examples

```
File Edit View Terminal Tabs Help
First Seen           Tue Sep  9 21:53:20 2008
Last Seen           Tue Sep  9 21:53:20 2008
Local ID            c32ca51d-02b1-431e-bdf3-c6598006a9ff
Line Numbers        83, 84, 85, 86

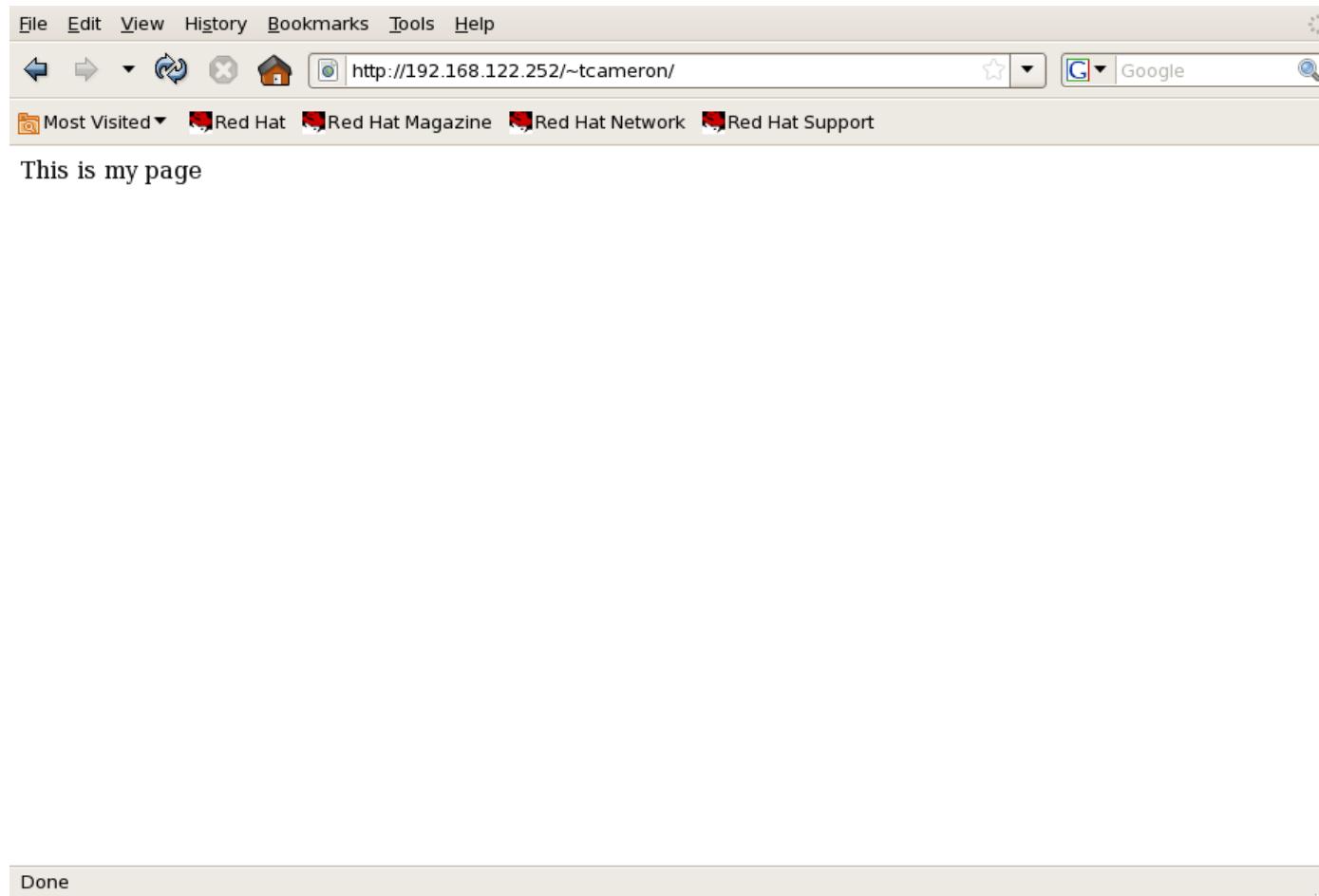
Raw Audit Messages

type=AVC msg=audit(1221015200.301:63): avc:  denied  { getattr } for  pid=4026 c
omm="httpd" path="/home/tcameron/public_html/index.html" dev=xvda3 ino=589315 sc
ontext=root:system_r:httpd_t:s0 tcontext=user_u:object_r:user_home_t:s0 tclass=f
ile

type=SYSCALL msg=audit(1221015200.301:63): arch=c000003e syscall=4 success=no ex
it=-13 a0=2ab385e68800 a1=7fff336c8180 a2=7fff336c8180 a3=2ab385e6b3a8 items=0 p
pid=4024 pid=4026 auid=0 uid=48 gid=48 euid=48 suid=48 egid=48 sgid=48
fsgid=48 tty=(none) ses=1 comm="httpd" exe="/usr/sbin/httpd" subj=root:system_r:
httpd_t:s0 key=(null)

[root@host175 ~]# restorecon -vR /home/tcameron/
restorecon reset /home/tcameron/public_html/index.html context user_u:object_r:u
ser_home_t:s0->user_u:object_r:httpd_sys_content_t:s0
[root@host175 ~]#
```

# SELinux Examples



# SELinux Examples

- What about something like setting up a virtual host?
  - Set up the virtual host in httpd.conf, but put it somewhere weird.

# SELinux Examples

```
File Edit View Terminal Tabs Help

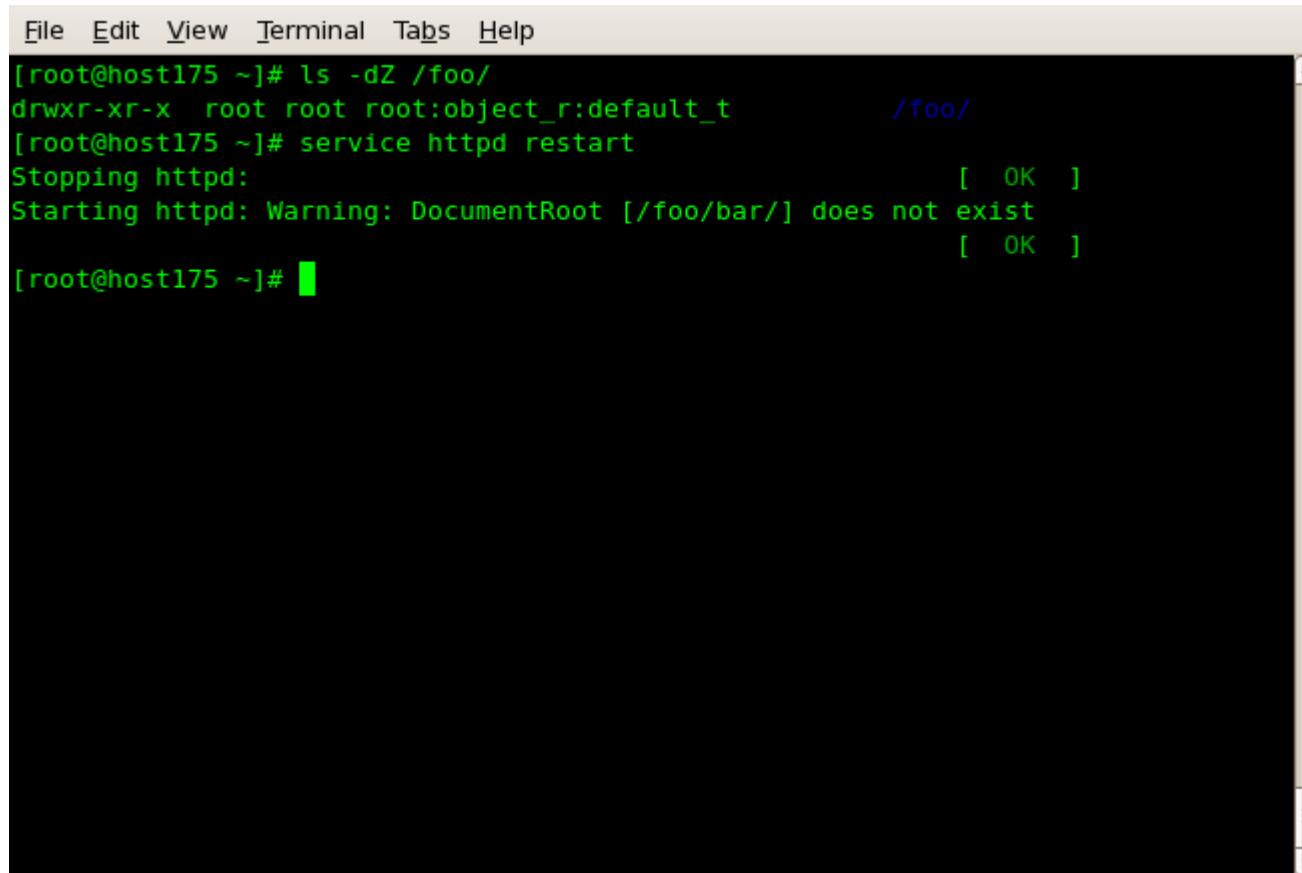
#
# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost container.
# The first VirtualHost section is used for requests without a known
# server name.
#
#<VirtualHost *:80>
#    ServerAdmin webmaster@dummy-host.example.com
#    DocumentRoot /www/docs/dummy-host.example.com
#    ServerName dummy-host.example.com
#    ErrorLog logs/dummy-host.example.com-error_log
#    CustomLog logs/dummy-host.example.com-access_log common
#</VirtualHost>

<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host.example.com
    DocumentRoot /foo/bar
    ServerName dummy-host.example.com
    ErrorLog logs/dummy-host.example.com-error_log
    CustomLog logs/dummy-host.example.com-access_log common
</VirtualHost>
[]
```

# SELinux Examples

- Restart the httpd service

# SELinux Examples



File Edit View Terminal Tabs Help

```
[root@host175 ~]# ls -dZ /foo/
drwxr-xr-x  root root root:object_r:default_t      /foo/
[root@host175 ~]# service httpd restart
Stopping httpd:                                         [  OK  ]
Starting httpd: Warning: DocumentRoot [/foo/bar/] does not exist
                                         [  OK  ]
[root@host175 ~]#
```

# SELinux Examples

- sealert to the rescue!
  - Or maybe not...?

# SELinux Examples

```
File Edit View Terminal Tabs Help
found 1 alerts in /var/log/audit/audit.log
-----
Summary:
SELinux is preventing access to files with the default label, default_t.

Detailed Description:
SELinux permission checks on files labeled default_t are being denied. These
files/directories have the default label on them. This can indicate a labeling
problem, especially if the files being referred to are not top level
directories. Any files/directories under standard system directories, /usr,
/var, /dev, /tmp, ..., should not be labeled with the default label. The default
label is for files/directories which do not have a label on a parent directory.
So if you create a new directory in / you might legitimately get this label.

Allowing Access:
If you want a confined domain to use these files you will probably need to
relabel the file/directory with chcon. In some cases it is just easier to
relabel the system, to relabel execute: "touch /.autorelabel; reboot"
:|
```

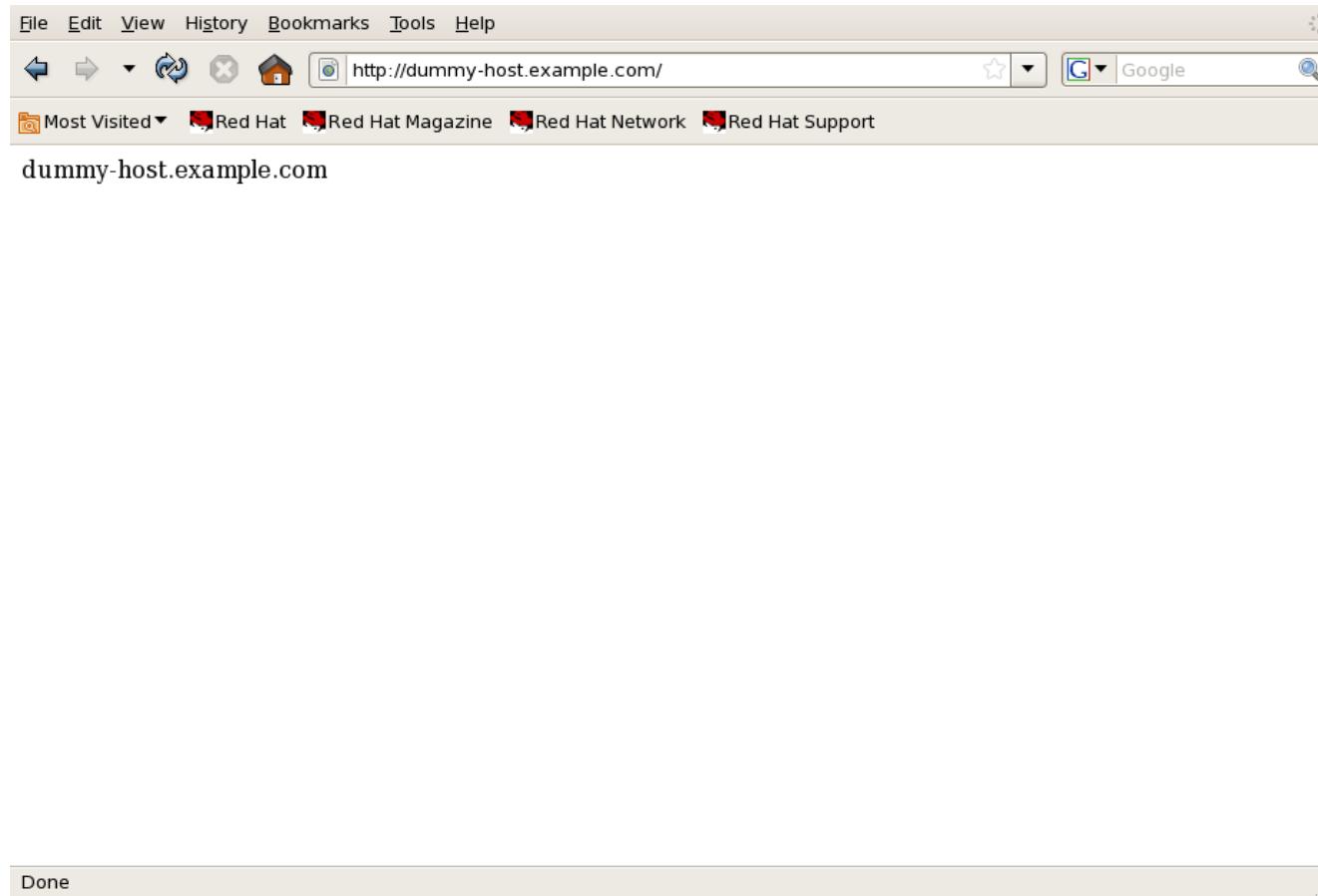
# SELinux Examples

- But wait, relabeling the filesystem won't help here!
- You can use chcon --reference to set context, then semanage fcontext to make it permanent (i.e. survive a relabel)

# SELinux Examples

```
File Edit View Terminal Tabs Help
[root@tct60 ~]# mkdir -p /foo/bar
[root@tct60 ~]# service httpd restart
Stopping httpd: [OK]
Starting httpd: Warning: DocumentRoot [/foo/bar] does not exist [OK]
[root@tct60 ~]# chcon -vR --reference /var/www/html/ /foo/
context of /foo/ changed to system_u:object_r:httdp_sys_content_t
context of /foo//bar changed to system_u:object_r:httdp_sys_content_t
[root@tct60 ~]# service httpd restart
Stopping httpd: [OK]
Starting httpd: [OK]
[root@tct60 ~]# semanage fcontext -a -t httpd_sys_content_t "/foo(/.*)?"
[root@tct60 ~]# rm -rf /foo/
[root@tct60 ~]# mkdir -p /foo/bar
[root@tct60 ~]# restorecon -vR /foo/
restorecon reset /foo context root:object_r:default_t:s0->system_u:object_r:httdp_sys_content_t:s0
restorecon reset /foo/bar context root:object_r:default_t:s0->system_u:object_r:httdp_sys_content_t:s0
[root@tct60 ~]#
```

# SELinux Examples



# SELinux Examples

- Some examples of booleans
- Setting up public\_html on NFS mounted home directories
- Mount an exported filesystem under /home and set up the home and public\_html directories and permissions

# SELinux Examples



# SELinux Examples

- In this case, there are two possible sets of booleans which might be interfering.
- `getsebool -a | grep http`
- `getsebool -a | grep nfs`

# SELinux Examples

```
File Edit View Terminal Tabs Help
[root@host175 ~]# getsebool -a | grep http
allow_httpd_anon_write --> off
allow_httpd_bugzilla_script_anon_write --> off
allow_httpd_mod_auth_pam --> off
allow_httpd_nagios_script_anon_write --> off
allow_httpd_squid_script_anon_write --> off
allow_httpd_sys_script_anon_write --> off
httpd_builtin_scripting --> on
httpd_can_network_connect --> off
httpd_can_network_connect_db --> off
httpd_can_network_relay --> off
httpd_disable_trans --> off
httpd_enable_cgi --> on
httpd_enable_ftp_server --> off
httpd_enable_homedirs --> on
httpd_rotatelogs_disable_trans --> off
httpd_ssi_exec --> off
httpd_suexec_disable_trans --> off
httpd_tty_comm --> on
httpd_unified --> on
[root@host175 ~]#
```

# SELinux Examples

```
File Edit View Terminal Tabs Help
[root@host175 ~]# getsebool -a | grep nfs
allow_ftpd_use_nfs --> off
allow_nfsd_anon_write --> off
nfs_export_all_ro --> on
nfs_export_all_rw --> on
nfsd_disable_trans --> off
samba_share_nfs --> off
use_nfs_home_dirs --> off
[root@host175 ~]#
```

# SELinux Examples

- You can always use sealert to check as well

# SELinux Examples

```
File Edit View Terminal Tabs Help
Summary:
SELinux prevented httpd from reading files stored on a NFS filesystem.

Detailed Description:
SELinux prevented httpd from reading files stored on a NFS filesystem. NFS
(Network Filesystem) is a network filesystem commonly used on Unix / Linux
systems. httpd attempted to read one or more files or directories from a mounted
filesystem of this type. As NFS filesystems do not support fine-grained SELinux
labeling, all files and directories in the filesystem will have the same
security context. If you have not configured httpd to read files from a NFS
filesystem this access attempt could signal an intrusion attempt.

Allowing Access:
Changing the "use_nfs_home_dirs" boolean to true will allow this access:
"setsebool -P use_nfs_home_dirs=1"

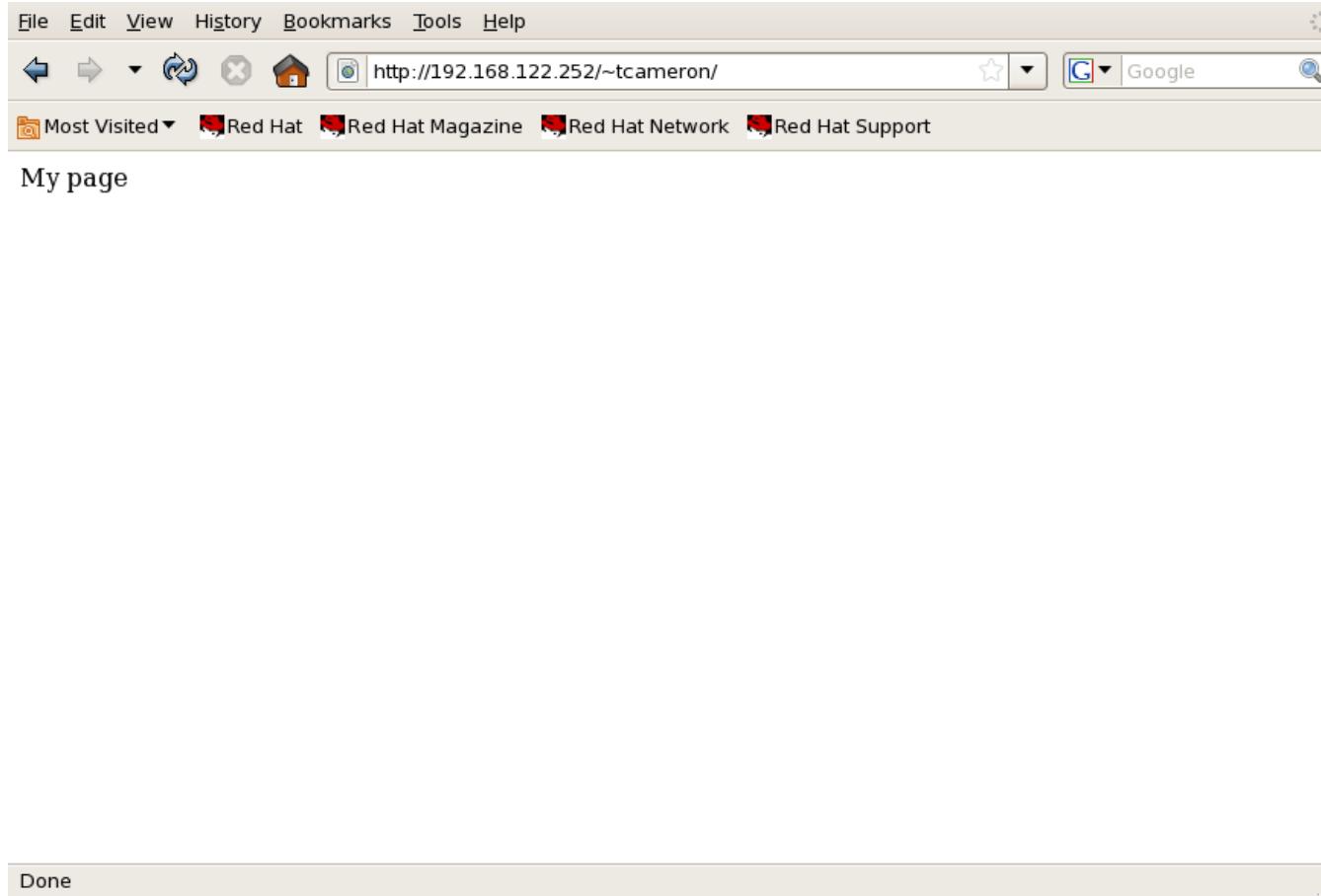
The following command will allow this access:
setsebool -P use_nfs_home_dirs=1

Additional Information:
```

# SELinux Examples

```
File Edit View Terminal Tabs Help
[root@host175 ~]# setsebool -P use_nfs_home_dirs=1
[root@host175 ~]#
```

# SELinux Examples



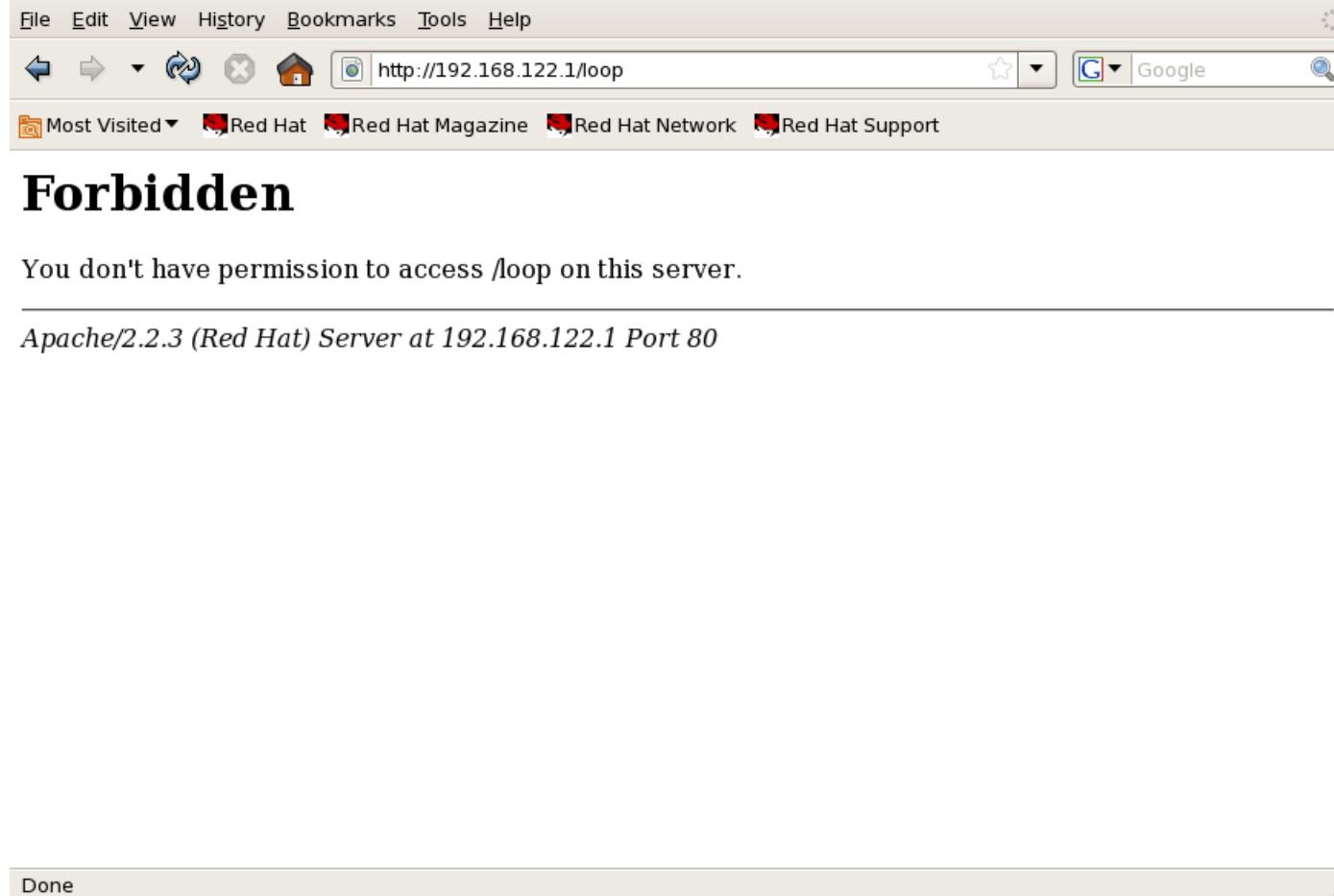
# SELinux Examples

- How about mounting an ISO image or drive so that Apache can export it?
- It's common to mount an ISO image under `/var/www/html` so that it can be used as an installation source

# SELinux Examples

```
File Edit View Terminal Tabs Help
[root@tct60 ~]# mount -o loop /media/500GB/isos/rhel5.2/x86_64/rhel-5-server-x86
_64-dvd.iso /var/www/html/loop/
[root@tct60 ~]# [ ]
```

# SELinux Examples



# SELinux Examples

```
File Edit View Terminal Tabs Help
Summary:
SELinux is preventing the httpd from using potentially mislabeled files
/var/www/html/loop (iso9660_t).

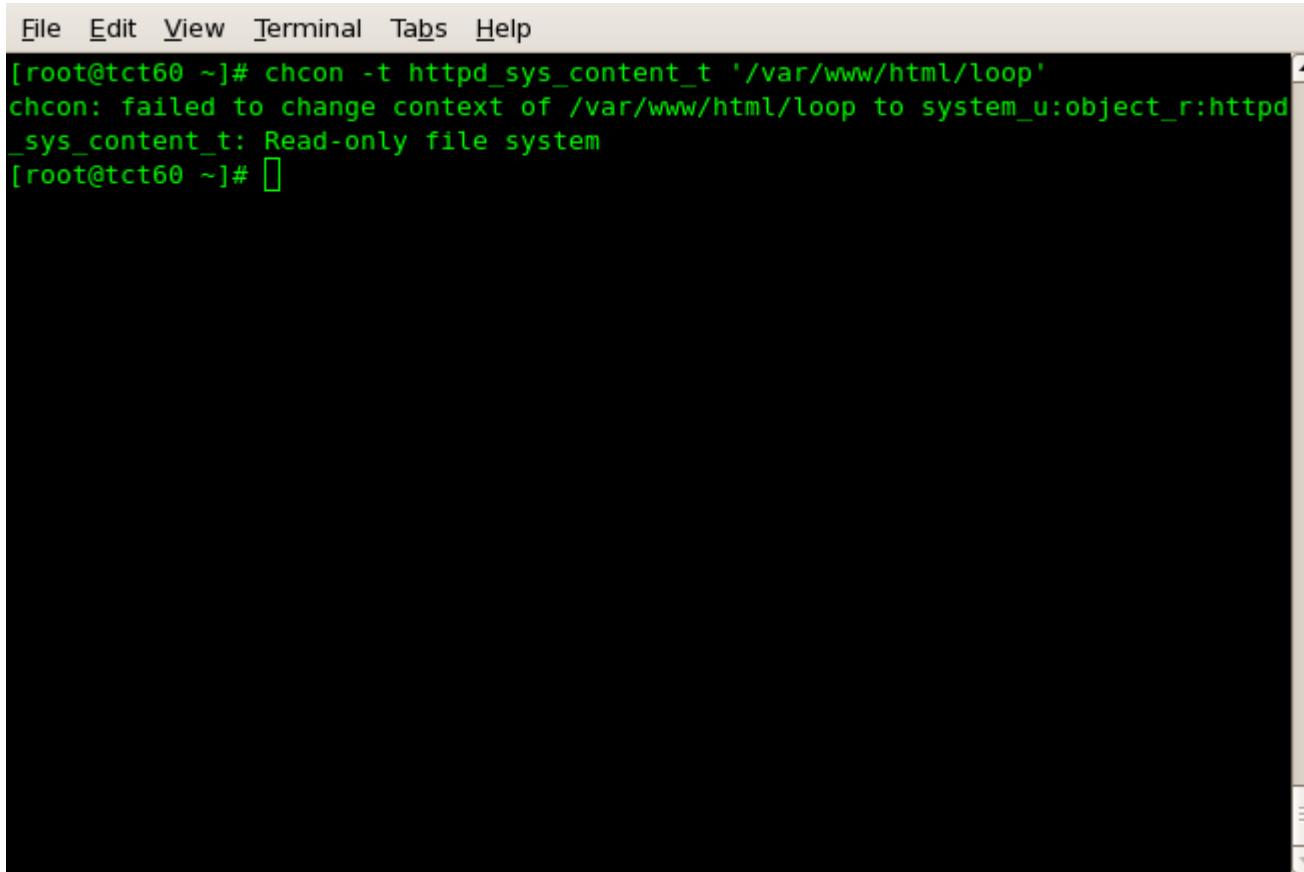
Detailed Description:
SELinux has denied the httpd access to potentially mislabeled files
/var/www/html/loop. This means that SELinux will not allow httpd to use these
files. Many third party apps install html files in directories that SELinux
policy cannot predict. These directories have to be labeled with a file context
which httpd can access.

Allowing Access:
If you want to change the file context of /var/www/html/loop so that the httpd
daemon can access it, you need to execute it using chcon -t httpd_sys_content_t
'/var/www/html/loop'. You can look at the httpd_selinux man page for additional
information.

Additional Information:
Source Context          system_u:system_r:httpd_t
Target Context          system_u:object_r:iso9660_t
```

# SELinux Examples

- Uh-oh!



A terminal window with a light gray header bar containing the menu items: File, Edit, View, Terminal, Tabs, Help. The main area is black and displays the following text in white:

```
[root@tct60 ~]# chcon -t httpd_sys_content_t '/var/www/html/loop'
chcon: failed to change context of /var/www/html/loop to system_u:object_r:httpd_sys_content_t: Read-only file system
[root@tct60 ~]#
```

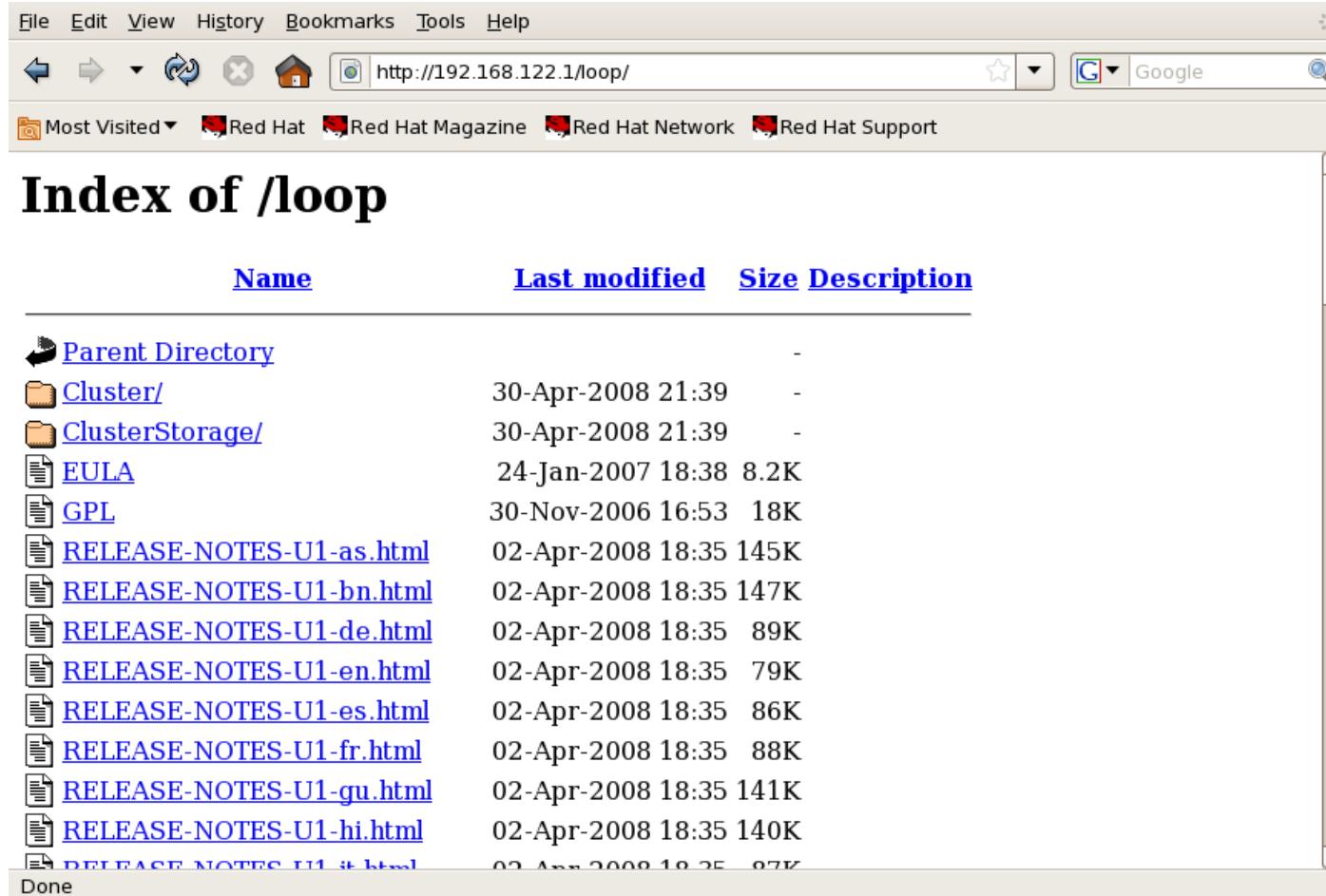
# SELinux Examples

- This will happen for things like USB drives and ISO images.
- The key is to tell mount under which context to mount the device

# SELinux Examples

```
File Edit View Terminal Tabs Help
[root@tct60 ~]# umount /var/www/html/loop/
[root@tct60 ~]# ls -dZ /var/www/html/
drwxr-xr-x  root root system_u:object_r:httpd_sys_content_t /var/www/html/
[root@tct60 ~]# mount -o loop,context=system_u:object_r:httpd_sys_content_t /media/500GB/isos/rhel5.2/x86_64/rhel-5-server-x86_64-dvd.iso /var/www/html/loop/
[root@tct60 ~]# 
```

# SELinux Examples



The screenshot shows a web browser window with the following details:

- Header:** File, Edit, View, History, Bookmarks, Tools, Help.
- Address Bar:** http://192.168.122.1/loop/
- Toolbar:** Back, Forward, Stop, Home, Refresh, Google search bar.
- Toolbar:** Most Visited, Red Hat, Red Hat Magazine, Red Hat Network, Red Hat Support.

The main content is a table titled "Index of /loop" with the following data:

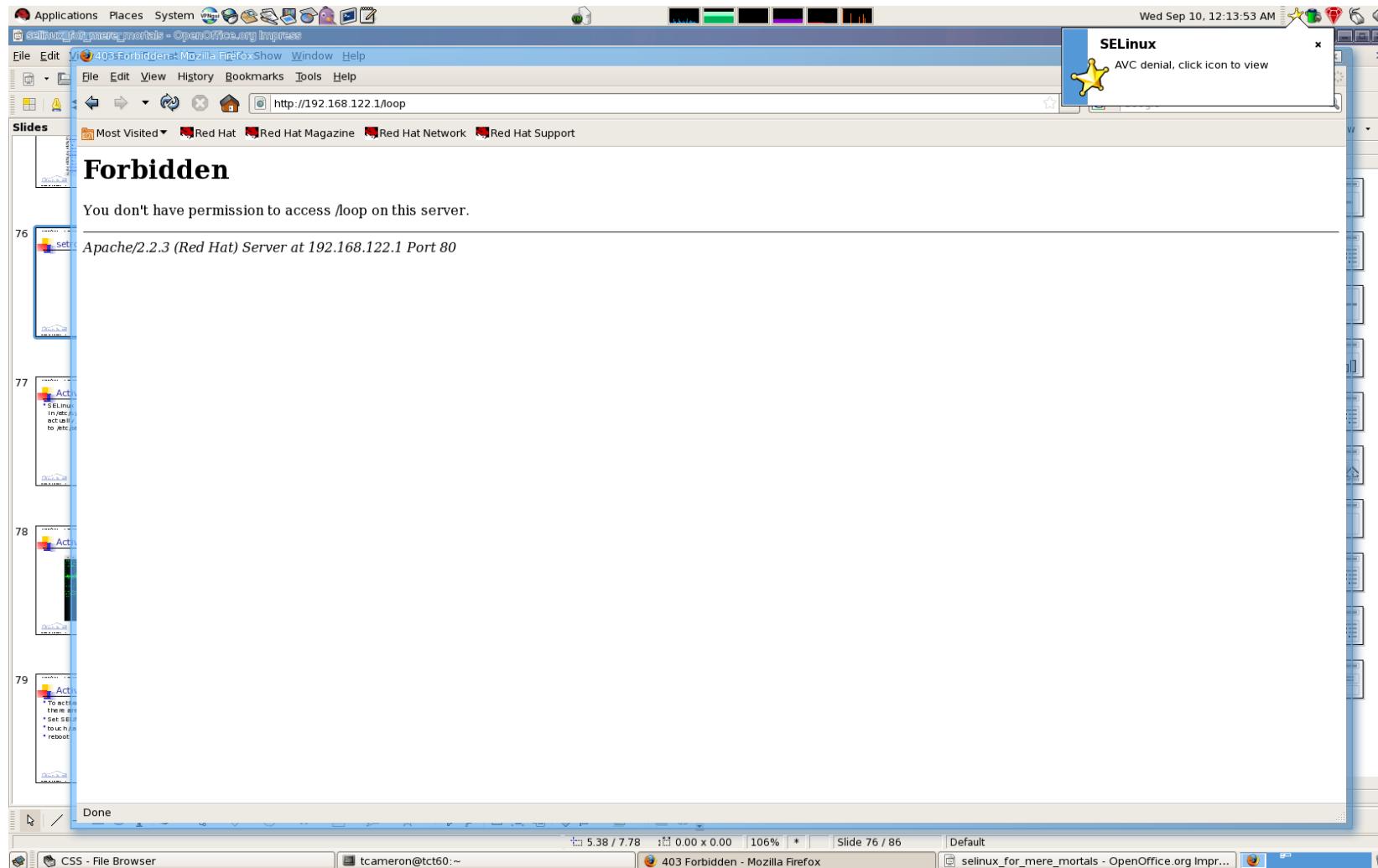
<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
<a href="#">Parent Directory</a>		-	
<a href="#">Cluster/</a>	30-Apr-2008 21:39	-	
<a href="#">ClusterStorage/</a>	30-Apr-2008 21:39	-	
<a href="#">EULA</a>	24-Jan-2007 18:38	8.2K	
<a href="#">GPL</a>	30-Nov-2006 16:53	18K	
<a href="#">RELEASE-NOTES-U1-as.html</a>	02-Apr-2008 18:35	145K	
<a href="#">RELEASE-NOTES-U1-bn.html</a>	02-Apr-2008 18:35	147K	
<a href="#">RELEASE-NOTES-U1-de.html</a>	02-Apr-2008 18:35	89K	
<a href="#">RELEASE-NOTES-U1-en.html</a>	02-Apr-2008 18:35	79K	
<a href="#">RELEASE-NOTES-U1-es.html</a>	02-Apr-2008 18:35	86K	
<a href="#">RELEASE-NOTES-U1-fr.html</a>	02-Apr-2008 18:35	88K	
<a href="#">RELEASE-NOTES-U1-gu.html</a>	02-Apr-2008 18:35	141K	
<a href="#">RELEASE-NOTES-U1-hi.html</a>	02-Apr-2008 18:35	140K	
<a href="#">RELEASE-NOTES-U1-ja.html</a>	02-Apr-2008 18:35	87K	

At the bottom of the table, there is a "Done" button.

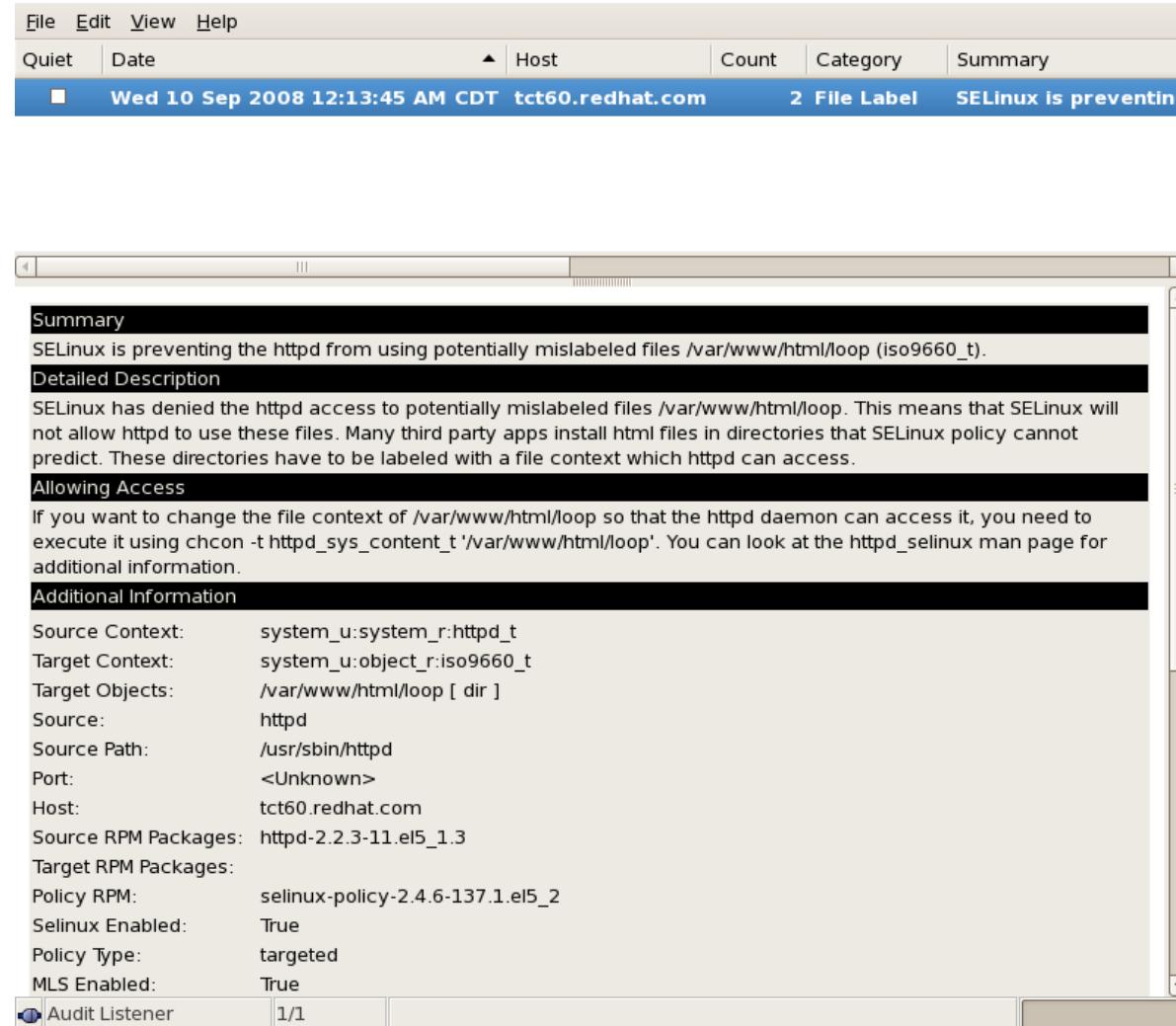
# setroubleshoot Browser

- Graphical tool which does the same thing as sealert.
  - Available from the desktop or via the command:
    - `sealert -b`

# setroubleshoot Browser



# setroubleshoot Browser



# Activating SELinux

- SELinux is enabled or disabled in `/etc/sysconfig/selinux` (which is actually just a link to `/etc/selinux/config`)

# Activating SELinux

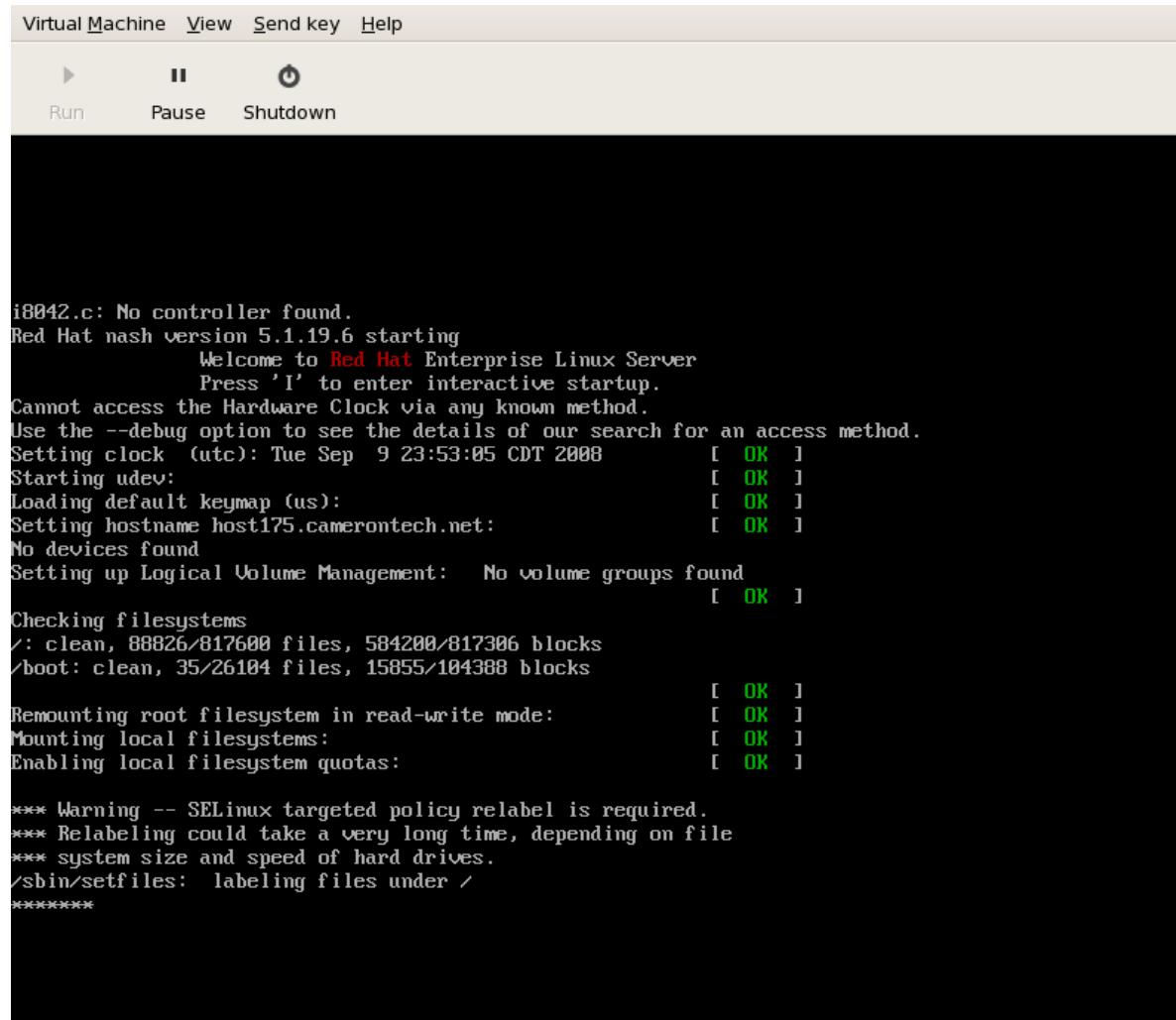
```
File Edit View Terminal Tabs Help
[root@tct60 ~]# cat /etc/sysconfig/selinux
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - SELinux is fully disabled.
SELINUX=enforcing
# SELINUXTYPE= type of policy in use. Possible values are:
#       targeted - Only targeted network daemons are protected.
#       strict - Full SELinux protection.
SELINUXTYPE=targeted

# SETLOCALDEFS= Check local definition changes
SETLOCALDEFS=0
[root@tct60 ~]#
```

# Activating SELinux

- To activate SELinux on your machine, there are a couple of ways to do it.
- Set SELINUX=enforcing
- touch /.autorelabel
- reboot

# Activating SELinux



Virtual Machine View Send key Help

Run Pause Shutdown

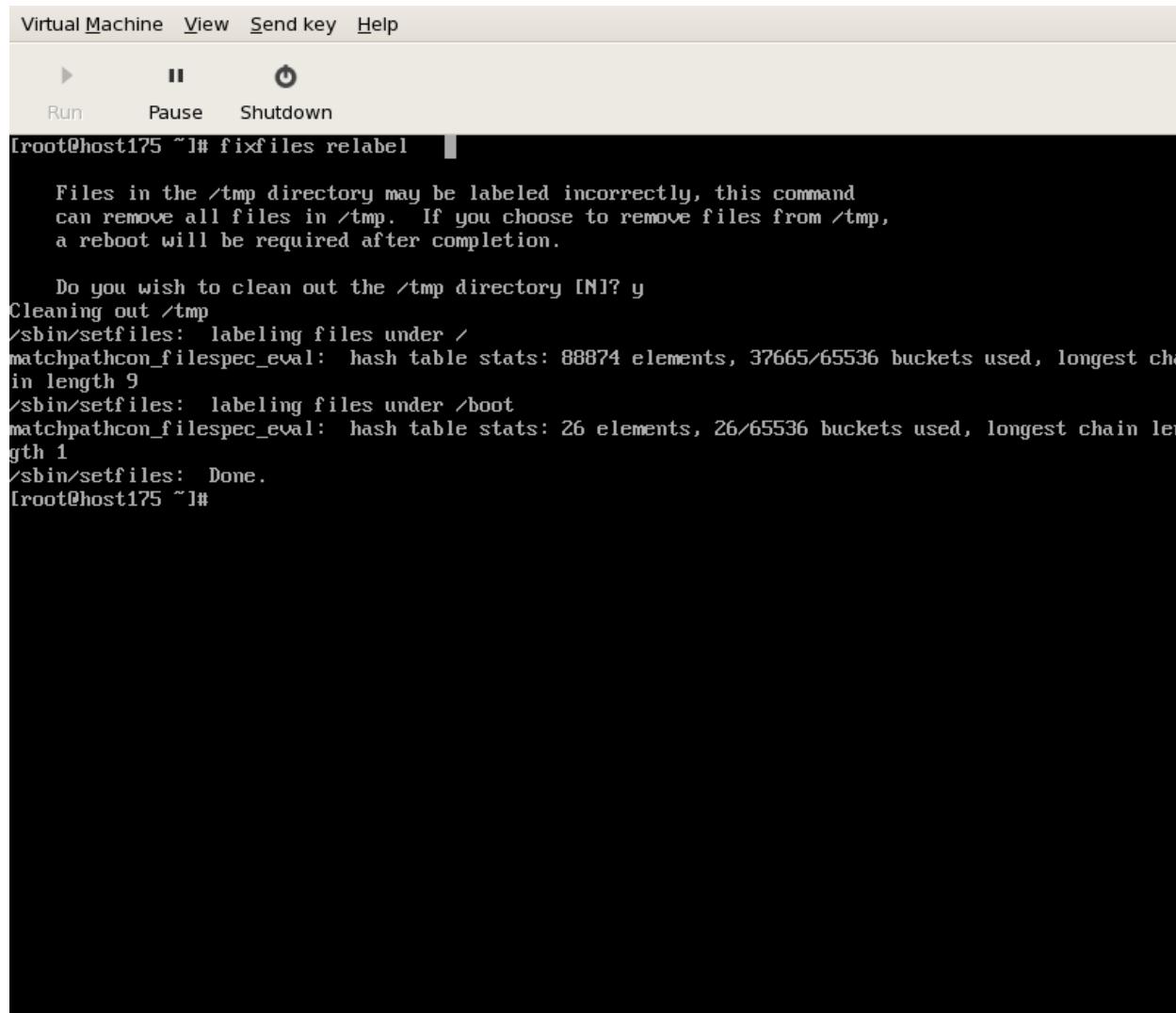
```
i8042.c: No controller found.
Red Hat nash version 5.1.19.6 starting
    Welcome to Red Hat Enterprise Linux Server
    Press 'I' to enter interactive startup.
Cannot access the Hardware Clock via any known method.
Use the --debug option to see the details of our search for an access method.
Setting clock (utc): Tue Sep  9 23:53:05 CDT 2008      [ OK ]
Starting udev:                                         [ OK ]
Loading default keymap (us):                         [ OK ]
Setting hostname host175.camerontech.net:            [ OK ]
No devices found
Setting up Logical Volume Management:  No volume groups found
                                            [ OK ]
Checking filesystems
/: clean, 88826/817600 files, 584200/817306 blocks
/boot: clean, 35/26104 files, 15855/104388 blocks
                                            [ OK ]
Remounting root filesystem in read-write mode:        [ OK ]
Mounting local filesystems:                          [ OK ]
Enabling local filesystem quotas:                   [ OK ]

*** Warning -- SELinux targeted policy relabel is required.
*** Relabeling could take a very long time, depending on file
*** system size and speed of hard drives.
/sbin/setfiles: labeling files under /
*****
```

# Activating SELinux

- Alternatively, you can issue the command "fixfiles relabel" as root
  - Reboot after it's done
  - Don't do it in runlevel 5 since it deletes everything in /tmp including files the X server needs

# Activating SELinux



Virtual Machine View Send key Help

Run Pause Shutdown

```
[root@host175 ~]# fixfiles relabel
Files in the /tmp directory may be labeled incorrectly, this command
can remove all files in /tmp. If you choose to remove files from /tmp,
a reboot will be required after completion.

Do you wish to clean out the /tmp directory [N]? y
Cleaning out /tmp
/sbin/setfiles: labeling files under /
matchpathcon_filespec_eval: hash table stats: 88874 elements, 37665/65536 buckets used, longest chain length 9
/sbin/setfiles: labeling files under /boot
matchpathcon_filespec_eval: hash table stats: 26 elements, 26/65536 buckets used, longest chain length 1
/sbin/setfiles: Done.
[root@host175 ~]#
```

# Activating SELinux

- You can also run SELinux in permissive mode, where it will not block anything but it will still log AVC errors.
- Do this in development environment and set policy or booleans as needed on production machines.

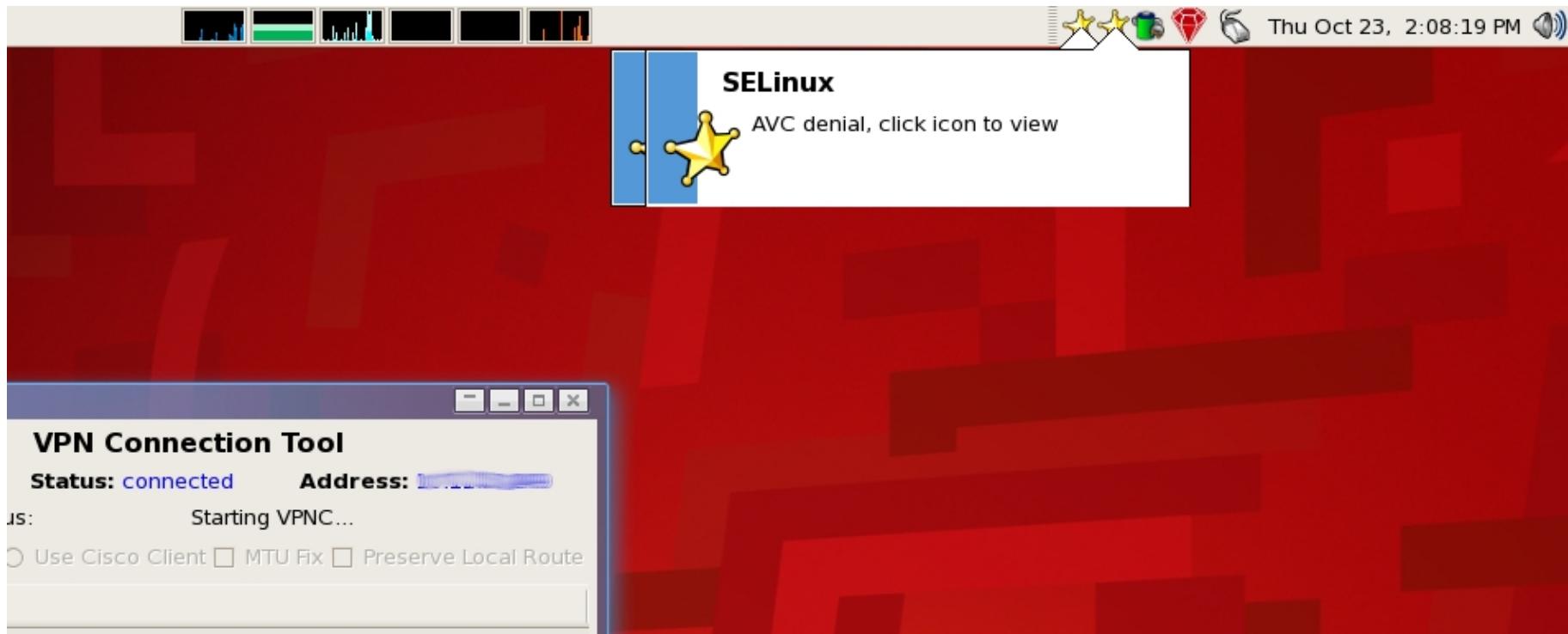
# Creating Basic Policies

- audit2why and audit2allow are two utilities to tell you why something was denied and how to allow it
- Note that just because audit2allow will create a policy, that does not mean it is the smartest thing to do! Consider security implications before applying policies.

# Creating Basic Policies

- In this example, the VPN client on my laptop causes an AVC denial

# Creating Basic Policies



# Creating Basic Policies

- Use audit2why to see why the alert was triggered

# Creating Basic Policies

```
File Edit View Terminal Tabs Help
[root@tct60 ~]# audit2why < /var/log/audit/audit.log
type=AVC msg=audit(1224792254.795:118): avc:  denied  { read } for  pid=12954 co
mm="sh" name="default" dev=sda2 ino=2222408 scontext=user_u:system_r:vpnc_t:s0 t
context=root:object_r:usr_t:s0 tclass=lnk_file
    Was caused by:
        Missing or disabled TE allow rule.
        Allow rules may exist but be disabled by boolean settings; check
boolean settings.
        You can see the necessary allow rules by running audit2allow wit
h this audit message as input.

type=AVC msg=audit(1224792254.807:119): avc:  denied  { read } for  pid=12955 co
mm="vpnc-script" name="default" dev=sda2 ino=2222408 scontext=user_u:system_r:vp
nc_t:s0 tcontext=root:object_r:usr_t:s0 tclass=lnk_file
    Was caused by:
        Missing or disabled TE allow rule.
        Allow rules may exist but be disabled by boolean settings; check
boolean settings.
        You can see the necessary allow rules by running audit2allow wit
h this audit message as input.

type=AVC msg=audit(1224792254.811:120): avc:  denied  { read } for  pid=12956 co
mm="vpnc-script" name="default" dev=sda2 ino=2222408 scontext=user_u:system_r:vp
nc_t:s0 tcontext=root:object_r:usr_t:s0 tclass=lnk_file
```

# Creating Basic Policies

- Use audit2allow see what needs to be changed

# Creating Basic Policies

```
File Edit View Terminal Tabs Help
[root@tct60 css]# audit2allow -i /var/log/audit/audit.log

===== ifconfig_t =====
allow ifconfig_t vpnc_t:udp_socket { read write };

===== vpnc_t =====
allow vpnc_t usr_t:lnk_file read;
[root@tct60 css]#
```

# Creating Basic Policies

- Use audit2allow -M [policynname] to create a .te and a .pp file

# Creating Basic Policies

```
File Edit View Terminal Tabs Help
[root@tct60 css]# audit2allow -i /var/log/audit/audit.log

#===== ifconfig_t =====
allow ifconfig_t vpnc_t:udp_socket { read write };

#===== vpnc_t =====
allow vpnc_t usr_t:lnk_file read;
[root@tct60 css]# audit2allow -i /var/log/audit/audit.log -M vpnstuff
***** IMPORTANT *****
To make this policy package active, execute:

semodule -i vpnstuff.pp

[root@tct60 css]# [
```

# Creating Basic Policies

- As indicated, run the command `semodule -i [policyname.pp]` to install the policy module

# Creating Basic Policies

```
File Edit View Terminal Tabs Help
[root@tct60 css]# audit2allow -i /var/log/audit/audit.log

#===== ifconfig_t =====
allow ifconfig_t vpnc_t:udp_socket { read write };

#===== vpnc_t =====
allow vpnc_t usr_t:lnk_file read;
[root@tct60 css]# audit2allow -i /var/log/audit/audit.log -M vpnstuff
***** IMPORTANT *****
To make this policy package active, execute:

semodule -i vpnstuff.pp

[root@tct60 css]# semodule -i vpnstuff.pp
[root@tct60 css]#
```

# Creating Basic Policies

```
File Edit View Terminal Tabs Help
[root@tct60 css]# cat vpnstuff.te

module vpnstuff 1.0;

require {
    type vpnc_t;
    type ifconfig_t;
    type usr_t;
    class lnk_file read;
    class udp_socket { read write };
}

===== ifconfig_t =====
allow ifconfig_t vpnc_t:udp_socket { read write };

===== vpnc_t =====
allow vpnc_t usr_t:lnk_file read;
[root@tct60 css]#
```

# Final Thoughts

- Don't turn it off!
- SELinux can really save you in the event of a breach.
- It's *\*much\** easier to use SELinux today than it was just a few months ago
- NSA grade security is available at no extra cost - use it!

**QUESTIONS?**

**TELL US WHAT YOU THINK:  
[REDHAT.COM/SUMMIT-SURVEY](http://REDHAT.COM/SUMMIT-SURVEY)**