SUMMIT

JBoss WORLD

PRESENTED BY RED HAT

LEARN. NETWORK. EXPERIENCE OPEN SOURCE.

www.theredhatsummit.com

FOLLOW US ON TWITTER

www.twitter.com/redhatsummit

TWEET ABOUT IT

#summitjbw

READ THE BLOG

http://summitblog.redhat.com/

SELinux For Mere Mortals

(Or, “Don't Turn It Off”)

Dan Walsh
Principal Software Engineer, Red Hat

Thomas Cameron, RHCA
Managing Solutions Architect, Red Hat

June 23rd, 2010

Agenda

About Us

What is SELinux?

What can I do with it?

SELinux Architecture

Real World Examples

About Us

Red Hat leads the way in SELinux development. John Dennis, Ulrich Drepper, Steve Grubb, Eric Paris, Roland McGrath, James Morris and Dan Walsh, all Red Hat staffers, 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.

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 trying to tell me?

The four key causes of SELinux Messages and how to deal with them

- First we will take a simplified view of

What is SELinux?

- When SELinux complains how can I deal with it, in a secure way

SELinux == LABELING

- Keep it simple stupid...
  - Process has labels
    - system_u:system_r:httpd_t:s0
  - Files/Directories have labels.
    - system_u:object_r:httpd_sys_content_t:s0
  - Kernel has rules controlling how labels interact.
    - allow httpd_t httpd_sys_content_t : file { ioctl read getattr lock open } ;
  - Simple?

DAC vs MAC

- Discretionary Access Control - Labeling
  - Label is file ownership/Group+ Permission Field
  - Processes has Ownership.
  - Hard coded policy.
  - Process Owner has discretion over files he owns.
- Mandatory Access Control
  - Flexible policy
  - Kernel governs all access
- Both required permissions in SELinux system

SELinux Label

- User Component
    - dwalsh:staff_r:passwd_t:s0
    - Not necessarily the same as the Linux user
    - Often ends in “_u”: system_u, user_u
    - Not currently used in the targeted policy
    - Files and directories:
        - user inherited from process
        - system process -> files created with system_u

SELinux Label

- Role Component
  - dwalsh_u:staff_r:passwd_t:s0
  - Used for RBAC
    - role further restricts available type transitions
    - in cooperation with TE (e.g., user_r / user_t)
  - Usually ends with “_r”
  - Resources have default “object_r” role
  - Used in strict and MLS policies
    - user_r, staff_r, secadm_r

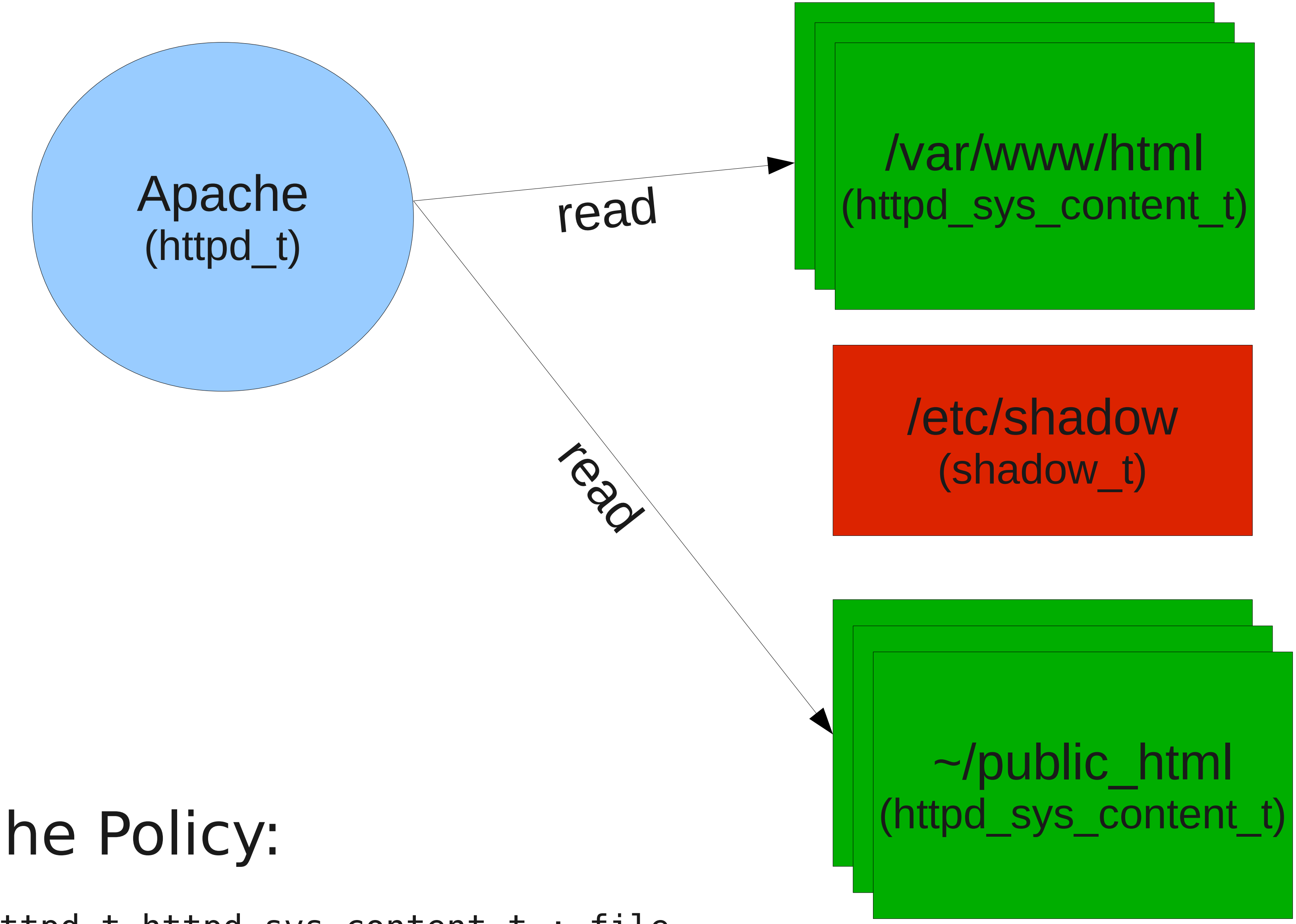
SELinux Label

- MLS/MCS Component
  - dwalsh_u:staff_r:passwd_t:s0-s15:c0.c1023
  - Identifies one level or range
    - single level: s0
    - range: so-s15:c0.c1023
  - Usually translated
    - s15:c0.c1023 -> “SystemHigh”
  - Mainly used for separation in “targeted” policy
    - Svirt, sandbox

SELinux Label

- Type Component
  - dwalsh_u:staff_r:passwd_t:s0
  - Most important field
  - SELinux is a type enforcement system.
  - RBAC and MLS are built on top of type enforcement.

Type Enforcement Overview



Apache Policy:

```
allow httpd_t httpd_sys_content_t : file
read;
```

SELinux == LABELING

- How do I see the labels?
  - -Z is your friend.

```
ls -Z /etc/resolv.conf
-rw-r--r--. root root system_u:object_r:net_conf_t:s0  /etc/resolv.conf

id -Z
staff_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

ps -eZ | grep httpd
staff_u:system_r:httpd_t:s0     13167 ?        00:00:00 httpd

find /etc/ -name shadow -printf "%p %Z\n"
/etc/chroot/oldrhelapp/etc/shadow system_u:object_r:etc_t:s0
/etc/shadow system_u:object_r:shadow_t:s0
```

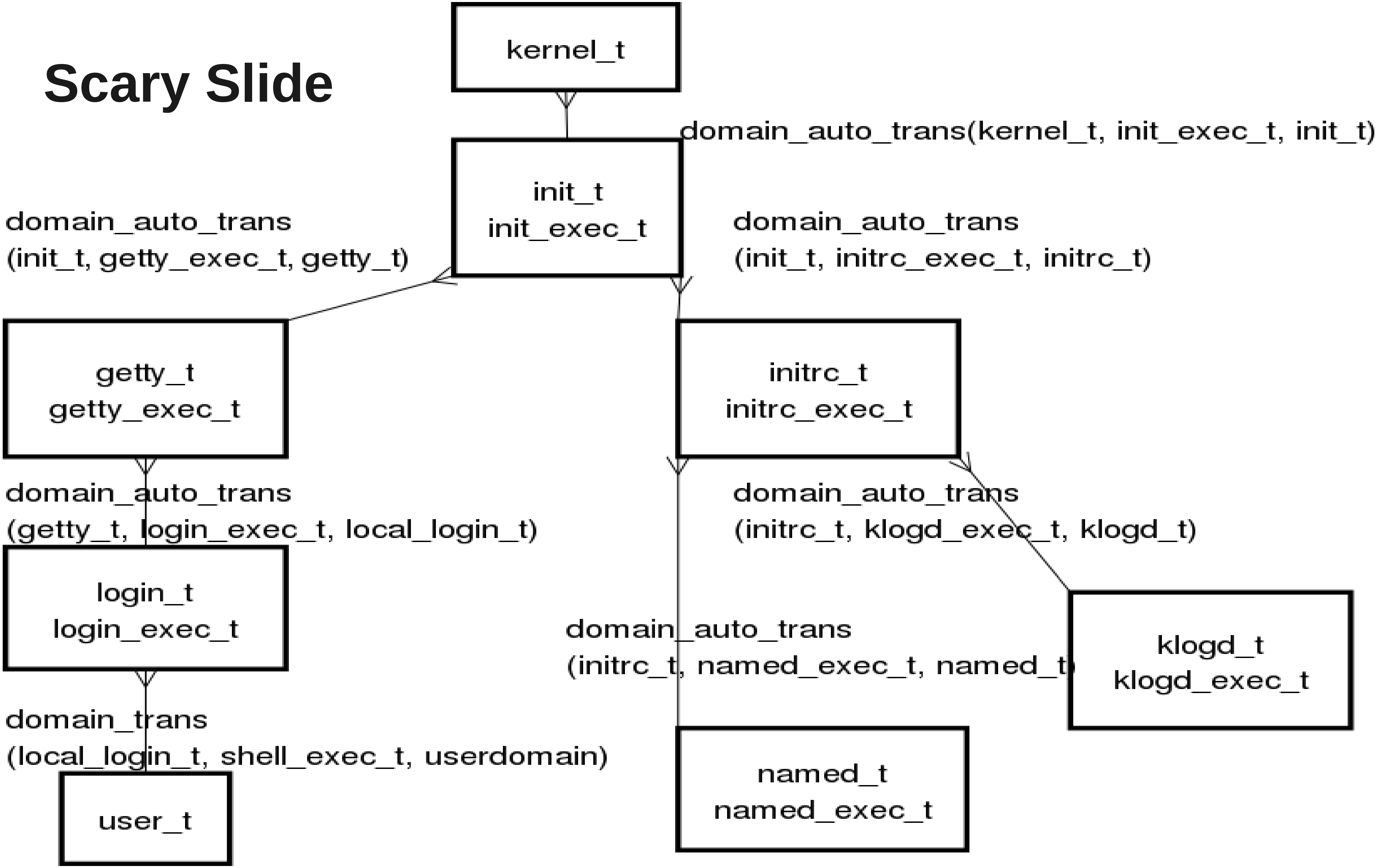
How do the labels get there?

- SELinux aware Applications
  - RPM
  - restorecon/chcon/semanage fcontext
    - /etc/selinux/targeted/contexts/files/file_contexts
- Users creating files
  - Default to parent directory
  - cp vs mv
- Login Programs
  - Sets the default process login label. Usually unconfined_t

Transitions

- File Transitions
  - Process A_t creates a FILE in directory B_t labeled C_t.
  - dhclient_t creates resolv.conf in directory etc_t labeled net_conf_t
- Process Transitions
  - Process A_t execute file B_exec_t, transitions new process B_t.
  - user_t executes passwd_exec_t transitions to passwd_t

Scary Slide


kernel_t
domain_auto_trans(kernel_t, init_exec_t, init_t)
init_t
init_exec_t
domain_auto_trans
(init_t, getty_exec_t, getty_t)
domain_auto_trans
(init_t, initrc_exec_t, initrc_t)
getty_t
getty_exec_t
initrc_t
initrc_exec_t
domain_auto_trans
(getty_t, login_exec_t, local_login_t)
domain_auto_trans
(initrc_t, klogd_exec_t, klogd_t)
login_t
login_exec_t
domain_auto_trans
(initrc_t, named_exec_t, named_t)
klogd_t
klogd_exec_t
domain_trans
(local_login_t, shell_exec_t, userdomain)
user_t
named_t
named_exec_t

#1 Cause of SELinux Messages
Something is wrong with the labeling.

- SELinux needs to know...
  - SELinux doesn't like admins changing defaults.
  - Changing default file locations means you have to set the labels, and tell SELinux about it.
- Permission denied means check the file ownership, permissions field AND **SELinux label**.

#2 Cause of SELinux Messages
SELinux Needs to know

- Least Privs versus Reasonable Privs
  - Policy writer decides default way most confined applications run.
  - If you run a confined application in a different way, you need to tell SELinux
  - Booleans
  - semanage
    - fcontext, ports

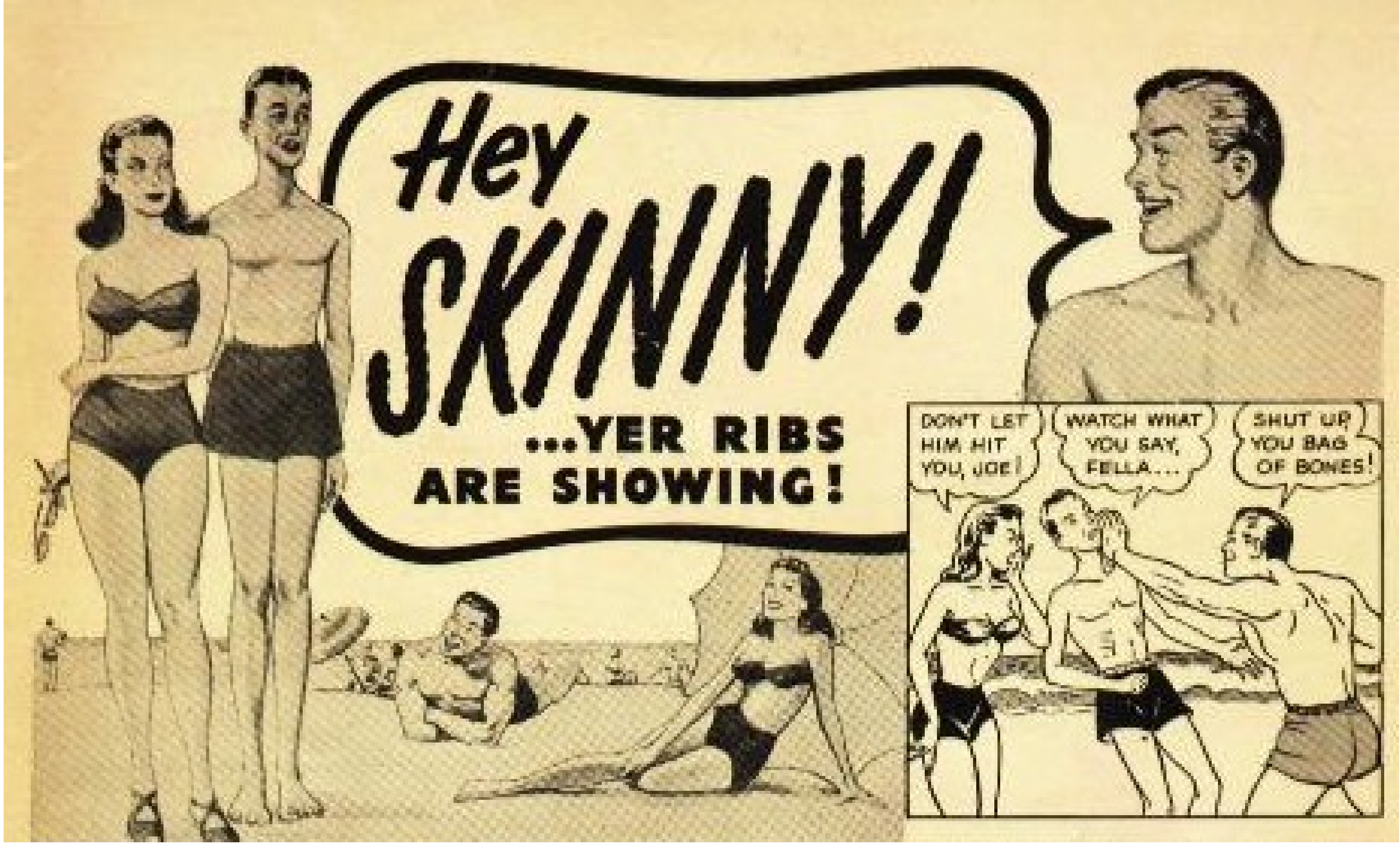
#3 Cause of SELinux Messages
SELinux/Apps still have bugs

- SELinux Policy might have a bug
  - Unusual Code Paths
  - Configurations
  - Redirection of stdout
- Apps have bugs
  - Leaked File Descriptors
  - Executable Memory
  - Badly built libraries
- Report the bugs so we can fix them!!!

#4 Cause of SELinux Messages
You have been hacked

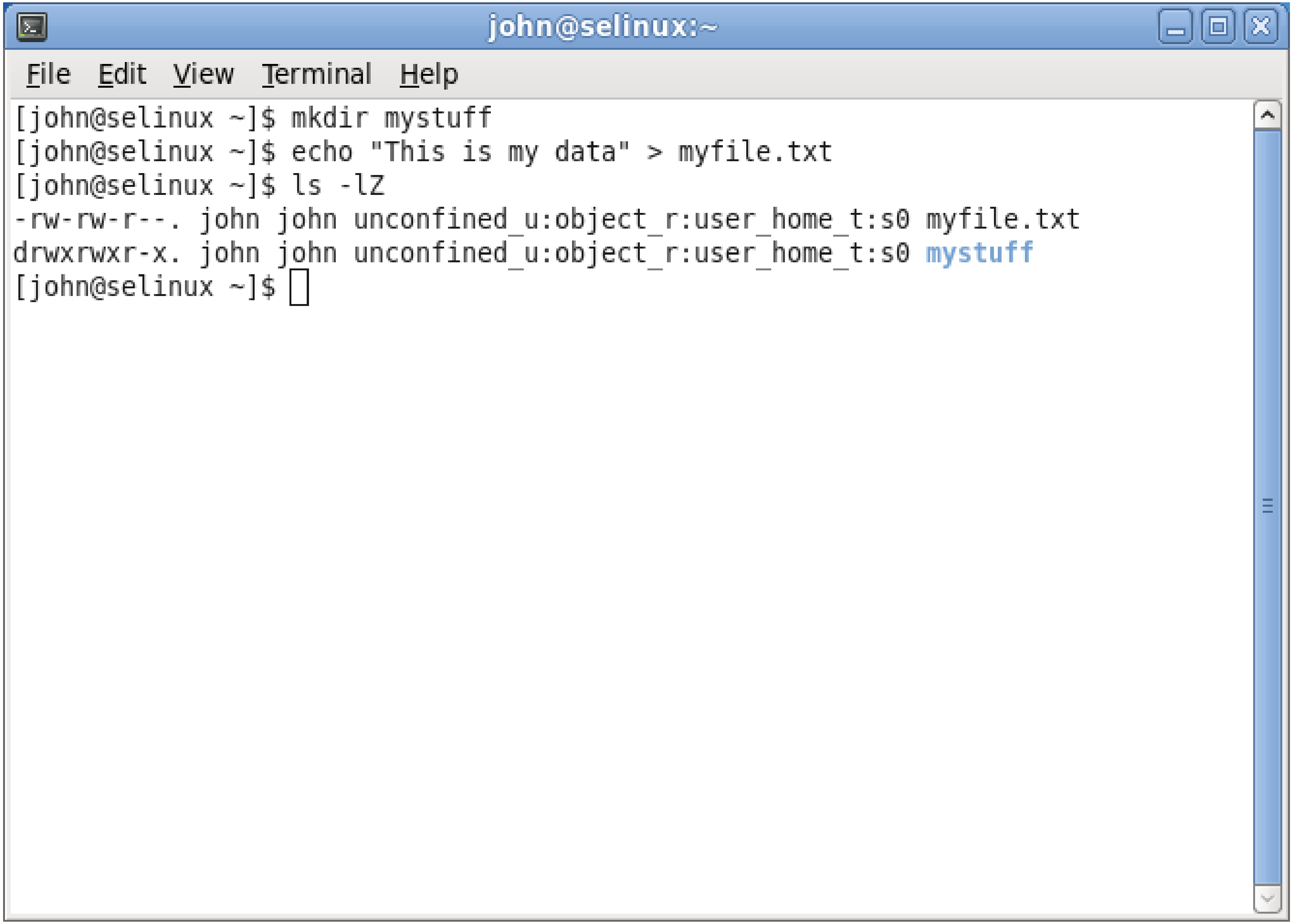
- Current tools do not do a good job of differentiating
  - If you have a confined domain that tries to:
    - Load a kernel module
    - Turn off SELinux enforcing mode
    - Write to etc_t? shadow_t
    - Modify iptables rules
    - Sendmail????
    - others
  - You might be compromised

What Thomas Thought SELinux Was


Hey SKINNY!
...YER RIBS ARE SHOWING!
DON'T LET HIM HIT YOU, JOE!
WATCH WHAT YOU SAY, FELLA....
SHUT UP YOU BAG OF BONES!

SELinux Examples

Creating a file and noting the context:

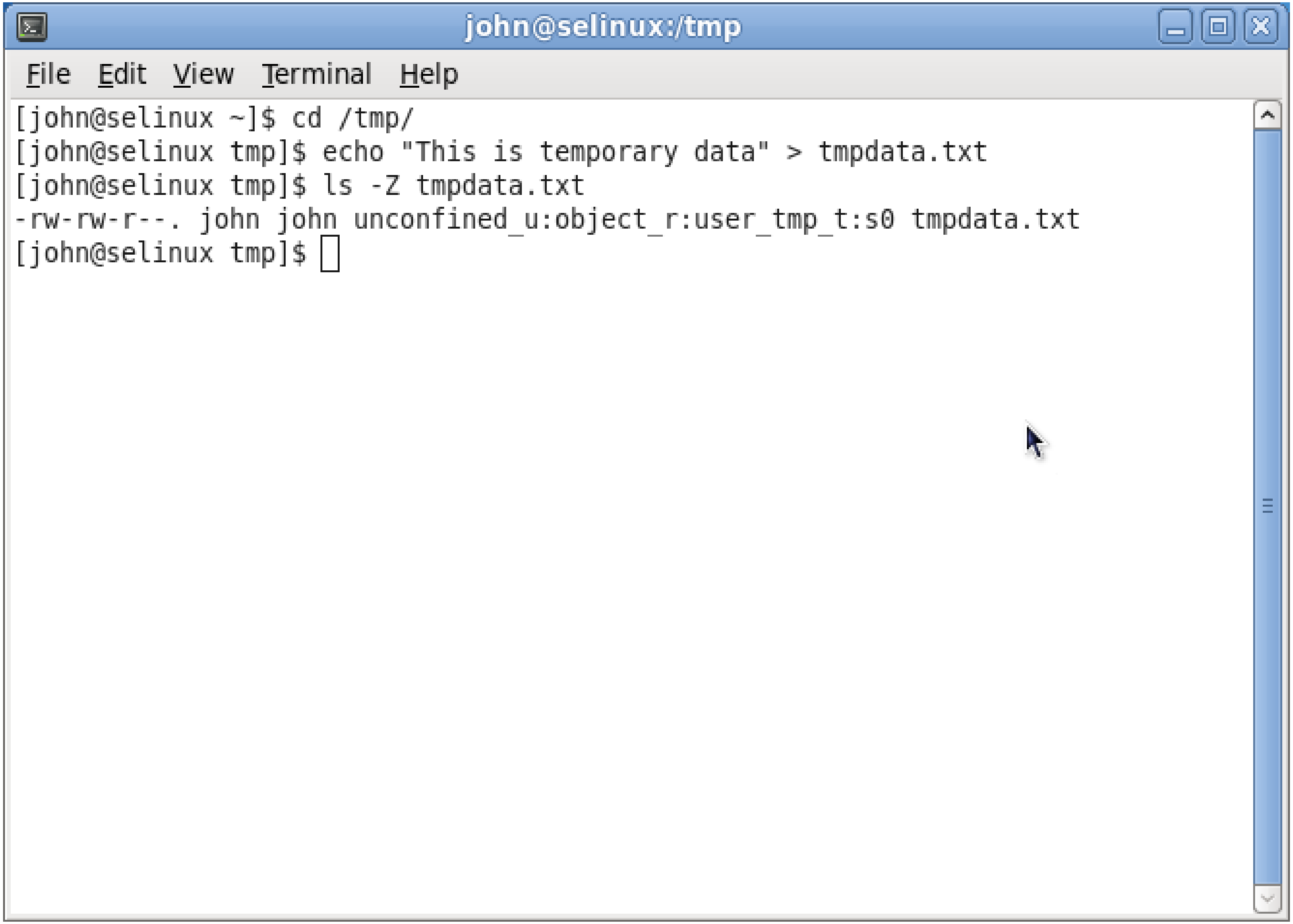


```
[john@selinux ~]$ mkdir mystuff
[john@selinux ~]$ echo "This is my data" > myfile.txt
[john@selinux ~]$ ls -lZ
-rw-rw-r--. john john unconfined_u:object_r:user_home_t:s0 myfile.txt
drwxrwxr-x. john john unconfined_u:object_r:user_home_t:s0 mystuff
[john@selinux ~]$ 
```

SELinux Examples

Changing the context of a file:

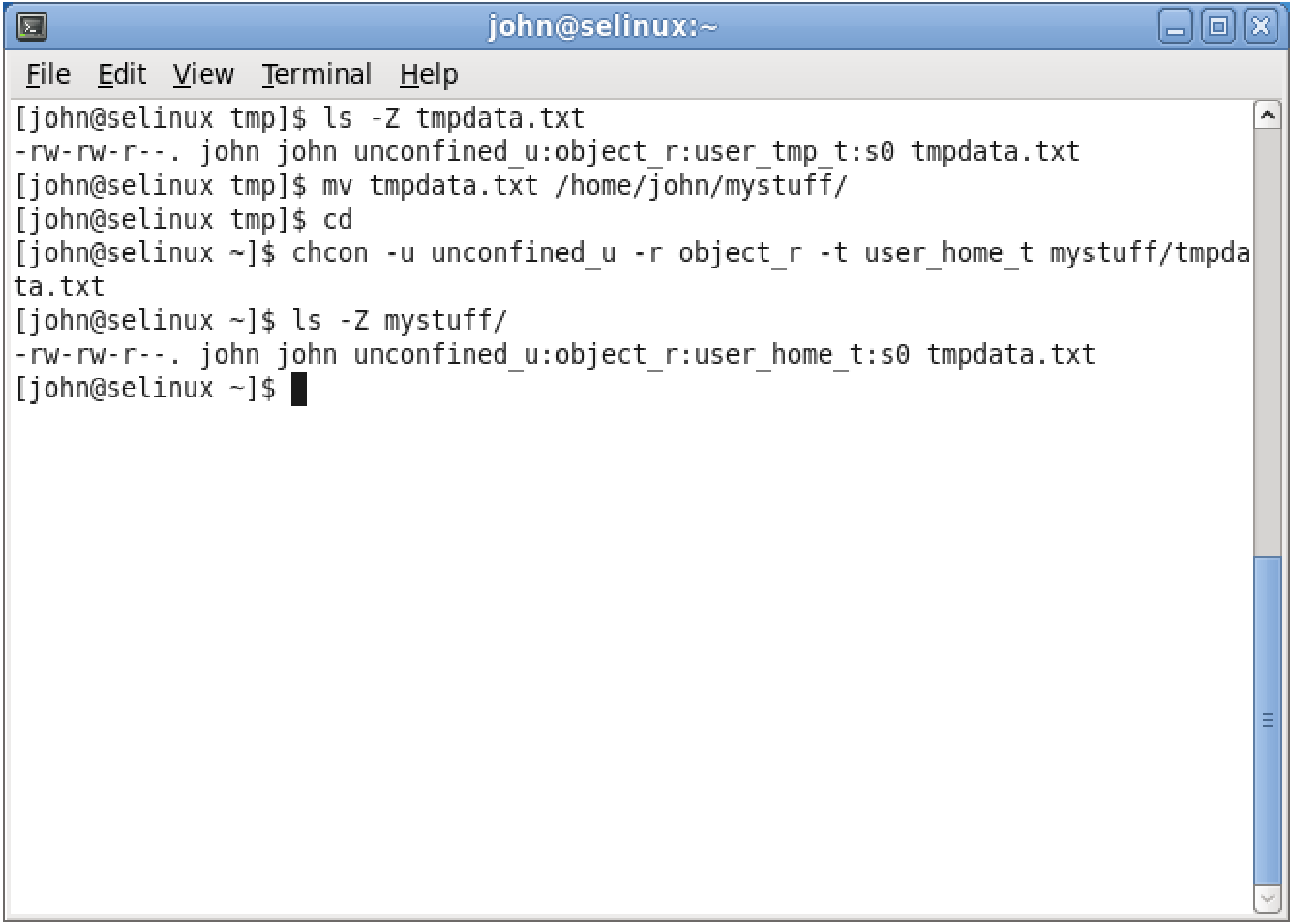
- First create data somewhere other than home
  - In this case, /tmp
- Note that the type is user_tmp_t, not user_home_t



```
[john@selinux ~]$ cd /tmp/
[john@selinux tmp]$ echo "This is temporary data" > tmpdata.txt
[john@selinux tmp]$ ls -Z tmpdata.txt
-rw-rw-r--. john john unconfined_u:object_r:user_tmp_t:s0 tmpdata.txt
[john@selinux tmp]$ 
```

SELinux Examples

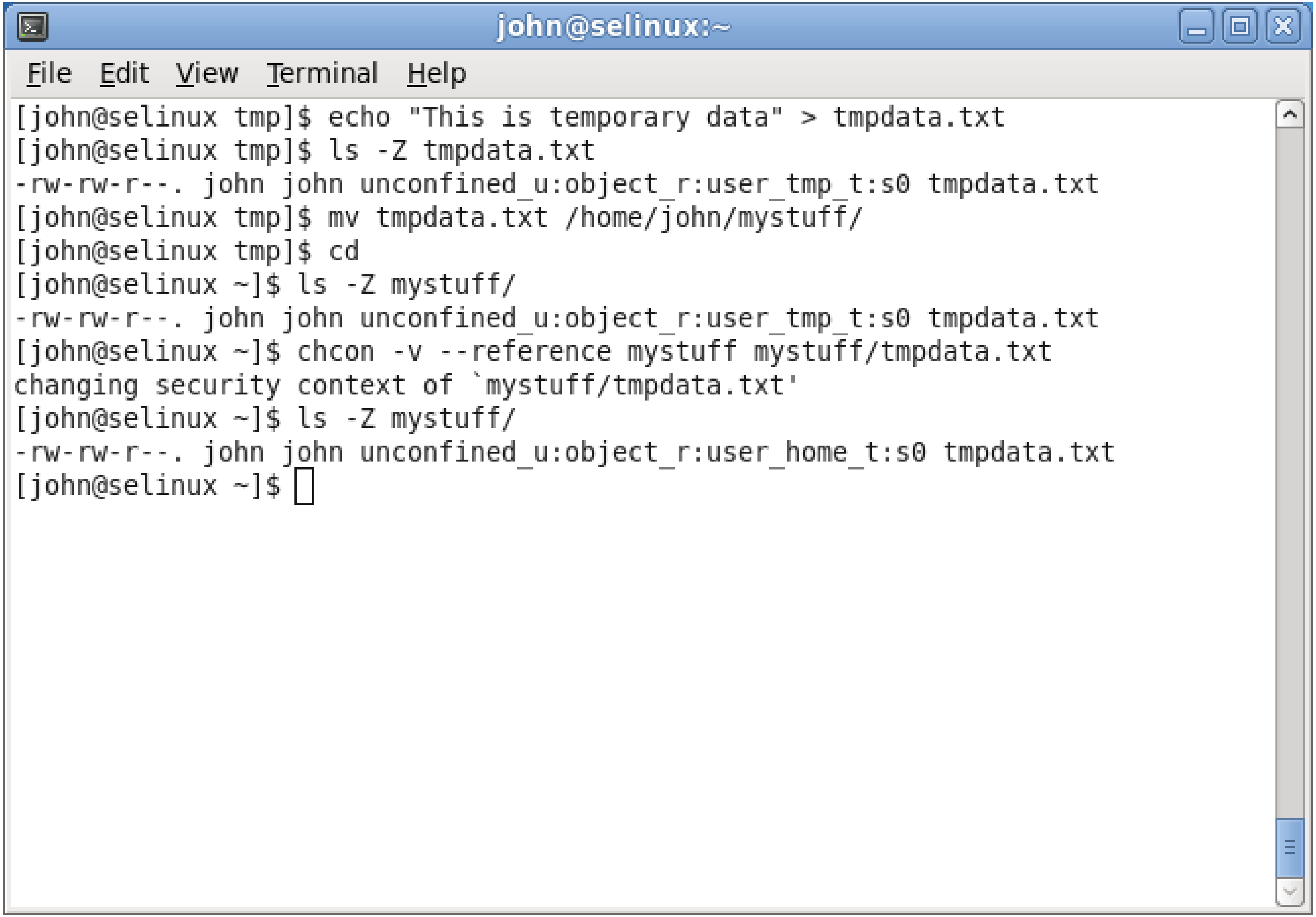
Hardest way: change the context manually using chcon, based on other files in /home/john:



```
[john@selinux tmp]$ ls -Z tmpdata.txt
-rw-rw-r--. john john unconfined_u:object_r:user_tmp_t:s0 tmpdata.txt
[john@selinux tmp]$ mv tmpdata.txt /home/john/mystuff/
[john@selinux tmp]$ cd
[john@selinux ~]$ chcon -u unconfined_u -r object_r -t user_home_t mystuff/tmpda
ta.txt
[john@selinux ~]$ ls -Z mystuff/
-rw-rw-r--. john john unconfined_u:object_r:user_home_t:s0 tmpdata.txt
[john@selinux ~]$ 
```

SELinux Examples

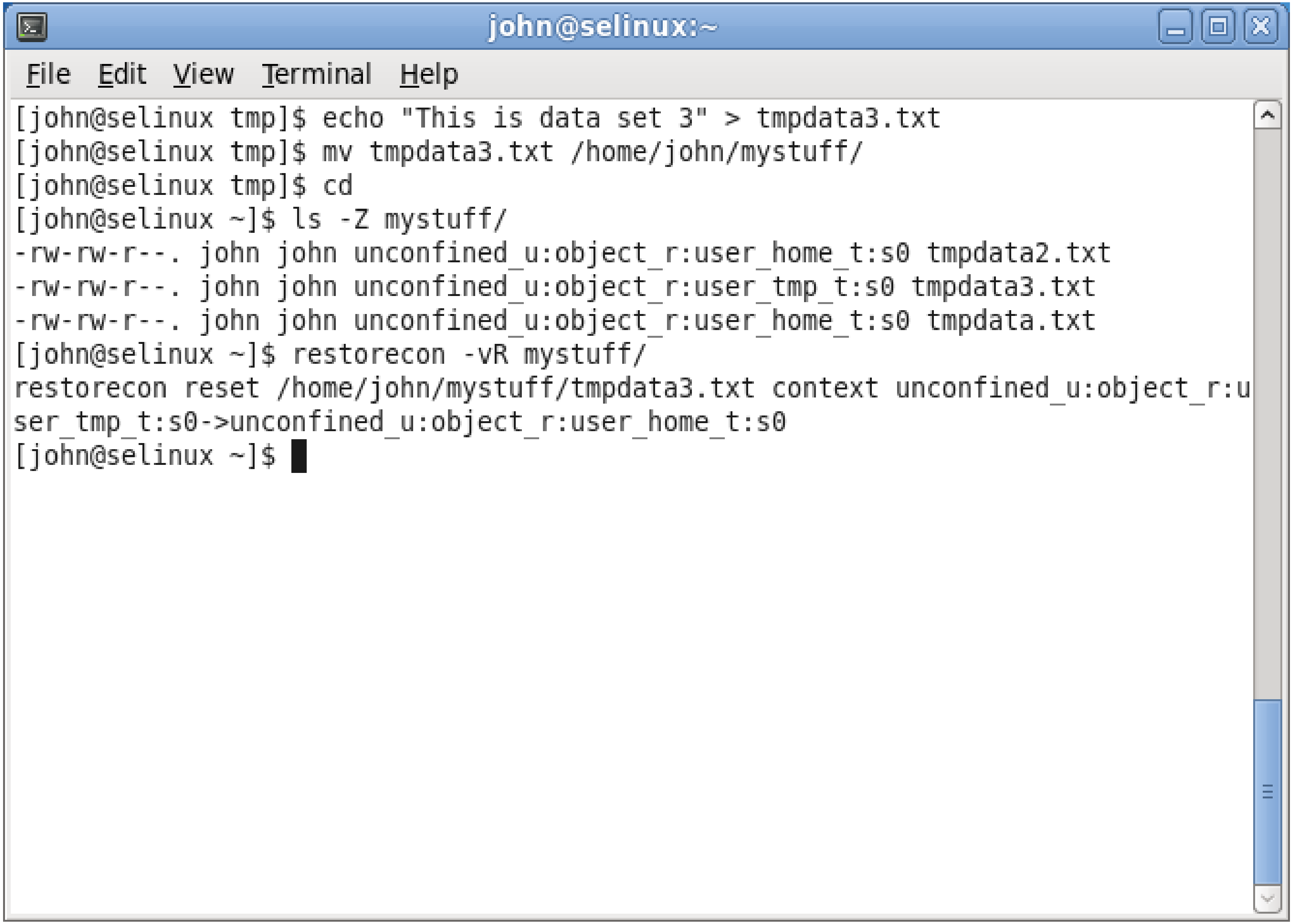
Easier way: chcon --reference



```
[john@selinux tmp]$ echo "This is temporary data" > tmpdata.txt
[john@selinux tmp]$ ls -Z tmpdata.txt
-rw-rw-r--. john john unconfined_u:object_r:user_tmp_t:s0 tmpdata.txt
[john@selinux tmp]$ mv tmpdata.txt /home/john/mystuff/
[john@selinux tmp]$ cd
[john@selinux ~]$ ls -Z mystuff/
-rw-rw-r--. john john unconfined_u:object_r:user_tmp_t:s0 tmpdata.txt
[john@selinux ~]$ chcon -v --reference mystuff mystuff/tmpdata.txt
changing security context of `mystuff/tmpdata.txt'
[john@selinux ~]$ ls -Z mystuff/
-rw-rw-r--. john john unconfined_u:object_r:user_home_t:s0 tmpdata.txt
[john@selinux ~]$ 
```

SELinux Examples

Easiest way: change the context with restorecon -vR:

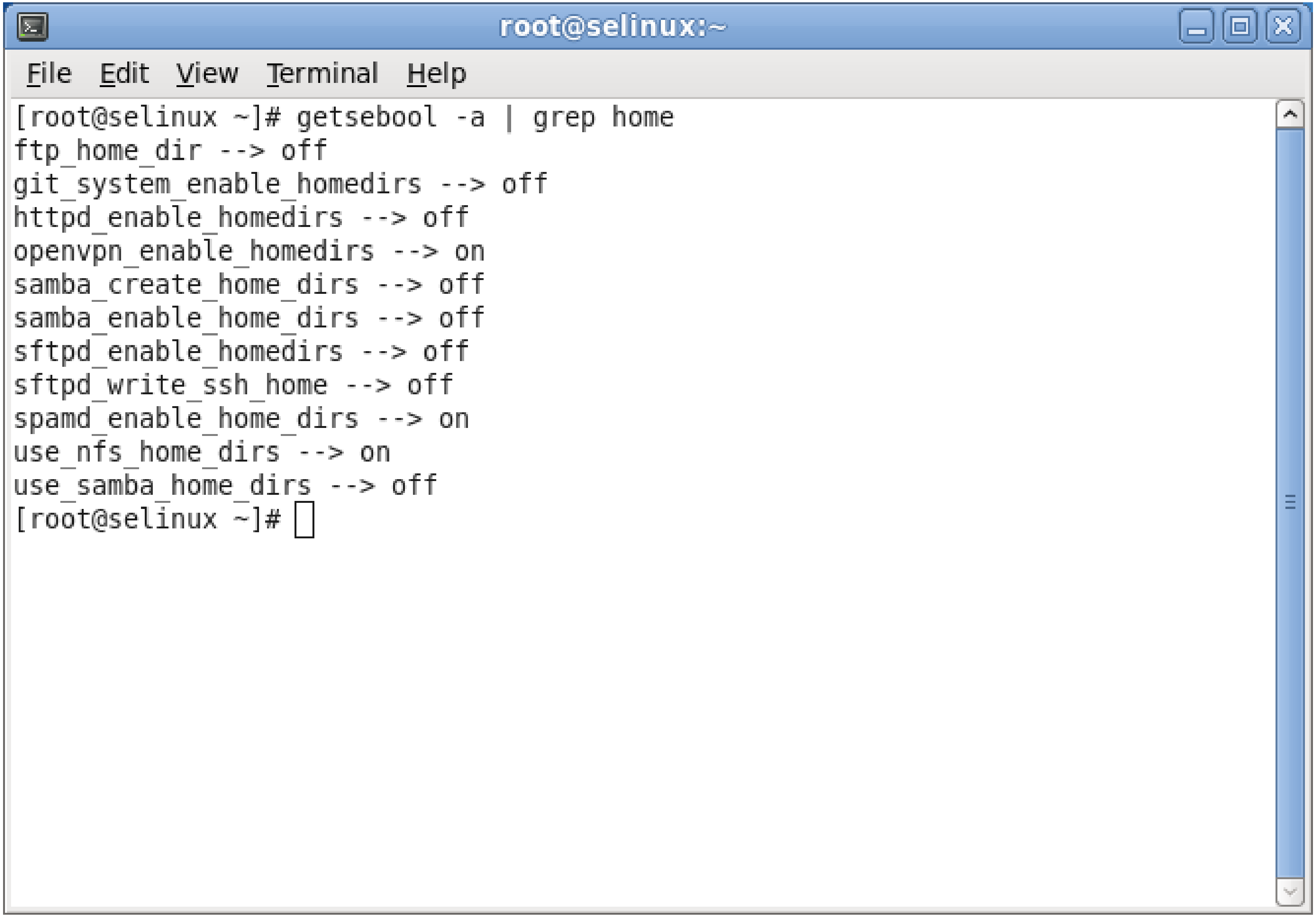


```
[john@selinux tmp]$ echo "This is data set 3" > tmpdata3.txt
[john@selinux tmp]$ mv tmpdata3.txt /home/john/mystuff/
[john@selinux tmp]$ cd
[john@selinux ~]$ ls -Z mystuff/
-rw-rw-r--. john john unconfined_u:object_r:user_home_t:s0 tmpdata2.txt
-rw-rw-r--. john john unconfined_u:object_r:user_tmp_t:s0 tmpdata3.txt
-rw-rw-r--. john john unconfined_u:object_r:user_home_t:s0 tmpdata.txt
[john@selinux ~]$ restorecon -vR mystuff/
restorecon reset /home/john/mystuff/tmpdata3.txt context unconfined_u:object_r:u
ser_tmp_t:s0->unconfined_u:object_r:user_home_t:s0
[john@selinux ~]$ 
```

SELinux Examples

Booleans

- If you have NFS mounted home directories, there are a couple of SELinux booleans you need to check.
- The default is to allow home directories on NFS.

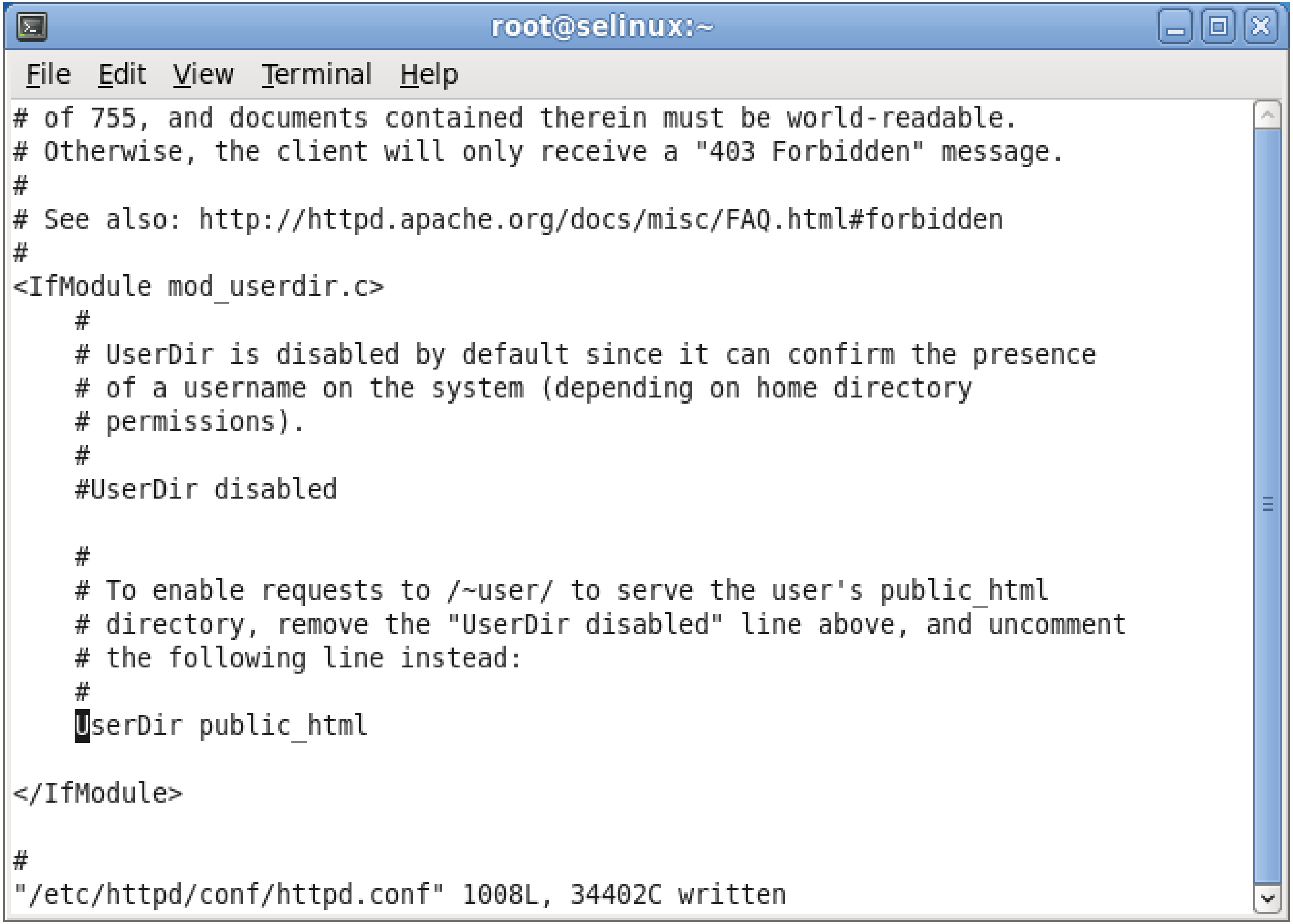


```
[root@selinux ~]# getsebool -a | grep home
ftp_home_dir --> off
git_system_enable_homedirs --> off
httpd_enable_homedirs --> off
openvpn_enable_homedirs --> on
samba_create_home_dirs --> off
samba_enable_home_dirs --> off
sftpd_enable_homedirs --> off
sftpd_write_ssh_home --> off
spamd_enable_home_dirs --> on
use_nfs_home_dirs --> on
use_samba_home_dirs --> off
[root@selinux ~]# 
```

SELinux Examples

SELinux vs. Apache. Enabling access in home directory

- Enable mod_userdir.c and uncomment “UserDir public_html” in /etc/httpd/conf/httpd.conf

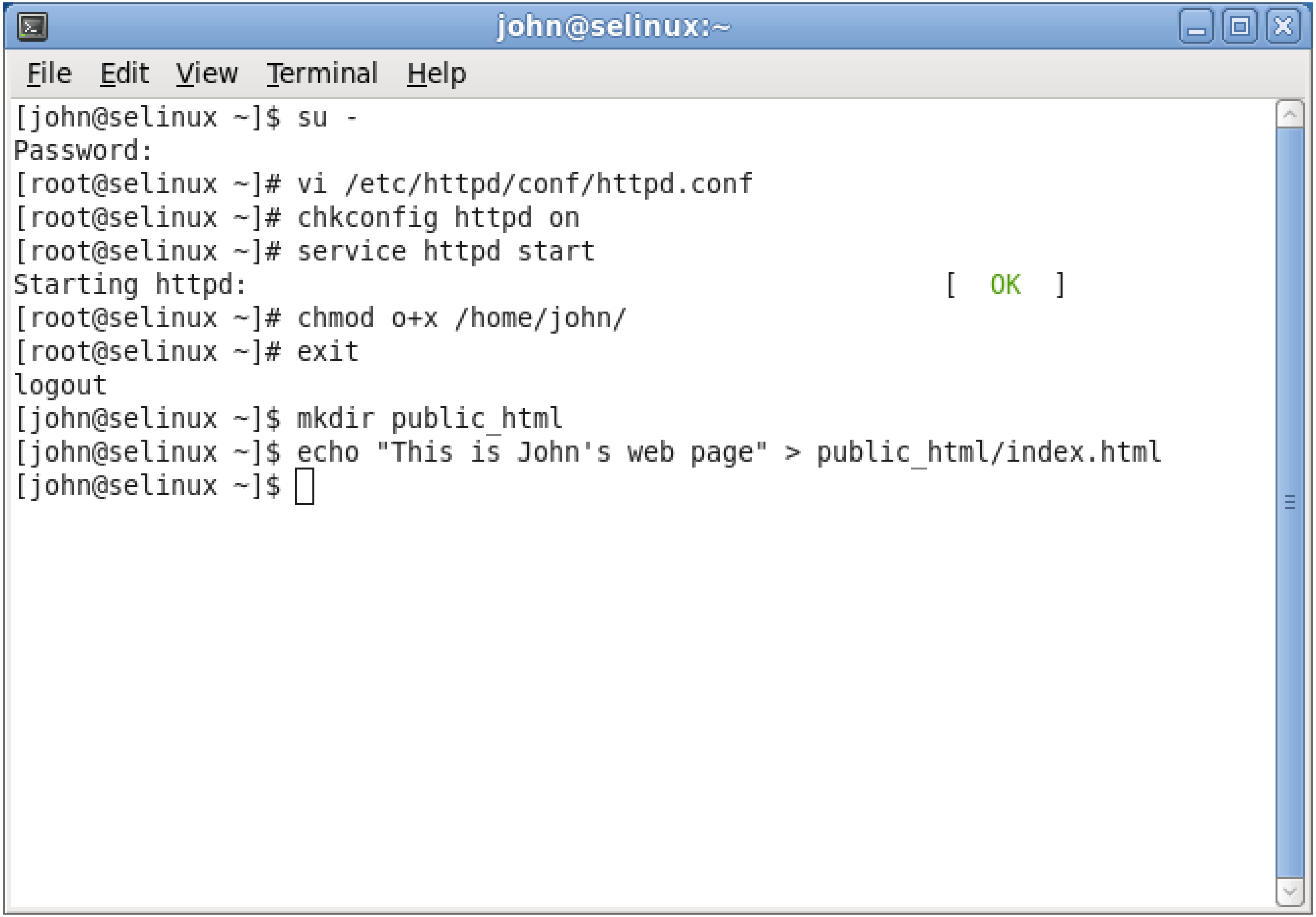


```
# of 755, and documents contained therein must be world-readable.
# Otherwise, the client will only receive a "403 Forbidden" message.
#
# See also: http://httpd.apache.org/docs/misc/FAQ.html#forbidden
#
<IfModule mod_userdir.c>
    #
    # UserDir is disabled by default since it can confirm the presence
    # of a username on the system (depending on home directory
    # permissions).
    #
    #UserDir disabled

    #
    # To enable requests to /~user/ to serve the user's public_html
    # directory, remove the "UserDir disabled" line above, and uncomment
    # the following line instead:
    #
    UserDir public_html

</IfModule>

#
"/etc/httpd/conf/httpd.conf" 1008L, 34402C written
```

SELinux Examples

SELinux vs. Apache. Enabling access in home directory

- As a user, create public_html in /home/[username] and “chmod o+x /home/[username]”
- Populate an index.html file

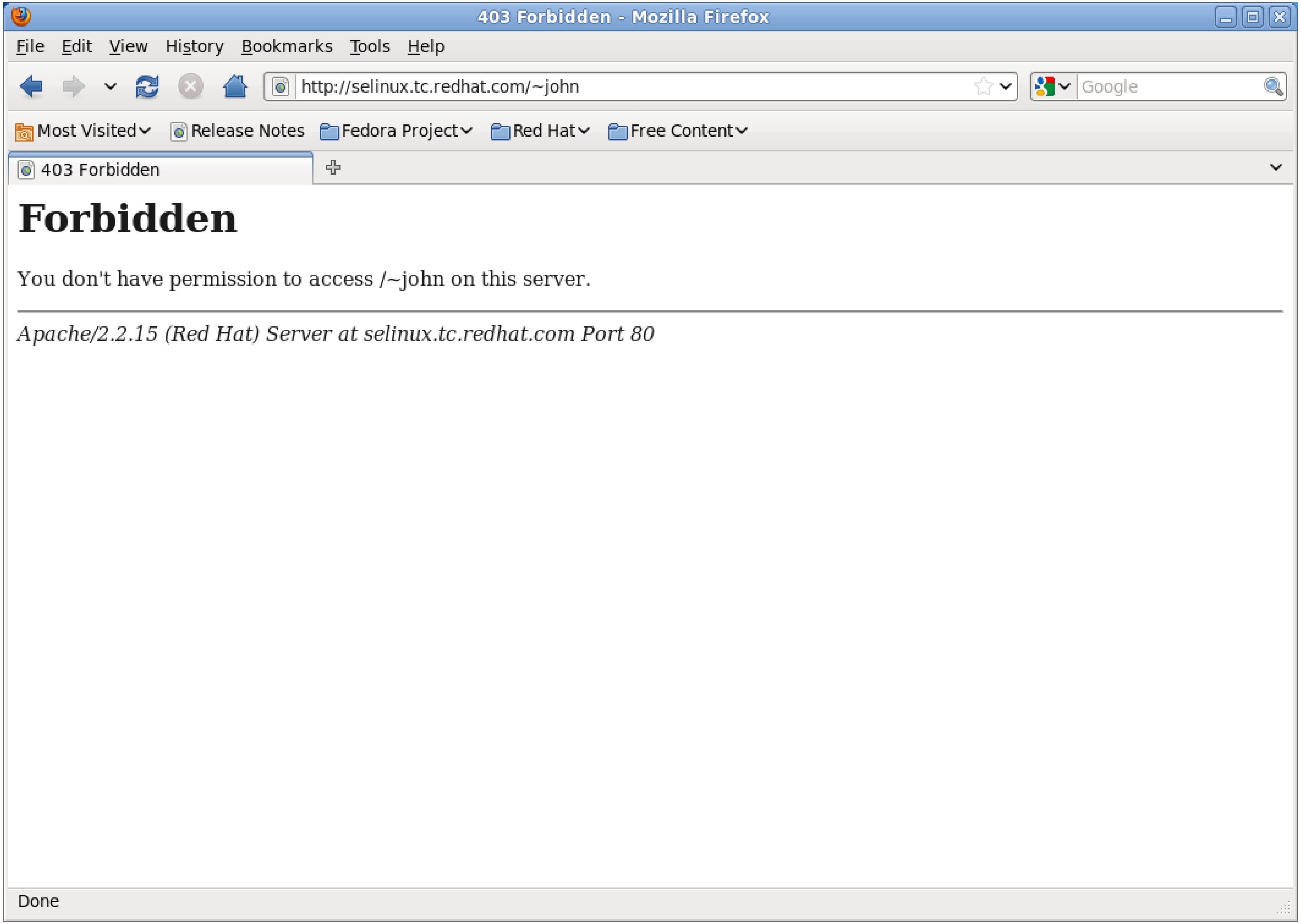


```
[john@selinux ~]$ su -
Password:
[root@selinux ~]# vi /etc/httpd/conf/httpd.conf
[root@selinux ~]# chkconfig httpd on
[root@selinux ~]# service httpd start
Starting httpd:                                            [  OK  ]
[root@selinux ~]# chmod o+x /home/john/
[root@selinux ~]# exit
logout
[john@selinux ~]$ mkdir public_html
[john@selinux ~]$ echo "This is John's web page" > public_html/index.html
[john@selinux ~]$ 
```

SELinux Examples

SELinux vs. Apache. Enabling access in home directory

- Connect with a browser



403 Forbidden - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://selinux.tc.redhat.com/~john
Google
Most Visited
Release Notes
Fedora Project
Red Hat
Free Content
403 Forbidden

Forbidden

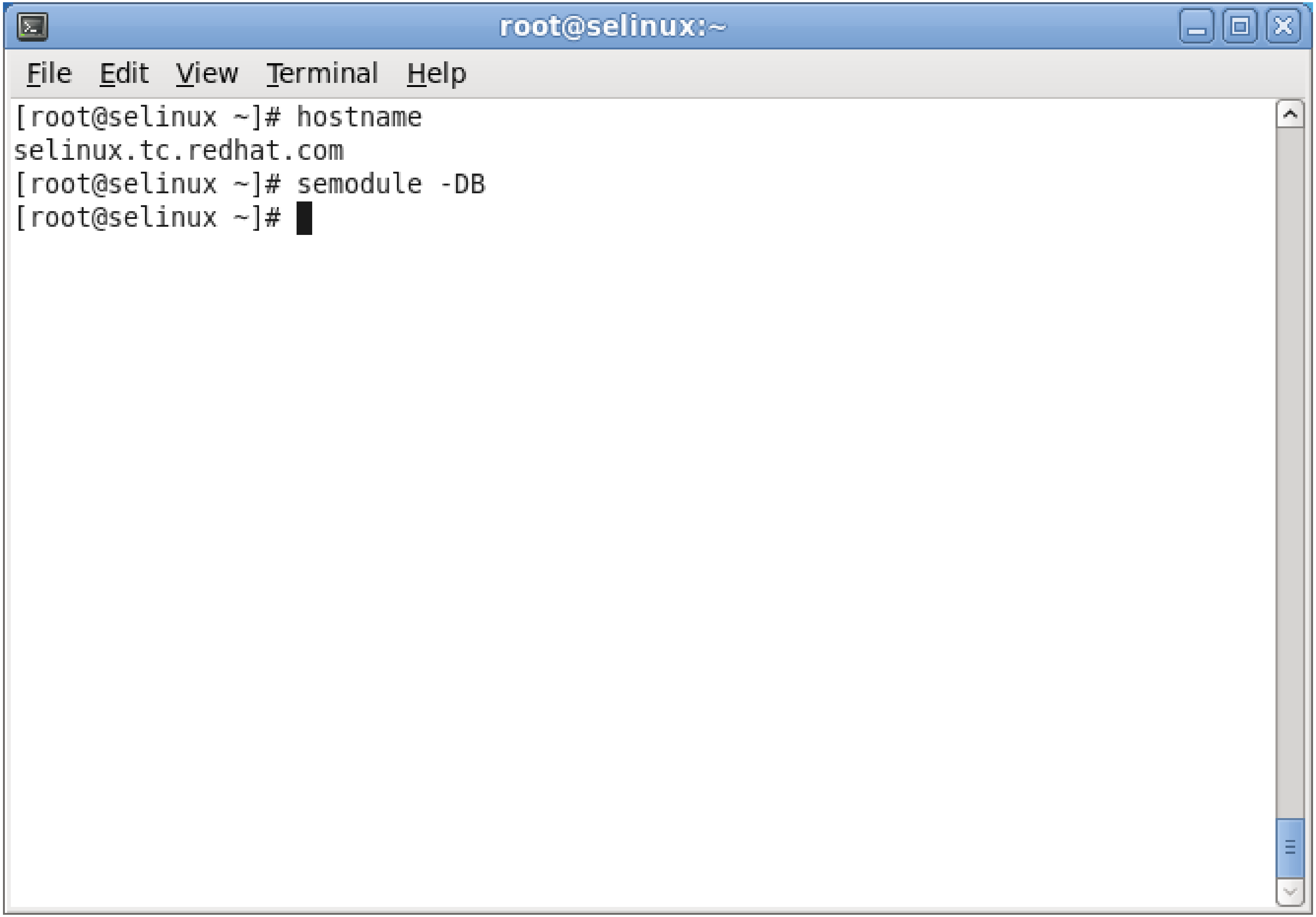
You don't have permission to access /~john on this server.

Apache/2.2.15 (Red Hat) Server at selinux.tc.redhat.com Port 80

SELinux Examples

SELinux vs. Apache. Enabling access in home directory

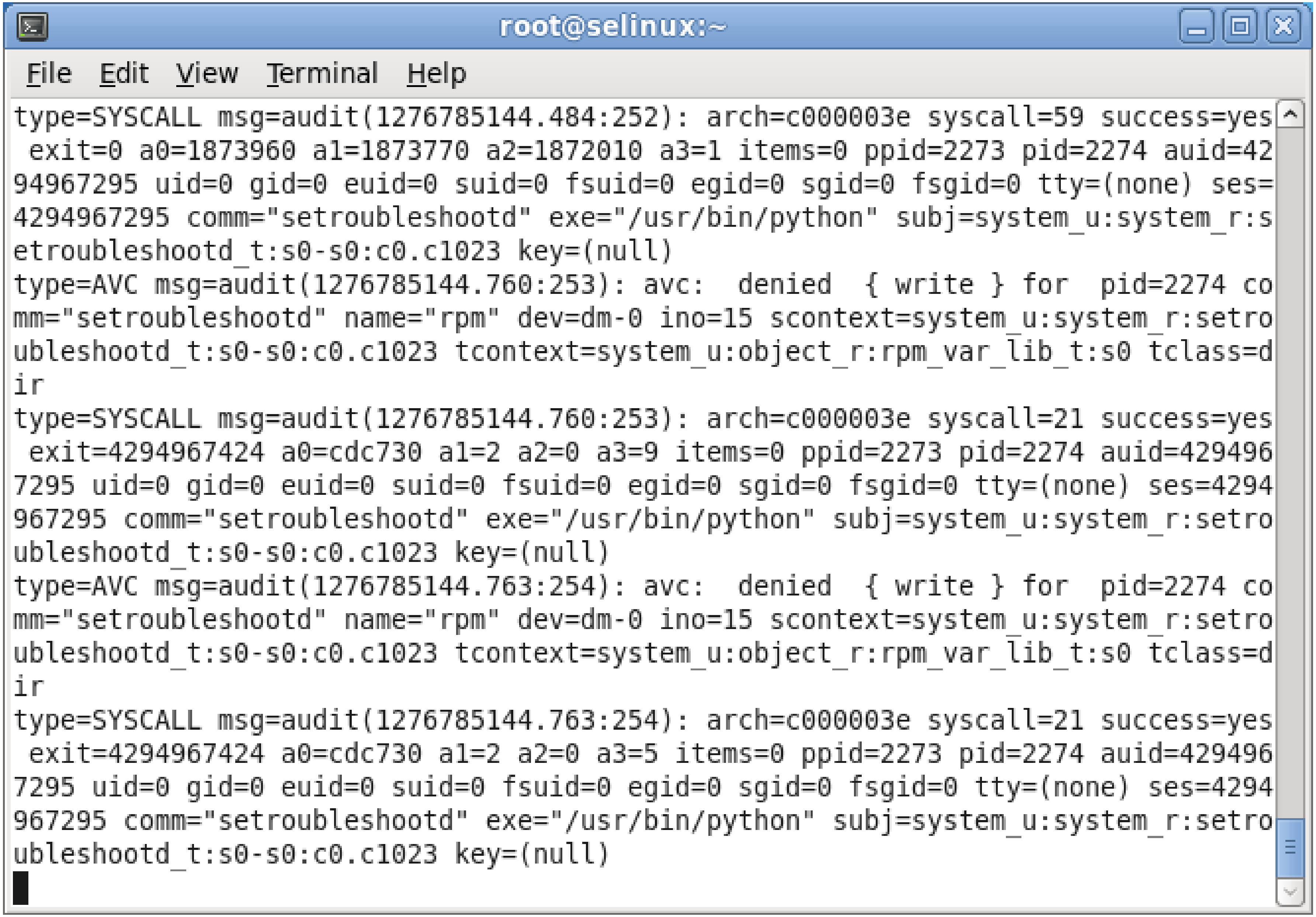
- How do we know if this is an SELinux denial?
  - Check /var/log/audit/audit.log
  - Check /var/log/messages
  - Check application logs
  - Temporarily disable SELinux
- Note that some really common errors are not audited to avoid filling the audit.log file. To turn on all auditing, run “semodule -DB”



```
[root@selinux ~]# hostname
selinux.tc.redhat.com
[root@selinux ~]# semodule -DB
[root@selinux ~]# 
```

SELinux Examples

Once full audit logging is on, you can watch /var/log/audit/audit.log when you see errors which aren't related to regular permissions.

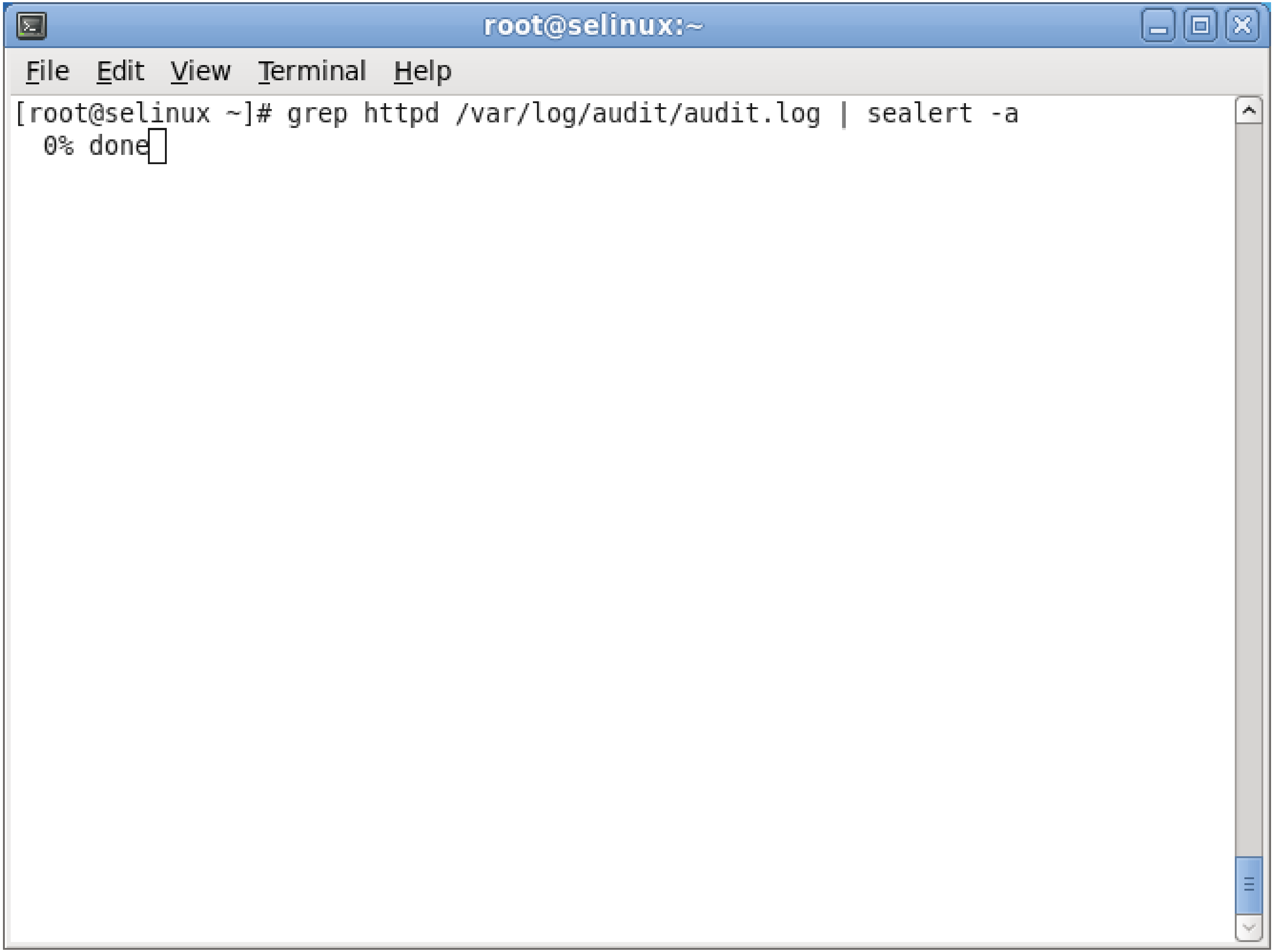


```
type=SYSCALL msg=audit(1276785144.484:252): arch=c000003e syscall=59 success=yes
 exit=0 a0=1873960 a1=1873770 a2=1872010 a3=1 items=0 ppid=2273 pid=2274 auid=42
94967295 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=
4294967295 comm="setroubleshootd" exe="/usr/bin/python" subj=system_u:system_r:s
etroubleshootd_t:s0-s0:c0.c1023 key=(null)
type=AVC msg=audit(1276785144.760:253): avc:  denied  { write } for  pid=2274 co
mm="setroubleshootd" name="rpm" dev=dm-0 ino=15 scontext=system_u:system_r:setro
ubleshootd_t:s0-s0:c0.c1023 tcontext=system_u:object_r:rpm_var_lib_t:s0 tclass=d
ir
type=SYSCALL msg=audit(1276785144.760:253): arch=c000003e syscall=21 success=yes
 exit=4294967424 a0=cdc730 a1=2 a2=0 a3=9 items=0 ppid=2273 pid=2274 auid=429496
7295 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=4294
967295 comm="setroubleshootd" exe="/usr/bin/python" subj=system_u:system_r:setro
ubleshootd_t:s0-s0:c0.c1023 key=(null)
type=AVC msg=audit(1276785144.763:254): avc:  denied  { write } for  pid=2274 co
mm="setroubleshootd" name="rpm" dev=dm-0 ino=15 scontext=system_u:system_r:setro
ubleshootd_t:s0-s0:c0.c1023 tcontext=system_u:object_r:rpm_var_lib_t:s0 tclass=d
ir
type=SYSCALL msg=audit(1276785144.763:254): arch=c000003e syscall=21 success=yes
 exit=4294967424 a0=cdc730 a1=2 a2=0 a3=5 items=0 ppid=2273 pid=2274 auid=429496
7295 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=4294
967295 comm="setroubleshootd" exe="/usr/bin/python" subj=system_u:system_r:setro
ubleshootd_t:s0-s0:c0.c1023 key=(null)
```

SELinux Examples

The log entries aren't terribly intuitive, so we use tools like “sealert -a /var/log/audit/audit.log”

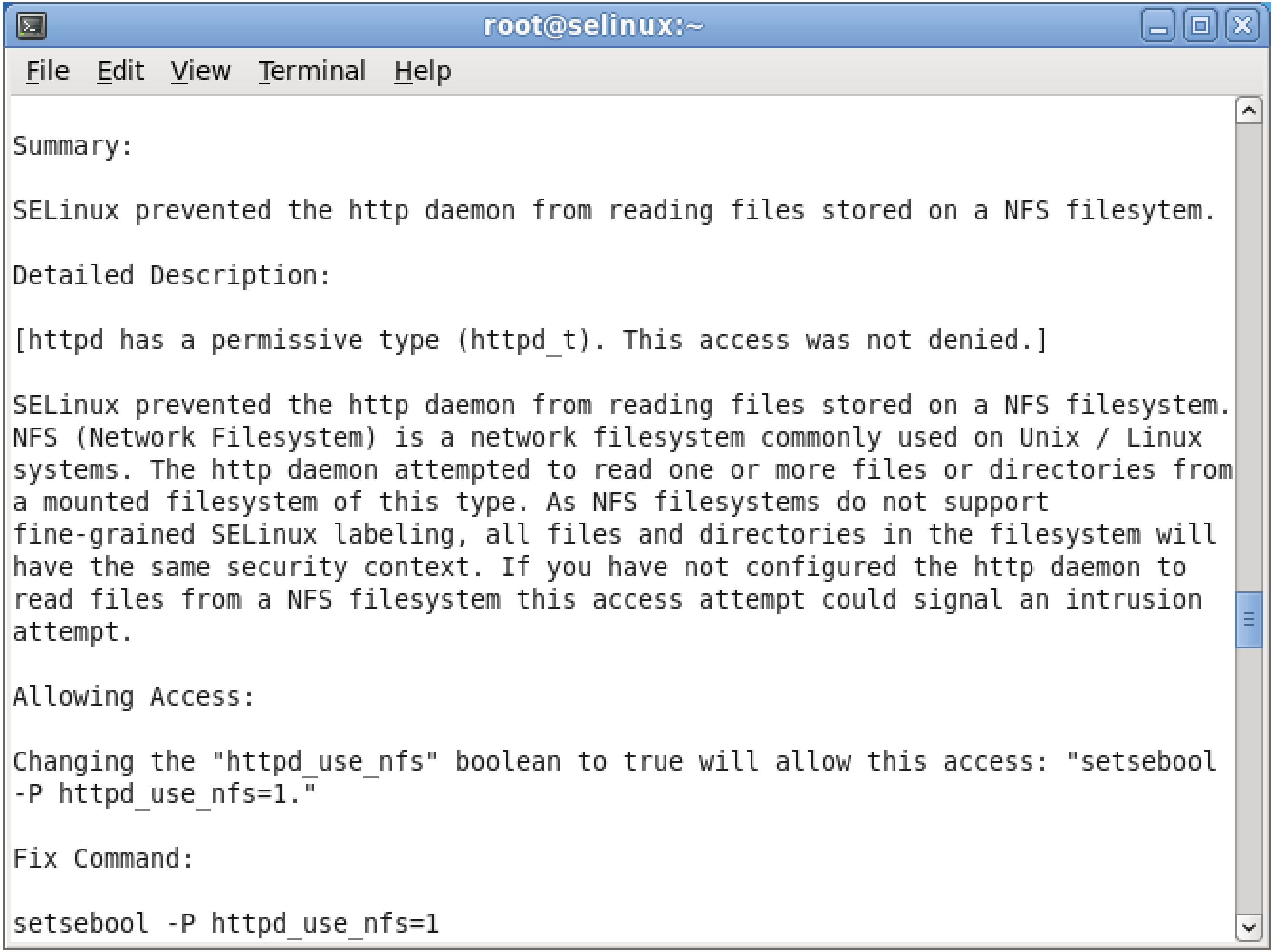
To make it easier to read, since we know it's a problem with httpd, you can issue “grep httpd /var/log/audit/audit.log | sealert -a”



```
[root@selinux ~]# grep httpd /var/log/audit/audit.log | sealert -a
  0% done
```

SELinux Examples

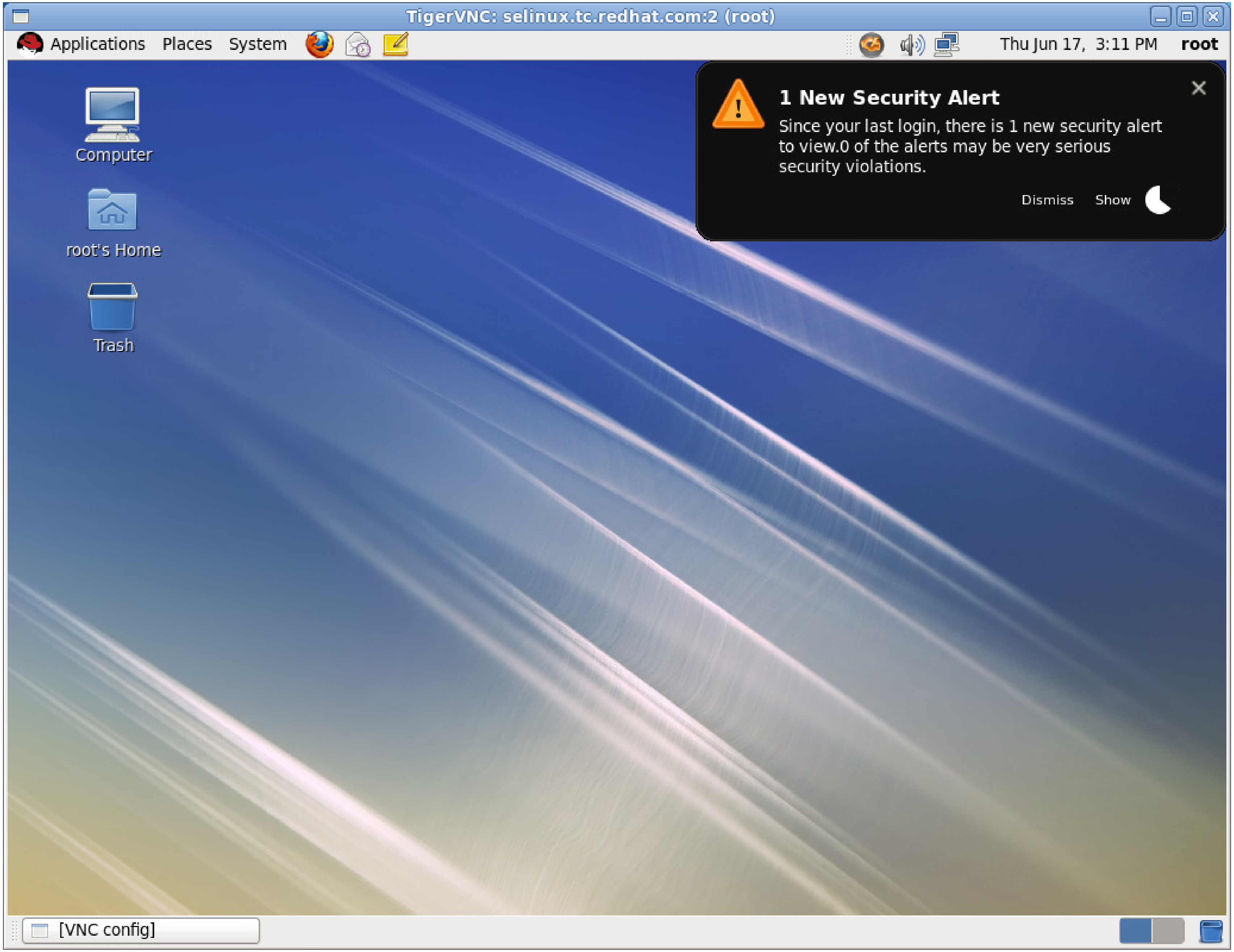
The output is human readable, and includes recommendations for how to allow the blocked access:



```
Summary:

SELinux prevented the http daemon from reading files stored on a NFS filesytem.

Detailed Description:

[httpd has a permissive type (httpd_t). This access was not denied.]

SELinux prevented the http daemon from reading files stored on a NFS filesystem.
NFS (Network Filesystem) is a network filesystem commonly used on Unix / Linux
systems. The http daemon 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 the http daemon to
read files from a NFS filesystem this access attempt could signal an intrusion
attempt.

Allowing Access:

Changing the "httpd_use_nfs" boolean to true will allow this access: "setsebool
-P httpd_use_nfs=1."

Fix Command:

setsebool -P httpd_use_nfs=1
```

SELinux Examples

For a graphical login, you'll get an setroubleshoot browser alert:



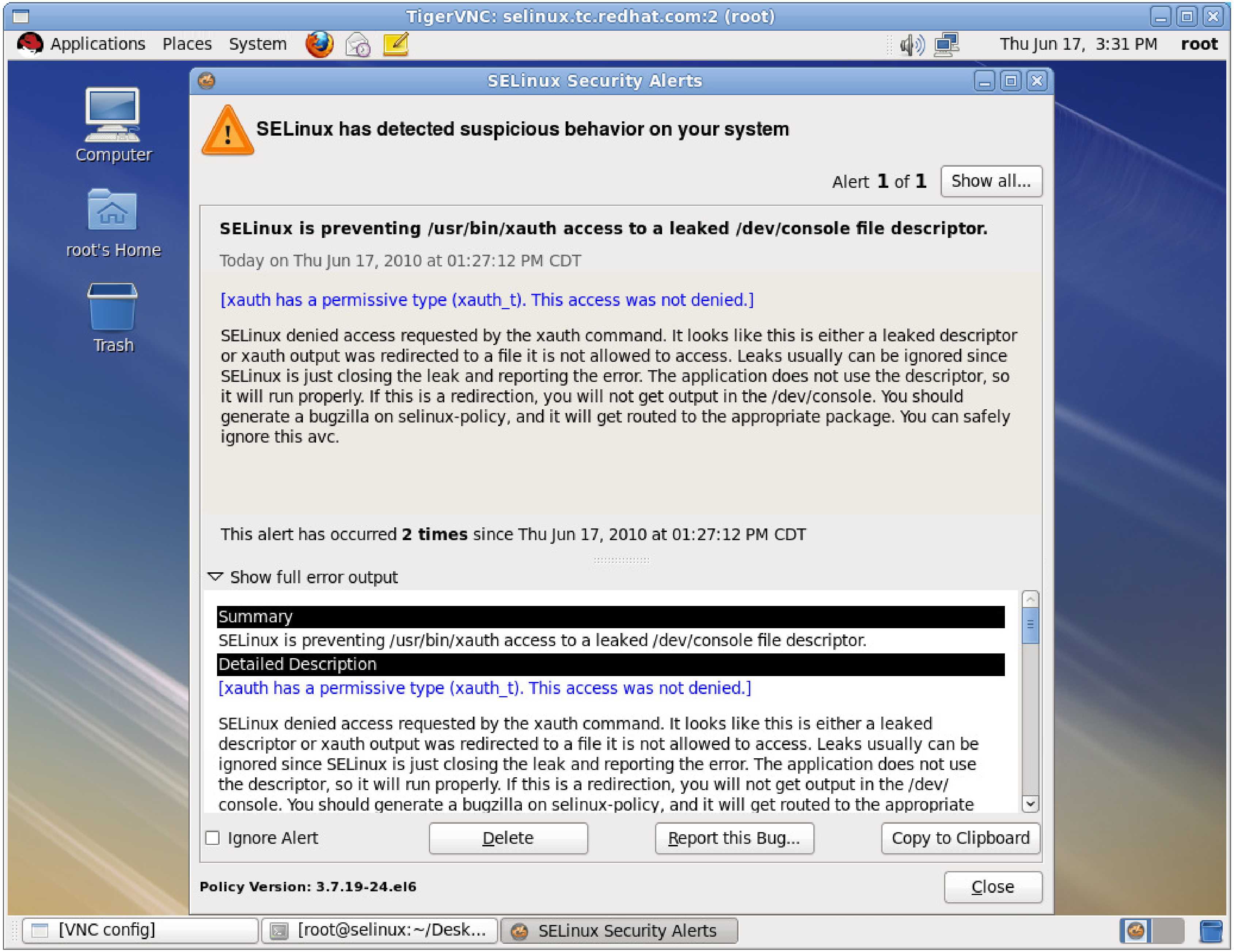
TigerVNC: selinux.tc.redhat.com:2 (root)

1 New Security Alert

Since your last login, there is 1 new security alert to view.0 of the alerts may be very serious security violations.

SELinux Examples

To use the graphical version of the SELinux troubleshooting browser, either click on the star or run “sealert -b”



SELinux has detected suspicious behavior on your system

Alert **1** of **1** Show all...

SELinux is preventing /usr/bin/xauth access to a leaked /dev/console file descriptor.

Today on Thu Jun 17, 2010 at 01:27:12 PM CDT

[xauth has a permissive type (xauth_t). This access was not denied.]

SELinux denied access requested by the xauth command. It looks like this is either a leaked descriptor or xauth output was redirected to a file it is not allowed to access. Leaks usually can be ignored since SELinux is just closing the leak and reporting the error. The application does not use the descriptor, so it will run properly. If this is a redirection, you will not get output in the /dev/console. You should generate a bugzilla on selinux-policy, and it will get routed to the appropriate package. You can safely ignore this avc.

This alert has occurred **2 times** since Thu Jun 17, 2010 at 01:27:12 PM CDT

▽ Show full error output

Summary

SELinux is preventing /usr/bin/xauth access to a leaked /dev/console file descriptor.

Detailed Description

[xauth has a permissive type (xauth_t). This access was not denied.]

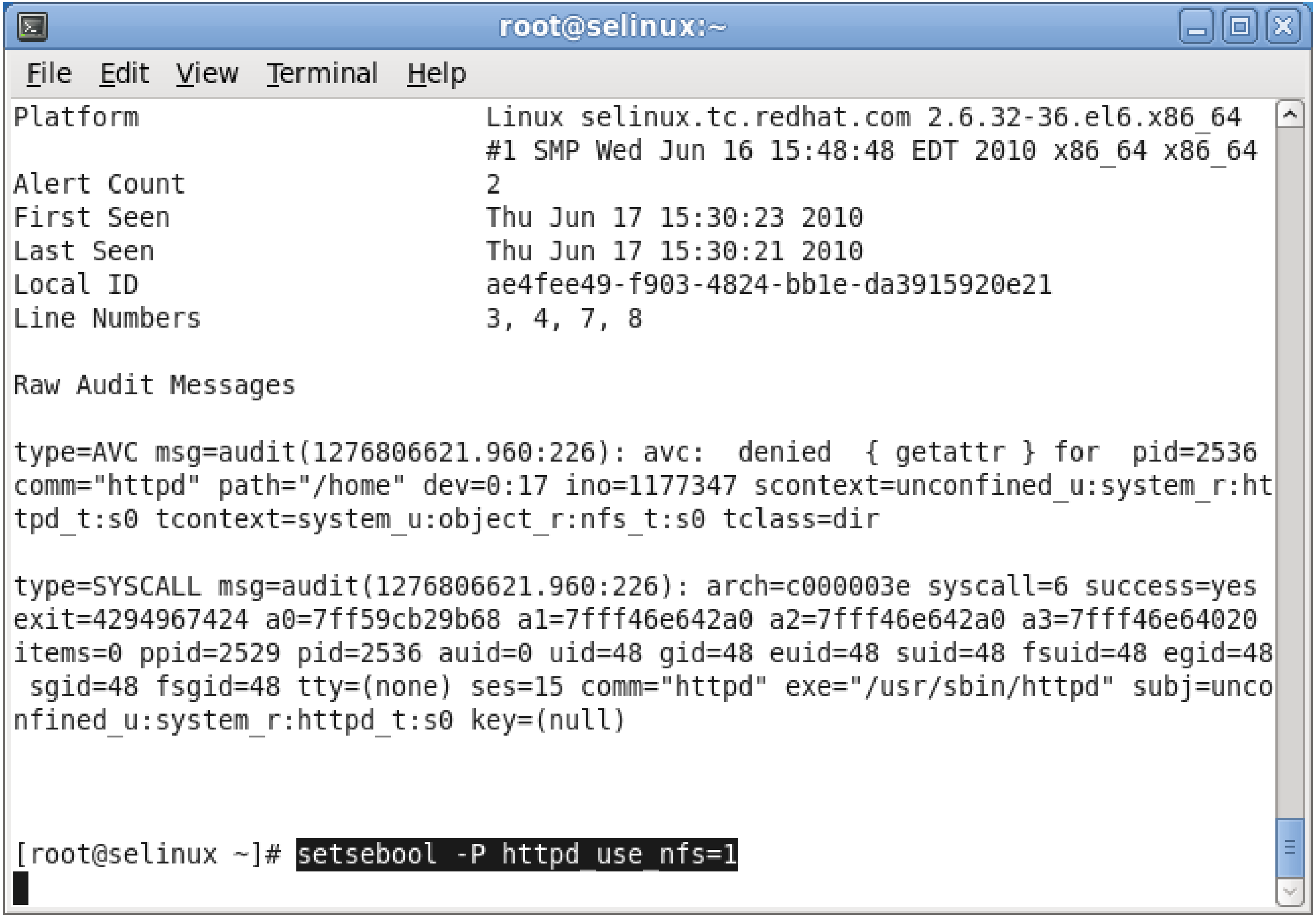
SELinux denied access requested by the xauth command. It looks like this is either a leaked descriptor or xauth output was redirected to a file it is not allowed to access. Leaks usually can be ignored since SELinux is just closing the leak and reporting the error. The application does not use the descriptor, so it will run properly. If this is a redirection, you will not get output in the /dev/console. You should generate a bugzilla on selinux-policy, and it will get routed to the appropriate

□ Ignore Alert   Delete   Report this Bug...   Copy to Clipboard

Policy Version: 3.7.19-24.el6   Close

SELinux Examples

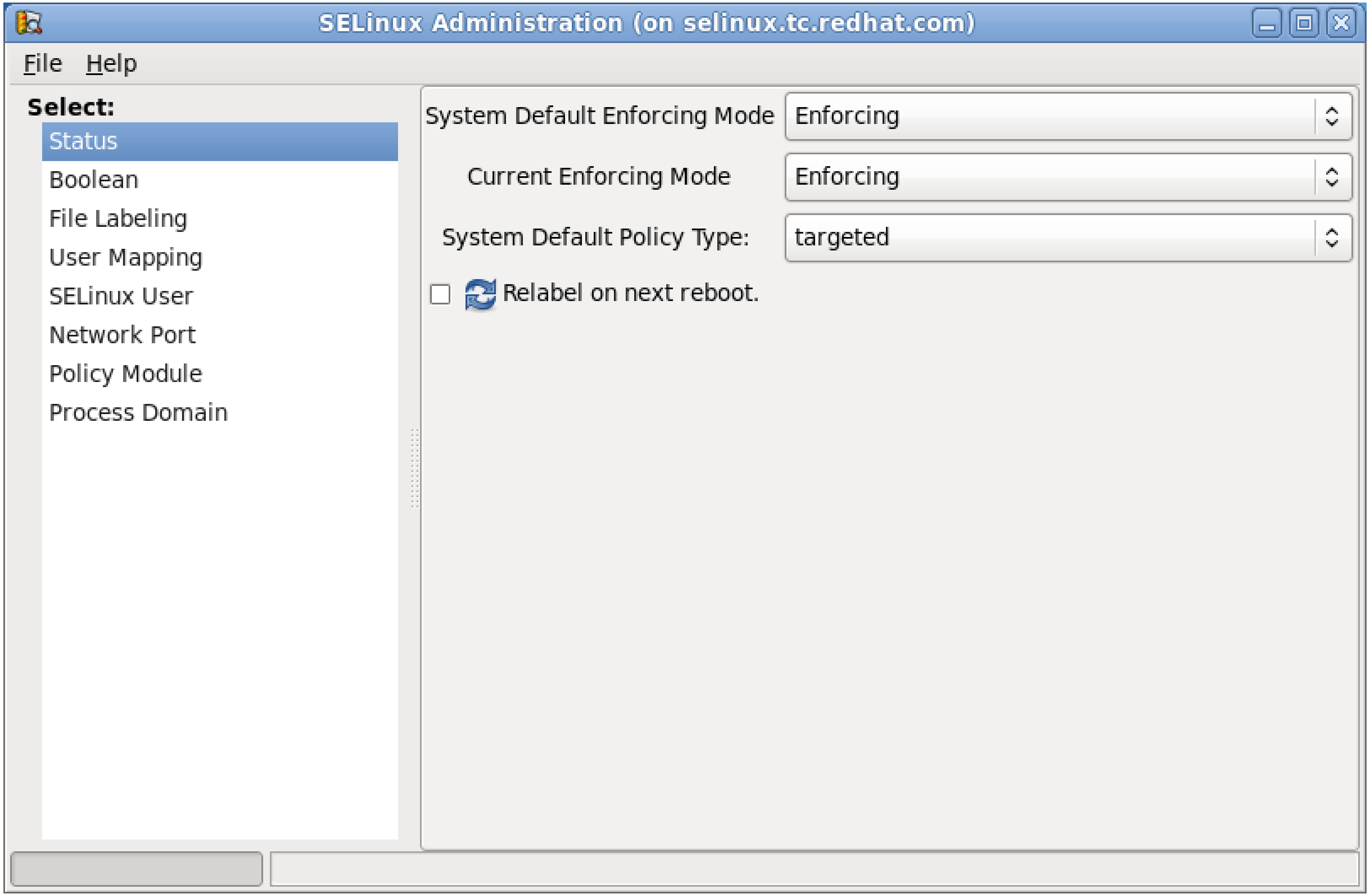
To allow the access, we set the appropriate boolean, in this case “setsebool -P httpd_use_nfs=1”



```
Platform                      Linux selinux.tc.redhat.com 2.6.32-36.el6.x86_64
                              #1 SMP Wed Jun 16 15:48:48 EDT 2010 x86_64 x86_64
Alert Count                   2
First Seen                    Thu Jun 17 15:30:23 2010
Last Seen                     Thu Jun 17 15:30:21 2010
Local ID                      ae4fee49-f903-4824-bb1e-da3915920e21
Line Numbers                  3, 4, 7, 8

Raw Audit Messages

type=AVC msg=audit(1276806621.960:226): avc:  denied  { getattr } for  pid=2536
comm="httpd" path="/home" dev=0:17 ino=1177347 scontext=unconfined_u:system_r:ht
tpd_t:s0 tcontext=system_u:object_r:nfs_t:s0 tclass=dir

type=SYSCALL msg=audit(1276806621.960:226): arch=c000003e syscall=6 success=yes
exit=4294967424 a0=7ff59cb29b68 a1=7fff46e642a0 a2=7fff46e642a0 a3=7fff46e64020
items=0 ppid=2529 pid=2536 auid=0 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48
 sgid=48 fsgid=48 tty=(none) ses=15 comm="httpd" exe="/usr/sbin/httpd" subj=unco
nfined_u:system_r:httpd_t:s0 key=(null)



[root@selinux ~]# setsebool -P httpd_use_nfs=1
```

SELinux Examples

You can also use system-config-selinux to set booleans



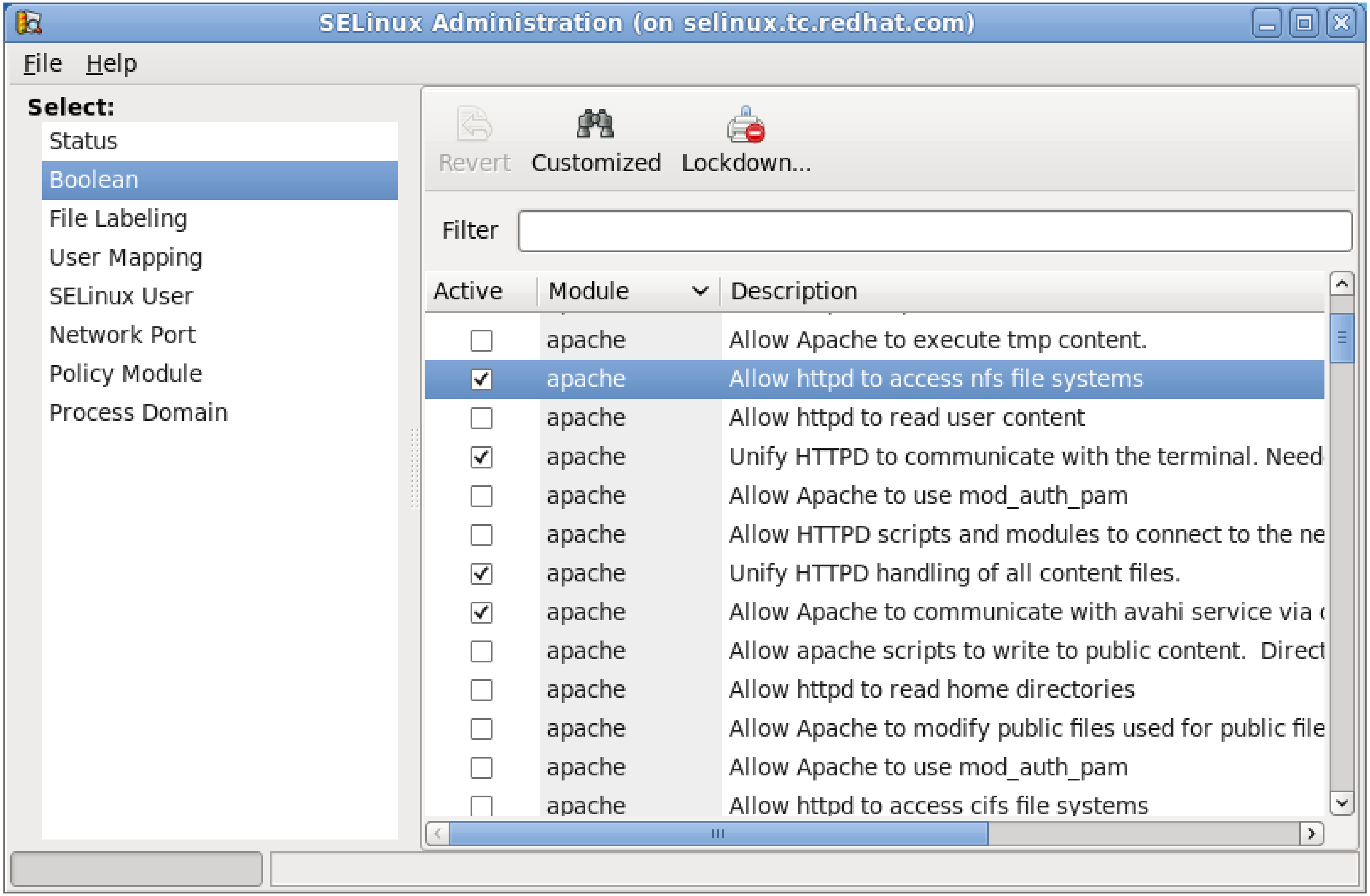
SELinux Administration (on selinux.tc.redhat.com)

System Default Enforcing Mode: Enforcing

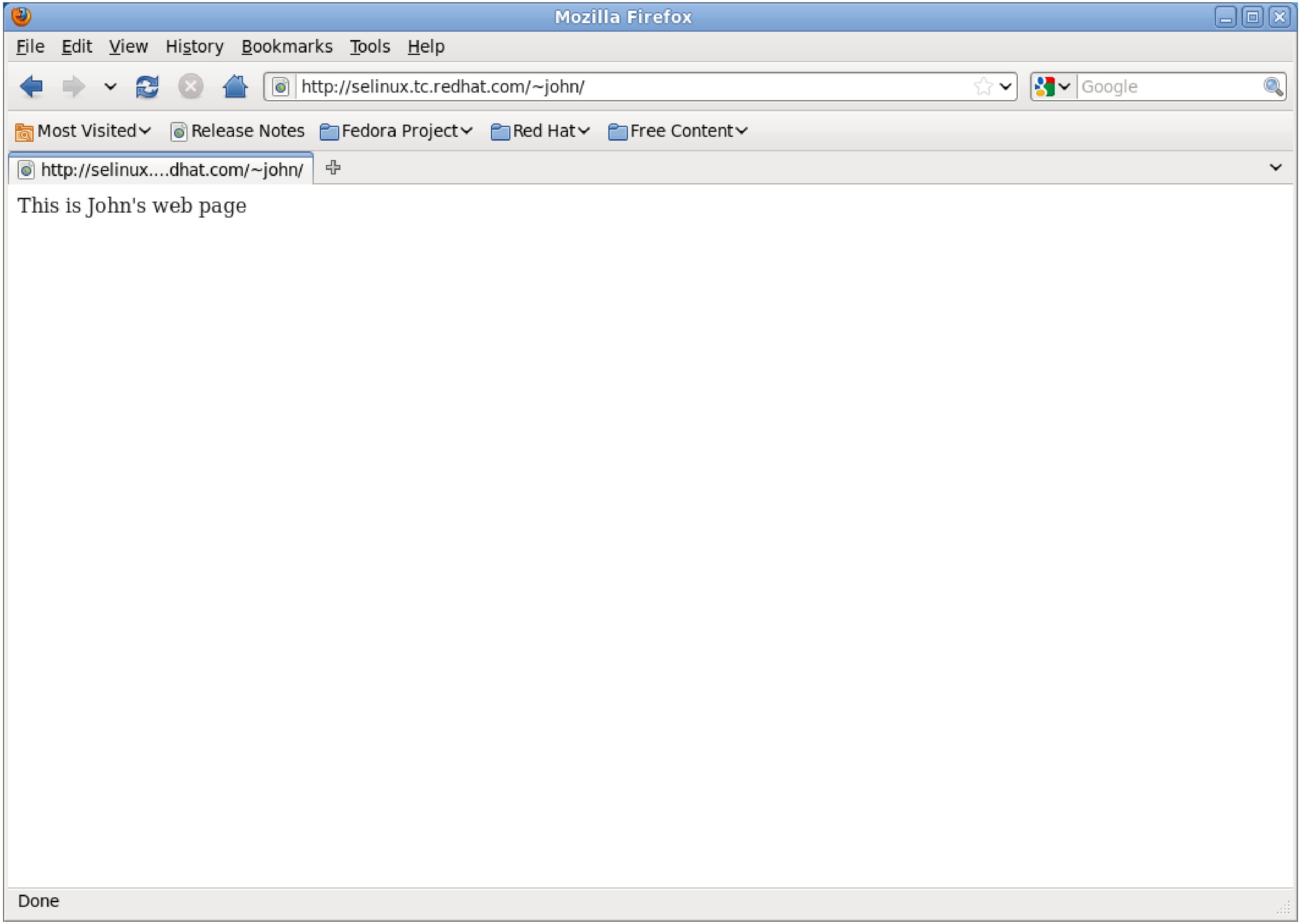
Current Enforcing Mode: Enforcing

System Default Policy Type: targeted

□ Relabel on next reboot.



Active	Module	Description
□	apache	Allow Apache to execute tmp content.
☑	apache	Allow httpd to access nfs file systems
□	apache	Allow httpd to read user content
☑	apache	Unify HTTPD to communicate with the terminal. Need
□	apache	Allow Apache to use mod_auth_pam
□	apache	Allow HTTPD scripts and modules to connect to the ne
☑	apache	Unify HTTPD handling of all content files.
☑	apache	Allow Apache to communicate with avahi service via (
□	apache	Allow apache scripts to write to public content. Direct
□	apache	Allow httpd to read home directories
□	apache	Allow Apache to modify public files used for public file
□	apache	Allow Apache to use mod_auth_pam
□	apache	Allow httpd to access cifs file systems



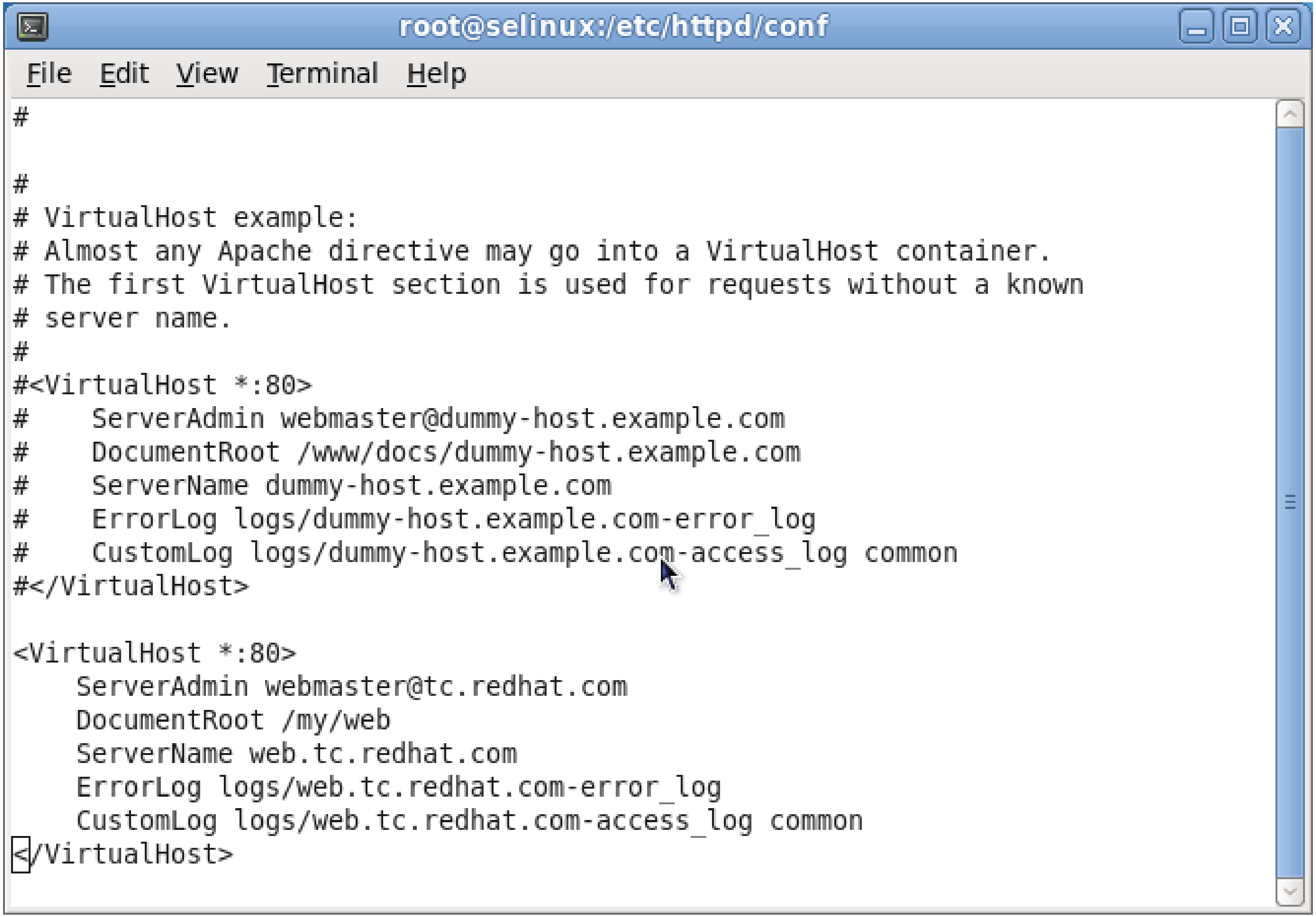
Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://selinux.tc.redhat.com/~john/
Google
Most Visited
Release Notes
Fedora Project
Red Hat
Free Content
http://selinux....dhat.com/~john/

This is John's web page

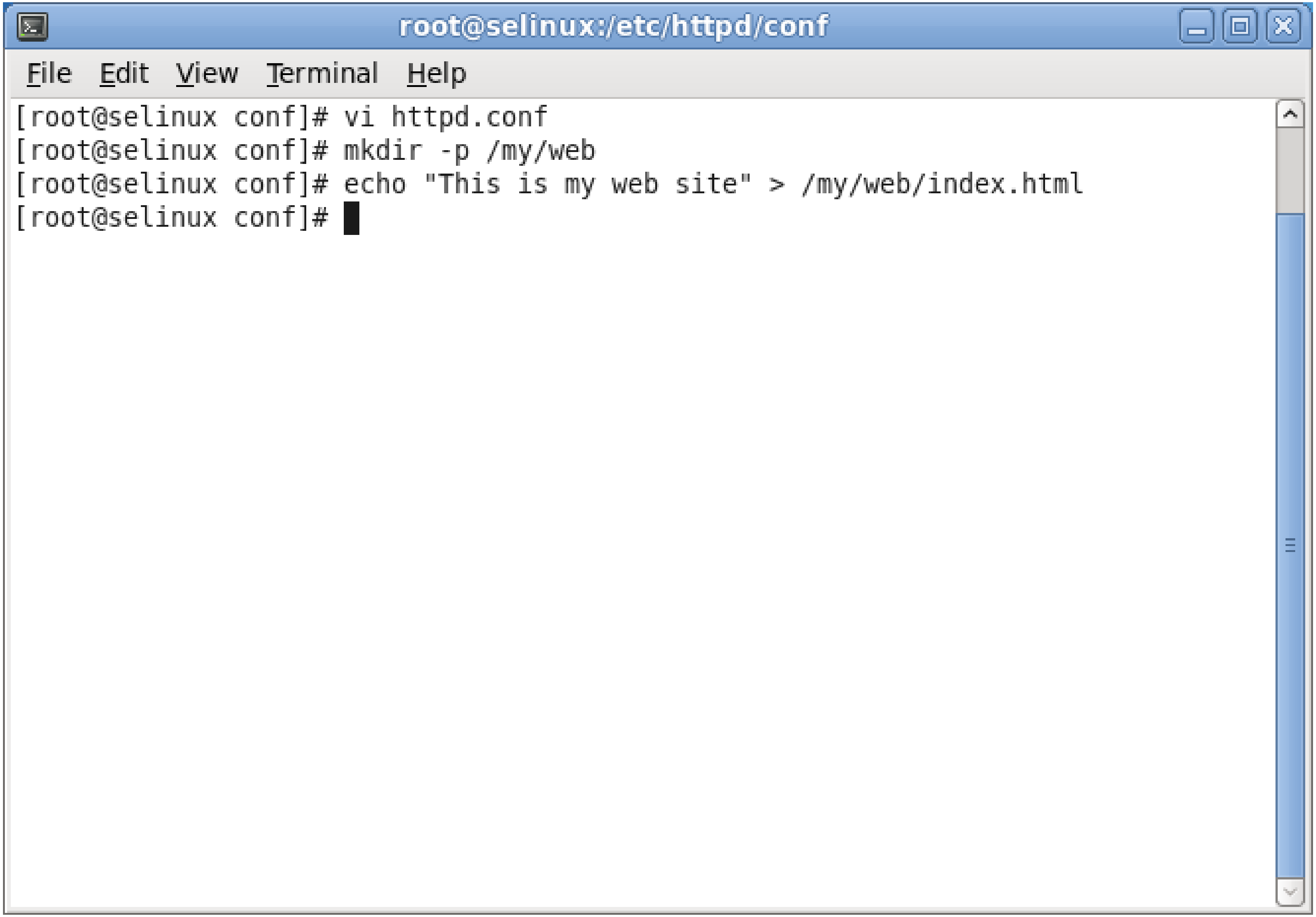
Done

SELinux Examples

Setting up an Apache virtual host in a weird place on the filesystem (/my/web).



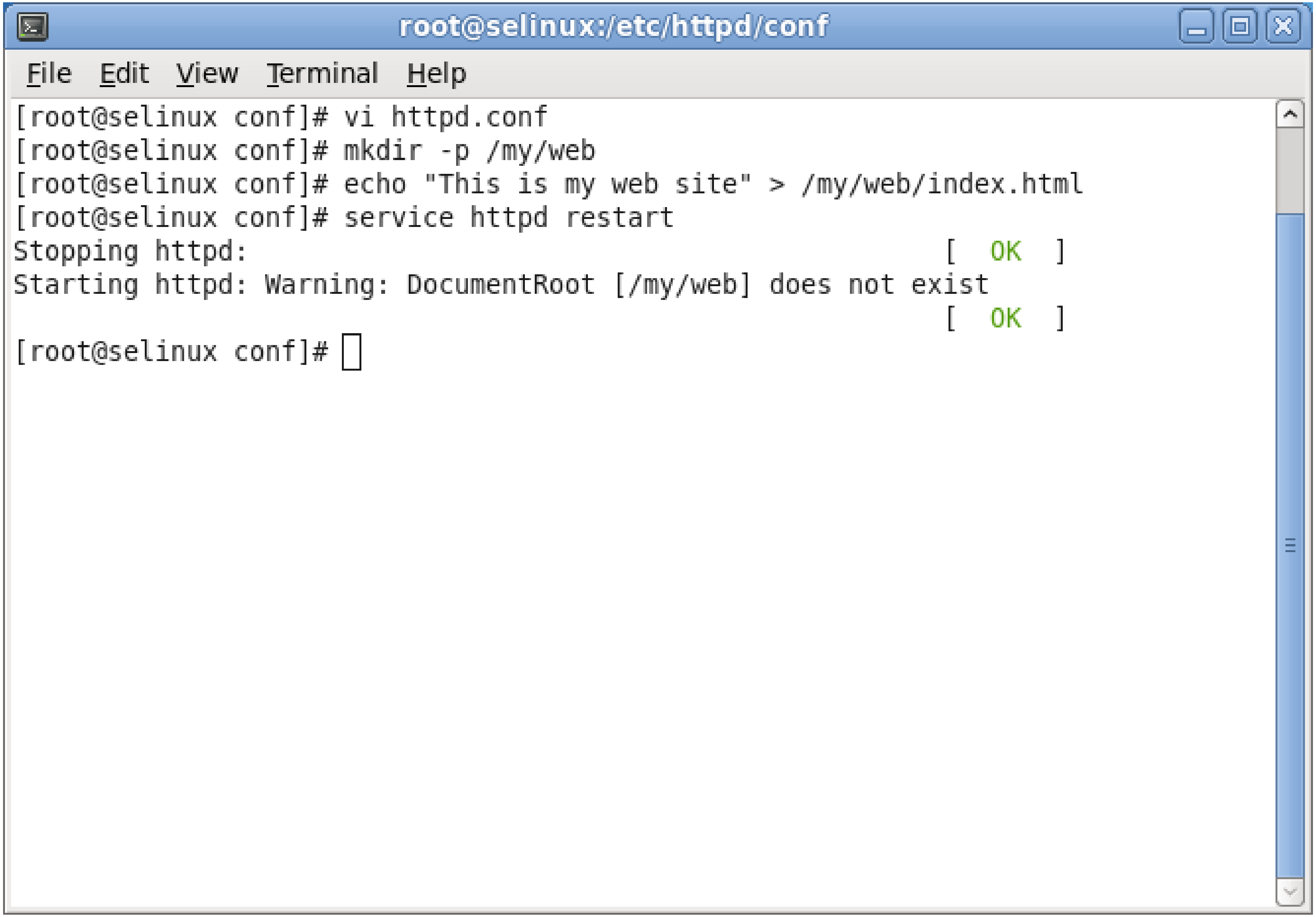
```
#


#
# 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@tc.redhat.com
    DocumentRoot /my/web
    ServerName web.tc.redhat.com
    ErrorLog logs/web.tc.redhat.com-error_log
    CustomLog logs/web.tc.redhat.com-access_log common
</VirtualHost>
```



```
[root@selinux conf]# vi httpd.conf
[root@selinux conf]# mkdir -p /my/web
[root@selinux conf]# echo "This is my web site" > /my/web/index.html
[root@selinux conf]# 
```

SELinux Examples

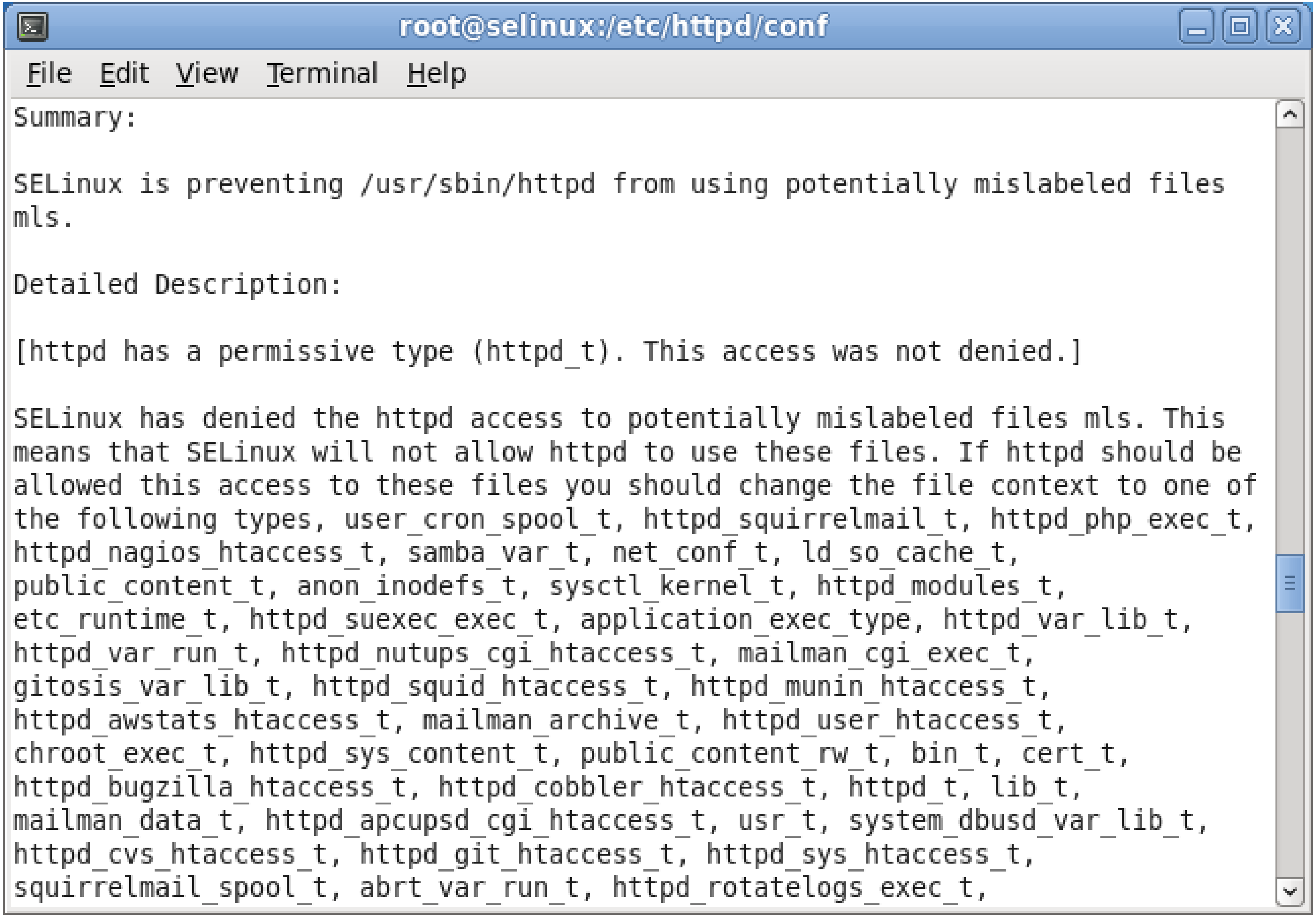
Restart Apache to start serving up the new site.



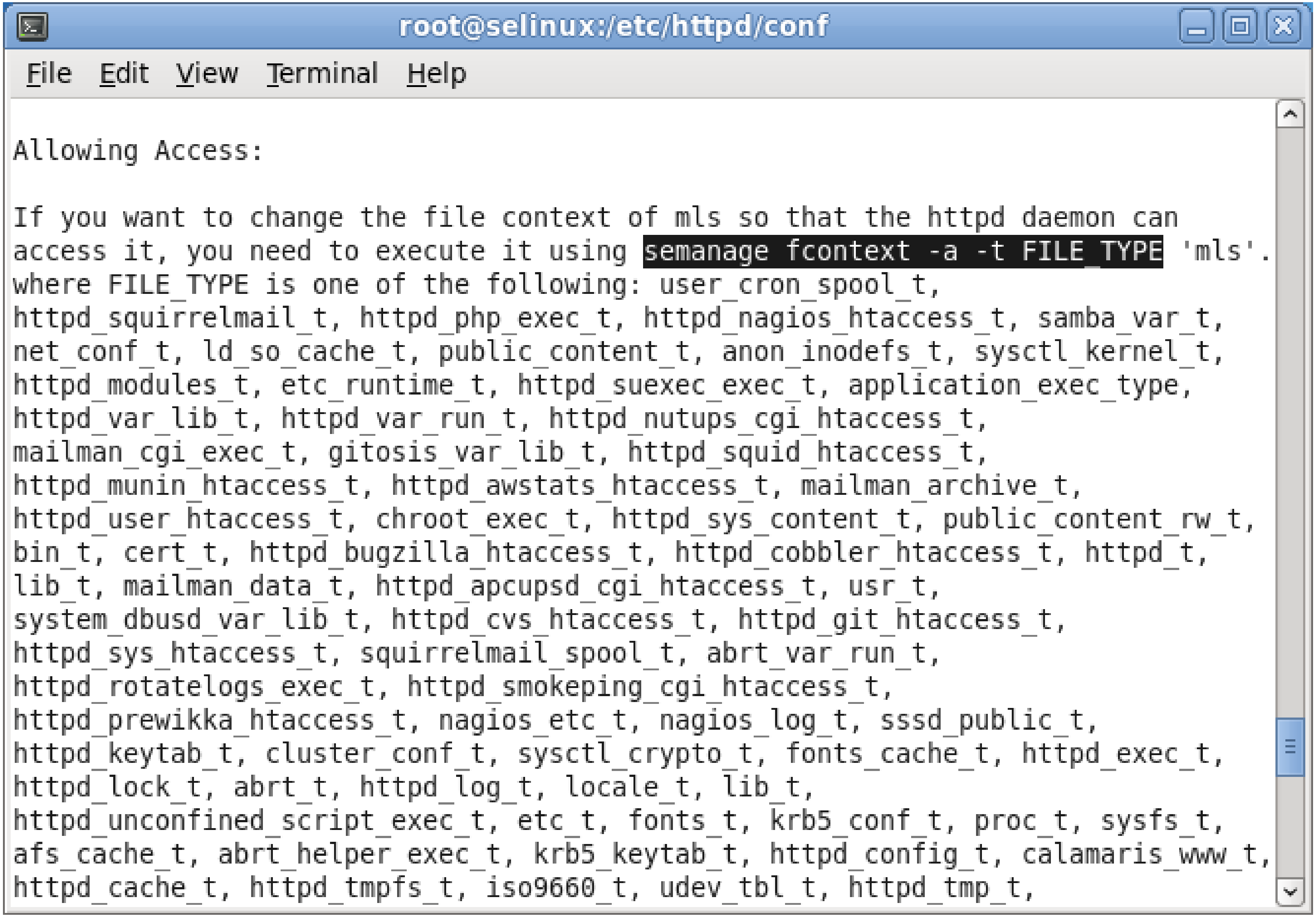
```
[root@selinux conf]# vi httpd.conf
[root@selinux conf]# mkdir -p /my/web
[root@selinux conf]# echo "This is my web site" > /my/web/index.html
[root@selinux conf]# service httpd restart
Stopping httpd:                                            [  OK  ]
Starting httpd: Warning: DocumentRoot [/my/web] does not exist
                                                           [  OK  ]
[root@selinux conf]# 
```

SELinux Examples

We know the directory does exist, you can see it in the screen shot! We can run “grep httpd /var/log/audit/audit.log | sealert -a” to see what's going on:



```
Summary:

SELinux is preventing /usr/sbin/httpd from using potentially mislabeled files
mls.

Detailed Description:

[httpd has a permissive type (httpd_t). This access was not denied.]

SELinux has denied the httpd access to potentially mislabeled files mls. This
means that SELinux will not allow httpd to use these files. If httpd should be
allowed this access to these files you should change the file context to one of
the following types, user_cron_spool_t, httpd_squirrelmail_t, httpd_php_exec_t,
httpd_nagios_htaccess_t, samba_var_t, net_conf_t, ld_so_cache_t,
public_content_t, anon_inodefs_t, sysctl_kernel_t, httpd_modules_t,
etc_runtime_t, httpd_suexec_exec_t, application_exec_type, httpd_var_lib_t,
httpd_var_run_t, httpd_nutups_cgi_htaccess_t, mailman_cgi_exec_t,
gitosis_var_lib_t, httpd_squid_htaccess_t, httpd_munin_htaccess_t,
httpd_awstats_htaccess_t, mailman_archive_t, httpd_user_htaccess_t,
chroot_exec_t, httpd_sys_content_t, public_content_rw_t, bin_t, cert_t,
httpd_bugzilla_htaccess_t, httpd_cobbler_htaccess_t, httpd_t, lib_t,
mailman_data_t, httpd_apcupsd_cgi_htaccess_t, usr_t, system_dbusd_var_lib_t,
httpd_cvs_htaccess_t, httpd_git_htaccess_t, httpd_sys_htaccess_t,
squirrelmail_spool_t, abrt_var_run_t, httpd_rotatelogs_exec_t,
```



```
Allowing Access:

If you want to change the file context of mls so that the httpd daemon can
access it, you need to execute it using semanage fcontext -a -t FILE_TYPE 'mls'.
where FILE_TYPE is one of the following: user_cron_spool_t,
httpd_squirrelmail_t, httpd_php_exec_t, httpd_nagios_htaccess_t, samba_var_t,
net_conf_t, ld_so_cache_t, public_content_t, anon_inodefs_t, sysctl_kernel_t,
httpd_modules_t, etc_runtime_t, httpd_suexec_exec_t, application_exec_type,
httpd_var_lib_t, httpd_var_run_t, httpd_nutups_cgi_htaccess_t,
mailman_cgi_exec_t, gitosis_var_lib_t, httpd_squid_htaccess_t,
httpd_munin_htaccess_t, httpd_awstats_htaccess_t, mailman_archive_t,
httpd_user_htaccess_t, chroot_exec_t, httpd_sys_content_t, public_content_rw_t,
bin_t, cert_t, httpd_bugzilla_htaccess_t, httpd_cobbler_htaccess_t, httpd_t,
lib_t, mailman_data_t, httpd_apcupsd_cgi_htaccess_t, usr_t,
system_dbusd_var_lib_t, httpd_cvs_htaccess_t, httpd_git_htaccess_t,
httpd_sys_htaccess_t, squirrelmail_spool_t, abrt_var_run_t,
httpd_rotatelogs_exec_t, httpd_smokeping_cgi_htaccess_t,
httpd_prewikka_htaccess_t, nagios_etc_t, nagios_log_t, sssd_public_t,
httpd_keytab_t, cluster_conf_t, sysctl_crypto_t, fonts_cache_t, httpd_exec_t,
httpd_lock_t, abrt_t, httpd_log_t, locale_t, lib_t,
httpd_unconfined_script_exec_t, etc_t, fonts_t, krb5_conf_t, proc_t, sysfs_t,
afs_cache_t, abrt_helper_exec_t, krb5_keytab_t, httpd_config_t, calamaris_www_t,
httpd_cache_t, httpd_tmpfs_t, iso9660_t, udev_tbl_t, httpd_tmp_t,
```

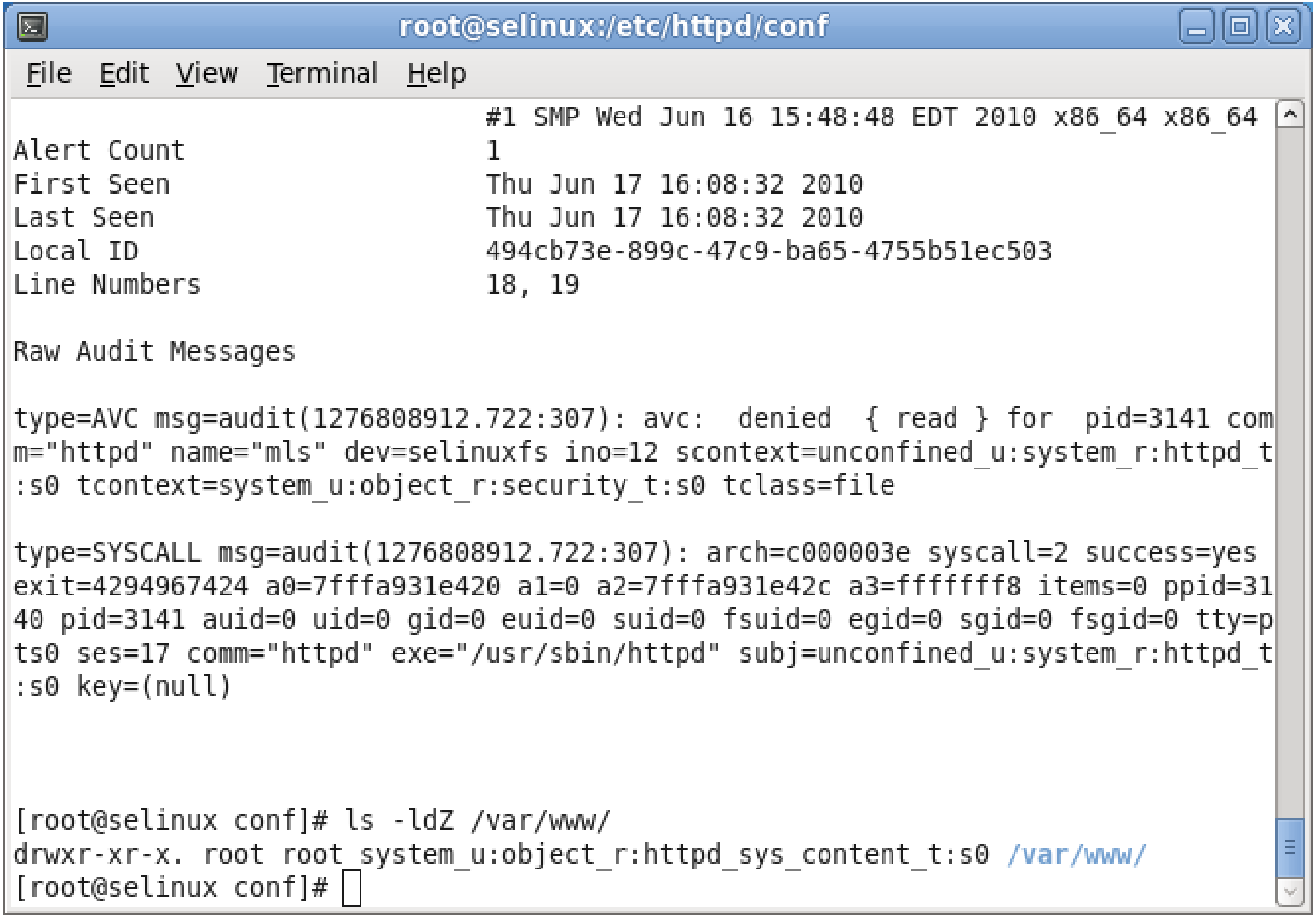
SELinux Examples

In this case, it's not as clear exactly what change to make. The key here is that it's telling you the filesystem is not labeled correctly for a process running in httpd_t.

We can look under the “Allow Access” section to see how to fix this. Run:

- semanage fcontext -a -t FILE_TYPE “/my(/.*)?”

To find out the FILE_TYPE, we can just look at another directory we know works with httpd_t, /var/www

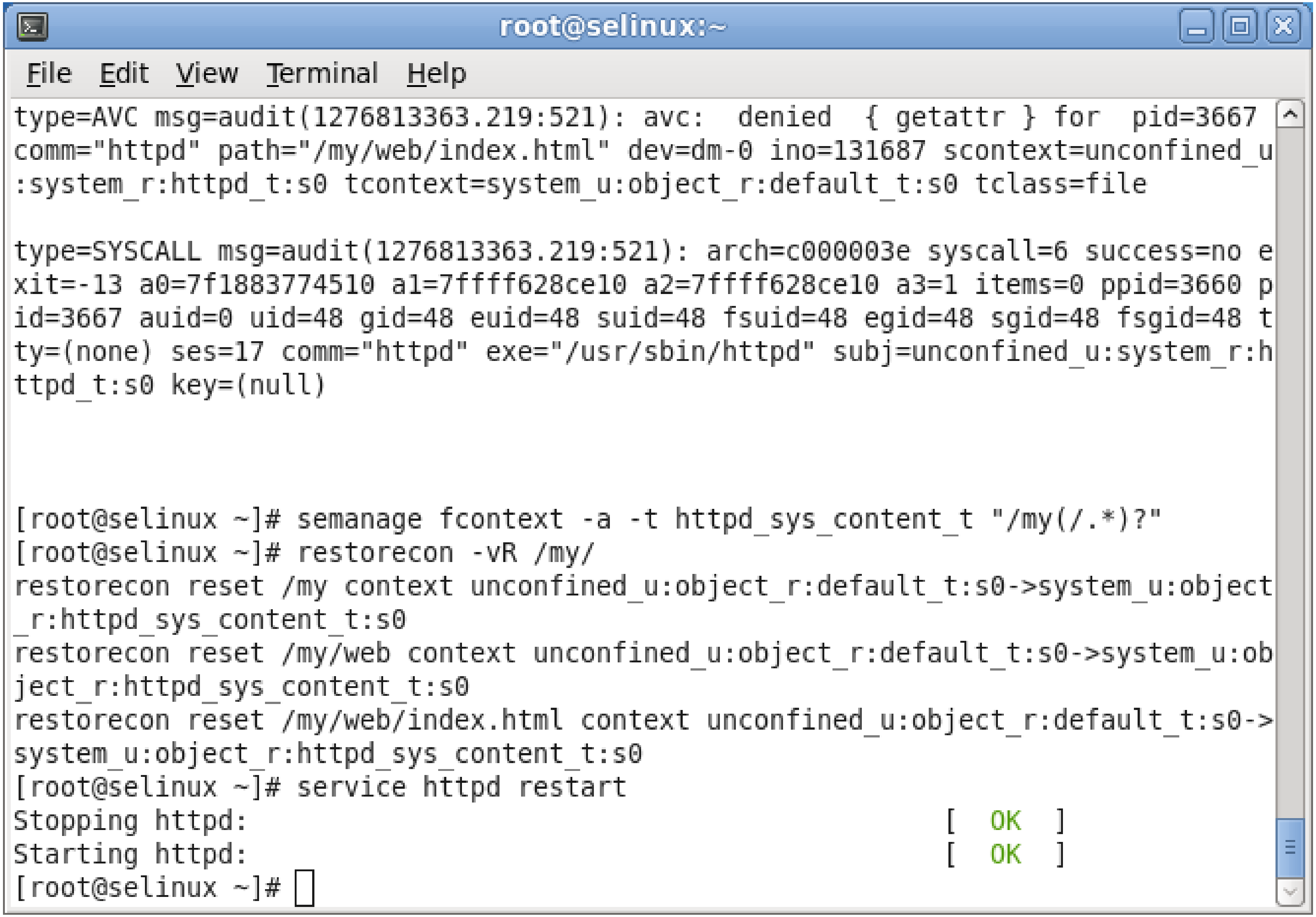


```
                              #1 SMP Wed Jun 16 15:48:48 EDT 2010 x86_64 x86_64
Alert Count                   1
First Seen                    Thu Jun 17 16:08:32 2010
Last Seen                     Thu Jun 17 16:08:32 2010
Local ID                      494cb73e-899c-47c9-ba65-4755b51ec503
Line Numbers                  18, 19

Raw Audit Messages

type=AVC msg=audit(1276808912.722:307): avc:  denied  { read } for  pid=3141 com
m="httpd" name="mls" dev=selinuxfs ino=12 scontext=unconfined_u:system_r:httpd_t
:s0 tcontext=system_u:object_r:security_t:s0 tclass=file

type=SYSCALL msg=audit(1276808912.722:307): arch=c000003e syscall=2 success=yes
exit=4294967424 a0=7fffa931e420 a1=0 a2=7fffa931e42c a3=fffffff8 items=0 ppid=31
40 pid=3141 auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=p
ts0 ses=17 comm="httpd" exe="/usr/sbin/httpd" subj=unconfined_u:system_r:httpd_t
:s0 key=(null)



[root@selinux conf]# ls -ldZ /var/www/
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /var/www/
[root@selinux conf]# 
```

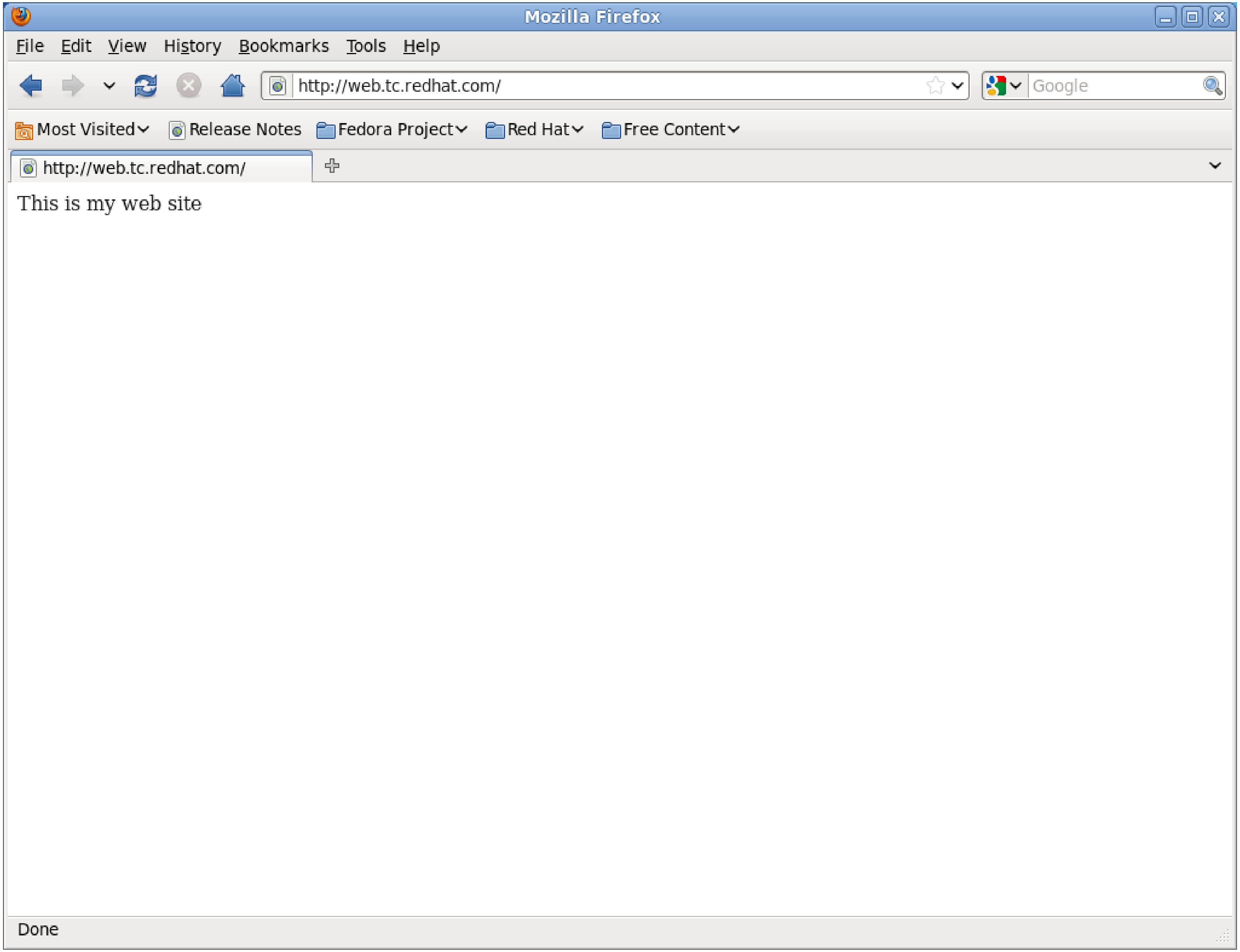
SELinux Examples

Now we define the filesystem context with the command “semanage fcontext -a -t httpd_sys_content_t /my(/.*)?” - remember we are just updating the definition of the file context under /etc/selinux. That way if the filesystem gets relabeled, the context will be set correctly.

Afterwards, we need to actually set the context of the directory with chcon or restorecon



```
type=AVC msg=audit(1276813363.219:521): avc:  denied  { getattr } for  pid=3667
comm="httpd" path="/my/web/index.html" dev=dm-0 ino=131687 scontext=unconfined_u
:system_r:httpd_t:s0 tcontext=system_u:object_r:default_t:s0 tclass=file

type=SYSCALL msg=audit(1276813363.219:521): arch=c000003e syscall=6 success=no e
xit=-13 a0=7f1883774510 a1=7ffff628ce10 a2=7ffff628ce10 a3=1 items=0 ppid=3660 p
id=3667 auid=0 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 t
ty=(none) ses=17 comm="httpd" exe="/usr/sbin/httpd" subj=unconfined_u:system_r:h
ttpd_t:s0 key=(null)



[root@selinux ~]# semanage fcontext -a -t httpd_sys_content_t "/my(/.*)?"
[root@selinux ~]# restorecon -vR /my/
restorecon reset /my context unconfined_u:object_r:default_t:s0->system_u:object
_r:httpd_sys_content_t:s0
restorecon reset /my/web context unconfined_u:object_r:default_t:s0->system_u:ob
ject_r:httpd_sys_content_t:s0
restorecon reset /my/web/index.html context unconfined_u:object_r:default_t:s0->
system_u:object_r:httpd_sys_content_t:s0
[root@selinux ~]# service httpd restart
Stopping httpd:                                            [  OK  ]
Starting httpd:                                            [  OK  ]
[root@selinux ~]# 
```

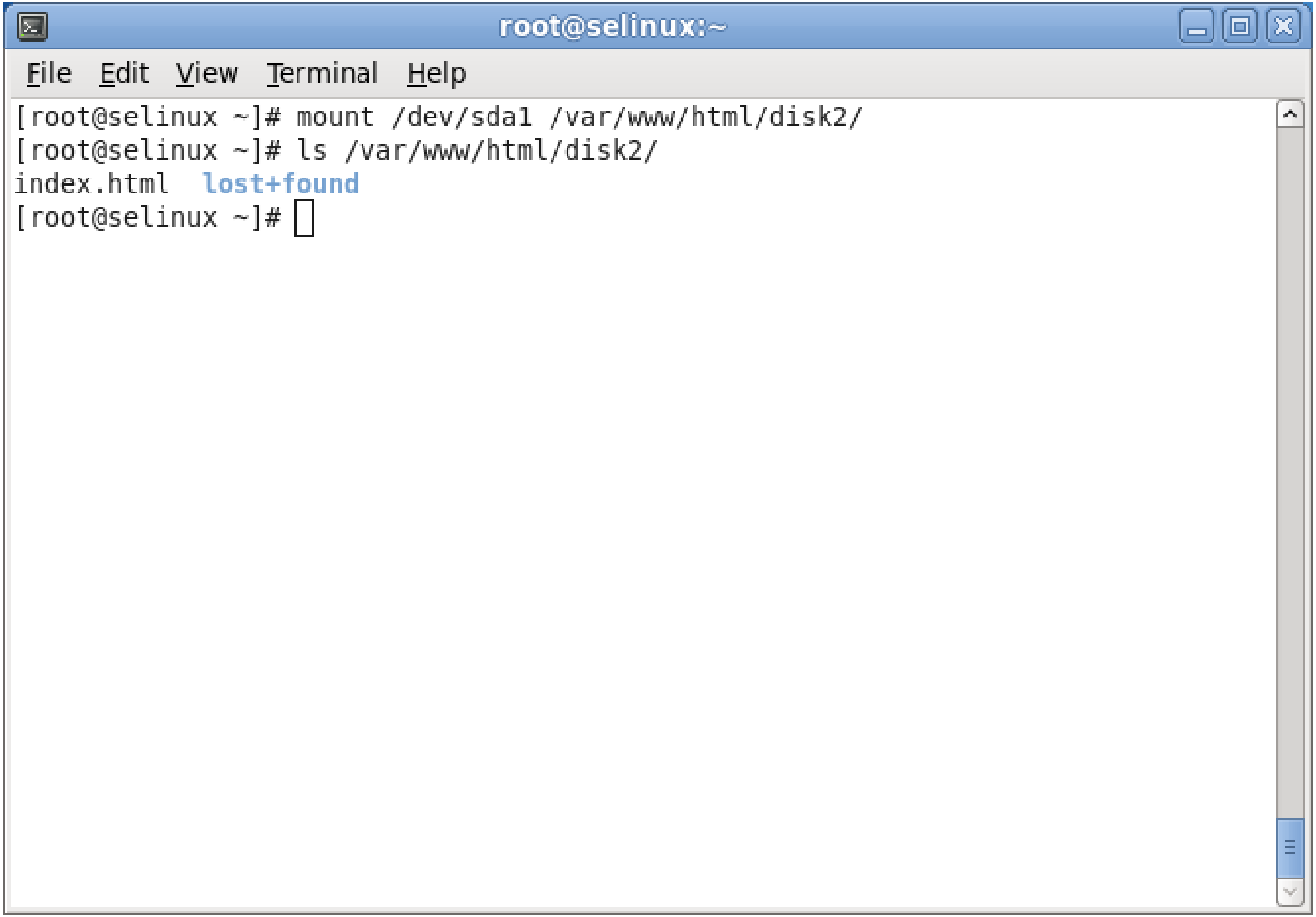


Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://web.tc.redhat.com/
Google
Most Visited
Release Notes
Fedora Project
Red Hat
Free Content
http://web.tc.redhat.com/

This is my web site

SELinux Examples

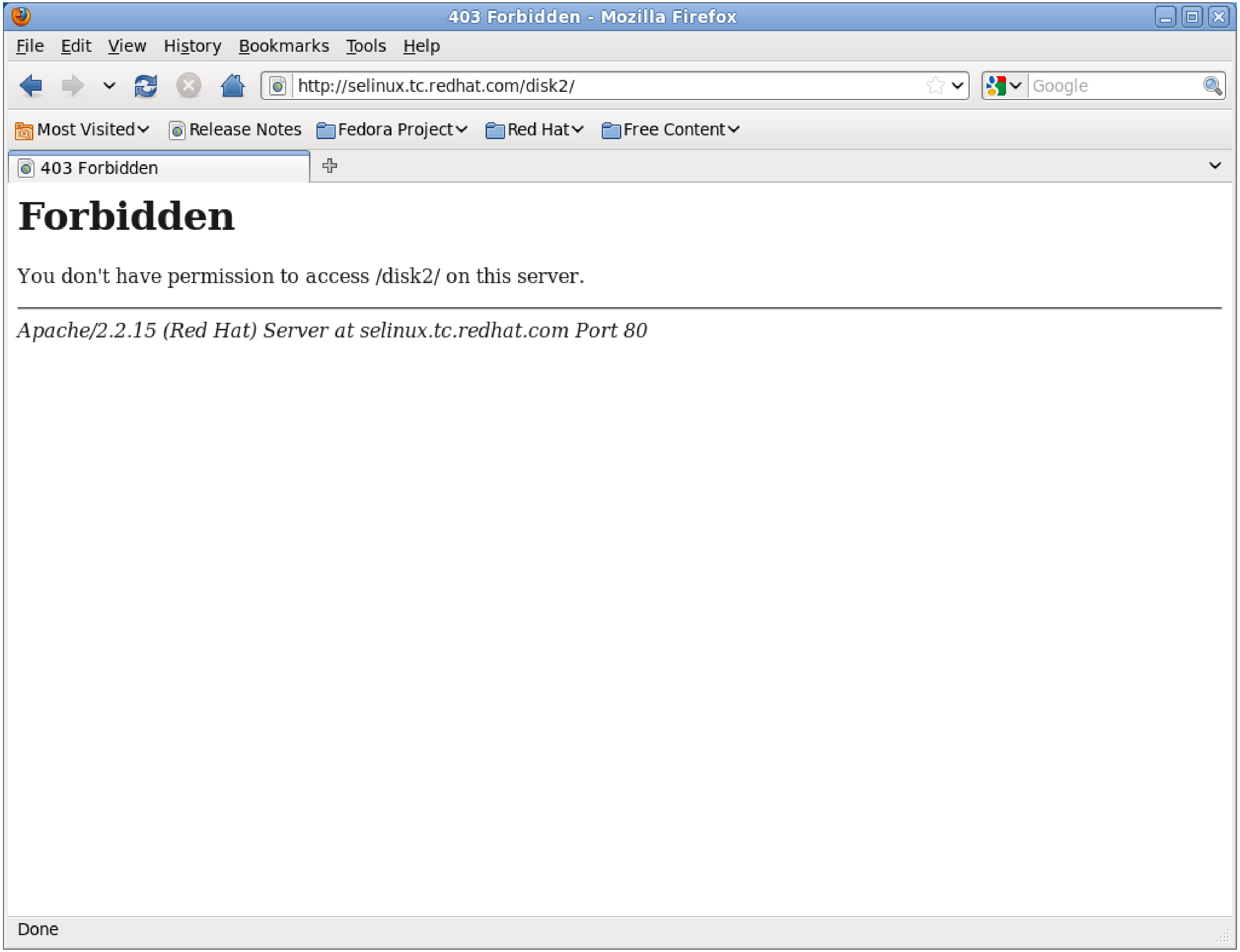
Mount a drive (USB, ISO file) under /var/www to share its content.



```
[root@selinux ~]# mount /dev/sda1 /var/www/html/disk2/
[root@selinux ~]# ls /var/www/html/disk2/
index.html  lost+found
[root@selinux ~]# 
```

SELinux Examples

Now try to view the contents via web browser



403 Forbidden - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://selinux.tc.redhat.com/disk2/
Google
Most Visited Release Notes Fedora Project Red Hat Free Content
403 Forbidden

Forbidden

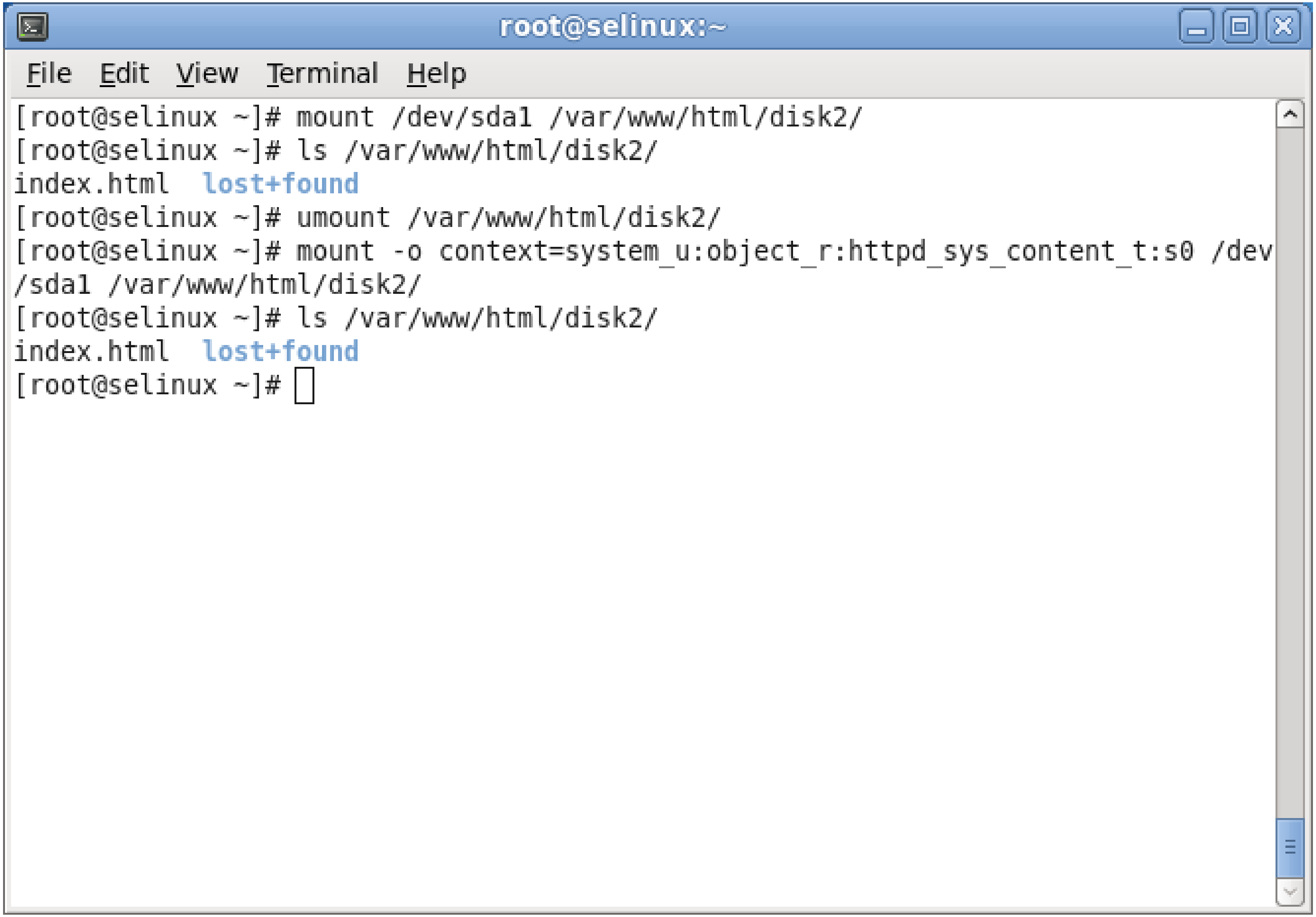
You don't have permission to access /disk2/ on this server.

Apache/2.2.15 (Red Hat) Server at selinux.tc.redhat.com Port 80

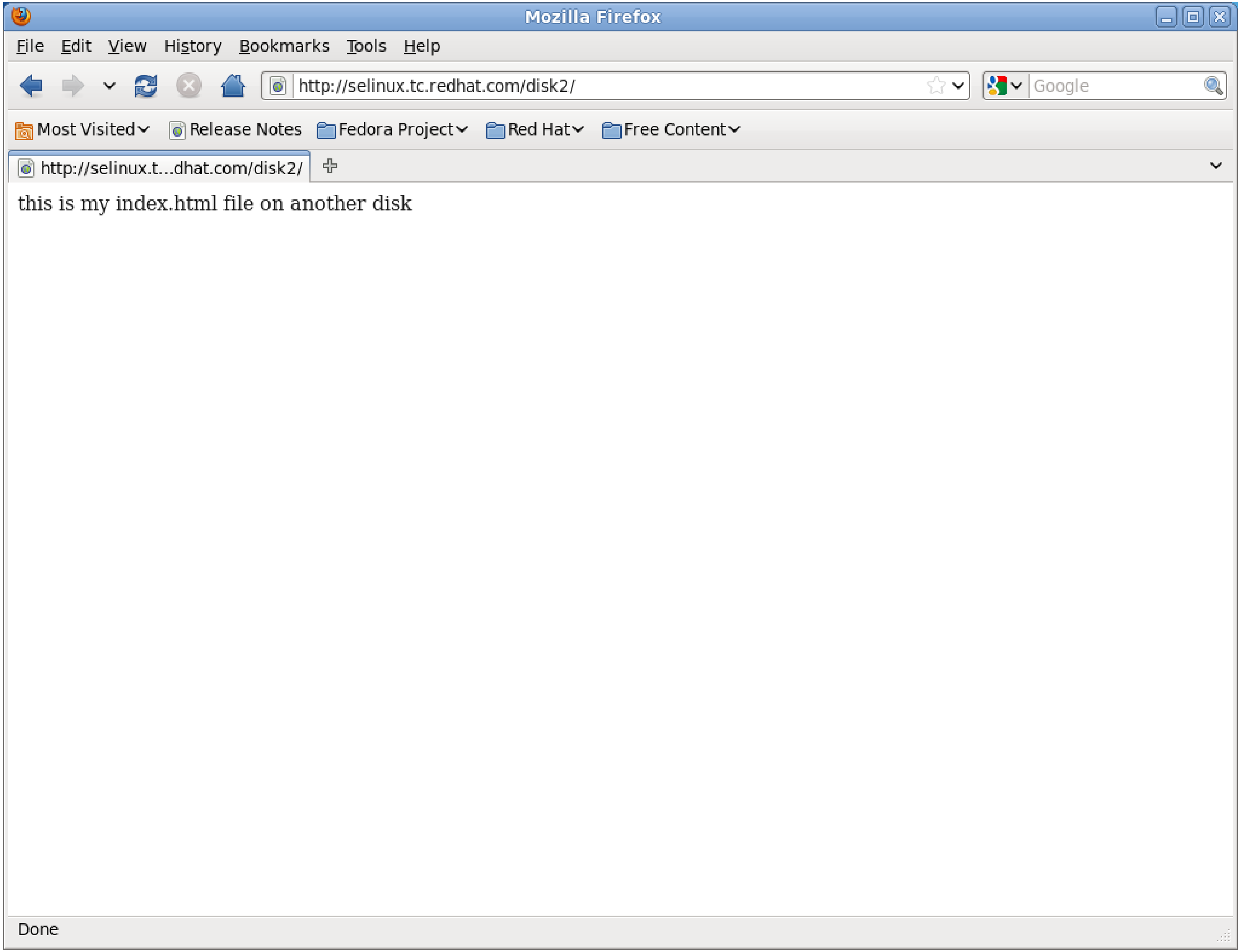
SELinux Examples

You can relabel the filesystem with restorecon, since it's writable media, but do you want the context permanently changed? What if it's an ISO file or other read-only media?

Instead, mount it with the -o context argument:



```
[root@selinux ~]# mount /dev/sda1 /var/www/html/disk2/
[root@selinux ~]# ls /var/www/html/disk2/
index.html  lost+found
[root@selinux ~]# umount /var/www/html/disk2/
[root@selinux ~]# mount -o context=system_u:object_r:httpd_sys_content_t:s0 /dev
/sda1 /var/www/html/disk2/
[root@selinux ~]# ls /var/www/html/disk2/
index.html  lost+found
[root@selinux ~]# 
```



Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://selinux.tc.redhat.com/disk2/
Google
Most Visited
Release Notes
Fedora Project
Red Hat
Free Content
http://selinux.t...dhat.com/disk2/

this is my index.html file on another disk

Creating Basic Policies

audit2why and audit2allow are two utlities 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!

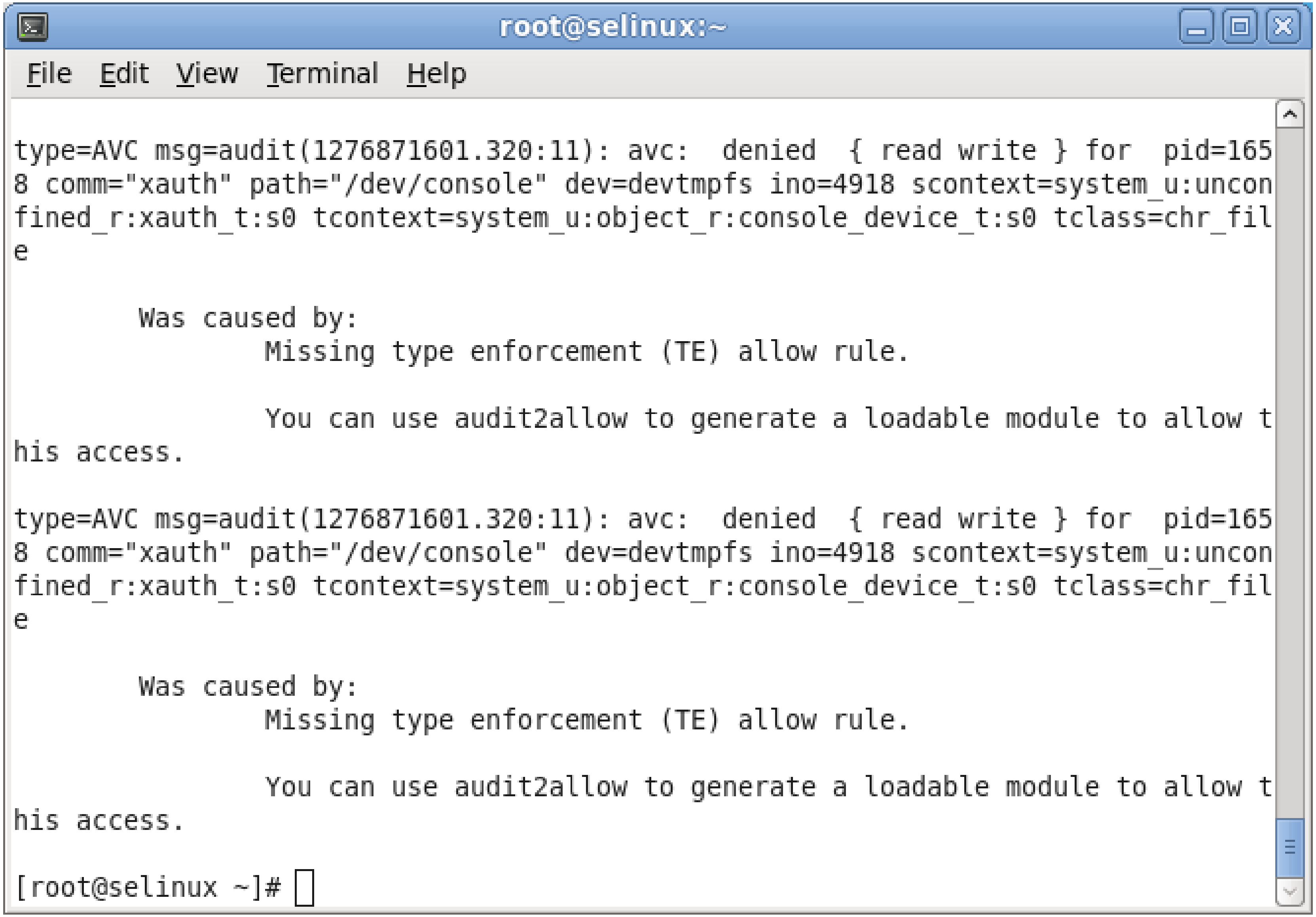
Creating Basic Policies

In the following example, xauth is leaking file descriptors and SELinux is blocking it (well, it would be if it didn't have a permissive type).

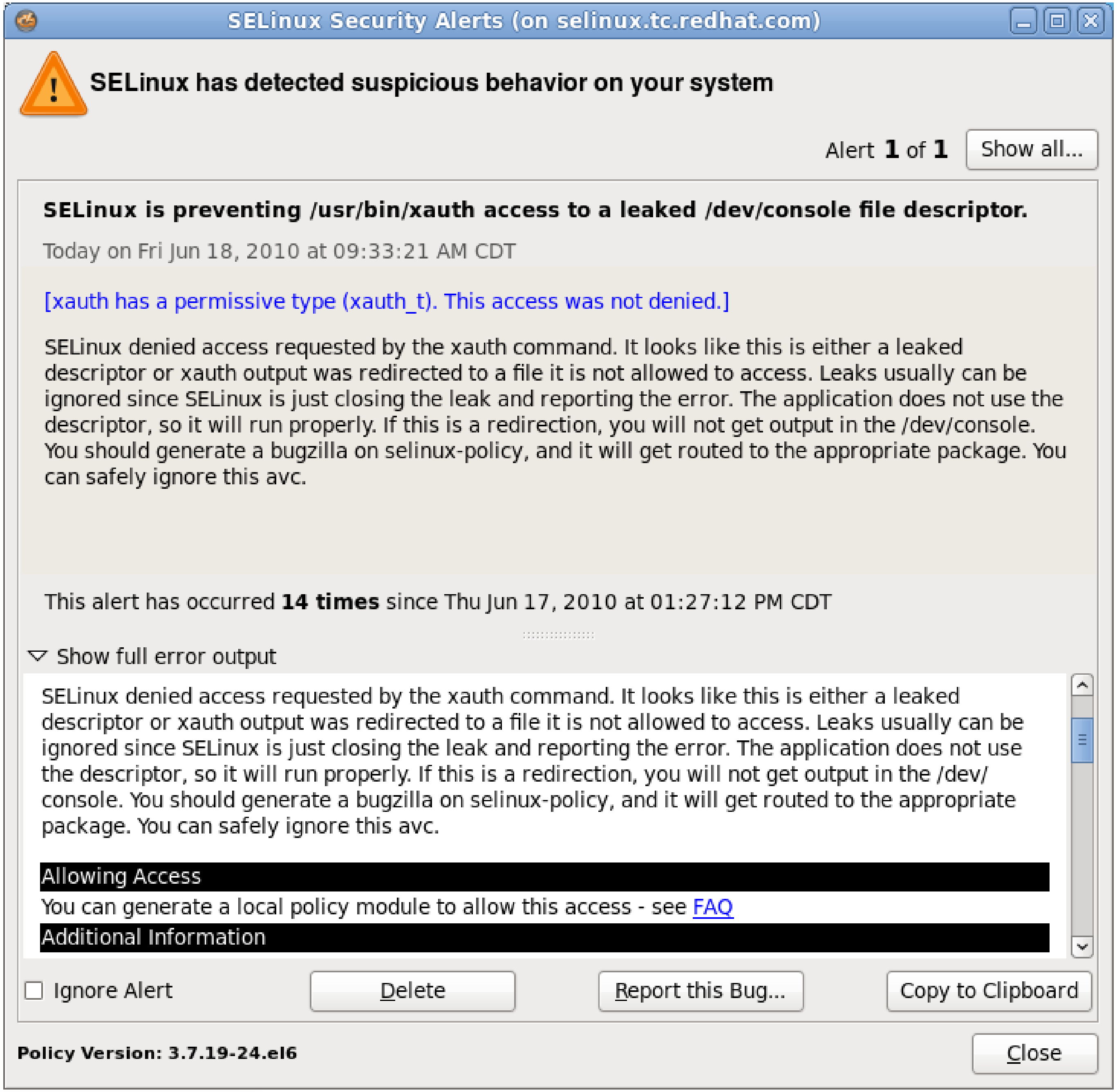
Per MITRE, leaking file descriptors is dangerous - “A process does not close sensitive file descriptors before invoking a child process, which allows the child to perform unauthorized I/O operations using those descriptors.”

Creating Basic Policies

You can use audit2why or sealert -b to see why this was blocked:



```
type=AVC msg=audit(1276871601.320:11): avc:  denied  { read write } for  pid=165
8 comm="xauth" path="/dev/console" dev=devtmpfs ino=4918 scontext=system_u:uncon
fined_r:xauth_t:s0 tcontext=system_u:object_r:console_device_t:s0 tclass=chr_fil
e

        Was caused by:
                Missing type enforcement (TE) allow rule.

                You can use audit2allow to generate a loadable module to allow t
his access.

type=AVC msg=audit(1276871601.320:11): avc:  denied  { read write } for  pid=165
8 comm="xauth" path="/dev/console" dev=devtmpfs ino=4918 scontext=system_u:uncon
fined_r:xauth_t:s0 tcontext=system_u:object_r:console_device_t:s0 tclass=chr_fil
e

        Was caused by:
                Missing type enforcement (TE) allow rule.

                You can use audit2allow to generate a loadable module to allow t
his access.

[root@selinux ~]# 
```



SELinux has detected suspicious behavior on your system

Alert **1** of **1** Show all...

SELinux is preventing /usr/bin/xauth access to a leaked /dev/console file descriptor.

Today on Fri Jun 18, 2010 at 09:33:21 AM CDT

[xauth has a permissive type (xauth_t). This access was not denied.]

SELinux denied access requested by the xauth command. It looks like this is either a leaked descriptor or xauth output was redirected to a file it is not allowed to access. Leaks usually can be ignored since SELinux is just closing the leak and reporting the error. The application does not use the descriptor, so it will run properly. If this is a redirection, you will not get output in the /dev/console. You should generate a bugzilla on selinux-policy, and it will get routed to the appropriate package. You can safely ignore this avc.

This alert has occurred **14 times** since Thu Jun 17, 2010 at 01:27:12 PM CDT

▽ Show full error output

SELinux denied access requested by the xauth command. It looks like this is either a leaked descriptor or xauth output was redirected to a file it is not allowed to access. Leaks usually can be ignored since SELinux is just closing the leak and reporting the error. The application does not use the descriptor, so it will run properly. If this is a redirection, you will not get output in the /dev/console. You should generate a bugzilla on selinux-policy, and it will get routed to the appropriate package. You can safely ignore this avc.

Allowing Access

You can generate a local policy module to allow this access - see FAQ

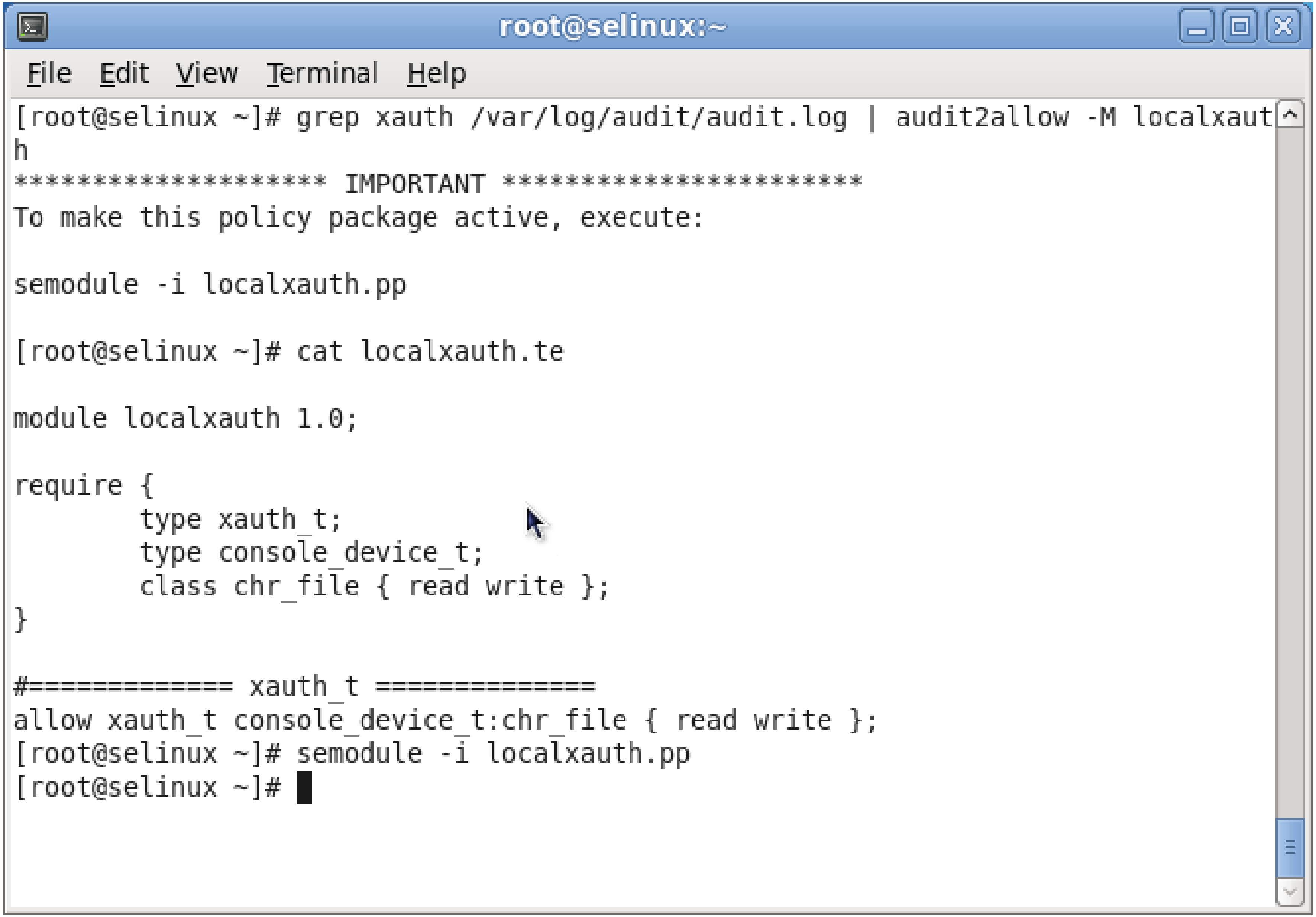
Additional Information

Policy Version: 3.7.19-24.el6

Creating Basic Policies

As indicated in the SE Troubleshoot Browser, you can read the SELinux FAQ at http://bit.ly/8XRSEh for more details about creating policy.

Grab all the xauth entries from /var/log/audit/audit.log and run them against audit2allow and output them to a policy called xauthlocal:



```
[root@selinux ~]# grep xauth /var/log/audit/audit.log | audit2allow -M localxaut
h
******************** IMPORTANT ***********************
To make this policy package active, execute:

semodule -i localxauth.pp

[root@selinux ~]# cat localxauth.te

module localxauth 1.0;

require {
        type xauth_t;
        type console_device_t;
        class chr_file { read write };
}

#============= xauth_t ==============
allow xauth_t console_device_t:chr_file { read write };
[root@selinux ~]# semodule -i localxauth.pp
[root@selinux ~]# █
```

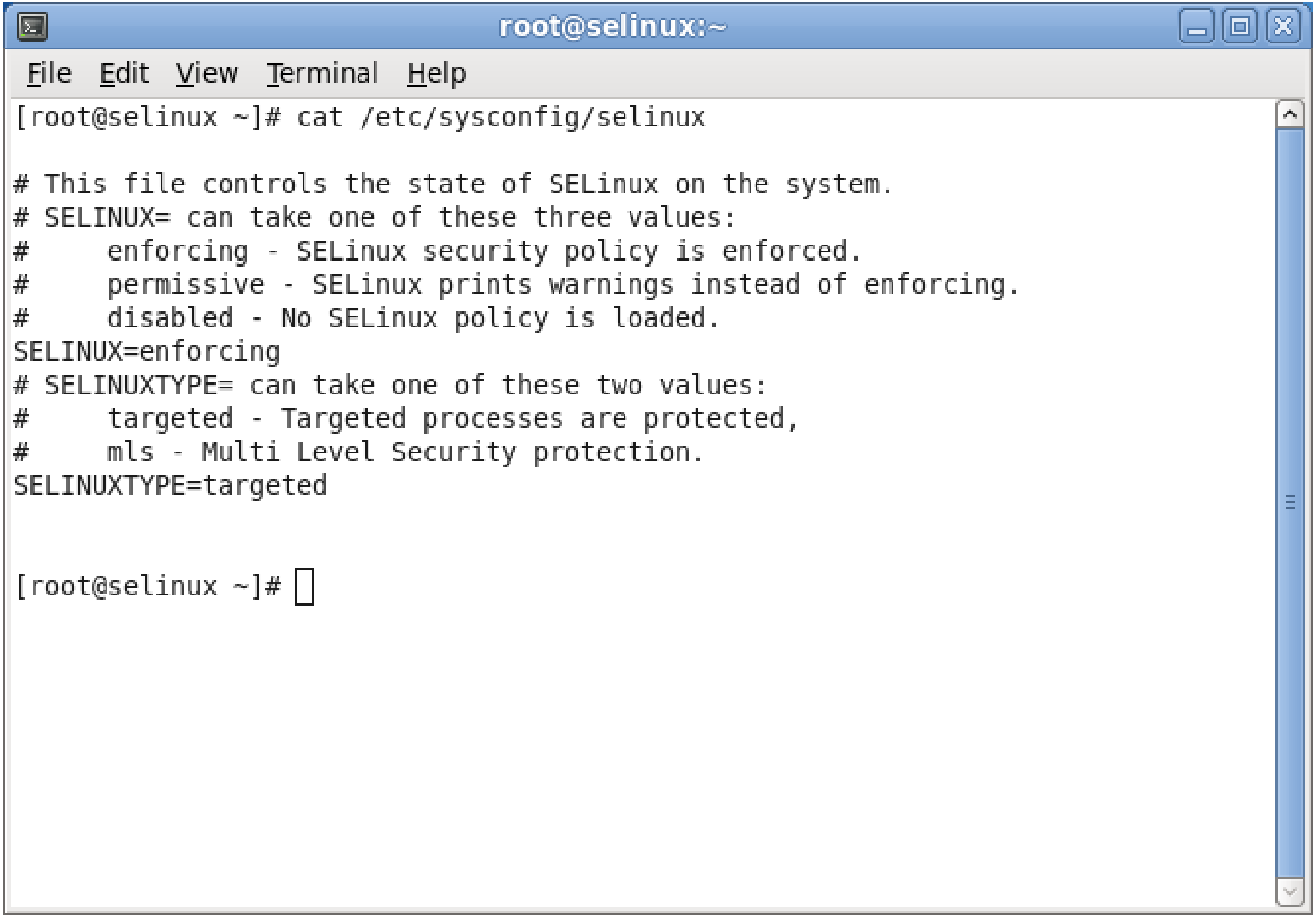
Creating Basic Policies

Now SELinux will allow the leaked descriptors. This method can be used to allow anything that SELinux is blocking.

BE CAREFUL. UNDERSTAND WHAT YOU'RE DOING BEFORE YOU ALLOW BLOCKED ACCESS!

Activating SELinux

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

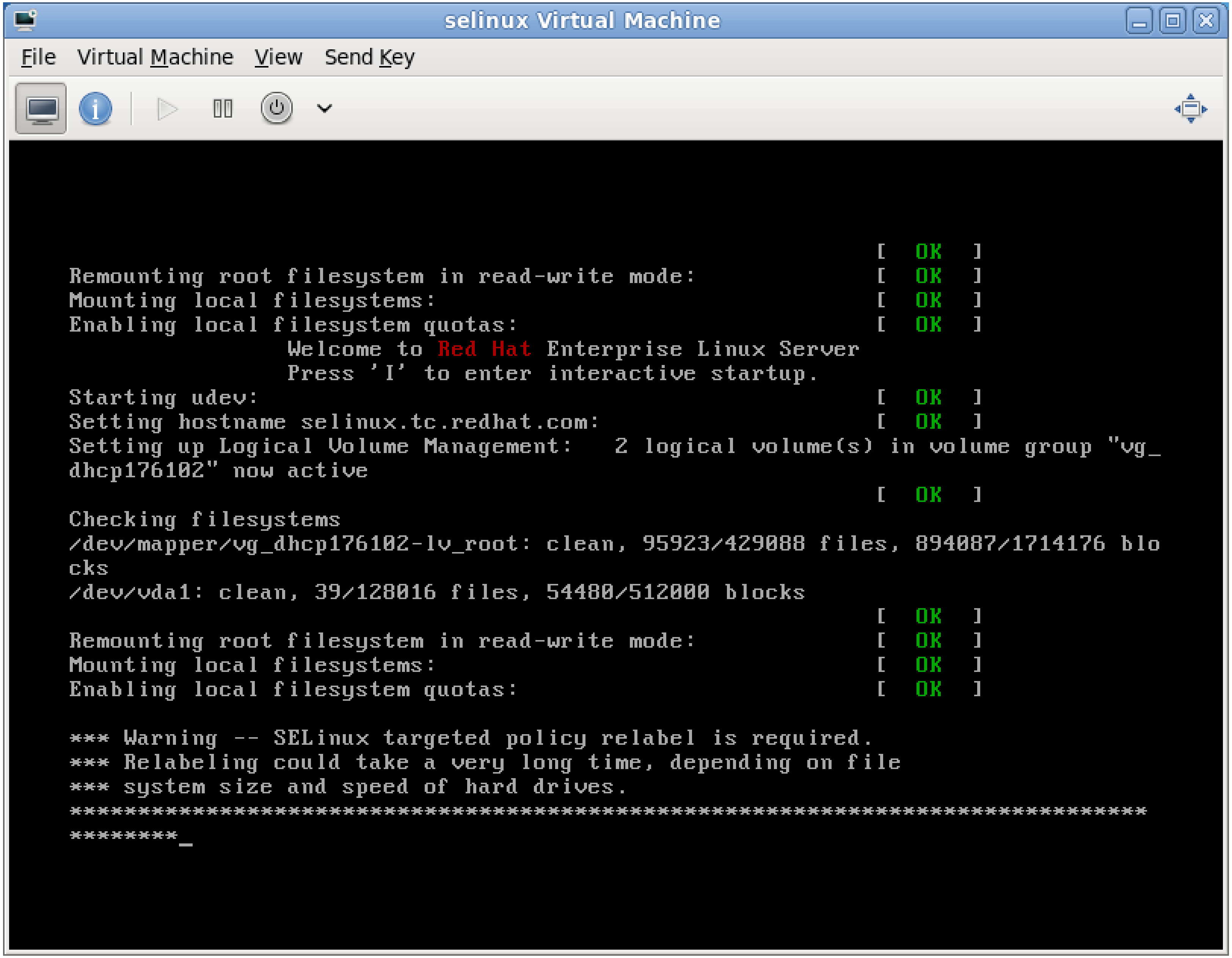


```
[root@selinux ~]# cat /etc/sysconfig/selinux

# 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 - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
#      targeted - Targeted processes are protected,
#      mls - Multi Level Security protection.
SELINUXTYPE=targeted


[root@selinux ~]# 
```

Activating SELinux

To activate SELinux on your machine, there are a couple of ways to do it.

- Set SELINUX=enforcing in /etc/sysconfig/selinux
- touch /.autorelabel
- reboot

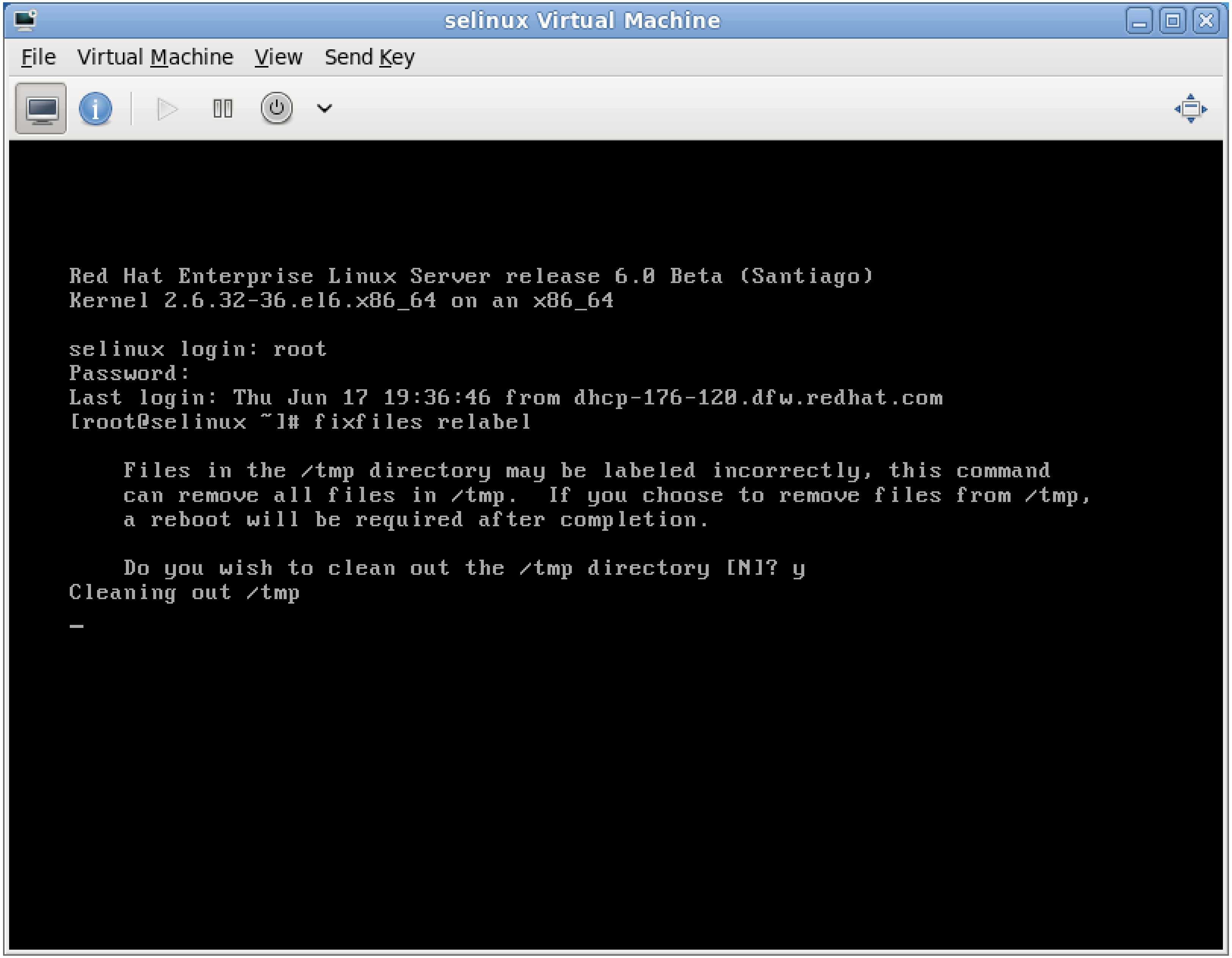


```
                                                           [  OK  ]
Remounting root filesystem in read-write mode:             [  OK  ]
Mounting local filesystems:                                [  OK  ]
Enabling local filesystem quotas:                          [  OK  ]
                Welcome to Red Hat Enterprise Linux Server
                Press 'I' to enter interactive startup.
Starting udev:                                             [  OK  ]
Setting hostname selinux.tc.redhat.com:                    [  OK  ]
Setting up Logical Volume Management:   2 logical volume(s) in volume group "vg_
dhcp176102" now active
                                                           [  OK  ]
Checking filesystems
/dev/mapper/vg_dhcp176102-lv_root: clean, 95923/429088 files, 894087/1714176 blo
cks
/dev/vda1: clean, 39/128016 files, 54480/512000 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.
********************************************************************************
********_
```

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



```
Red Hat Enterprise Linux Server release 6.0 Beta (Santiago)
Kernel 2.6.32-36.el6.x86_64 on an x86_64

selinux login: root
Password:
Last login: Thu Jun 17 19:36:46 from dhcp-176-120.dfw.redhat.com
[root@selinux ~]# 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

_
```

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.

Reporting Errors

Please note – if you are getting denials, it means **there is something wrong!**

It's either a configuration issue, which is fairly straight forward, or a problem with code, which **needs to be reported**, or a problem with SELinux policy, which **needs to be reported.**

Please file bug reports! If it's a configuration issue, we'll tell you how to fix it. If it's a code issue, we'll fix it (patches cheerfully accepted).

http://bugzilla.redhat.com/

How Thomas Feels (And Hopefully You Feel) Now

DARN IT! I'M TIRED OF BEING A SKINNY SCARECROW. CHARLES ATLAS SAYS HE CAN MAKE ME A NEW MAN! I'LL GAMBLE A STAMP AND GET HIS FREE BOOK.
LATER
BOY! IT DIDN'T TAKE LONG. WHAT A BUILD. NOW I'LL TAKE CARE OF THAT BULLY.
HERE'S A LOVE-TAP--- FROM THAT "BAG OF BONES," REMEMBER?
OH, JOE! YOU ARE A REAL HE-MAN, AFTER ALL.
WHAT A MAN
AND HE USED TO BE SO SKINNY!

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!

More Information

Section 44 of the RHEL Deployment Guide:

- http://www.redhat.com/docs/manuals/enterprise/

Fedora Project SELinux Documentation:

- http://fedoraproject.org/wiki/SELinux

fedora-selinux-list (mailing list):

- https://www.redhat.com/mailman/listinfo

Red Hat Training - Red Hat Enterprise SELinux Policy Administration:

- http://bit.ly/aoRDyr

Thank You!

If you enjoyed today's presentation, please let us know!

You can follow up with us:

Thomas - thomas@redhat.com, choirboy on Freenode and thomasdcameron on Twitter.

Dan - dwalsh@redhat.com