



Introduction

RHS429: SELinux Policy Administration

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.



-
- The contents of this course and all its modules and related materials, including handouts to audience members, are Copyright © 2010 Red Hat, Inc.
 - No part of this publication may be stored in a retrieval system, transmitted or reproduced in any way, including, but not limited to, photocopy, photograph, magnetic, electronic or other record, without the prior written permission of Red Hat, Inc.
 - This instructional program, including all material provided herein, is supplied without any guarantees from Red Hat, Inc. Red Hat, Inc. assumes no liability for damages or legal action arising from the use or misuse of contents or details contained herein.
 - If you believe Red Hat training materials are being used, copied, or otherwise improperly distributed please email training@redhat.com or phone toll-free (USA) +1 866 626 2994 or +1 919 754 3700.

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email training@redhat.com or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.



- By nature, available to the world
 - Including crackers!
- SELinux protections include:
 - Special security contexts for different types of files
 - Configuration files
 - Log files
 - Binaries
 - Web content
 - Ability to disable Apache features through SELinux

One of the most important services protected by the targeted policy is the Apache web server. Web servers by their nature are open to the world and consequently, they present a conveniently open port for crackers to attempt an attack. Furthermore, they present to the public multiple methods of manipulating data, from reading data, to running commands (CGIs) on the client to running commands on the server (server side includes).

As a consequence, the web server is protected by SELinux, and also the most configurable of the targeted services. SELinux protections include special security contexts for different various types of files. Indeed multiple security contexts are used for SELinux depending nature of the data being manipulated.

Configuration files, log files, binary files, and content each have their own SELinux security context. As you will find, the startup scripts also have their own file type:

Files	Security Contexts
<code>/usr/sbin/httpd</code>	<code>system_u:object_r:httpd_exec_t</code>
<code>/etc/httpd/conf/*</code>	<code>system_u:object_r:httpd_conf_t</code>
<code>/var/log/httpd/*</code>	<code>system_u:object_r:httpd_log_t</code>
<code>/var/www/html/*</code>	<code>system_u:object_r:httpd_sys_content_t</code>
<code>/var/www/cgi-bin/*</code>	<code>system_u:object_r:httpd_sys_script_exec_t</code>
<code>/etc/init.d/httpd</code>	<code>system_u:object_r:initrc_exec_t</code>

These are only the most common security contexts used by Apache.

Peruse the `/etc/selinux/targeted/context/files/file_contexts` file to identify other contexts used by the web server.

SELinux also provides configuration booleans: configuration items that turn on or off the ability of the web server to perform certain types of actions, regardless of the configuration of the server itself.



- Web content must be set to a special context
- Including web content in the users' home directory!
 - By default, the `httpd_enable_homedirs` boolean is enabled
 - By default, **restorecon** knows about `~/public_html`, `~/www` and `~/web`
 - By default, **restorecond** is running and will set the context on these directories accordingly (as long as the user is logged in)

Transitioning to SELinux involves setting the proper context for web content. Assuming web content is in the default location, `/var/www/html`, the update or the installation should have set the files to the proper context: `system_u:object_r:httpd_sys_content_t`. However, it may be necessary to reset the context for the tree if it is in an unexpected location, if it is added later, or if the content is moved from some other location. This can be accomplished with the **chcon** command. For example, to set the contents of the `/apache/content/` directory to the proper context, you can run either one of the commands below:

```
[root@stationX ~]# chcon -R system_u:object_r:httpd_sys_content_t /apache/content
```

(OR)

```
[root@stationX ~]# chcon -R --reference=/var/www/html /apache/content
```

Alternately, you could set the context when creating the parent directory:

```
[root@stationX ~]# install -d -o root -g root -m 755 --context=system_u:object_r:httpd_sys_content_t:s0 /apache/
```

(OR)

```
[root@stationX ~]# mkdir --context system_u:object_r:httpd_sys_content_t /apache/
```

When running the web server with the `UserDir` function enabled, allowing URLs such as `http://server1.example.com/~digby`, the user's content must be set to the security context `system_u:object_r:httpd_user_content_t`. Note that this is different from the security context of files in `/var/www`, however, `httpd_user_content_t` is an alias for `httpd_sys_content_t`. Inevitably, this will cause confusion to users who create their own web content, as it is unlikely that the average user will understand, or even know about, SELinux file contexts. To ease the transition, **restorecond** will set the security contexts for your users. We will cover this topic in a later unit.



- Again, by nature, diverse in the types of content deliverable
- Can enable or disable a number of Apache features within SELinux
 - CGI support
 - *~username* support
 - Server side includes (SSI) support
- Must still configure support for the above in Apache

By its nature, Apache has many forms of content that is deliverable. Each content type must be configured in Apache and the content must be put in place. But the ability to serve certain types of content must also be set in SELinux. These are the Apaches special SELinux configuration booleans. These booleans all default to being enabled. They include:

`httpd_enable_cgi`

CGIs execute a command on the server. This is an inherently dangerous action, particularly if combined with input from the users.

`httpd_ssi_exec`

Server Side Includes (SSIs), similar to CGIs, are instructions embedded within a web page to execute a command on the server. As with CGIs, these can be risky elements in a web page.

`httpd_enable_homedirs`

This permits the use of the *~username* feature of the web server, allowing all users to create their own content in a subdirectory of their home directory.

`httpd_tty_comm`

This permits the web server to communicate with the console.

`httpd_unified`

This treats all httpd files in a unified manner. Typically, you will want this turned on.

Turn off any booleans for features that you are not using. And should you decide to configure one or more of these features on your web server, remember to configure both Apache and SELinux.



- Most top-level directories (and many lower level directories) have their own security contexts
- Resetting security contexts for an entire directory tree
 - Use **restorecon**, **fixfiles** or **setfiles**
 - Be wary of **chcon**

Most system directories have a special security context for themselves and their children. It is important to understand, though, that there are wide variations within directories of the security contexts for the children. The `/etc` directory is only the most obvious example. The security context for the directory is `system_u:object_r:etc_t`, but, as we have seen, many files within this directory have security contexts dependent on the item being configured, rather than the directory default. Similarly, the files in home directories have security contexts of `user_u:object_r:user_home_t` generally, but web content in `~/public_html/` typically will have a different security context. Therefore, it is virtually always dangerous to recursively change the security contexts of entire directory trees unless the content of the directories is narrowly focused. For example, this is a very bad idea:

```
[root@stationX ~]# chcon -R system_u:object_r:etc_t /etc # WRONG!!!
```

When you need to set the security context for an entire directory tree, use the **setfiles** command:

```
[root@stationX ~]# setfiles /etc/selinux/targeted/context/files/file_contexts /etc
```

Optionally, you can use the **restorecon** command:

```
[root@stationX ~]# restorecon -R /etc
```

These commands will evaluate the files in `/etc/` recursively, and set the security contexts to the values identified in the `file_contexts` file.



- Identifying an SELinux Denial
- `avc: denied` messages appear when SELinux disallows an operation
- Check the audit log which is `/var/log/audit/audit.log` if **auditd** is running, otherwise `/var/log/messages`

When SELinux prevents a program from running, we rely heavily on AVC messages to help us figure out what has happened and how to fix the problem.

If the system is not booting up, it is necessary to figure out whether this is a problem caused by SELinux. By turning off SELinux during the boot process, this can be determined. At the grub prompt, go to the kernel line and add `enforcing=0` (turns the system into permissive mode) or boot into rescue environment.

Login problems can be caused by having a mislabeled home directory, or possibly no labels at all. Try to relabel it with **chcon** or **restorecon**.

If an application does not work as expected, again, try to relabel it to see whether it does the job it is supposed to. Look at the `avc: denied` message to figure out what is wrong.

Daemons can have problems running if the label is not correct, again look at `avc: denied` messages, and try to relabel it properly.

When SELinux disallows an operation, a denial message is generated for the audit logs. In Red Hat Enterprise Linux, the audit log is `/var/log/audit/audit.log` if **auditd** is running. Otherwise, the audit messages will be found in `/var/log/messages`. This section explains the format of these log messages.



- Read `avc: denied` messages
- What process is blocked
- What is the target object
- **fixfiles check** [*path ...*]

This presents a brief methodology for troubleshooting problems that your users might have with SELinux.

1. Deciphering the denial message is the first step in troubleshooting.
 - What is the process that is being blocked? You can find its context from the **scontext=** portion of the message.
 - What is the target object? The `path=` and the `tclass=` tell you where and what the object is. You get its context from `tcontext=`. You may need the `ino=` to find an object if its path is not evident. This may happen because SELinux reports the path as relative to the device node `dev=`.
2. Knowing these essential who, what, where, and how questions should help you in determining the why. At this point it may be obvious, such as the `tcontext=` being set to a context that the process clearly should not be writing to. This may point back to troubles in the application or script, or troubles in the type for the subject or object.
3. If you need to analyze the policy further, you can try using the source and target contexts as search parameters with the **apol** tool.
4. If you think the interaction should be allowed and represents a policy bug, file a bug report at <http://bugzilla.redhat.com>.

fixfiles can be used to check the context on a given directory recursively. If a directory is given **fixfiles check** will check each object in the path for incorrect context and print the information to standard output. **fixfiles restore** will actually change the context. If no directory is given as an argument, it checks the entire directory tree starting at `/`. The output is sent to **syslog** and should appear in `/var/log/messages`.



```
audit(1105758604.519:420): avc: denied { getattr } for
pid=5962 comm="httpd" path="/home/auser/public_html"
dev=sdb2 ino=921135 scontext=system_u:system_r:httpd_t:s0
tcontext=user_u:object_r:user_home_t:s0 tclass=dir
```

The following is an example of an error that occurred attempting to access a web page:

```
audit(1105758604.519:420): avc: denied { getattr } for pid=5962
exe=/usr/sbin/httpd path=/home/auser/public_html dev=hdb2 ino=921135
scontext=root:system_r:httpd_t tcontext=user_u:object_r:user_home_t
tclass=dir
```

This is the kernel audit log message pointer. The timestamp consists of a long number, which is the unformatted current time (Epoch time), and a short number, which is the milliseconds, that is, *current_time.milliseconds_past_current_time*. The third number is the serial number, which helps in stitching together the full audit trail from multiple messages. Multiple messages for the same event occur when full audit logging is enabled using an audit daemon, which logs various kernel events.

To translate the Epoch time into a more manageable format, use **date -d @EPOCH_TIME**.

```
scontext=root:system_r:httpd_t
```

The security context of the source, that is, the process being denied access.

```
tcontext=user_u:object_r:user_home_t
```

The security context of the target, that is, the file or directory that is denied.

```
tclass=dir
```

The object class of the target, indicating that it was the directory `/home/auser/public_html/` that was being blocked.



- Included in the `setroubleshoot` RPM
- Watches for AVC errors and logs them in `/var/log/messages`
- **sealert** connects to the **setroubleshootd** daemon
- Provides information on the error and possible resolution

The **setroubleshootd** daemon is part of the `setroubleshoot` RPM. **setroubleshootd** connects to **auditd** using a UNIX domain socket and watches for AVC errors. **setroubleshootd** sends an alert to `/var/log/messages`. These alerts look something like the following:

```
setroubleshoot: SELinux is preventing the /usr/sbin/httpd from using
potentially mislabeled files (/var/www/html/file). For complete SELinux
messages. run sealert -l 977d2339-1d14-46cc-b1f5-ac7d8d2f6db0
```

Running the **sealert** command above gives:

Summary

```
SELinux is preventing the /usr/sbin/httpd from using potentially
mislabeled
files (/var/www/html/file).
```

Detailed Description

```
SELinux has denied /usr/sbin/httpd access to potentially mislabeled
file(s)
(/var/www/html/file). This means that SELinux will not allow
/usr/sbin/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 /usr/sbin/httpd to access this files, you need to relabel
them
using restorecon -v /var/www/html/file. You might want to relabel the
entire
directory using restorecon -R -v /var/www/html.
```

Additional Information

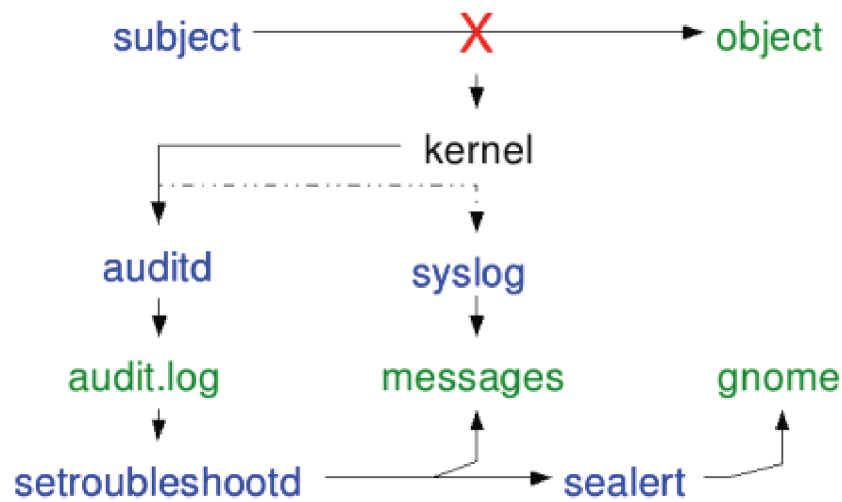
```
Source Context      root:system_r:httpd_t
Target Context      root:object_r:tmp_t
Target Objects      /var/www/html/file [ file ]
Affected RPM Packages httpd-2.2.3-6.el5 [application]
Policy RPM          selinux-policy-2.4.6-30.el5
Selinux Enabled     True
Policy Type         targeted
MLS Enabled         True
```

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

```
Enforcing Mode      Enforcing
Plugin Name         plugins.home_tmp_bad_labels
Host Name           station1.example.com
Platform            Linux station1.example.com 2.6.18-8.el5 #1
                   SMP Thu Jul 26 14:15:21 EST 2007 i686 i686
Alert Count         1
Line Numbers
```

Raw Audit Messages

```
avc: denied { getattr } for comm="httpd" dev=dm-0 egid=48 euid=48
exe="/usr/sbin/httpd" exit=-13 fsgid=48 fsuid=48 gid=48 items=0
  name="file"
path="/var/www/html/file" pid=5816 scontext=root:system_r:httpd_t:s0
  sgid=48
subj=root:system_r:httpd_t:s0 suid=48 tclass=file
tcontext=root:object_r:tmp_t:s0 tty=(none) uid=48
```



Flow of information through the logging process:

1. Subject attempts to access an object.
2. Access is denied by the SELinux engine in the kernel.
3. The kernel sends the `avc: denied` message to the audit daemon. If **auditd** is not available, the denial is instead reported to syslog, where it is handled as a `kern` facility.
4. By default **auditd** should be available, and would write an error to `/var/log/audit/audit.log`.
5. If **setroubleshootd** is running, it will intercept the `avc: denied` message, and translate it into an **sealert** description. The **sealert** description will be written to `/var/log/messages`.
6. The **setroubleshootd** process will trigger **sealert**.
7. If the system is running the GNOME desktop, **sealert** will place an alert icon on the panel.

Sequence 3.1: Accessing the student's web page

Scenario: We noted in the last lab that the problems with the web server related to SELinux. Look in `/var/log/audit/audit.log` to find the AVC messages that relate to that problem. In fact, throughout this Lab, you should have a terminal open that is running **`tail -f /var/log/audit/audit.log`**, so you see the messages as they are being generated.

System Setup: Install `setroubleshoot`.

Instructions:

1. Install the `setroubleshoot` rpm. Consult the Appendix for more information on installing software. After it is installed, start the service.
2. Try to access the student web site again. What errors do you see in the logs?
3. Use **`getsebool`** to search for a boolean that may relate to serving web pages from user's home directories. Search the logs or use **`sealert`** again to find more information. Once you find it, enable it.
4. Now access the student web site and view the logs to determine the next problem. Once you have determined the problem, fix it.

Note that **`restorecond`** did not automatically fix this problem because we did not log in directly as the student user when we created the `public_html` directory. We will discuss **`restorecond`** in a later unit.

Sequence 3.3: Accessing the main web page

Scenario: The main web site is also not working. Find and fix the problem.

Instructions:

1. Try to access the main web page. Try to solve this problem by referencing the logs and other files.
2. Install the `rhs429-ts` rpm. Try to execute the cgi script with

```
[root@stationX ~]# links -dump http://stationX/cgi-bin/sel.sh
```

It should not work. Fix the problem. Ensure your fix survives after a reboot.

Sequence 3.1 Solutions

1. Install the `setroubleshoot` rpm. Consult the Appendix for more information on installing software. After it is installed, start the service.

If you can configure `yum` use that to install `setroubleshoot`. Do not forget to start the service.

a. `[root@stationX ~]# yum install -y setroubleshoot`

b. `[root@stationX ~]# service setroubleshoot start`

2. Try to access the student web site again. What errors do you see in the logs? Use `sealert` to view the full details.

a. `[root@stationX ~]# links -dump http://stationX/~student`

```
Forbidden
```

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

- b. `/var/log/messages` should have errors such as the ones below.

```
setroubleshoot: SELinux is preventing the http daemon from reading u
users home directories. For complete SELinux messages. run sealert
-l 499f9f30-a097-4912-b07f-1aad361013d9
```

```
setroubleshoot: SELinux is preventing the http daemon from reading u
users home directories. For complete SELinux messages. run sealert
-l fc87bfe2-66d1-4743-b1a9-a82ccc91cd24
```

- c. Now run `sealert` as mentioned in the above log entries. Compare the two entries to find the differences. You may want to redirect the output to a file and use `diff` to compare them.

```
[root@stationX ~]# sealert -l 499f9f30-a097-4912-b07f-1aad361013d9 > /tmp/error1
```

```
[root@stationX ~]# sealert -l fc87bfe2-66d1-4743-b1a9-a82ccc91cd24 > /tmp/error2
```

```
[root@stationX ~]# diff /tmp/error1 /tmp/error2
```

```
21c21
```

```
< Target Objects          student [ dir ]
```

```
---
```

```
> Target Objects          /home/student [ dir ]
```

```
37,39c37,39
```

```
< avc: denied { search } for comm="httpd" dev=sda3 egid=48...
```

```
---
```

```
> avc: denied { getattr } for comm="httpd" dev=sda3 egid=48...
```

Notice that one error was a `search` error, and the other was a `getattr` error. The web server attempted to search the `/home/student/` directory, and then attempted to get the attributes of that directory. Both access attempts were denied by SELinux.

3. Use **getsebool** to search for a boolean that may relate to serving web pages from user's home directories. Search the logs or use **sealert** again to find more information. Once you find it, enable it.

a. `[root@stationX ~]# getsebool -a | grep http`

b. **sealert** shows that the boolean needed is `httpd_enable_homedirs`.

c. Use the following command to explicitly set the boolean:

```
[root@stationX ~]# setsebool httpd_enable_homedirs 1
```

4. Now access the student web site and view the logs to determine the next problem. Once you have determined the problem, fix it.

Note that **restorecond** did not automatically fix this problem because we did not log in directly as the student user when we created the `public_html` directory. We will discuss **restorecond** in a later unit.

a. `[root@stationX ~]# links -dump http://stationX/~student`

b. `[root@stationX ~]# tail -1 /var/log/messages`

```
setroubleshoot: SELinux is preventing the /usr/sbin/httpd from using
potentially mislabeled files (/home/student/public_html). For
complete SELinux messages. run sealert -l
c33191f2-c1e4-415f-9e4b-edf228a1a14c
```

c. `[root@stationX ~]# sealert -l c33191f2-c1e4-415f-9e4b-edf228a1a14c`

Summary

SELinux is preventing the /usr/sbin/httpd from using potentially mislabeled files (/home/student/public_html).

Detailed Description

SELinux has denied /usr/sbin/httpd access to potentially mislabeled file(s) (/home/student/public_html). This means that SELinux will not allow /usr/sbin/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 /usr/sbin/httpd to access this files, you need to relabel them using `restorecon -v /home/student/public_html`. You might want to relabel the entire directory using `restorecon -R -v /home/student`.

- d. As mentioned in the output of the **sealert** command is that the directory and files all have the wrong context. Fix the problem and try to access the web page:

```
[root@stationX ~]# restorecon -R -v /home/student/
restorecon reset /home/student/public_html context
```

```
root:object_r:user_home_t:s0 -> ↵
user_u:object_r:httpd_sys_content_t:s0
restorecon reset /home/student/public_html/index.html context ↵
root:object_r:user_home_t:s0 -> ↵
user_u:object_r:httpd_sys_content_t:s0
[root@stationX ~]# links -dump http://stationX/~student
Student web site
```

Sequence 3.3 Solutions

1. Try to access the main web page. Try to solve this problem by referencing the logs and other files.

a. `[root@stationX ~]# links -dump http://stationX/index.html`
Forbidden

b. `[root@stationX ~]# tail -1 /var/log/messages`
setroubleshoot: SELinux is preventing the /usr/sbin/httpd from using
potentially mislabeled files (/var/www/html/index.html). For
complete SELinux messages. run `sealert -l`
d5771658-00c9-4b44-b951-a2e722b38e17

c. `[root@stationX ~]# sealert -l d5771658-00c9-4b44-b951-a2e722b38e17`
Summary

SELinux is preventing the /usr/sbin/httpd from using
potentially mislabeled files (/var/www/html/index.html).

Detailed Description

SELinux has denied /usr/sbin/httpd access to potentially
mislabeled file(s) (/var/www/html/index.html). This means that
SELinux will not allow /usr/sbin/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 /usr/sbin/httpd to access this files, you need
to relabel them using `restorecon -v /var/www/html/index.html`.
You might want to relabel the entire directory using
`restorecon -R -v /var/www/html`.

...

- d. The log files and **sealert** should show that the `index.html` file has the wrong security contexts (`tmp_t`). If you know the proper security contexts, simply set the proper type:

```
[root@stationX ~]# cd /var/www/html/  
[root@stationX html]# chcon -t httpd_sys_content_t index.html
```

If you have forgotten the type, there may be other files or directories that have the proper type. In fact, if you create a new file in `/var/www/html/`, it should get the proper type. You can then use that file as a reference point:

```
[root@stationX ~]# cd /var/www/html  
[root@stationX html]# touch tempfile  
[root@stationX html]# chcon --reference tempfile index.html
```

or simply use **restorecon**:

```
[root@stationX ~]# restorecon -vR /var/www/html/
```

Test the web site again and verify that this fixes the problem. If it does not, try running **fixfiles**.

```
[root@stationX ~]# fixfiles check
```

```
[root@stationX ~]# tail -50 /var/log/messages
```

```
[root@stationX ~]# 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]? *Enter*

```
[root@stationX ~]# links -dump http://stationX/index.html
stationX.example.com
```

2. Install the `rhs429-ts` rpm. Try to execute the cgi script with

```
[root@stationX ~]# links -dump http://stationX/cgi-bin/sel.sh
```

It should not work. Fix the problem. Ensure your fix survives after a reboot.

a.

```
[root@stationX ~]# yum install -y rhs429-ts
```

- b. The script installed by the rpm can be fixed in a similar fashion as described above. Enable the **CGI script boolean** and fix the context.

```
[root@stationX ~]# setsebool -P httpd_enable_cgi 1
```

```
[root@stationX ~]# restorecon -R /var/www/
```

```
[root@stationX ~]# reboot
```

```
[root@stationX ~]# links -dump http://stationX/cgi-bin/sel.sh
```