http://www.coincoin.fr.eu.org/?Chronicles-of-SELinux-Dealing-with

Chronicles of SELinux : Dealing with web content in unusual directories

- 6- Webographie -

Date de mise en ligne : jeudi 10 septembre 2015

https://major.io/wp-content/uploads...
[PNG - 63.8Â kio]

I've decided to start a series of posts called "Chronicles of SELinux" where I hope to educate more users on how to handle SELinux denials with finesse rather than simply disabling it entirely . To kick things off, I'll be talking about dealing with web content in the first post.

First steps

If you'd like to follow along, simply hop onto a system running Fedora 21 (or later), CentOS 7 or Red Hat Enterprise Linux 7. We need SELinux in enforcing mode on the host, so be sure to check the status with getenforce. Depending on what getenforce returns, you'll need to make adjustments : *Enforcing* : No adjustments needed you're all set ! *Permissive* : Run setenforce 1 and adjust SELinux configuration file (see below) *Disabled* : Adjust the SELinux configuration file and reboot (see below)

To enable enforcing mode in the SELinux configuration file, edit /etc/selinux/config and ensure your SELINUX line has enforcing :

| # 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

If getenforce returned *Disabled* earlier, you will need to reboot to get SELinux working. Also be sure that the selinux-policy-targeted package is installed and run fixfiles onboot -B to relabel the system on reboot *(thanks to immanetize for the comment*).

Let's install httpd and create a developer user :

| # For Fedora dnf -y install httpd # For CentOS/RHEL yum -y install httpd useradd developer systemctl enable httpd systemctl start httpd

On to the guide !

Hosting content in an unique directory

On Red Hat-based systems, httpd expects to find its content in /var/www/html, but some system administrators prefer to have content stored elsewhere on the system. It could be on a SAN or other remote storage, but it could also just be in a different directory to make things easier for the business.

Let's consider a situation where the web content is hosted from /web/. We can create the directory :

| [root@fedora22]# mkdir -v /web mkdir : created directory '/web'

We can edit /etc/httpd/conf/httpd.conf and set our new DocumentRoot :

| # # DocumentRoot : The directory out of which you will serve your # documents. By default, all requests are taken from this directory, but # symbolic links and aliases may be used to point to other locations. # DocumentRoot "/web"

Let's reload the httpd configuration :

| systemctl reload httpd

And now we can add some amazing web content :

| echo "Amazing web content !" > /web/index.html

It's time to test our web server :

| # curl -i localhost/index.html HTTP/1.1 403 Forbidden Date : Thu, 10 Sep 2015 12:54:19 GMT Server : Apache/2.4.16 (Fedora) Content-Length : 219 Content-Type : text/html ; charset=iso-8859-1

Oh, come on. What's with this 403 error ?

Investigating the 403

The first step for any situation like this is to review some logs. Let's check the logs for httpd :

| [Thu Sep 10 12:55:04.541789 2015] [core:error] [pid 16597] (13)Permission denied : [client ::1:49860] AH00035 : access to /index.html denied (filesystem path '/web/index.html') because search permissions are missing on a component of the path

Search permissions are missing ? What ? Let's check the permissions on our web directory :

| # ls -al /web total 12 drwxr-xr-x. 2 root root 4096 Sep 10 12:53 . dr-xr-xr-x. 19 root root 4096 Sep 10 12:51 .. -rw-râ€”râ€”. 1 root root 21 Sep 10 12:54 index.html

The httpd user has the ability to get into the directory (o+x is set on /web/) and the httpd user can read the file (o+r is set on /web/index.html). Let's check the system journal just in case :

| # journalctl -n 1 | tail â€” Logs begin at Thu 2015-09-10 12:31:37 UTC, end at Thu 2015-09-10 12:55:04 UTC. â€” Sep 10 12:55:04 fedora22 audit[16597] : <audit-1400> avc : denied *getattr* for pid=16597 comm="httpd" path="/web/index.html" dev="xvda1" ino=524290 scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:default_t:s0 tclass=file permissive=0

That's quite a long log line. Let's break it into pieces : avc : denied *getattr* for pid=16597 comm="httpd" : httpd tried to do something and was denied path="/web/index.html" dev="xvda1" ino=524290 : path to the file (index.html) involved

in the denial scontext=system_u:system_r:httpd_t:s0 : the SELinux context of the httpd process tcontext=unconfined_u:object_r:default_t:s0 : the SELinux contect that is actually applied to our index.html tclass=file : the denial came from accessing a file (index.html) permissive=0 : we're in enforcing mode, not permissive mode

Long story short, when httpd tried to access our /web/index.html file, the httpd process was labeled with httpd_t, but the kernel found that the HTML file was labeled with default_t. The httpd process (labeled with httpd_t) isn't allowed to read files that are labeled as default_t, so the access is denied.

Fixing it the right way

Since we know what SELinux expects for this file (from the log line in the journal), we can apply the right context and re-test. The chcon command has a handy argument that allows you to reference a file or directory, and apply the contexts from there. Since we know that /var/www/html has the right contexts already, we can use it as a reference :

| # chcon -v -R â€"reference=/var/www/html /web changing security context of '/web/index.html' changing security context of '/web'

Now we see some different contexts on /web :

| # ls -alZ /web/ total 12 drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 4096 Sep 10 13:19 . dr-xr-xr-x. 19 root root system_u:object_r:root_t:s0 4096 Sep 10 13:19 .. -rw-râ€"râ€". 1 root root system_u:object_r:httpd_sys_content_t:s0 21 Sep 10 13:19 index.html

Let's test again :

| # curl -I localhost/index.html HTTP/1.1 403 Forbidden Date : Thu, 10 Sep 2015 13:21:22 GMT Server : Apache/2.4.16 (Fedora) Content-Type : text/html ; charset=iso-8859-1

Darn ! What's in the httpd logs ?

| [Thu Sep 10 13:21:22.267719 2015] [authz_core:error] [pid 16593] [client ::1:49861] AH01630 : client denied by server configuration : /web/index.html

Ah, we cleared the SELinux problem but now httpd is upset. Just below the DocumentRoot line that we edited earlier, look for two Directory blocks. Change /var/www/ and /var/www/html to /web in those blocks. Reload the httpd configuration and try once more :

| # systemctl reload httpd # curl -I localhost/index.html HTTP/1.1 200 OK Date : Thu, 10 Sep 2015 13:25:16 GMT Server : Apache/2.4.16 (Fedora) Last-Modified : Thu, 10 Sep 2015 13:19:47 GMT ETag : "15-51f6474064d50" Accept-Ranges : bytes Content-Length : 21 Content-Type : text/html ; charset=UTF-8

Success !

Long term fix

The chcon method is good for fixing one-off issues and for testing, but we need a good long term fix. SELinux has some file contexts already configured for certain directories, but not for our custom web directory. You can examine the defaults here :

| # semanage fcontext -l | grep ^/var/www/html /var/www/html(/.*) ?/sites/default/files(/.*) ? all files system_u:object_r:httpd_sys_rw_content_t:s0 /var/www/html(/.*) ?/sites/default/settings\.php regular file system_u:object_r:httpd_sys_rw_content_t:s0 /var/www/html(/.*) ?/uploads(/.*) ? all files system_u:object_r:httpd_sys_rw_content_t:s0 /var/www/html(/.*) ?/wp-content(/.*) ? all files system_u:object_r:httpd_sys_rw_content_t:s0 /var/www/html/[^/]*/cgi-bin(/.*) ? all files system_u:object_r:httpd_sys_script_exec_t:s0 /var/www/html/cgi/munin.* all files system_u:object_r:munin_script_exec_t:s0 /var/www/html/configuration\.php all files system_u:object_r:httpd_sys_rw_content_t:s0 /var/www/html/munin(/.*) ? all files system_u:object_r:munin_content_t:s0 /var/www/html/munin/cgi(/.*) ? all files system_u:object_r:munin_script_exec_t:s0 /var/www/html/owncloud/data(/.*) ? all files system_u:object_r:httpd_sys_rw_content_t:s0

SELinux's tools have a concept of *equivalency*. This allows you to say that one directory is *equivalent* to another one in the long term. We already used chcon to apply contexts with a reference to a directory with valid contexts, but this equivalency concept gives us a longer term fix. Here's the command to use :

| semanage fcontext â€”add â€”equal /var/www /web

If we break this down, we're saying we want to add a new file context where /web is equal to /var/www. This means we want the same SELinux contexts applied in the same places and want them treated equally. After running the semanage command, let's make an index2.html file to test :

| # echo "Amazing web content !" > /web/index2.html # curl -I localhost/index2.html HTTP/1.1 200 OK Date : Thu, 10 Sep 2015 13:35:24 GMT Server : Apache/2.4.16 (Fedora) Last-Modified : Thu, 10 Sep 2015 13:34:11 GMT ETag : "15-51f64a78266c8" Accept-Ranges : bytes Content-Length : 21 Content-Type : text/html ; charset=UTF-8

Great ! We didn't have to use chcon this time around because we configured /web as an equivalent directory to /var/www. Let's double check the contexts :

| # ls -alZ /web total 16 drwxr-xr-x. 2 root root unconfined_u:object_r:httpd_sys_content_t:s0 4096 Sep 10 13:34 . dr-xr-xr-x. 19 root root system_u:object_r:root_t:s0 4096 Sep 10 13:33 .. -rw-râ€”râ€”. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 21 Sep 10 13:34 index2.html -rw-râ€”râ€”. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 21 Sep 10 13:33 index.html

Perfect ! We now have all of the security benefits of SELinux in a completely custom web directory.

The post Chronicles of SELinux : Dealing with web content in unusual directories appeared first on major.io .