System Security

Carles Mateu

Installing a Linux Server

Create a new virtual machine where we'll install a new Linux server. With the *minimum* following specs:

- Memory: 768 Mbytes of RAM (If memory is scarce, 512 MB should do the trick)
- Disk: 8 Gbytes is enough.
- CPU: minimum is enough.
- Network: Any setup with internet access (NAT or bridging). If using NAT all lab should be done on the virtual machine window, if using bridging (on your home network), you can SSH to the virtual machine.

SELinux Security

Enforcing or permissive

Check whether you are on enforcing or permissive mode:

```
[root@host ~]# getenforce
Enforcing
[root@host ~]# setenforce Permissive
[root@host ~]# getenforce
Permissive
[root@host ~]# setenforce Enforcing
[root@host ~]# getenforce
Enforcing
[root@host ~]#
```

Check first permissions

Add a user, and check which permissions it's home dir gets assigned, and how is it different than other directories (as /tmp):

```
[root@host ~]# useradd user
[root@host ~]# ls -Z /home /tmp
/home:
unconfined_u:object_r:user_home_dir_t:s0 user

/tmp:
system_u:object_r:tmp_t:s0 ks-script-3wx92dqy
system_u:object_r:tmp_t:s0 systemd-private-dd6839f9207545c2a86ec2d29cc83d90-chronyd.service-4jMsaw
[root@host ~]#
```

Now we can open a shell with that user, first we give him a password, then we can open a new shell.

```
[root@host ~]# passwd user
S'està canviant la contrasenya de l'usuari user.
Nova contrasenya de :
CONTRASENYA DOLENTA: La contrasenya és inferior als 8 caràcters
Torneu a escriure la nova contrasenya de :
passwd: tots els testimonis d'autenticació s'han actualitzat amb èxit.
[root@host ~]#
[root@host ~]# su - user
[user@host ~]$
```

We can check now that, our processes (as user), have special contexts assigned:

```

[user@host ~]$ ps Z
LABEL                             PID TTY      STAT   TIME COMMAND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 1430 pts/0 S   0:00 -bash
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 1454 pts/0 R+   0:00 ps Z
[user@host ~]$
```

We can see that our desktop user processes are “unconfined”, check all system processes:

```
  [user@host ~]$ ps axZ
  LABEL                              PID TTY      STAT   TIME COMMAND
  system_u:system_r:init_t:s0          1 ?        Ss     0:01 /usr/lib/systemd/systemd --switched-root --
       system --deserialize 17
4 system_u:system_r:kernel_t:s0        2 ?        S      0:00 [kthreadd]
  system_u:system_r:kernel_t:s0        3 ?        I<     0:00 [rcu_gp]
  system_u:system_r:kernel_t:s0        4 ?        I<     0:00 [rcu_par_gp]
  system_u:system_r:kernel_t:s0        6 ?        I<     0:00 [kworker/0:0H-kblockd]
  system_u:system_r:kernel_t:s0        8 ?        I<     0:00 [mm_percpu_wq]
9 system_u:system_r:kernel_t:s0        9 ?        S      0:00 [ksoftirqd/0]
  system_u:system_r:kernel_t:s0       10 ?        R      0:01 [rcu_sched]
  system_u:system_r:kernel_t:s0       11 ?        S      0:00 [migration/0]
  system_u:system_r:kernel_t:s0       12 ?        S      0:00 [watchdog/0]
  system_u:system_r:kernel_t:s0       13 ?        S      0:00 [cpuhp/0]
14 system_u:system_r:kernel_t:s0      15 ?        S      0:00 [kdevtmpfs]
  system_u:system_r:kernel_t:s0       16 ?        I<     0:00 [netns]
  system_u:system_r:kernel_t:s0       17 ?        S      0:00 [kauditd]
  [....]
  system_u:system_r:chronyd_t:s0      711 ?        S      0:00 /usr/sbin/chronyd
19 system_u:system_r:rngd_t:s0        713 ?        Ssl    0:03 /sbin/rngd -f --fill-watermark=0
  system_u:system_r:sssd_t:s0         723 ?        S      0:00 /usr/libexec/sssd/sssd_be --domain
       implicit_files --uid 0 --gid 0 --logg
  system_u:system_r:firewalld_t:s0    726 ?        Ssl    0:00 /usr/libexec/platform-python -s /usr/sbin/
       firewalld --nofork --nopid
  system_u:system_r:sssd_t:s0         727 ?        S      0:00 /usr/libexec/sssd/sssd_nss --uid 0 --gid 0 --
       logger=files
  system_u:system_r:systemd_logind_t:s0 728 ?      Ss     0:00 /usr/lib/systemd/systemd-logind
24 system_u:system_r:NetworkManager_t:s0 736 ?     Ssl    0:00 /usr/sbin/NetworkManager --no-daemon
  system_u:system_r:sshd_t:s0-s0:c0.c1023 745 ?    Ss     0:00 /usr/sbin/sshd -D -oCiphers=aes256-
       gcm@openssh.com,chacha20-poly1305@ope
  system_u:system_r:tuned_t:s0        747 ?        Ssl    0:00 /usr/libexec/platform-python -Es /usr/sbin/
       tuned -l -P
  system_u:system_r:crond_t:s0-s0:c0.c1023 760 ?   Ss     0:00 /usr/sbin/crond -n
  system_u:system_r:local_login_t:s0-s0:c0.c1023 766 ? Ss   0:00 login -- root
29 system_u:system_r:syslogd_t:s0     968 ?        Ssl    0:00 /usr/sbin/rsyslogd -n
  [....]
```

We can see that most daemon processes run on special contexts defined for them, as NetworkManager_t, etc. This allows policies to have very fine granularity on defining what can or cannot a process do.

File contexts

Now we will create a file, as user, on two different places, and we'll see that, albeit it has the same permissions, files inherit their contexts from where they are created:

```
  [user@host ~]$ touch /tmp/temporary
  [user@host ~]$ ls -Zl /tmp/temporary
  -rw-rw-r--. 1 user user unconfined_u:object_r:user_tmp_t:s0 0 14 oct. 05:32 /tmp/temporary
  [user@host ~]$ touch /home/user/homefile
5 [user@host ~]$ ls -Zl /home/user/homefile
  -rw-rw-r--. 1 user user unconfined_u:object_r:user_home_t:s0 0 14 oct. 05:32 /home/user/homefile
  [user@host ~]$
```

If we move the temporary file to our home dir, it keeps the context information:

```
[user@host ~]$ mv /tmp/temporary /home/user/
[user@host ~]$ ls -Zl /home/user/temporary
-rw-rw-r--. 1 user user unconfined_u:object_r:user_tmp_t:s0 0 14 oct. 05:32 /home/user/temporary
```

We can, if allowed to, set the context to what it should be by location:

```
[user@host ~]$ restorecon -Rv /home/user/
Relabeled /home/user/temporary from unconfined_u:object_r:user_tmp_t:s0 to unconfined_u:object_r:
    user_home_t:s0
[user@host ~]$ ls -Zl /home/user/
total 0
-rw-rw-r--. 1 user user unconfined_u:object_r:user_home_t:s0 0 14 oct. 05:32 homefile
-rw-rw-r--. 1 user user unconfined_u:object_r:user_home_t:s0 0 14 oct. 05:32 temporary
```

SELinux Daemon protection

First we'll install a web server (httpd server) and a web client (wget):

```
[root@host ~]# yum -y install vim httpd wget curl
Última comprovació del venciment de les metadades: fa 0:12:06 el dijous, 14 'doctubre de 2021, 06:20:04
    EDT.
El paquet curl-7.61.1-12.el8.x86_64 ja està instal·lat.
[...]
Instal·lat:
  apr-1.6.3-11.el8.x86_64                                  apr-util-1.6.1-6.el8.x86_64
  apr-util-bdb-1.6.1-6.el8.x86_64                          apr-util-openssl-1.6.1-6.el8.x86_64
  centos-logos-httpd-85.8-1.el8.noarch                     gpm-libs-1.20.7-17.el8.x86_64
  httpd-2.4.37-39.module_el8.4.0+778+c970deab.x86_64       httpd-filesystem-2.4.37-39.module_el8
      .4.0+778+c970deab.noarch
  httpd-tools-2.4.37-39.module_el8.4.0+778+c970deab.x86_64 mailcap-2.1.48-3.el8.noarch
  mod_http2-1.15.7-3.module_el8.4.0+778+c970deab.x86_64    vim-common-2:8.0.1763-15.el8.x86_64
  vim-enhanced-2:8.0.1763-15.el8.x86_64                    vim-filesystem-2:8.0.1763-15.el8.noarch
  wget-1.19.5-10.el8.x86_64

S'ha completat!
[root@host ~]#
```

Once installed, we start the web server, and check it:

```
[root@host httpd]# systemctl restart httpd
[root@host httpd]# touch /var/www/html/index.html
[root@host httpd]# wget http://127.0.0.1
--2021-10-14 06:49:52--  http://127.0.0.1/
S'està connectant a …127.0.0.1:80 connectat.
HTTP: s'ha enviat la petició, s'està esperant una …resposta 200 OK
Mida: 0 [text/html]
S'està desant a: «index.html»

index.html                          [ <=>                                                                ]
     0  --.-KB/s    in 0s

2021-10-14 06:49:52 (0,00 B/s) - s'ha desat «index.html» [0/0]
```

SELinux file creation protections

We now download two PHP small programs, a.php and b.php:

```
[root@host httpd]# cd /var/www/html/
[root@host html]# wget http://codeandbeer.org/virtual/SegAC/selinux/b.php
--2021-10-14 06:50:52--  http://codeandbeer.org/virtual/SegAC/selinux/b.php
S'està resolent codeandbeer.org (codeandbeer.org)… 193.144.12.5
[...]
2021-10-14 06:50:52 (8,01 MB/s) - s'ha desat «b.php» [103/103]
[root@host html]# wget http://codeandbeer.org/virtual/SegAC/selinux/a.php
--2021-10-14 06:51:06--  http://codeandbeer.org/virtual/SegAC/selinux/a.php
[...]
2021-10-14 06:51:06 (7,62 MB/s) - s'ha desat «a.php» [102/102]
[root@host html]#
```

If we look at them, we see that a.php opens a file on a subdir on the directory where it resides, writes to it, and then closes it.

```
<?php
$content = "Prova";
$fp = fopen("test/test.txt","wb");
fwrite($fp, $content);
fclose($fp);

?>
```

Whereas b.php does the same on but on a subdirectory of the root filesystem.

```
<?php
$content = "Prova";
$fp = fopen("/test/test.txt","wb");
fwrite($fp, $content);
fclose($fp);

?>
```

Provided that both directories have the right permissions, it should work, don't it?

We start by installing php:

```
[root@host html]# dnf install php
Última comprovació del venciment de les metadades: fa 0:38:40 el dijous, 14 'doctubre de 2021, 06:20:04 EDT.
[...]
  nginx-filesystem-1:1.14.1-9.module_el8.0.0+184+e34fea82.noarch       php-7.2.24-1.module_el8.2.0+313+b04d0a66.x86_64
  php-cli-7.2.24-1.module_el8.2.0+313+b04d0a66.x86_64                  php-common-7.2.24-1.module_el8.2.0+313+b04d0a66.x86_64
  php-fpm-7.2.24-1.module_el8.2.0+313+b04d0a66.x86_64

S'ha completat!
```

And adding a writeable directory, and enabling the webserver to write files.

```
[root@host ~]# cd /var/www/html/
[root@host html]# mkdir test
[root@host html]# chown apache test
[root@host html]# setsebool -P httpd_unified 1
[root@host html]# systemctl restart httpd ; systemctl restart php-fpm
```

And we test:

```
[root@host html]# wget http://127.0.0.1/a.php
--2021-10-14 08:24:00--  http://127.0.0.1/a.php
S'està connectant a …127.0.0.1:80 connectat.
HTTP: s'ha enviat la petició, s'està esperant una …resposta 200 OK
Mida: no especificada [text/html]
S'està desant a: «a.php.1»

a.php.1                             [ <=>                                                                   ]
     10  --.-KB/s    in 0s

2021-10-14 08:24:00 (579 KB/s) - s'ha desat «a.php.1» [10]

[root@host html]# cat test/test.txt
Prova[root@host html]#
```

Now we create a similar directory on the root folder, for b.php to write there:

```
[root@host html]# mkdir /test
[root@host html]# chown apache /test
[root@host html]# chmod a+w /test/
[root@host html]#
```

We test, but we do not get any resulting file:

```
[root@host html]# wget http://127.0.0.1/b.php
--2021-10-14 08:25:10--  http://127.0.0.1/b.php
S'està connectant a …127.0.0.1:80 connectat.
HTTP: s'ha enviat la petició, s'està esperant una …resposta 200 OK
Mida: 0 [text/html]
S'està desant a: «b.php.1»

b.php.1                             [ <=>                                                                 ]
     0  --.-KB/s    in 0s

2021-10-14 08:25:10 (0,00 B/s) - s'ha desat «b.php.1» [0/0]

[root@host html]# ls -l /test/
total 0
```

We can see the SELinux actions with ausearch:

```
[root@host html]# ausearch -m AVC,USER_AVC
----
time->Thu Oct 14 08:29:30 2021
type=AVC msg=audit(1634214570.592:330): avc:  denied  { getattr } for  pid=12568 comm="php-fpm" path="/
     test/test.txt" dev="sda3" ino=9272655 scontext=system_u:system_r:httpd_t:s0 tcontext=system_u:object_r
     :default_t:s0 tclass=file permissive=0
----
time->Thu Oct 14 08:29:30 2021
type=AVC msg=audit(1634214570.592:331): avc:  denied  { write } for  pid=12568 comm="php-fpm" name="test.
     txt" dev="sda3" ino=9272655 scontext=system_u:system_r:httpd_t:s0 tcontext=system_u:object_r:default_t
     :s0 tclass=file permissive=0
```

SELinux file read protections

SELinux prevents things like accidental access to unwanted files.

We create a file, on the /root directory, readable by all, and we move it to the web root:

```
[root@host html]# echo "Root file" > /root/rootfile
[root@host html]# mv /root/rootfile /var/www/html/
[root@host html]# ls -Z rootfile
unconfined_u:object_r:admin_home_t:s0 rootfile
```

If we try to access the file, SELinux prevents it:

```
[root@host html]# wget -O - http://127.0.0.1/rootfile
--2021-10-14 08:32:57--  http://127.0.0.1/rootfile
S'està connectant a …127.0.0.1:80 connectat.
HTTP: s'ha enviat la petició, s'està esperant una …resposta 403 Forbidden
2021-10-14 08:32:57 ERROR: 403 Forbidden.
```

That's because the context of the file is not right for a web accessible file.

```
[root@host html]# ausearch -m AVC,USER_AVC
time->Thu Oct 14 08:32:57 2021
type=AVC msg=audit(1634214777.679:332): avc:  denied  { getattr } for  pid=12349 comm="httpd" path="/var/
    www/html/rootfile" dev="sda3" ino=8924907 scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:
    object_r:admin_home_t:s0 tclass=file permissive=0
----
time->Thu Oct 14 08:32:57 2021
type=AVC msg=audit(1634214777.679:333): avc:  denied  { getattr } for  pid=12349 comm="httpd" path="/var/
    www/html/rootfile" dev="sda3" ino=8924907 scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:
    object_r:admin_home_t:s0 tclass=file permissive=0
```

We can make that file accessible:

```
[root@host html]# restorecon -Rv /var/www/html/rootfile
Relabeled /var/www/html/rootfile from unconfined_u:object_r:admin_home_t:s0 to unconfined_u:object_r:
    httpd_sys_content_t:s0
[root@host html]# wget -O - http://127.0.0.1/rootfile
--2021-10-14 08:38:15--  http://127.0.0.1/rootfile
S'està connectant a …127.0.0.1:80 connectat.
HTTP: s'ha enviat la petició, s'està esperant una …resposta 200 OK
Mida: 10
S'està desant a: «STDOUT»

-                                   0%[                                                                    ]
    0  --.-KB/s               Root file
-                                  100%[=========================================================>]
    10  --.-KB/s    in 0s

2021-10-14 08:38:15 (1,01 MB/s) - escrit a la sortida estàndard [10/10]

[root@host html]#
```

Creating rules.

We can see why things happen, first by looking at the log:

```
[root@host html]# grep httpd /var/log/audit/audit.log
type=AVC msg=audit(1634214540.300:326): avc:  denied  { write open } for  pid=12567 comm="php-fpm" path="/
    test/test.txt" dev="sda3" ino=9272655 scontext=system_u:system_r:httpd_t:s0 tcontext=system_u:object_r
    :default_t:s0 tclass=file permissive=1
type=AVC msg=audit(1634214540.300:327): avc:  denied  { getattr } for  pid=12567 comm="php-fpm" path="/
    test/test.txt" dev="sda3" ino=9272655 scontext=system_u:system_r:httpd_t:s0 tcontext=system_u:object_r
    :default_t:s0 tclass=file permissive=1
type=AVC msg=audit(1634214570.592:330): avc:  denied  { getattr } for  pid=12568 comm="php-fpm" path="/
    test/test.txt" dev="sda3" ino=9272655 scontext=system_u:system_r:httpd_t:s0 tcontext=system_u:object_r
    :default_t:s0 tclass=file permissive=0
type=AVC msg=audit(1634214570.592:331): avc:  denied  { write } for  pid=12568 comm="php-fpm" name="test.
    txt" dev="sda3" ino=9272655 scontext=system_u:system_r:httpd_t:s0 tcontext=system_u:object_r:default_t
    :s0 tclass=file permissive=0
type=AVC msg=audit(1634214777.679:332): avc:  denied  { getattr } for  pid=12349 comm="httpd" path="/var/
    www/html/rootfile" dev="sda3" ino=8924907 scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:
    object_r:admin_home_t:s0 tclass=file permissive=0
type=AVC msg=audit(1634214777.679:333): avc:  denied  { getattr } for  pid=12349 comm="httpd" path="/var/
    www/html/rootfile" dev="sda3" ino=8924907 scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:
    object_r:admin_home_t:s0 tclass=file permissive=0
[root@host html]#
```

Then ask why that denegation:

```
[root@host html]# grep httpd /var/log/audit/audit.log  | audit2why
type=AVC msg=audit(1634214777.679:332): avc:  denied  { getattr } for  pid=12349 comm="httpd" path="/var/
     www/html/rootfile" dev="sda3" ino=8924907 scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:
     object_r:admin_home_t:s0 tclass=file permissive=0

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

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

type=AVC msg=audit(1634214777.679:333): avc:  denied  { getattr } for  pid=12349 comm="httpd" path="/var/
     www/html/rootfile" dev="sda3" ino=8924907 scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:
     object_r:admin_home_t:s0 tclass=file permissive=0

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

         You can use audit2allow to generate a loadable module to allow this access.
```

And even creating policies to allow for it:

```
[root@host html]# grep httpd /var/log/audit/audit.log  | audit2allow -M mynewpol
****************** IMPORTANT ***********************
To make this policy package active, execute:

semodule -i mynewpol.pp
```

We'll have two files mynewpol.pp and mynewpol.te, with the policies in binary and in text format.

```
[root@host html]# cat  mynewpol.te

module mynewpol 1.0;

require {
    type default_t;
    type admin_home_t;
    type httpd_sys_content_t;
    type httpd_t;
    class dir { add_name write };
    class file { create getattr open write };
}

#============= httpd_t ==============
allow httpd_t admin_home_t:file getattr;
allow httpd_t default_t:dir add_name;
allow httpd_t default_t:file { create getattr open write };

#!!!! This avc is allowed in the current policy
allow httpd_t httpd_sys_content_t:dir { add_name write };

#!!!! This avc is allowed in the current policy
allow httpd_t httpd_sys_content_t:file { create write };
```