3.2. UNCONFINED PROCESSES

Unconfined processes run in unconfined domains, for example, unconfined services executed by `init` end up running in the `unconfined_service_t` domain, unconfined services executed by kernel end up running in the `kernel_t` domain, and unconfined services executed by unconfined Linux users end up running in the `unconfined_t` domain. For unconfined processes, SELinux policy rules are applied, but policy rules exist that allow processes running in unconfined domains almost all access. Processes running in unconfined domains fall back to using DAC rules exclusively. If an unconfined process is compromised, SELinux does not prevent an attacker from gaining access to system resources and data, but of course, DAC rules are still used. SELinux is a security enhancement on top of DAC rules – it does not replace them.

To ensure that SELinux is enabled and the system is prepared to perform the following example, complete the Procedure 3.1, “How to Verify SELinux Status” (chap-Security-Enhanced_Linux-Targeted_Policy#proc-How_to_Verify_SELinux_Status) described in Section 3.1, “Confined Processes” (chap-Security-Enhanced_Linux-Targeted_Policy#sect-Security-Enhanced_Linux-Targeted_Policy-Confined_Processes).

The following example demonstrates how the Apache HTTP Server (`httpd`) can access data intended for use by Samba, when running unconfined. Note that in Red Hat Enterprise Linux, the `httpd` process runs in the confined `httpd_t` domain by default. This is an example, and should not be used in production. It assumes that the httpd, wget, dbus and audit packages are installed, that the SELinux targeted

policy is used, and that SELinux is running in enforcing mode.

Procedure 3.3. An Example of Unconfined Process

1. The `chcon` command relabels files; however, such label changes do not survive when the file system is relabeled. For permanent changes that survive a file system relabel, use the `semanage` utility, which is discussed later. As the root user, enter the following command to change the type to a type used by Samba:

```
~]# chcon -t samba_share_t /var/www/html/testfile
```

View the changes:

```
~]$ ls -Z /var/www/html/testfile -rw-r--r-- root root
unconfined_u:object_r:samba_share_t:s0 /var/www/html/testfile
```

2. Enter the following command to confirm that the `httpd` process is not running:

```
~]$ systemctl status httpd.service httpd.service - The Apache
HTTP Server Loaded: loaded (/usr/lib/systemd/system
/httpd.service; disabled) Active: inactive (dead)
```

If the output differs, enter the following command as root to stop the `httpd` process:

```
~]# systemctl stop httpd.service
```

3. To make the `httpd` process run unconfined, enter the following command as root to change the type of the `/usr/sbin/httpd` file, to a type that does not transition to a confined domain:

```
~]# chcon -t bin_t /usr/sbin/httpd
```

4. Confirm that `/usr/sbin/httpd` is labeled with the `bin_t` type:

```
~]$ ls -Z /usr/sbin/httpd -rwxr-xr-x. root root
system_u:object_r:bin_t:s0 /usr/sbin/httpd
```

5. As root, start the `httpd` process and confirm, that it started successfully:

```
~]# systemctl start httpd.service
```

```
~]# systemctl status httpd.service httpd.service - The Apache
HTTP Server Loaded: loaded (/usr/lib/systemd/system
/httpd.service; disabled) Active: active (running) since Thu
2013-08-15 11:17:01 CEST; 5s ago
```

6. Enter the following command to view `httpd` running in the `unconfined_service_t` domain:

```
~]$ ps -eZ | grep httpd system_u:system_r:unconfined_service_t:s0
11884 ? 00:00:00 httpd system_u:system_r:unconfined_service_t:s0
11885 ? 00:00:00 httpd system_u:system_r:unconfined_service_t:s0
11886 ? 00:00:00 httpd system_u:system_r:unconfined_service_t:s0
11887 ? 00:00:00 httpd system_u:system_r:unconfined_service_t:s0
11888 ? 00:00:00 httpd system_u:system_r:unconfined_service_t:s0
11889 ? 00:00:00 httpd
```

7. Change into a directory where your Linux user has write access to, and enter the following command. Unless there are changes to the default configuration, this command succeeds:

```
~]$ wget http://localhost/testfile --2009-05-07 01:41:10--
http://localhost/testfile Resolving localhost... 127.0.0.1
Connecting to localhost|127.0.0.1|:80... connected. HTTP request
sent, awaiting response... 200 OK Length: 0 [text/plain] Saving
to: `testfile' [ <=> ]--.-K/s in 0s 2009-05-07 01:41:10 (0.00
B/s) - `testfile' saved [0/0]
```

Although the `httpd` process does not have access to files labeled with the `samba_share_t` type, `httpd` is running in the unconfined

`unconfined_service_t` domain, and falls back to using DAC rules, and as such, the `wget` command succeeds. Had `httpd` been running in the confined `httpd_t` domain, the `wget` command would have failed.

8. The `restorecon` utility restores the default SELinux context for files. As root, enter the following command to restore the default SELinux context for `/usr/sbin/httpd`:

```
~]# restorecon -v /usr/sbin/httpd restorecon reset /usr/sbin
/httpd context
system_u:object_r:unconfined_exec_t:s0->system_u:object_r:httpd_e
xec_t:s0
```

Confirm that `/usr/sbin/httpd` is labeled with the `httpd_exec_t` type:

```
~]$ ls -Z /usr/sbin/httpd -rwxr-xr-x root root
system_u:object_r:httpd_exec_t:s0 /usr/sbin/httpd
```

9. As root, enter the following command to restart `httpd`. After restarting, confirm that `httpd` is running in the confined `httpd_t` domain:

```
~]# systemctl restart httpd.service
```

```
~]$ ps -eZ | grep httpd system_u:system_r:httpd_t:s0 8883 ?
00:00:00 httpd system_u:system_r:httpd_t:s0 8884 ? 00:00:00 httpd
system_u:system_r:httpd_t:s0 8885 ? 00:00:00 httpd
system_u:system_r:httpd_t:s0 8886 ? 00:00:00 httpd
system_u:system_r:httpd_t:s0 8887 ? 00:00:00 httpd
system_u:system_r:httpd_t:s0 8888 ? 00:00:00 httpd
system_u:system_r:httpd_t:s0 8889 ? 00:00:00 httpd
```

10. As root, remove `testfile`:

```
~]# rm -i /var/www/html/testfile rm: remove regular empty file
`/var/www/html/testfile'? y
```

11. If you do not require `httpd` to be running, as root, enter the following

command to stop `httpd` :

```
~]# systemctl stop httpd.service
```

The examples in these sections demonstrate how data can be protected from a compromised confined-process (protected by SELinux), as well as how data is more accessible to an attacker from a compromised unconfined-process (not protected by SELinux).

3.1. Confined Processes (/documentation/en-us/red_hat_enterprise_linux/7/html/selinux_users_and_administrators_guide/chap-security-enhanced_linux-targeted_policy#sect-Security-Enhanced_Linux-Targeted_Policy-Confined_Processes)

3.3. Confined and Unconfined Users (/documentation/en-us/red_hat_enterprise_linux/7/html/selinux_users_and_administrators_guide/sect-security-enhanced_linux-targeted_policy-confined_and_unconfined_users)

Where did the comment section go?

Red Hat's documentation publication system recently went through an upgrade to enable speedier, more mobile-friendly content. We decided to re-evaluate our commenting platform to ensure that it meets your expectations and serves as an optimal feedback mechanism. During this redesign, we invite your input on providing feedback on Red Hat documentation via the discussion platform (/node/add/discussion?field_tags[]=docs-feedback&field_product[]=red_hat_enterprise_linux).