Confining Users with Predefined SELinux Security Policies in Red Hat Enterprise Linux 6

By David Egts and Krista M. Guglielmetti, with Daniel Walsh
2/17/11

On a Red Hat Enterprise Linux system, you can use SELinux to confine applications and to confine users. Confining an application limits the extent to which a hacker can exploit security vulnerabilities in that application to gain unauthorized access to files or to execute malicious code on the system. Confining an authorized user limits the extent to which that user can compromise the system – whether deliberately or accidentally – by executing malicious code, by editing or deleting system files, or by compromising the security of sensitive information, such as credit card billing statements.

SELinux automatically enforces predefined security policies to confine many commonly used applications, making the system less vulnerable to damage from external attacks. However, all Linux users are unconfined by default. To prevent malicious, or more often, ignorant Linux users from compromising your system, you should configure SELinux to confine all Linux users.

In Red Hat Enterprise Linux 5, the default SELinux policy, the Targeted Policy, did not include predefined security policies for confined users, and many system administrators found it difficult to configure SELinux to use the Strict Policy or the MLS Policy to confine users. As a result, the application of user confinement was generally limited to special purpose deployments implemented by SELinux experts.

In Red Hat Enterprise Linux 6, SELinux has been redesigned to make it much easier for system administrators to configure SELinux to confine Linux users. The default Targeted Policy now includes four predefined, role-based security policies for confined users. In addition, SELinux now includes a Policy Generator GUI that allows system administrators to quickly and easily customize existing SELinux policy modules or create new ones to define custom role-based security policies for confined users.

This paper describes the simplest way to confine users with SELinux in Red Hat Enterprise Linux 6 – using the predefined, role-based security policies that are included with the default Targeted Policy.

This paper covers the following topics:

CHOOSING AN APPROPRIATE SELINUX SECURITY POLICY FOR A LINUX USER

In Red Hat Enterprise Linux, each time you create a new user, the system maps the new Linux user to an SELinux user for which a role-based security policy is defined in an SELinux policy module. SELinux applies the security policy defined for an SELinux user to **all** Linux users that are mapped to that SELinux user.

In Red Hat Enterprise Linux 6, the default SELinux policy – the Targeted Policy – includes five SELinux user policy modules, which define role-based security policies for four confined SELinux users and one unconfined SELinux user. By default, Red Hat Enterprise Linux maps all new Linux users to the unconfined SELinux user, `unconfined_u`. You must manually configure the system to map a Linux user to a confined SELinux user.

Each SELinux user represents a common role or a common user profile that you may assign to one or more Linux users. Each security policy is designed to allow Linux users who are mapped to an SELinux user to perform **only** those tasks associated with their assigned role or user profile.

Table 1 describes the five predefined SELinux security policies that are included in the default SELinux policy in Red Hat Enterprise Linux 6.

For **each** Linux user that you want to confine, use Table 1 to identify the SELinux user whose security policy is most appropriate for the Linux user's assigned role or user profile. Then, use one of the three methods described in the next section to map the Linux user to the SELinux user.

Table 1: Predefined Role-Based Security Policies for Confined Users

SELinux User Name	Predefined SELinux Security Policy	Linux User Role/Profile	Example Use Case
`unconfined_u` **NOTE:** By default, all Linux users are mapped to this SELinux user.	Default, unconfined user. User's access to system resources is restricted only by standard UNIX® file permissions settings. User processes are **also** unconfined. They can use `setuid` applications, such as `su` and `sudo`, to become root, so they can access files and processes owned by root. **NOTE:** A *user process* is any process that is started by a user login program, such as `sshd`, `gdm`, or `login`.	System owner/Initial root user	Default user configuration enables a system administrator to become root and configure SELinux to confine all Linux users – including themself.

Table continued on next page...

SELinux User Name	Predefined SELinux Security Policy	Linux User Role/Profile	Example Use Case
`guest_u`	User is confined. User can log into the system **only** by using a non-graphical console or by using `ssh`. User **cannot** access network ports or run X Window. User **cannot** execute files in his home directory or in the `/tmp` directory. **NOTE:** You can enable `guest_u` to execute files in his home directory and in the `/tmp` directory by issuing the following command as root: `# usr/sbin/setsebool -P allow_guest_exec_content on` User **cannot** use `setuid` applications, such as `su` and `sudo`, to become root. User processes **cannot** use `setuid` applications to operate as root, and they cannot access files or processes owned by root. **NOTE:** A *user process* is any process that is started by a user login program, such as `sshd`, `gdm`, or `login`.	Anonymous/guest user accounts, client user accounts Security policy is designed for remote users who need to log into the system **solely** to access files stored in their home directories.	Many colleges and universites have faculty profile pages posted on their websites. In most cases, a university's faculty members are expected to create and maintain their own profile pages on the university's web server. To create and edit their profiles, faculty members need to be able to log into their Linux user accounts on their university's web server **solely** to upload, modify, and/or delete files stored in their personal profile (home) directories. These users should not be allowed to access any other system resources on the university's web server, they should not be allowed to use the server to open a network connection, and they should not be allowed to become root on the server. *Table continued on next page...*

SELinux User Name	Predefined SELinux Security Policy	Linux User Role/Profile	Example Use Case
`xguest_u`	Default `xguest_u` security policy: User is confined. User has the same privileges as `guest_u` with the following major exceptions: • User can run X Window applications. • User can connect to WWW network ports. User can access any website and can run any web application. • User can use a few other types of Internet applications, such as instant messaging software. **NOTE:** You can disable network access for `xguest_u` by executing the following command as root: **`# usr/sbin/setsebool -P xguest_connect_network off`** **NOTE:** You can enable `xguest_u` to execute files in his home directory and in the `/tmp` directory by issuing the following command as root: `#` **`usr/sbin/setsebool -P allow_xguest_exec_content on`** Kiosk user security policy: If you install the `xguest` package for Red Hat Enterprise Linux 6, you can configure a Linux user account mapped to `xguest_u` as a kiosk user account. A *kiosk user account* is a non-password-protected, anonymous guest user account whose security policy is designed for users logging into a public system **solely** to access the web. A kiosk user can **only** log into the system locally. He can use a web browser to view websites and run web applications, but he **cannot** use any other Internet applications. In addition, any changes that a kiosk user makes to the system (such as creating files or changing settings) are lost when the kiosk user logs out..	Default `xguest_u_` security policy: Non-technical application users and web users who log into shared systems to run applications and/or access the web Security policy is designed to enable a system administrator to configure a shared system used primarily to provided non-technical users with access to shared resources run specific X Window or web applications or to provide users with web access. a group of non-technical users who all use the system to perform similar tasks, and who all need to run the same X Window applications and/or web applications while they are logged into the system. Kiosk user security policy: Anonymous public web kiosk user Security policy is designed for public systems used primarily for web access. Security policy enables users to log into an anonymous guest user account to access the web.	Default `xguest_u` security policy: A system administrator may install and configure a shared Red Hat Enterprise Linux 6 system for the receptionist desk. Receptionists need to be able to use the shared system to run X Window applications and/or web applications, so they can create badges, schedule appointments, etc. when they are working at the front desk. Receptionists do not need to use their system to open a network connection to another machine because the shared system should provide them with access to all the resources they need to do their jobs. Receptionists also do not need to have root access to their machine, since the system administrator is responsible for configuring and maintaining the system. Kiosk user security policy: Many hotels and libraries have installed public web kiosks in their lobbies. Hotel and library customers need to be able to use the kiosks **solely** to access the web. These users need access to WWW network ports, but they should not be allowed to access any other network ports. These users should not be allowed to access system resources not required to run X Window applications or web applications, and they should not be allowed to become root. *Table continued on next page...*

SELinux User Name	Predefined SELinux Security Policy	Linux User Role/Profile	Example Use Case
`user_u`	User is confined. User has the same privileges as `xguest_u`, with two major exceptions: • User can access **all** network ports, not just WWW network ports. • User can execute files in his home directory and in the `/tmp` directory. **NOTE:** You can prevent `user_u` from executing files in his home directory and in the `/tmp` directory by issuing the following command as root: `# usr/sbin/setsebool -P allow_user_exec_content off` User is restricted in only two ways: • User **cannot** execute `setuid` applications, such as `su` and `sudo`, to become root. • User processes **cannot** use `setuid` applications to operate as root, and they cannot access files or processes owned by root. **NOTE:** A *user process* is any process that is started by a user login program, such as `sshd`, `gdm`, or `login`.	Non-technical system owners/users who are not responsible for configuring and maintaining their own systems Security policy is designed for systems that companies provide to non-technical employees. Non-technical employees can use their systems to do their jobs, but they cannot become root on the system, so they cannot inadvertently access (or compromise) important system files and processes.	Many non-technical employees are currently using company-owned Red Hat Enterprise Linux 6 systems to do their work. If the company's IT department is responsible for configuring and maintaining all company-owned systems, there is no reason for non-technical employees to have root access to their machines. To prevent these users from compromising their systems, IT staff members can configure each employee's system to map his Linux user account to the confined SELinux user, `user_u`. *Table continued on next page...*

SELinux User Name	Predefined SELinux Security Policy	Linux User Role/Profile	Example Use Case

`staff_u`	User is confined.
	
	User has the same privileges as `user_u`, with two major exceptions:
	• SELinux **can** be configured to allow Linux users mapped to `staff_u` to use `sudo` (though **not** `su`) to become root.
	
	NOTE: By default, SELinux prevents **all** confined users from using `sudo` to execute commands as root. You can configure SELinux to change this default behavior, but **only** for Linux users mapped to `staff_u`. To do this, execute the following command as `root` to add an access rule to the file, `/etc/sudoers`, for **each** Linux user mapped to `staff_u`:
	`# echo "<Linux_user_name> ALL=(ALL) ROLE=sysadm_r TYPE=sysadm_t ALL" >> /etc/sudoers`
	
	• User processes **can** use `setuid` applications to operate as root, and they **can** access files or processes owned by root.
	
	NOTE: A *user process* is any process that is started by a user login program, such as `sshd`, `gdm`, or `login`.
	
	NOTE: You can prevent `staff_u` from executing files in his home directory and in the `/tmp` directory by issuing the following command as root:
	`# usr/sbin/setsebool -P allow_staff_exec_content off`
		Red Hat Enterprise Linux 6 system administrators
		
		Users are trustworthy, security-conscious, and technically competent.
		
		Users need to execute commands as root for one or both of the following two reasons:
		• Users are responsible for configuring and maintaining their own systems.
		• Users need to execute commands as root to do their jobs.
			A company's IT staff members are responsible for configuring and maintaining the company-owned systems that they use to do their jobs. Therefore, each IT staff member can configure his company-owned system to map his Linux user account to the confined SELinux user, `staff_u`.

CONFIGURING A LINUX USER AS A CONFINED SELINUX USER

To confine a Linux user with an SELinux security policy defined for a confined SELinux user, you configure the system to map the Linux user to the SELinux user. SELinux applies the security policy defined for an SELinux user to **all** Linux users that are mapped to that SELinux user.

You can map a Linux user to a confined SELinux user in one of three ways:

- Create a new Linux user account and map it to a confined SELinux user:
  # **/usr/sbin/useradd -Z <*SELinux_user_name*> <*Linux_user_name*>**
- Map an existing Linux user to a confined SELinux user:
  # **/usr/sbin/semanage login -a -s <*SELinux_user_name*> <*Linux_user_name*>**
- Change the default SELinux user mapping from `unconfined_u` to one of the four confined SELinux users. The following command maps the specified SELinux user to all new Linux users **and** to all existing Linux users that are not explicitly mapped to another SELinux user:
  # **/usr/sbin/semanage login -m -S targeted -s "<*SELinux_user_name*>" -r s0 __default__**

TUTORIAL: CONFINING LINUX USERS WITH PREDEFINED SELINUX SECURITY POLICIES

This tutorial guides you through the process of configuring Linux users as confined SELinux users, and demonstrates the application of each of the four predefined, role-based SELinux security policies available with SELinux in Red Hat Enterprise Linux 6.

System Requirements

Before beginning this tutorial, make sure your system meets the following software requirements:

- System must be configured as a Red Hat Enterprise Linux 6.0 server.
  NOTE: Optionally, you can run Red Hat Enterprise Linux 6.0 as a guest of any hypervisor.
- X Window and GNOME must be installed on the system.
- The Red Hat Enterprise Linux 6 `policycoreutils-python` and `xguest` packages must be installed on the system.
- The `httpd` daemon must be running on the system.
- System must be able to access the Internet.

Configure Linux Users as Confined SELinux Users

Create four demo Linux user accounts, and map each Linux user account to one of the four confined SELinux users, as follows.

1. Login to the system as root.

2. Create a new Linux user account with the login name `joe_guest` and the password `redhat`, and map the new Linux user to the confined SELinux user, `guest_u`:
   `# /usr/sbin/useradd -Z guest_u joe_guest && echo redhat | passwd --stdin joe_guest`
3. Create a new Linux user account with the login name `joe_xguest` and the password `redhat`, and map the new Linux user to the confined SELinux user, `xguest_u`:
   `# /usr/sbin/useradd -Z xguest_u joe_xguest && echo redhat | passwd --stdin joe_xguest`
4. Create a new Linux user account with the login name `joe_user` and the password `redhat`, and map the new Linux user to the confined SELinux user, user_u:
   `# /usr/sbin/useradd -Z user_u joe_user && echo redhat | passwd --stdin joe_user`
5. Create a new Linux user account with the login name `joe_staff` and the password `redhat`, and map the new Linux user to the confined SELinux user, staff_u:
   `# /usr/sbin/useradd -Z staff_u joe_staff && echo redhat | passwd --stdin joe_staff`
6. Use the following command to add an access rule to your `/etc/sudoers` file that allows `joe_staff` to use `sudo` to execute commands as root:
   `# echo "joe_staff ALL=(ALL) ROLE=sysadm_r TYPE=sysadm_t ALL" >> /etc/sudoers`
7. Log out of the system as root.

SELinux in Action: Predefined SELinux Security Policies for Confined SELinux Users

Use the four demo Linux user accounts that you created above to test the predefined role-based security policies for the four confined SELinux users:

- guest_u
- xguest_u
- user_u
- staff_u

SELinux Security Policy for guest_u

1. Take a moment to review the predefined SELinux security policy for Linux users mapped to the confined SELinux user, guest_u.
2. Log into the system with user name `joe_guest` and password `redhat`.
3. Use the `id -Z` command to confirm that `joe_guest` is mapped to the SELinux user, `guest_u`.
4. Try to do the following tasks, and note which tasks you cannot do when logged in as `joe_guest`:
   i. From the `gdm` login screen, log in to X Window as `joe_guest`.

ii. Open a virtual console and attempt to log into the system as `joe_guest`.
If you are running Red Hat Enterprise Linux 6 as a VM, to open a virtual console, select **Send Key** from the virt-manager pull-down menu and press `Ctrl-Alt-F2`. If you are running Red Hat Enterprise Linux 6 on bare metal, to open a virtual console, just press `Ctrl-Alt-F2`.

iii. From the virtual console, use `ssh` to connect to the localhost:
`ssh localhost`

iv. From the virtual console, ping the localhost:
`ping localhost`

v. Execute the following command on the system:
`/bin/echo hi`

vi. Enter the following commands to copy the above executable into your home directory and into the `/tmp` directory, and then run the executable in each directory:
`cp /bin/echo ~`
`cp /bin/echo /tmp`
`~/echo hi`
`/tmp/echo hi`

You should be able to complete tasks ii and v when logged in as `joe_guest`. You should not be able to do any of the other tasks because `joe_guest` is mapped to the confined SELinux user, `guest_u`, so the security policy for `guest_u` is applied to `joe_guest`.

As `joe_guest`, you cannot do task i because `guest_u` cannot run X Window, you cannot do tasks iii and iv because `guest_u` cannot access network ports, and you cannot do task vi because `guest_u` cannot run executables in his home directory or in the `/tmp` directory.

5. Log out of the virtual console as `joe_guest` and log out of the system as `joe_guest`.

SELinux Security Policy for xguest_u

1. Take a moment to review the predefined SELinux security policy for Linux users mapped to the confined SELinux user, xguest_u.
2. Log into the system with user name `joe_xguest` and password `redhat`.
3. Use the `id -Z` command to confirm that `joe_xguest` is mapped to the SELinux user, `xguest_u`.
4. Try to do the following tasks, and note which tasks you cannot do when logged in as `joe_xguest`:
   i. From the `gdm` login screen, log in to X Window as `joe_xguest`.
   ii. Open a terminal console by choosing **Application > System Tools > Terminal** from the desktop menu.
   iii. From the terminal console, use `ssh` to connect to the localhost:
   `ssh localhost`
   iv. From the terminal console, ping the localhost:
   `ping localhost`

v. Start a web browser and go to http://localhost, or go to any website to which you have access.

vi. Execute the following command on the system:
`/bin/echo hi`

vii. Enter the following commands to copy the above executable into your home directory and into the `/tmp` directory, and then run the executable in each directory:
`cp /bin/echo ~`
`cp /bin/echo /tmp`
`~/echo hi`
`/tmp/echo hi`

You should be able to complete tasks i, ii, v, and vi when logged in as `joe_xguest`. You should not be able to do any of the other tasks because `joe_xguest` is mapped to the confined SELinux user, `xguest_u`, so the security policy for `xguest_u` is applied to `joe_xguest`.

As `joe_xguest`, you can do task i because unlike `guest_u`, `xguest_u` can run X Window. You cannot do tasks iii and iv, but you can do task v. This is because `xguest_u` can access WWW network ports, but cannot access any other network ports. You cannot do task vii because `xguest_u` cannot run executables in his home directory or in the `/tmp` directory.

5. Log out of the terminal console as `joe_xguest`, log out of X Window as `joe_xguest`, and log out of the system as `joe_xguest`.

SELinux Security Policy for user_u

1. Take a moment to review the predefined SELinux security policy for Linux users mapped to the confined SELinux user, user_u.
2. Log into the system with user name `joe_user` and password `redhat`.
3. Use the `id -Z` command to confirm that `joe_user` is mapped to the SELinux user, `user_u`.
4. Try to do the following tasks, and note which tasks you cannot do when logged in as `joe_user`:
   i. From the `gdm` login screen, log in as `joe_user` with a password of `redhat`.
   ii. Open a terminal console by choosing **Application > System Tools > Terminal** from the desktop menu.
   iii. From the terminal console, ping the localhost:
   `ping localhost`
   iv. From the terminal console, use `ssh` to connect to the localhost:
   `ssh localhost`
   v. From the terminal console, use ssh to connect to the localhost and log in as `joe_guest` with password `redhat`:
   `ssh joe_guest@localhost`
   vi. From the `ssh` session, use the `id -Z` command to confirm that you are logged in as `joe_guest`, and `joe_guest` is mapped to the confined SELinux user, `guest_u`.

vii. From the `ssh` session, use `ssh` to open a connection to the localhost:
`ssh localhost`

viii. Log out of the `ssh` session as `joe_guest`.

ix. From the `ssh` session, use the `id -Z` command to confirm that you are logged in as `joe_user`, and `joe_user` is mapped to the SELinux user, `user_u`.

x. Log out of the `ssh` session as `joe_user`.

xi. From the terminal console, use the `id -Z` command to confirm that you are still logged in as `joe_user`, and `joe_user` is mapped to the SELinux user, `user_u`.

xii. From the terminal console, use the `su` command to become the root user.

xiii. From the terminal console, use the `sudo root` command to become the root user.

xiv. Close the terminal console.

xv. Start a web browser, and go to http://www.redhat.com.

You should be able to complete all of the above tasks except tasks iii, vii, xii, and xiii.

You cannot complete task iii because root owns the `ping` application. Since `user_u` cannot access `setuid` applications, he cannot run processes owned by root.

You cannot complete task vii because at that point, you are logged into the `ssh` session as `joe_guest`, and joe_guest is mapped to the confined SELinux user, `guest_u`, which cannot access network ports.

You cannot complete tasks xii and xiii because `user_u` cannot use `setuid` applications such as `su` and `sudo` to become root.

Log out of X Window, and log out of the system as `joe_user`.

SELinux Security Policy for staff_u

1. Take a moment to review the predefined SELinux security policy for Linux users mapped to the confined SELinux user, staff_u.
2. Log into the system with user name `joe_staff` and password `redhat`.
3. Use the `id -Z` command to confirm that `joe_staff` is mapped to the SELinux user, `staff_u`.
4. Try to do the following tasks, and note which tasks you cannot do when logged in as `joe_staff`:
   i. From the `gdm` login screen, log in as `joe_staff` with a password of `redhat`.
   ii. Open a terminal console by choosing **Application > System Tools > Terminal** from the desktop menu.
   iii. From the terminal console, use the `su` command to become the root user.
   iv. From the terminal console, use the `sudo whoami` command to run `whoami` as the root user.
   v. Close the terminal console.

You should be able to complete all tasks except task iii. You cannot complete task iii because while SELinux has been configured to allow `joe_staff`, who is mapped to `staff_u`, to use `sudo` to execute commands as `root`, Linux users mapped to `staff_u` cannot use `su` to become `root`.

5. Log out of X Window and log out of the system as `joe_staff`.

CONCLUSION

In Red Hat Enterprise Linux 6, SELinux is designed to make it easy for system administrators to confine each Linux user with an appropriate SELinux security policy.

This paper has demonstrated the simplest way to configure SELinux to confine users – by mapping each Linux user to a predefined SELinux security policy for a confined SELinux user.

However, if none of the predefined SELinux security policies is ideal for one or more Linux users, you can define your own custom role-based SELinux security policies for confined users. The new Red Hat Enterprise Linux 6 SELinux Policy Generator GUI enables system administrators to quickly and easily customize predefined SELinux policy modules or create new ones. Advanced SELinux users can also create and edit their own SELinux policy files to define more complex SELinux security policies for confined users.

For more information on defining custom role-based SELinux security policies for confined users, refer to Dan Walsh's FUDCon presentation, "Writing SELinux Policy". For more information on using the SELinux Policy Generator GUI, refer to the February 4th, 2011 entry in Dan Walsh's SELinux Blog.

REFERENCES

- *Red Hat Enterprise Linux 6 Security Enhanced Linux User Guide*
- Red Hat Magazine Article: "Fedora 9 and Summit preview: Confining the user with SELinux" by Dan Walsh
- Dan Walsh's SELinux Blog
- FUDCon 2011 Presentation: "Writing SELinux Policy" by Dan Walsh