Semester Project Spring 2011

SELinux And Confined Users

Report – 3

By – Akhil Vij - 200702003

Prince Rai – 200701061

Under the guidance of –

Professor Saurabh Barjatiya

ABSTRACT

This report deals with creating SELinux policies and access rules for confined users. We approached the solution via two ways. First, we used the concept of Multi-Category Security to set access rules for new/existing users. Here, we label files with categories & only users those who have permissions to access those categories can access these files. Then, we also used an automatic policy generation tool called SELinux-polgengui to write policies for new/ existing users.

INTRODUCTION

Till now, we have studied about the basics of SELinux, security contexts, MAC, DAC, its architecture and about targeted policy writing. Then we also used the SElinux security framework to secure daemons like httpd etc. Now our major task for the final leg of the project was to write policy for confined SELinux users.

SELINUX AND CONFINED USERS

Problem Statement

In, essence the problem is a classic one in OS security literature. Our main goal here is we to see what happens when a (new or already existing) user logs in the system. What all files can the user access. So, the main motivation here is to have some sort of access control mechanism for each user. We tried to solve the problem by two ways. First, we used the concept of Multi-Category Security and then we used policy generation tool called SELinux-polgengui.

MULTI-CATEGORY SECURITY(MCS)

MCS is an enhancement to SELinux which allows users to label files with categories. Only users with access to a particular category can access files of that category. Any user which does not have access to that category will not be able to access the file. An example of a category is “Company_Confidential”. Only users which are allowed to access this category can access files of this category. Also, it is assumed that the normal DAC (file permissions in basic linux) and Type Enforcement (access via security contexts) rules also are satisfied before checking for MCS. So, we can say that the file categories are used to further constraint DAC and TE logic.

The names of the categories and their meanings are set by the sysadmin, and can be set to whatever is required for the specific deployment. A system at home can simply have one category of "Private", and be configured so that only trusted local users are assigned to this category. Once users are assigned categories, they may label any of their own files, with any of their own categories. For example, if our home system user labels his file as “Private” , daemons like httpd cannot access those files as they don’t have permissions to access “Private” category files.

Summarizing, MCS is a very simple idea : to access a file, a user needs to be assigned to **all** of the categories with which the file is labeled.

Testing MCS

Now we are going to define 4 different categories and then assign different categories to 4 different users. We will show how MCS solves the problem at hand.

Creating SELinux users

Here, we create 4 new linux users and map them to SELinux users.

First create 4 users using useradd command .

```
$ useradd james
$ useradd daniel
$ useradd olga
$ useradd karl
```

Assigning Linux User to SELinux User

Use the semanage login -a command to assign Linux users to SELinux user identities:

```
$ semanage  login -a  james
$ semanage  login -a daniel
$ semanage login -a olga
$ semanage login -a karl
```

Listing Categories available on System

SELinux maintains a mapping between internal sensitivity and category levels and their human-readable representations in the setrans.conf file. The system administrator can edit this file to manage and maintain the required categories. Use the "*chcat -L*" command to list the current categories.

```
$ chcat -L
```

```
s0                      SystemLow
s0-s0:c0.c1023          SystemLow-SystemHigh
s0:c0.c1023             SystemHigh
```

Creating SELinux Categories

Modify the /etc/selinux/targeted/setrans.conf file to add 4 new categories :

```
$ vi /etc/selinux/targeted/setrans.conf
```

```
s0:c0=Marketing
s0:c1=Finance
s0:c2=Payroll
s0:c3=Personnel
```

Use the chcat -L command to check the newly-added categories:

```
$ chcat -L
```

```
s0:c0                   Marketing
s0:c1                   Finance
s0:c2                   Payroll
s0:c3                   Personnel
s0                      SystemLow
s0-s0:c0.c1023          SystemLow-SystemHigh
s0:c0.c1023             SystemHigh
```

Assigning Categories to users

Now that we have added Categories to the system , its time to start assigning them to Selinux users and files.

```
$chcat -l  +Marketing james
$chcat -l  +Finance,+Payroll daniel
$chcat -l  +Personnel olga
$chcat  -l +Marketing,+Finance,+Payroll,+Personnel karl
```

Listing Categories for a user

We can also use the chcat command with additional command-line arguments to list the categories that are assigned to users:

$ chcat -L -l daniel james olga karl

Assigning Categories to Files

Now it is time to assign category to a file. For this example, we create a file in Daniel's home directory:

[daniel] $ echo "Financial Records 2006" > financeRecords.txt

Use the ls -Z command to check the initial security context of the file. Assign Category to the file financeRecords.txt.

[daniel] $ chcat +Finance financeRecords.txt

Each of the categories that have been assigned to the file is displayed in the security context. We can add and delete categories to files as required.

Testing Categories

Only users assigned to those categories can access that file, assuming that Linux DAC and TE permissions would already allow the access. For this we would give permission for the other user to access other user's home directory. If a user who is assigned to a different category tries to access the file, they receive an error message:

[olga]$ cat financeRecords.txt
cat: financeRecords.txt: Permission Denied

But if karl access this file he can access it because he has the same category as the file. In this way we can categorize the users and their accessibility to the files.

WRITING POLICY USING SELINUX-POLGENGUI

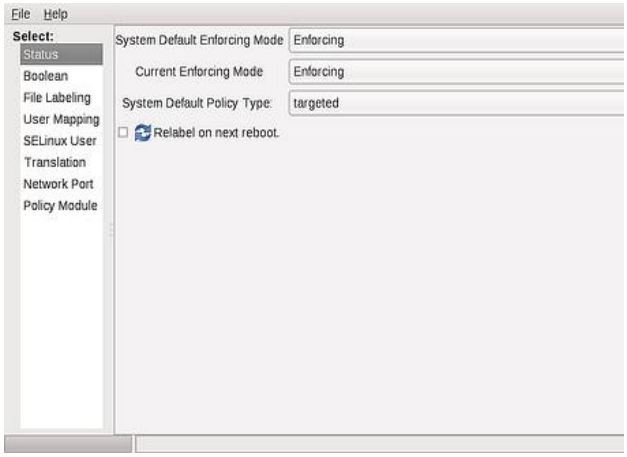
Generating policies manually in SELinux is quite tough. There are many security scenarios and the policy gets complicated really fast. That is why, people tend to use automatic policy generation tools to do the job. We also used an automatic policy generation tool called selinux-polgengui to solve the above problem of access control.

Selinux-polgengui makes policy writing a little bit easier. Polgengui is a template based policy framework that asks the user (or sysadmin) few questions and then, depending on the response, generates the initial policy files to allow the policy writer to get started. It is nowhere close to a complete policy editor. We can use tools like slide, audit2allow etc. for further editing of the initial policy. Selinux-polgengui just gives us a good headstart.

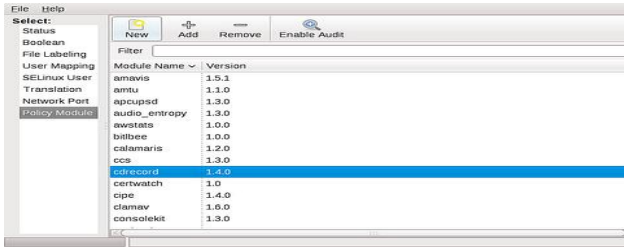
Experimenting with Polgengui tool

Lets take up an example problem of **creating a limited privilege terminal login user with the ability to send mails and read/write files in the /var/maildir directory**. We will use polgengui to create policy for such a user.

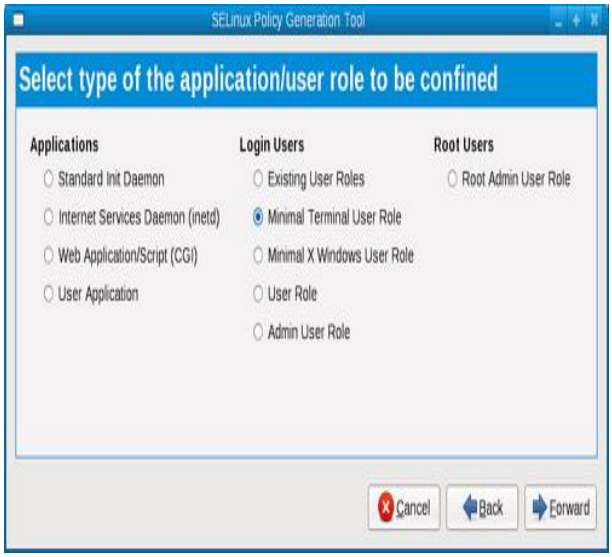
STEP 1. Login as root and start system-config-selinux(System->Administration->SELinux Management)



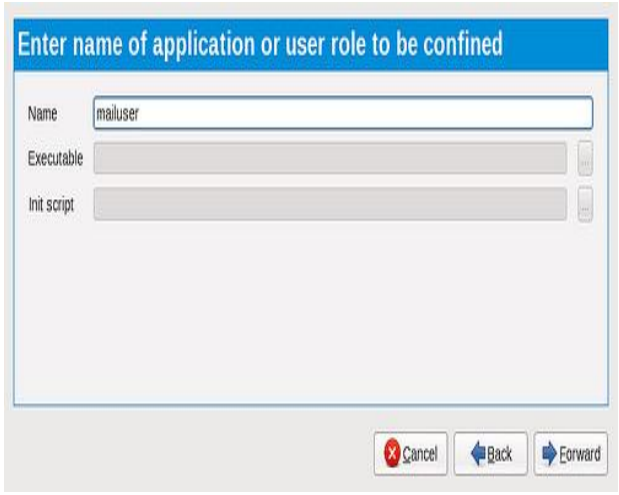
STEP 2. Select policy module and then select *the new button*



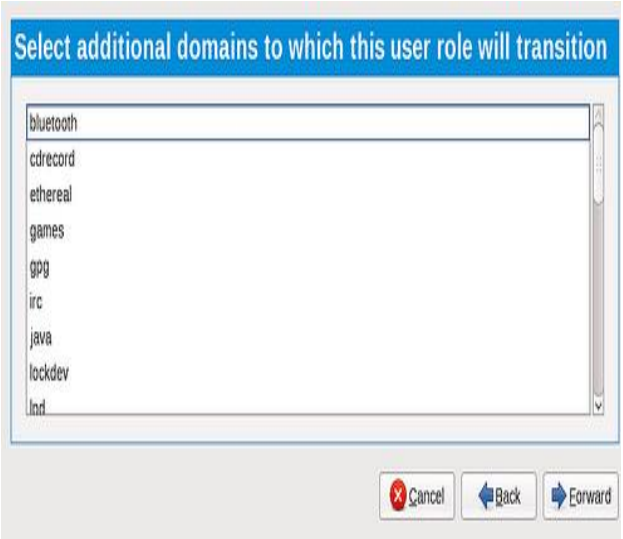
STEP 3. This will start the selinux polgengui tool. Select *create a new minimal terminal user role.*



STEP 4. After selecting minimal terminal user role press Forward and enter the name of the new user role as '*mailuser*'. Click Forward

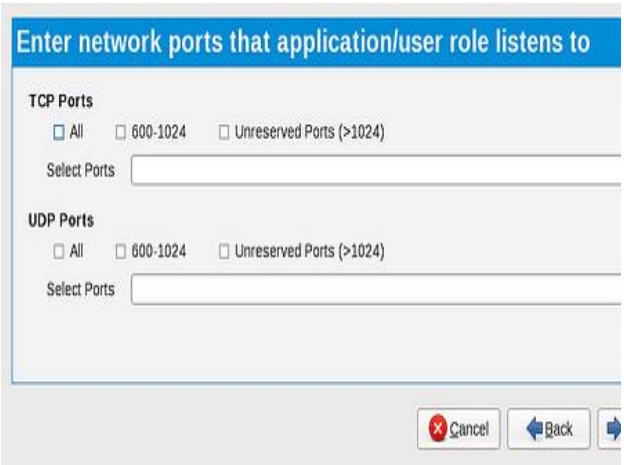


STEP 5. Now we will see a screen that displays a list of confined domains to which this new user role might transition. For example, if you wanted the mailuser to transition to the ethereal domain you would select this now. Since we do not want any transitions we will just hit forward.

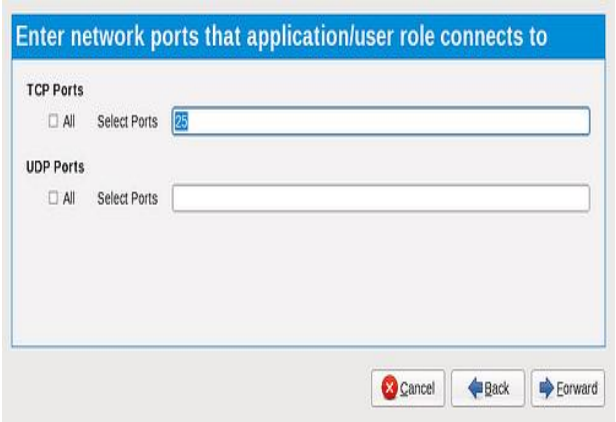


STEP 6. Now we will get a screen that allows us to select roles to which the current role can transition. Skip it and click Forward.

STEP 7. Now we get a screen that allows us to select the ports that the user can listen to. If the confined user was going to run a network server you could select the ports here. We will just select Forward.

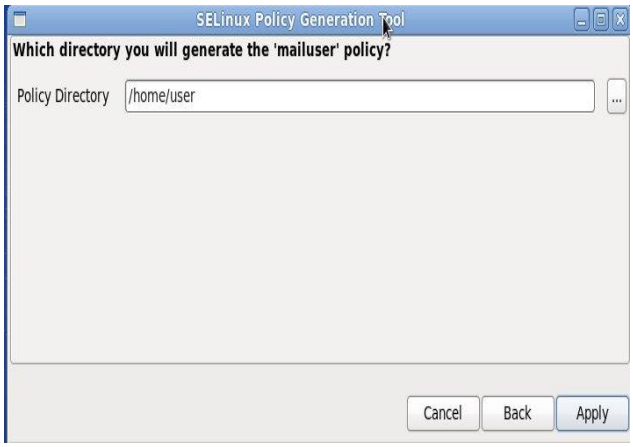


STEP 8. Finally, we get to a screen which defines the ports the confined user can connect to. We will select the smtp port #25 and then go forward again.



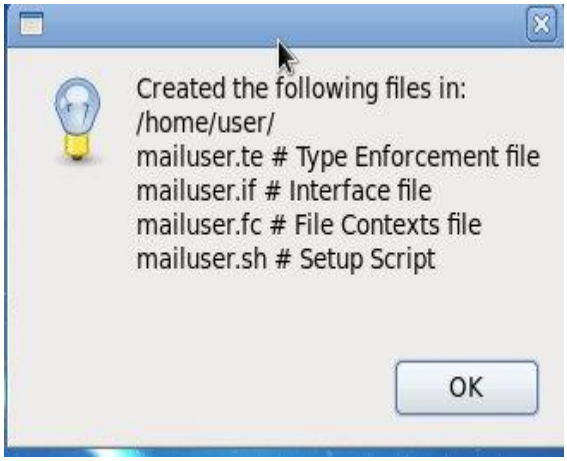
STEP 9 Then , we will get a screen that allows you to define a boolean. We will skip this & click Forward.

STEP 10. Then we select the directory in which we need the policy files. Enter *"/home/user"*



STEP 11. At last we will get four files in the directory . These four files are

1. mailuser.fc
2. mailuser.te
3. mailuser.sh
4. mailuser.if



But among above files only *mailuser.fc* , *mailuser.te* and *mailuser.sh* are of interest to us.

Configuring policy for a new user(say dwalsh)

Open the file *mailtuser.te*. This is the file that is used to create new types for the users and the files. Now in this file we are going to create a new type mailuser_rw_t for the files , which need to be accessed by the user dwlash.

The user **dwalsh** can only read or write those files which are of type **mailuser_rw_t** . So append the following lines in this file.

```
type mailuser_rw_t;
files_type(mailuser_rw_t);

manage_dirs_pattern(mailuser_t,mailuser_rw_t,mailuser_rw_t);

manage_files_pattern(mailuser_t,mailuser_rw_t,mailuser_rw_t);
```

Now open the *mailuser.fc* . This file is used to describe the file contexts of the files . In this file we are going to describe the file contexts of the files in the directory /var/maildir . So append the following line in this file.

```
/var/maildir(/.*)?
gen_context(system_u:object_r:mailuser_rw_t,s0)
```

Compiling policy and installing it

Now we can run the shell script *mailuser.sh* to compile the policy and install it to the test system. Run the following command .

```
$sh mailuser.sh
```

This will create the new role mailuser_r and user mailuser_u.

Mapping SELinux user to linux user

At this point we have a new SELinux user mailuser_u installed on the machine. We will assign a linux user to this type:

```
$ semanage login -a -s mailuser_u dwalsh
```

We also want to create the directories /var/maildir:

```
$ mkdir /var/maildir
$ restorecon /var/maildir
$ chown dwalsh:dwalsh /maildir
```

Now dwalsh can log in and use the /var/maildir directories. As a normal user , he is the only person who can use this directory to read and write .

CONCLUSION

In this project, we studied the very basics of SELinux security contexts, securing some basic daemons with SELinux, basic SELinux working and architecture, MAC, DAC and some very basic targeted policy writing. Then, we also wrote SELinux policies for confined users. We studied about the concept of Multi-Level Security (MLS) and Multi-Category Security. Then we used the policy generation tool SELinux-polgengui to generate policies for new users.

So, in essence, we learned about the working of SELinux.

FUTURE WORK

Since, writing the policy manually and compiling is a very complex task and requires much more deeper understanding of how to use SELinux in various security scenarios. So, manual policy writing and compiling is one task which is on top of to-do list. Then, we could also enable SELinux on campus servers to make sure that they are protected by SELinux.

REFERENCES

- Articles by Dan Walsh taken from his blog http://danwalsh.livejournal.com/
- Main SELinux documentation for Fedora http://fedoraproject.org/wiki/SELinux
- CentOS documentation for SELinux http://www.centos.org/docs/5/html/Deployment_Guide-en-US/selg-overview.html
- Other articles from Wikipedia and other resources