Access Control Systems Lab
Sommersemester 2017

Praktikumsbericht 1

Jean-Marc Hendrikse - 1751591

Prof. Dr. Hannes Hartenstein, Alexander Degitz, Jan Grashöfer, Till Neudecker
Forschungsgruppe Dezentrale Systeme und Netzdienste
Karlsruher Institut für Technologie (KIT)

Introduction

This is the solution sheet for the first Access Control Systems Lab exercise sheet. The main intention of this exercise is to get in touch with the Reference Monitor and Security-Enhanced Linux (SELinux) one of the oldest and most popular Linux Security Module (LSM) to implement Reference Monitors for the Linux operating system.

In the first section we will discuss the background and history of SELinux. In the second section we will explain the basic concepts of SELinux. The third section will describe how you could use SELinux concepts practically and the last section discusses how to implement a Reference Monitor.

1 History and Background

In this section we will give you a brief background and history of SELinux. SELinux was originally a development project from the National Security Agency (NSA) which was implemented as a Flask operating system security architecture. The Flask architecture implements Mandatory Access Control (MAC) and focuses on the concept of least privilege. [Sm02]

1.1 License

In December 2000 SELinux was released by the NSA as a general access software product with the source code distributed under a GPL license. Since then it is still under the GNU GPL license available.[1]

1.2 Who are the SELinux contributors?

The key concept underlying the SELinux based on several earlier projects by NSA who is also the main contributor. Other significant contributors include *Red Hat*, *Network Associates*, *Secure Computing Corporation*, *Tresys Technology*, and *Trusted Computer Solutions*.[1]

1.3 Who is in charge of SELinux?

As we mentioned earlier SELinux was originally developed by the NSA and Red Hat, but today Red Hat is mainly in charge of SELinux.[1]

2 Basic Concepts of SELinux

In the following we will discuss the basic concepts of SELinux and especially introduce concepts of *Policy*, *Labels* and *Modes*.

2.1 SELinux Policy

SELinux follows the model of least-privilege that means that by default everything is denied and only policies give each element of the system access required to its function. SELinux allows different policies to be written that are interchangeable. [5] SELinux policy is a set of rules that guide the SELinux security engine by defining some types for file objects and domains for processes. Types and domains are equivalent with the difference that types are used to apply to objects while domains apply to processes. In order to limit entering the domain SElinux policy uses *roles* and these roles are specified by user identities which can be attained to the roles. The default policy in Fedora is the so called *target* policy which targets and restricts selected system processes. [4]

2.2 SELinux Labels

As in [6] described SELinux labels are mostly stored as extended attributes. But you have to be careful this is not standard, because some file systems do not support extended attributes. In these cases, all files on the file system get assigned the same context, usually provided through the mount option of the file system. On systems running SELinux, all processes and files are labelled in a way that represents security-relevant information which is called the SELinux *context*. E.g. by running the command `ls -Z` the information of a certain file will be printed.

2.3 SELinux Modes

SELinux provides three basic modes of operation:

- **Enforcing** - is the installation default mode which enforces the SELinux security policy on the system, denying access and logging actions.[2]
- **Permissive** - this mode enables SELinux but will not enforce the security policy, only warn and log actions. This mode is also very helpful for troubleshooting SELinux issues.[2]
- **Disabled** - in this mode SELinux is turned off.[2]

The SELinux Management UI enables to view the SELinux mode and make changes possible. From the command line the GUI is available by running `system-config-selinux`.

3 Using SELinux

After we finished the theoretical part of the basic concepts of SELinux we would like to have a more practical view on SELinux.

3.1 Using sestatus

First of all we would like to check the *status* of SELinux. This can be done by printing the status of SELinux on the running system using `sestatus`:

```
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deniy_unknown status:    allowed
Max kernel policy version:      30
```

Listing 1: sestatus ouput

As in [3] described the command displays data whether SELinux is enabled, disabled, the loaded policy and whether it is in enforcing or permissive mode (we will describe in later section what that means). It can also be used to display the security context of files and processes listed in the /etc/sestatus.conf file. In this case SELinux is **enabled**, SELinux is mounted at `/sys/fs/selinux`, the root directory of SELinux is `/etc/selinux`, the default policy name is `targeted` and the current mode is `enforcing` the same as from config file, Multi Level Security protection is enabled, the state for unknown permissions is allowed which means everything is allowed to perform this action and the system support a maximum version of 30 that will also support loading policy modules of a lower binary version.

3.2 Starting the Webserver

In the next step we start the webserver by using the command `sudo systemctl start httpd`. If you are not sure whether the webserver is running or not you can simply check it by `systemctl status httpd`. After the webserver started we would like to see what the server actually serves when calling the index.html file: `curl http://localhost/index.html`. In our case this command prints out:

```
"Security = Access Control"
```

Listing 2: SELinux index.html output

Which is basically the content of `index.html`.

As mentioned in the section about labels we get attributes and detail information from a file by running `ls -lZ`:

```
total 4
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 26 May  3
    15:32 /var/www/html/index.html
```

Listing 3: SELinux context of index.html

By executing the `ls -lZ` command we get the security context of the file /var/www/html/index.html. In addition to the standard file permissions and ownership, we can see the SELinux security context fields: unconfined_u:object_r:httpd_sys_content_t:s0. The scheme is user:role:type:mls which means in that case that the user is unconfined logged in, the role is object and the type is `httpd_sys_content`. The MLS has a low security level (s0), associated with no compartments. [5]

In the next task we change the type label of the file to `samba_share_t` using `chcon`. After trying to retrieve the file again we get the following output:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
  <head>
    <title>403 Forbidden</title>
  </head>
  <body>
    <h1>Forbidden</h1>
    <p>You don't have permission to access /index.html on this server.<br />
    </p>
  </body>
</html>
```

Listing 4: index.html output access permission denied

This means we have no permission to access that file. But why did the permission changed and why do we have no permission rights? By using `chcon` users provide all or part of the SELinux context to change. An incorrect file type causes access deny which is why we get the error message from above.

3.3 The `audit.log` File

After we performed some actions on our SELinux access decisions are logged by *auditd* to /var/log/audit/audit.log. Locating the log message of the previous access decision gives us:

```
type=AVC msg=audit(1494689834.117:338): avc:  denied  { getattr } for  pid=2980
     comm="httpd" path="/var/www/html/index.html" dev="dm-0" ino=1180274
    scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:
    samba_share_t:s0 tclass=file permissive=0
```

Listing 5: Access decision from audit.log file.

The following Table is inspired by [7] and describes the parts of that message:

Log part	Description
type=AVC	This log part describes the audit log type. In our example the log type is set to Access Vector Cache (AVC) log entry.
msg=audit(1494689834.117:338)	Is the timestamp in seconds since January 1st 1970.
avc:	Again we get a log type and again it shows up a AVC log type.
denied	Describes the **state** of previous denied or granted decisions by SELinux.
getattr	Is simply a permission request, which is in this case a read operation getattr.
for pid=2980	The process identifier (PID) is a unique of an active process that performed the action for an access.
comm = httpd	The process command
path = /var/www/html/index.html	The path to the file which we tried to access
dev = dm-0	Describes the device where the target is archived.
ino=1180274	Every file has a inode number. Thus the operating system is able to identify the specific inode of the target file.
scontext	s in scontext stands for source which means that scontext describes the securit context of the process. The scontext is set to system_u:system_r:httpd_t:s0
tcontext	t in tcontext stands for security context of the target
tclass	Describes the class of the target. In our case we have a file as target class (index.html).
permissive	The permissive mode here is disabled.

Because of avc:denied for pid2980 in the message above we can say for sure that SELinux denied the access. We get a corresponding entry in the *system's journal* by `journalctl`:

```
~$ journalctl
```

The oldest entries will be up top, where we get the corresponding entry:

```
...
May 13 19:02:37 DSN-ACSLab-Master setroubleshoot[5005]: failed to retrieve rpm
    info for /var/www/html/index.html
May 13 19:02:38 DSN-ACSLab-Master setroubleshoot[5005]: SELinux is preventing
    httpd from getattr access on the file /var/www/html/index.html. For complete
     SELinux messages. run sealert -l 5e0840eb-ba33-414d-a23a-8d75a8f4a31a
May 13 19:02:38 DSN-ACSLab-Master python3[5005]: SELinux is preventing httpd
    from getattr access on the file /var/www/html/index.html.

                                                    *****  Plugin restorecon (92.2
     confidence) suggests   ************************

                                                    If you want to fix the label.
                                                    /var/www/html/index.html
    default label should be httpd_sys_content_t.
                                                    Then you can run restorecon.
                                                    Do
                                                    # /sbin/restorecon -v /var/www
    /html/index.html

                                                    *****  Plugin public_content
    (7.83 confidence) suggests   *******************

                                                    If you want to treat index.
    html as public content
```

```
                                                  Then you need to change the
    label on index.html to public_content_t or public_content_rw_t.
                                                  Do
                                                  # semanage fcontext -a -t
    public_content_t '/var/www/html/index.html'
                                                  # restorecon -v '/var/www/html
    /index.html'

                                                  *****  Plugin catchall (1.41
    confidence) suggests   **************************

                                                  If you believe that httpd
    should be allowed getattr access on the index.html file by default.
                                                  Then you should report this as
     a bug.
                                                  You can generate a local
    policy module to allow this access.
                                                  Do
                                                  allow this access for now by
    executing:
                                                  # ausearch -c 'httpd' --raw |
    audit2allow -M my-httpd
                                                  # semodule -X 300 -i my-httpd.
    pp
...
```

Listing 6: Output from file.

In listing 6 we are informed “for complete SELinux messages. run sealert -l 5e0840eb-ba33-414d-a23a-8d75a8f4a31a“ in order to get the complete message for that access decision. This shows us denials are assigned IDs. In our case we get the ID = `5e0840eb-ba33-414d-a23a-8d75a8f4a31a` for the access denial.

```
[student@DSN-ACSLab-Master html]$ sealert -l 5e0840eb-ba33-414d-a23a-8
    d75a8f4a31a
SELinux is preventing httpd from getattr access on the file /var/www/html/index
    .html.

*****  Plugin restorecon (92.2 confidence) suggests   ************************

If you want to fix the label.
/var/www/html/index.html default label should be httpd_sys_content_t.
Then you can run restorecon.
Do
# /sbin/restorecon -v /var/www/html/index.html

*****  Plugin public_content (7.83 confidence) suggests   ********************

If you want to treat index.html as public content
Then you need to change the label on index.html to public_content_t or
    public_content_rw_t.
Do
# semanage fcontext -a -t public_content_t '/var/www/html/index.html'
# restorecon -v '/var/www/html/index.html'

*****  Plugin catchall (1.41 confidence) suggests   **************************

If you believe that httpd should be allowed getattr access on the index.html
    file by default.
Then you should report this as a bug.
You can generate a local policy module to allow this access.
Do
allow this access for now by executing:
```

```
# ausearch -c 'httpd' --raw | audit2allow -M my-httpd
# semodule -X 300 -i my-httpd.pp

Additional Information:
Source Context                system_u:system_r:httpd_t:s0
Target Context                unconfined_u:object_r:samba_share_t:s0
Target Objects                /var/www/html/index.html [ file ]
Source                        httpd
Source Path                   httpd
Port                          <Unknown>
Host                          DSN-ACSLab-Master
Source RPM Packages
Target RPM Packages
Policy RPM                    selinux-policy-3.13.1-224.fc25.noarch
Selinux Enabled               True
Policy Type                   targeted
Enforcing Mode                Enforcing
Host Name                     DSN-ACSLab-Master
Platform                      Linux DSN-ACSLab-Master 4.8.6-300.fc25.x86_64 #1
                             SMP Tue Nov 1 12:36:38 UTC 2016 x86_64 x86_64
Alert Count                   4
First Seen                    2017-05-13 17:37:14 CEST
Last Seen                     2017-05-13 19:02:33 CEST
Local ID                      5e0840eb-ba33-414d-a23a-8d75a8f4a31a

Raw Audit Messages
type=AVC msg=audit(1494694953.911:493): avc:  denied  { getattr } for  pid=2981
     comm="httpd" path="/var/www/html/index.html" dev="dm-0" ino=1180274
    scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:
    samba_share_t:s0 tclass=file permissive=0


Hash: httpd,httpd_t,samba_share_t,file,getattr

[student@DSN-ACSLab-Master html]$
```

Listing 7: SELinux Context von index.html

Compared to /var/log/audit/audit.log `sealert` gives us the same message but associated with the denial and further detailed information.

4 SELinux as Reference Monitor

The Linux Security Module (LSM) defines a reference monitor interface for Linux so that many implementations of reference monitors such as SELinux exist.

In the lecture three fundamental requirements were introduced that a Reference Monitor implementation has to fulfill (c.f. Lecture 1, Slide 30):

1. Evaluable - small enough to be subject to analysis
2. Always invoked - no alterative access method
3. Tamper-proof - mechanism cannot be altered

Furthermore, the extended Reference Monitor concepts have been introduced (c.f. Lecture 1, Slide 31), i.e. *Authorization Database*, *Monitor Interface* and *Audit Trails*. To proof that LSM is an extended reference monitor interface and SELinux an implementation of it we have to map the concepts of an extended reference monitor to their Linux counterparts. A corresponding counterpart of the Reference Monitor Interface in LSM is the concept of LSM hooks which are upcalls to modules when ever a user-level system call tries to acces an internal kernel object. The LSM labels are implemented as Authorization Databases and auditd and the `sealert` command can be mapped to Audit Trails.

5 Discussion and Conclusion

Because SELinux is one of the oldest and most popular Linux Security Module (LSM), to implement Reference Monitors for the Linux operating system it provides all the concepts of a Reference Monitor such as hooks, labels for a Authorization Database and auditd for Audit Trails. SELinux is a mandatory access control (MAC) security mechanism implemented in the kernel. It is a comprehensive Linux Security Module that goal is tamperproofing of system's trusted computing base and therefore it implements Reference Monitors for the Linux operating system. It provides concepts i.e. *Authorization Database* by labels, *Monitor Interface* hooks and *Audit Trails* by commands as `sealert`.

Attachement A: Abbrevation

GNU GPL - Gneral Public License
GUI - Graphical User Interface
LSM - Linux Security Module
MAC - Mandatory Access Control
NSA - National Security Agency
SELinux - Security-Enhanced Linux

Literatur

[1] Wikipedia-Web,`https://en.wikipedia.org/wiki/Security-Enhanced_Linux`, visited on 21/05/2017.

[2] Documentation of Fedora SELinux Modes,`https://docs.fedoraproject.org/en-US/Fedora/12/html/Security-Enhanced_Linux/sect-Security-Enhanced_Linux-Working_with_SELinux-SELinux_Modes.html`, visited on 21/05/2017.

[3] Documentation of Fedora SELinux Enabling and Disabling SELinux, `https://docs.fedoraproject.org/en-US/Fedora/11/html/Security-Enhanced_Linux/sect-Security-Enhanced_Linux-Working_with_SELinux-Enabling_and_Disabling_SELinux.html`, visited on 19/05/2017.

[4] Documentation of Fedora SELinux Policies,`https://fedoraproject.org/wiki/De_DE/SELinux/Policies`, visted on 22/05/2017.

[5] Centos SELinux Documentation,`https://wiki.centos.org/HowTos/SELinux`, visited on 22/05/2017.

[6] Wikipedia Gentoo SELinux Documentation, chapter about SELinux labels, `https://wiki.gentoo.org/wiki/SELinux/Labels`, visted on 21/05/2017.

[7] Wikipedia Gentoo Documentation about SELinux permission Denial Details, https://wiki.gentoo.org/wiki/SELinux/Tutorials/Where_to_find_SELinux_permission_denial_details, visited on 21/05/2017.