MUNI
FI

Access Control Mechanisms – Practice

PA211 Advanced Topics of Cyber Security

November 15, 2022

Petr Velan, Pavel Čeleda and Jan Vykopal

Prepare Virtual Environment

Be sure to have an up-to date copy of the repository

- `git clone gitlab.fi.muni.cz:cybersec/pa211/hardening.git`
- `cd hardening/acm`
- `vagrant up`
- `vagrant ssh`

1) Warm-up Exercise

Delete file `/home/vagrant/practice/file_to_delete`

1) Warm-up Exercise – Solution

```
$ sudo chattr -ai /home/vagrant/practice/file_to_delete
$ rm /home/vagrant/practice/file_to_delete
```

Discretionary Access Control

2) DAC Exercise

- Alice wants to share a secret file with Bob so that no one else can read it or even see the file. Bob should not be able to write there.
- Set correct permissions to `/home/alice/shared_with_bob/for_bob.txt` to ensure this
- Test that `bob` can read it and `eve` cannot.
- Change users using `sudo su username` command.

2) DAC Exercise – Users and groups

- Group `security: alice, bob, eve`
- Group `accounting: jane, henry`

Do not modify membership of these users in groups.

2) DAC Exercise – Solution

```
vagrant@server:~$ ll /home/alice/shared_with_bob/
total 12
drwxr-xr-x 2 alice alice 4096 Nov 15 02:08 .
drwxr-xr-x 3 alice alice 4096 Nov 15 02:07 ..
-rw-r--r-- 1 alice alice   72 Nov 15 02:08 for_bob.txt
vagrant@server:~$ sudo chgrp bob -R /home/alice/shared_with_bob/
vagrant@server:~$ sudo chmod 750 /home/alice/shared_with_bob/
vagrant@server:~$ sudo chmod 640 /home/alice/shared_with_bob/for_bob.txt

bob@server:/home/vagrant$ cat /home/alice/shared_with_bob/for_bob.txt
Hey Bob, this is a very secret message that nobody but you should read.
bob@server:/home/vagrant$ exit

vagrant@server:~$ sudo su eve
eve@server:/home/vagrant$ cat /home/alice/shared_with_bob/for_bob.txt
cat: /home/alice/shared_with_bob/for_bob.txt: Permission denied
eve@server:/home/vagrant$ ls /home/alice/shared_with_bob/
ls: cannot open directory '/home/alice/shared_with_bob/': Permission denied
```

3) Shared Folder Exercise

- Alice wants to share files with everybody in the `security` group and no one else
- Everybody in the group should be able to read all files

  in `/home/alice/shared_with_security/`
- Everybody in the group should be able to delete their files, but not files created by others
- New files in the directory should automatically have these properties
- Test all the above-mentioned properties and also that nobody from `accounting` has access

3) Shared Folder Exercise – Solution

```
vagrant@server:~$ sudo chmod 3770 /home/alice/shared_with_security/          (alternative solution: chmod 1770 …)
vagrant@server:~$ sudo chgrp security /home/alice/shared_with_security/
vagrant@server:~$ sudo ls -la /home/alice/shared_with_security/
drwxrws—T 2 alice security 4096 Nov 15 02:26 .
drwxr-xr-x 4 alice alice    4096 Nov 15 02:21 ..

bob@server:/home/vagrant$ echo content > /home/alice/shared_with_security/bobs_file.txt

eve@server:/home/vagrant$ cat /home/alice/shared_with_security/bobs_file.txt
content
eve@server:/home/vagrant$ echo ‘more content’ >> /home/alice/shared_with_security/bobs_file.txt
bash: /home/alice/shared_with_security/bobs_file.txt: Permission denied
eve@server:/home/vagrant$ rm /home/alice/shared_with_security/bobs_file.txt
rm: remove write-protected regular file ‘/home/alice/shared_with_security/bobs_file.txt’? Y
rm: cannot remove ‘/home/alice/shared_with_security/bobs_file.txt’: Operation not permitted
eve@server:/home/vagrant$ echo ‘eve was here’ > /home/alice/shared_with_security/eves_file.txt
eve@server:/home/vagrant$ rm /home/alice/shared_with_security/eves_file.txt

jane@server:/home/vagrant/example$ cat /home/alice/shared_with_security/bobs_file.txt
cat: /home/alice/shared_with_security/bobs_file.txt: Permission denied
```

4) ACL Exercise

- Now Alice wants to share files with her friends, Bob and Jane.
- Set permissions of `/home/alice/shared_with_friends` to allow read and write access only to `bob` and `jane`.
- Both friends must be able to create and delete any file in this directory.
- Do not forget to test that others do not have access.

4) ACL Exercise – Solution 1/2

```
vagrant@server:~$ sudo chmod og-rwx /home/alice/shared_with_friends/
vagrant@server:~$ sudo setfacl -m d:u:bob:rwx -m u:bob:rwx -m d:u:jane:rwx -m u:jane:rwx -m d:g::--- -m
d:o:--- -m m::rwx /home/alice/shared_with_friends/
vagrant@server:~$ getfacl /home/alice/shared_with_friends/
getfacl: Removing leading '/' from absolute path names
# file: home/alice/shared_with_friends/
# owner: alice
# group: alice
user::rwx
user:bob:rwx
user:jane:rwx
group::---
mask::rwx
other::---
default:user::rwx
default:user:bob:rwx
default:user:jane:rwx
default:group::---
default:mask::rwx
default:other::---
```

4) ACL Exercise – Solution 2/2

```
jane@server:/home/vagrant/example$ echo 'hello Alice!' > /home/alice/shared_with_friends/message.txt

bob@server:/home/vagrant/example$ echo 'hello!' >> /home/alice/shared_with_friends/message.txt
bob@server:/home/vagrant/example$ ls -l /home/alice/shared_with_friends/message.txt
-rw-rw----+ 1 jane jane 20 Nov 15 03:00 /home/alice/shared_with_friends/message.txt
bob@server:/home/vagrant/example$ rm /home/alice/shared_with_friends/message.txt
bob@server:/home/vagrant/example$

eve@server:/home/vagrant/example$ ls /home/alice/shared_with_friends/
ls: cannot open directory '/home/alice/shared_with_friends/': Permission denied
eve@server:/home/vagrant/example$ echo 'evil message' > /home/alice/shared_with_friends/from_a_friend.txt
bash: /home/alice/shared_with_friends/from_a_friend.txt: Permission denied
```

AppArmor

5) Delete File Exercise

Delete file `/home/vagrant/practice/another_file_to_delete`

5) Delete File Exercise – Solution

```
$ grep another_file_to_delete /etc/apparmor.d/ -R
/etc/apparmor.d/usr.bin.rm:  deny /home/vagrant/practice/another_file_to_delete w,

$ sudo aa-disable /etc/apparmor.d/usr.bin.rm
Disabling /etc/apparmor.d/usr.bin.rm.

$ rm /home/vagrant/practice/another_file_to_delete
```

6) Profile for example.sh

- Create and load an AppArmor profile for `/home/vagrant/example/example.sh`
- Use `aa-genprof`

6) Profile for example.sh – Solution 1/3

- `sudo aa-genprof example.sh`
- Execute example.sh in another terminal, return back

```
[(S)can system log for AppArmor events] / (F)inish
```

- Select `(S)can system log for AppArmor events`

```
Reading log entries from /var/log/syslog.
Updating AppArmor profiles in /etc/apparmor.d.

Profile:  /home/vagrant/example/example.sh
Execute:  /usr/bin/touch
Severity: 3

(I)nherit / (C)hild / (N)amed / (X) ix On / (D)eny / Abo(r)t / (F)inish
```

- Select `(I)nherit`

6) Profile for example.sh – Solution 2/3

```
Profile:  /home/vagrant/example/example.sh
Execute:  /usr/bin/rm
Severity: unknown

(I)nherit / (C)hild / (N)amed / (X) ix On / (D)eny / Abo(r)t / (F)inish
```

– Select (I)nherit

```
Profile:  /home/vagrant/example/example.sh
Path:     /dev/tty
New Mode: rw
Severity: 9

[1 - #include <abstractions/consoles>]
  2 - /dev/tty rw,
(A)llow / [(D)eny] / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Audi(t) / Abo(r)t / (F)inish
```

– Select (A)llow

6) Profile for example.sh – Solution 3/3

```
Profile:  /home/vagrant/example/example.sh
Path:     /home/vagrant/practice/sample.txt
New Mode: owner w
Severity: 6

[1 - owner /home/*/practice/sample.txt w,]
  2 - owner /home/vagrant/practice/sample.txt w,
(A)llow / [(D)eny] / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Audi(t) / (O)wner permissions off /
Abo(r)t / (F)inish
```

- Select `(A)llow`

```
The following local profiles were changed. Would you like to save them?

[1 - /home/vagrant/example/example.sh]
(S)ave Changes / Save Selec(t)ed Profile / [(V)iew Changes] / View Changes b/w (C)lean profiles / Abo(r)t
```

- Select `(S)ave Changes`

- Check aa-status to see that the profile is loaded
- Example profile is in `/home/vagrant/practice/home.vagrant.example.example.sh`

7) Man Pages ping

- Run: `less /home/vagrant/.bash_aliases`
- Write: `!ping 8.8.8.8 -c 3`
- Ping is executed
- Repeat the same with less opened by `man apparmor`

- Question: Why does it not work?
- Task: Update local man profile to allow ping from a man page

7) Man Pages ping – Solution

- Capability new_raw is necessary for ping (see /etc/apparmor.d/bin.ping)

```
$ echo 'capability net_raw,' | sudo tee /etc/apparmor.d/local/usr.bin.man
$ sudo apparmor_parser -r /etc/apparmor.d/usr.bin.man
```

HW3 Assignment

Your task is to create a SUID/SGID binary that will allow privilege escalation to root user. Then you will limit the capabilities of this binary using AppArmor.

1. Create SUID/SGID binary **`getrootbash`** which, when executed, will drop any user into a `bash` with root privileges (real and effective user and group will be set to root). Provide source code for this binary as well as a `Makefile` for compilation and setting of necessary permissions and ownership (`chown` and `chmod`).
2. Create an AppArmor profile called **`usr.local.sbin.getrootbash`** that will limit access of this binary and all its children to **`/tmp/`** directory, everywhere else, it must be read only, if necessary.
   - Process must not be able to remove files of other users in `/tmp`
   - Put the binary into **`/usr/local/sbin/`** directory and prepare the AppArmor profile accordingly.
   - Provide the profile file that can be parsed and applied with `apparmor_parser`. You can use `bash` and `base` AppArmor abstractions.
   - You can ignore permission denied messages (e.g. `bash: /usr/etc/profile.d/ls.bash: Permission denied`) unless they interfere with desired functionality.

HW3 Usage Example

The submitted binary and profile should be usable as shown in the following example:

```
$ ls
getrootbash.c  Makefile

$ make
gcc -o getrootbash getrootbash.c
sudo chown root:root getrootbash
sudo chmod 6775 getrootbash

$ sudo mv getrootbash /usr/local/sbin/getrootbash
$ /usr/local/sbin/getrootbash

root@server:~# id
uid=0(root) gid=0(root) groups=0(root),1000(vagrant)

root@server:~# ls /
bin  boot  dev  etc  home  initrd.img  initrd.img.old
lib  lib32  lib64  libx32  lost+found  media  mnt  opt
proc  root  run  sbin  srv  sys  tmp  usr  vagrant  var
  vmlinuz  vmlinuz.old

root@server:~# exit
exit
```

```
# Install AppArmor profile:
$ sudo mv usr.local.sbin.getbashroot /etc/apparmor.d/
$ sudo apparmor_parser -r
  /etc/apparmor.d/usr.local.sbin.getbashroot

# Try again
$ /usr/local/sbin/getrootbash
bash: /usr/share/bash-completion/bash_completion: Permission
  denied

I have no name!@server:~# id
uid=0 gid=0 groups=0,1000
I have no name!@server:~# ls
ls: cannot open directory '.': Permission denied
I have no name!@server:~# touch /tmp/file
I have no name!@server:~# ls -l /tmp/file
-rw-r--r-- 1 0 0 0 Nov  8 15:23 /tmp/file
I have no name!@server:~# rm /tmp/file
I have no name!@server:~#
```

HW3 Points

- SUID binary: 7 points
  - 5 points: correctly working SUID binary
  - 2 points: correctly working Makefile
- AppArmor profile: 8 points
  - 3 points: Write access only into /tmp
  - 3 points: profile applies to child processes
  - 2 points: cannot remove files of another user in /tmp

- Failure to run the application or install the profile will result in 0 points for the respective parts.
- Incorrect submission format will result in 0 points.

HW3 Submission

- Submit your solution to the Homework Vault:

  https://is.muni.cz/auth/el/fi/podzim2022/PA211/ode/ode_hw3/

- Submit ONLY a single **tar** archive with following files:
  - `getbashroot.*`: Source code for `getbashroot` binary
  - `Makefile`: Makefile to build the `getbashroot` binary and set permissions
  - `usr.local.sbin.getbashroot`: AppArmor profile limiting privileges of `getbashroot` processes
- Deadline for submission is **November 29, 2022,10:00**

How was it today?

Please fill in an **anonymous** exit ticket:

https://muni.cz/go/pa211-22-10

MUNI FI

MUNI FI