Лабораторная работа 2
Политики безопасности Linux

Выполнила: Батомункуева Виктория

Группа: P34101

Программные и аппаратные средства:

Процессор – AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx, 2.10 GHz
Видеокарта – AMD Radeon(TM) Vega 8 Graphics
Объем оперативной памяти – 16,0 ГБ.

Лабораторная работа выполнялась в Oracle VM VirtualBox (Ubuntu 20.04.6). Основная ОС – Windows 10.

Основная часть:

1) Установите утилиту AppArmor sudo apt install apparmor-utils apparmor-profiles. Напишите bash-скрипт который будет создавать файл в директории log , записывать в него что-то, читать из него и затем удалять.

Установим AppArmor

```
ubuntu@ubuntu:~$ sudo apt install apparmor-utils apparmor-profiles
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apparmor libapparmor1 python3-apparmor python3-libapparmor
Suggested packages:
  apparmor-profiles-extra vim-addon-manager
The following NEW packages will be installed:
  apparmor-profiles apparmor-utils python3-apparmor python3-libapparmor
The following packages will be upgraded:
  apparmor libapparmor1
2 upgraded, 4 newly installed, 0 to remove and 204 not upgraded.
Need to get 892 kB of archives.
After this operation, 1442 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libapparmor1 amd64 4.0.1really4.0.1-0u
buntu0.24.04.3 [50.3 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 apparmor amd64 4.0.1really4.0.1-0ubunt
u0.24.04.3 [641 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 python3-libapparmor amd64 4.0.1really4
.0.1-0ubuntu0.24.04.3 [30.1 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 python3-apparmor all 4.0.1really4.0.1-
0ubuntu0.24.04.3 [84.5 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 apparmor-utils all 4.0.1really4.0.1-0u
buntu0.24.04.3 [46.4 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 apparmor-profiles all 4.0.1really4.0.1
-0ubuntu0.24.04.3 [39.6 kB]
Fetched 892 kB in 1s (973 kB/s)
Preconfiguring packages ...
```

Создадим файл, напишем в нем bash-скрипт и дадим права на выполнение

```
ubuntu@ubuntu:~$ which bash
/usr/bin/bash
ubuntu@ubuntu:~$ touch file
ubuntu@ubuntu:~$ gedit file
cat file
ubuntu@ubuntu:~$ cat file
#!/usr/bin/bash
touch log/myfile.txt
echo "UNIX Lab 2" > log/myfile.txt
cat log/myfile.txt
rm log/myfile.txt
```

2) Создайте директорию log. Выдайте файлу права на исполнение. Запустите файл, покажите вывод ./file

Создадим директорию log и не забудем указать права для file. Запустим file.

```
ubuntu@ubuntu:~$ mkdir log
ubuntu@ubuntu:~$ ./file
bash: ./file: Permission denied
ubuntu@ubuntu:~$ chmod u+x file
ubuntu@ubuntu:~$ ./file
UNIX Lab 2
```

3) Создайте профиль безопасности для данной программы sudo aa-genprof ./file. Покажите результат выполнения программы

После выполнения sudo aa-genprof ./file вот что пишет консоль. Надо параллельно запустить file.

```
Updating AppArmor profiles in /etc/apparmor.d.
Writing updated profile for /home/ubuntu/file.
Setting /home/ubuntu/file to complain mode.

Before you begin, you may wish to check if a
profile already exists for the application you
wish to confine. See the following wiki page for
more information:
https://gitlab.com/apparmor/apparmor/wikis/Profiles

Profiling: /home/ubuntu/file

Please start the application to be profiled in
another window and exercise its functionality now.

Once completed, select the "Scan" option below in
order to scan the system logs for AppArmor events.

For each AppArmor event, you will be given the
opportunity to choose whether the access should be
allowed or denied.

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

Сканирование:

```
Profile:  /home/ubuntu/file
Execute:  /usr/bin/touch
Severity: 3

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

Profile:  /home/ubuntu/file
Execute:  /usr/bin/cat
Severity: unknown

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

Profile:  /home/ubuntu/file
Execute:  /usr/bin/rm
Severity: unknown

(I)nherit / (C)hild / (N)amed / (X) ix On / (D)eny / Abo(r)t / (F)inish
Complain-mode changes:

Profile:  /home/ubuntu/file
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
Adding include <abstractions/consoles> to profile.
```

```
Profile:  /home/ubuntu/file
Path:     /home/ubuntu/log/myfile.txt
New Mode: owner w
Severity: 6

 [1 - owner /home/*/log/myfile.txt w,]
  2 - owner /home/ubuntu/log/myfile.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
Adding owner /home/*/log/myfile.txt w, to profile.

Profile:  /home/ubuntu/file
Path:     /etc/ld.so.cache
New Mode: r
Severity: 1

 [1 - /etc/ld.so.cache r,]
(A)llow / [(D)eny] / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Audi(t) / Abo(r)t / (F)inish
Adding /etc/ld.so.cache r, to profile.

Profile:  /home/ubuntu/file
Path:     /etc/locale.alias
New Mode: r
Severity: unknown

 [1 - /etc/locale.alias r,]
(A)llow / [(D)eny] / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Audi(t) / Abo(r)t / (F)inish
Adding /etc/locale.alias r, to profile.

Profile:  /home/ubuntu/file
Path:     /home/ubuntu/log/myfile.txt
Old Mode: owner w
New Mode: owner r
Severity: 4


Profile:  /home/ubuntu/file
Path:     /home/ubuntu/log/myfile.txt
Old Mode: owner rw
New Mode: owner rw
Severity: 6

 [1 - owner /home/*/log/myfile.txt rw,]
  2 - owner /home/ubuntu/log/myfile.txt rw,
(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
Adding owner /home/*/log/myfile.txt rw, to profile.
Deleted 2 previous matching profile entries.
Enforce-mode changes:

= Changed Local Profiles =

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

 [1 - /home/ubuntu/file]
(S)ave Changes / Save Selec(t)ed Profile / [(V)iew Changes] / View Changes b/w (C)lean profiles / Abo(r)t
Writing updated profile for /home/ubuntu/file.

Profiling: /home/ubuntu/file

Please start the application to be profiled in
another window and exercise its functionality now.

Once completed, select the "Scan" option below in
order to scan the system logs for AppArmor events.

For each AppArmor event, you will be given the
opportunity to choose whether the access should be
allowed or denied.

[(S)can system log for AppArmor events] / (F)inish
Setting /home/ubuntu/file to enforce mode.
```

4) Запустите утилиту aa-logprof и настройте разрешения так, чтобы при выполнении программы не было ошибок. Запустите файл еще раз. Покажите, что теперь ошибок нет.

Настроим aa-logprof

```
ubuntu@ubuntu:~$ sudo aa-logprof
Updating AppArmor profiles in /etc/apparmor.d.
Reading log entries from /var/log/syslog.
Complain-mode changes:
Enforce-mode changes:

Profile:    /usr/lib/snapd/snap-confine
Capability: net_admin
Severity:   8

 [1 - capability net_admin,]
(A)llow / [(D)eny] / (I)gnore / Audi(t) / Abo(r)t / (F)inish
Adding capability net_admin, to profile.

Profile:    /usr/lib/snapd/snap-confine
Capability: perfmon
Severity:   7

 [1 - capability perfmon,]
(A)llow / [(D)eny] / (I)gnore / Audi(t) / Abo(r)t / (F)inish
Adding capability perfmon, to profile.

= Changed Local Profiles =

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

 [1 - /usr/lib/snapd/snap-confine]
(S)ave Changes / Save Selec(t)ed Profile / [(V)iew Changes] / View Changes b/w (C)lean profiles / Abo(r)t
Writing updated profile for /usr/lib/snapd/snap-confine.
```

Проверим корректное выполнение программы и вывод aa-logprof

```
ubuntu@ubuntu:~$ ./file
UNIX Lab 2
ubuntu@ubuntu:~$ sudo aa-logprof
Updating AppArmor profiles in /etc/apparmor.d.
Reading log entries from /var/log/syslog.
Complain-mode changes:
Enforce-mode changes:
```

5) В программе, измените местоположение создаваемого файла с /log на /logs.

```
ubuntu@ubuntu:~$ cat file
#!/usr/bin/bash
touch logs/myfile.txt
echo "UNIX Lab 2" > logs/myfile.txt
cat logs/myfile.txt
rm logs/myfile.txt
```

6) Создайте директорию logs. Запустите программу, покажите, что AppArmor блокирует попытку получить доступ к пути за пределами границ.

```
ubuntu@ubuntu:~$ mkdir logs
ubuntu@ubuntu:~$ ./file
touch: cannot touch 'logs/myfile.txt': Permission denied
./file: line 3: logs/myfile.txt: Permission denied
cat: logs/myfile.txt: No such file or directory
rm: cannot remove 'logs/myfile.txt': No such file or directory
```

7) Верните изначальное значение /log. Покажите, что программа работает корректно.

```
ubuntu@ubuntu:~$ gedit file
ubuntu@ubuntu:~$ cat file
#!/usr/bin/bash
touch log/myfile.txt
echo "UNIX Lab 2" > log/myfile.txt
cat log/myfile.txt
rm log/myfile.txt
ubuntu@ubuntu:~$ ./file
UNIX Lab 2
ubuntu@ubuntu:~$
```

8) Отключите и удалите профиль безопасности из системы.

```
ubuntu@ubuntu:~$ sudo aa-disable /etc/apparmor.d/home.ubuntu.file
Disabling /etc/apparmor.d/home.ubuntu.file.
ubuntu@ubuntu:~$ sudo rm /etc/apparmor.d/home.ubuntu.file
ubuntu@ubuntu:~$ ./file
UNIX Lab 2
ubuntu@ubuntu:~$ gedit file
ubuntu@ubuntu:~$ cat file
#!/usr/bin/bash
touch logs/myfile.txt
echo "UNIX Lab 2" > logs/myfile.txt
cat logs/myfile.txt
rm logs/myfile.txt
ubuntu@ubuntu:~$ ./file
UNIX Lab 2
```

Дополнительная часть:

1) Опишите отличия SELinux vs AppArmor?

И SELinux, и AppArmor используют мандатная модель управления доступом (Mandatory Access Control, MAC)

AppArmor:

- Модель безопасности основана на профилях приложений и путях к файлам. Для каждой программы создается профиль, который содержит правила, определяющие, к каким файлам, сетевым ресурсам и другим системным ресурсам программа может получить доступ.
- Для обеспечения безопасности использует схему принудительного применения типов, которое определяет, может ли процесс, работающий с определенным типом, получить доступ к файлу, помеченному определенным типом.
- Имеет два основных режима для профилей: Enforce (принудительный) и Complain (жалобный).
- Чтобы начать работать достаточно установить утилиту

SELinux:

- Модель безопасности основана на типах и контекстах безопасности файлов. Каждому объекту присваивается контекст безопасности, который включает информацию о типе, роли, пользователе и уровне безопасности.
- Используемые схемы безопасности: как многоуровневая система безопасности, так и много категорийная система безопасности
- Имеет три режима: Enforcing (принудительный), Permissive (разрешающий), Disabled (отключенный).
- Чтобы начать работать надо настроить контексты и метки

2) Опишите режимы профилей Enforce и Complain? Их различия для чего нужны?

Режим Enforce

- Если приложение пытается выполнить действие, которое запрещено профилем, это действие будет блокировано, то есть профили AppArmor применяются строго
- Основная цель этого режима — обеспечить защиту системы. Он предотвращает выполнение небезопасных операций, даже если приложение содержит уязвимости или было скомпрометировано.

Режим Complain

- В этом режиме AppArmor не блокирует действия, которые нарушают правила профиля, все нарушения только записываются в журналы
- Основная цель этого режима — отладка и тестирование профилей. Он позволяет вам наблюдать за поведением приложения и понять, какие действия оно выполняет, чтобы корректно настроить профиль.

Вывод:

Я познакомилась с утилитой AppArmor, создала при помощи нее профиль безопасности. Помимо этого я узнала различия утилит AppArmor и SELinux и режимов Enforce и Complain в AppArmor.