Информационная безопасность. Лабораторная работа №2. Политики безопасности Linux.

Группа:	P34121
Студенты:	Гиниятуллин Арслан Рафаилович
Преподаватель:	Маркина Татьяна Анатольевна

Санкт-Петербург
2024

Содержание

1 Требования к выполнению работы

Примечание: выполнение всех пунктов лабораторной должно сопровождаться скриншотами с результатами работы команд и изменений в конфигурационных файлах.

2 Текст задания

2.1 Основная часть

2.1.1 Этап 1. Написание скрипта

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.

2.1.2 Этап 2. Запуск скрипта

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

2.1.3 Этап 3. Создание профиля безопасности

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

2.1.4 Этап 4. Настройка разрешений

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

2.1.5 Этап 5. Изменить местоположение создаваемого файла

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

2.1.6 Этап 6. Проверка блокирования доступа AppArmor

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

2.1.7 Этап 7. Проверка работоспособности программы до изменений

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

2.1.8 Этап 8. Отключение и удаление профиля

Отключите и удалите профиль безопасности из системы.

2.2 Дополнительная часть

2.2.1

Опишите отличия SELinux vs AppArmor?

2.2.2

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

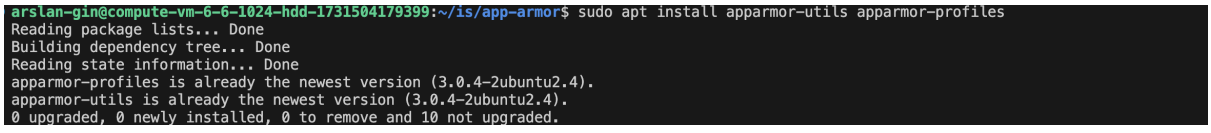
3 Этапы выполнения работы

3.1 Основная часть

3.1.1 Этап 1. Написание скрипта

Для начала установим утилиту AppArmor.

```
sudo apt install apparmor-utils apparmor-profiles
```



Далее напишем bash-скрипт, который будет создавать файл в директории log , записывать в него что-то, читать из него и затем удалять.

```
#!/bin/bash

log_dir="./log"
mkdir -p "$log_dir"
log_file="$log_dir/sample.log"

echo "Это тестовая запись в файл журнала" > "$log_file"

echo "Новое содержимое файла" > "$log_file"
echo "Данные записаны в файл $log_file"

echo "Содержимое файла $log_file:"

cat "$log_file"

rm "$log_file"
~
```

3.1.2 Этап 2. Запуск скрипта

Создадим директорию *log*, выдадим права на исполнение и запустим скрипт.

```
chmod +x file;
./file;
```

```
arslan-gin@compute-vm-6-6-1024-hdd-1731504179399:~/is/app-armor$ chmod +x file;
./file;
Данные записаны в файл ./log/sample.log
Содержимое файла ./log/sample.log:
Новое содержимое файла
```

3.1.3 Этап 3. Создание профиля безопасности

Создадим профиль безопасности для данной программы.

```
sudo aa-logprof ./file
```

```
arslan-gin@compute-vm-6-6-1024-hdd-1731504179399:~/is/app-armor$ sudo aa-genprof ./file
Updating AppArmor profiles in /etc/apparmor.d.
Writing updated profile for /home/arslan-gin/is/app-armor/file.
Setting /home/arslan-gin/is/app-armor/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/arslan-gin/is/app-armor/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
```

Теперь проверим результат выполнения программы.

```
arslan-gin@compute-vm-6-6-1024-hdd-1731504179399:~/is/app-armor$ ./file
./file: line 4: /usr/bin/mkdir: Permission denied
./file: line 7: ./log/sample.log: Permission denied
./file: line 9: ./log/sample.log: Permission denied
Данные записаны в файл ./log/sample.log
Содержимое файла ./log/sample.log:
./file: line 13: /usr/bin/cat: Permission denied
./file: line 15: /usr/bin/rm: Permission denied
```

3.1.4 Этап 4. Настройка разрешений

Модифицируем профиль из логов так, чтобы при дальнейшем запуске *./file* отрабатывал.

```
sudo aa-logprof
```

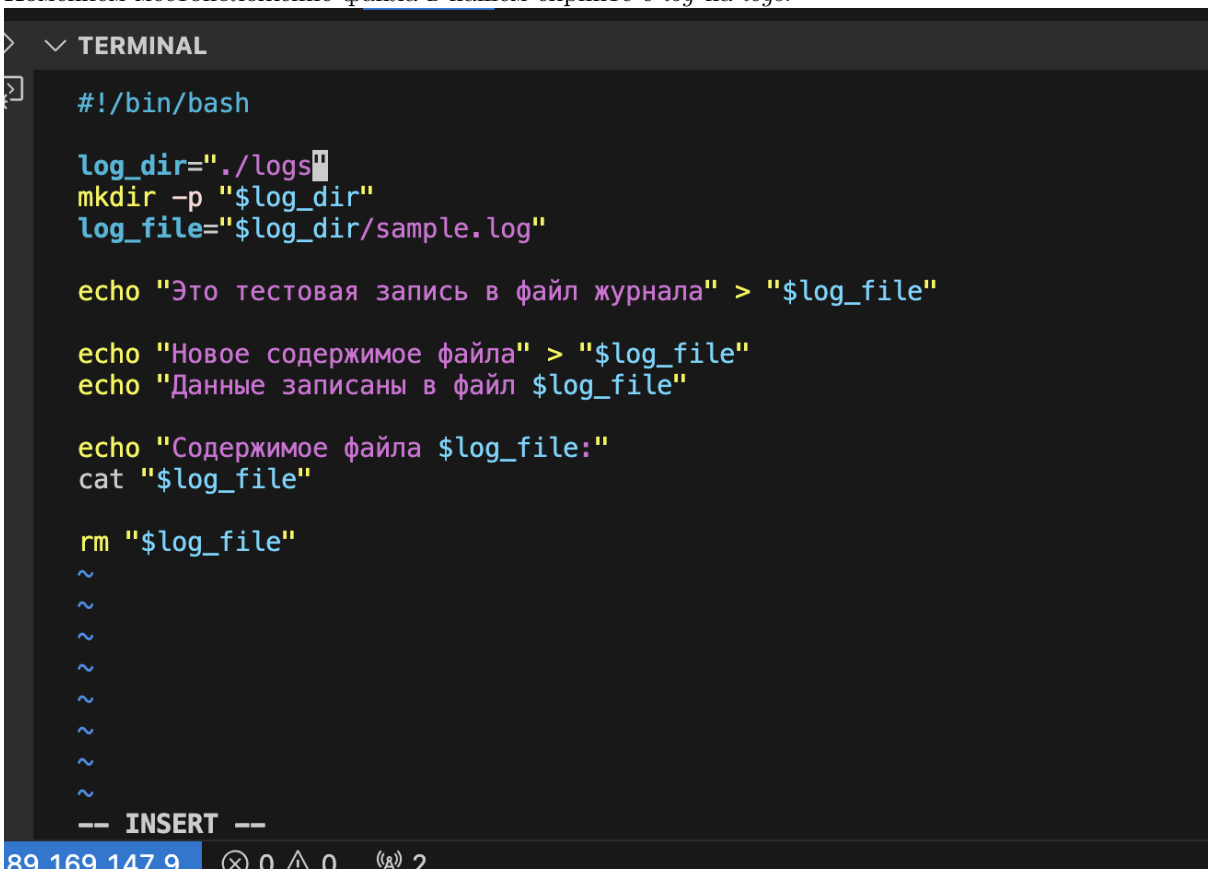
```
[1 - include <abstractions/consoles>]
  2 - /dev/tty rw,
(A)llow / [(D)eny] / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Audi(t) / Abo(r)t / (F)inish
Adding include <abstractions/consoles> to profile.

Profile:  /home/arslan-gin/is/app-armor/file
Path:     /home/arslan-gin/is/app-armor/log/sample.log
New Mode: owner w
Severity: 6

 [1 - owner /home/*/is/app-armor/log/sample.log w,]
  2 - owner /home/arslan-gin/is/app-armor/log/sample.log 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/*/is/app-armor/log/sample.log w, to profile.

= Changed Local Profiles =

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

 [1 - /home/arslan-gin/is/app-armor/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/arslan-gin/is/app-armor/file.
arslan-gin@compute-vm-6-6-1024-hdd-1731504179399:~/is/app-armor$
```

Теперь проверим результат выполнения программы. Видим, что ошибок нет.

```
arslan-gin@compute-vm-6-6-1024-hdd-1731504179399:~/is/app-armor$ ./file
Данные записаны в файл ./log/sample.log
Содержимое файла ./log/sample.log:
Новое содержимое файла
arslan-gin@compute-vm-6-6-1024-hdd-1731504179399:~/is/app-armor$
```

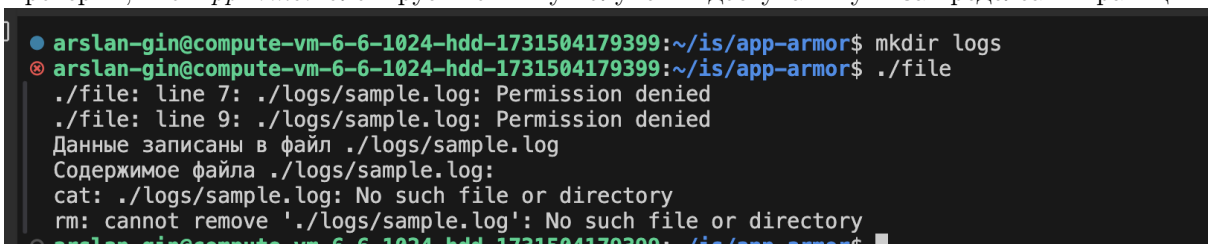
3.1.5 Этап 5. Изменить местоположение создаваемого файла

Поменяем местоположение файла в нашем скрипте с *log* на *logs*.



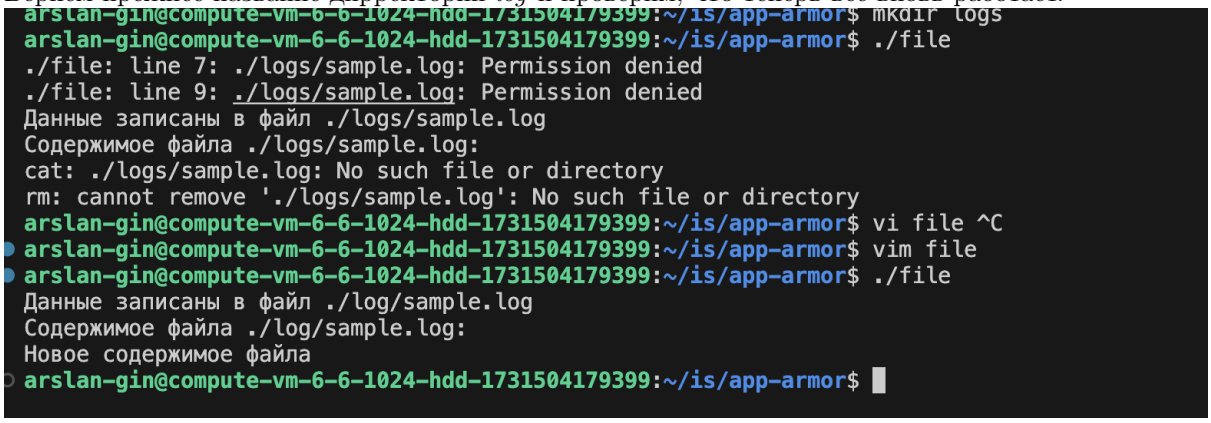
3.1.6 Этап 6. Проверка блокирования доступа AppArmor

Проверим, что *AppArmor* блокирует попытку получения доступа к пути за пределеами границ.



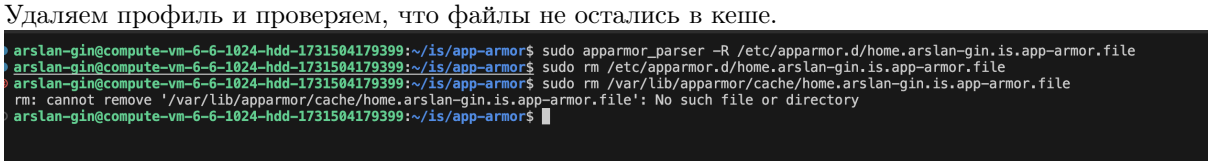
3.1.7 Этап 7. Проверка работоспособности программы до изменений

Вернем прежнее название дирректории *log* и проверим, что теперь все вновь работает.



3.1.8 Этап 8. Отключение и удаление профиля

Удаляем профиль и проверяем, что файлы не остались в кеше.



4 Дополнительная часть

4.1 Различия между SELinux и AppArmor

SELinux

- **Модель безопасности:** SELinux основывается на концепции меточных списков управления доступом (MAC), что позволяет точно определять, какие действия имеют право выполнять приложения и процессы.
- **Контейнеры:** Отлично работает с контейнерами, такими как Docker и Kubernetes, благодаря эффективному управлению процессами.
- **Политики:** Довольно сложные.

AppArmor

- **Модель безопасности:** Основывается на профилях, ориентированных на путь, что делает его менее подробным, но достаточно эффективным.
- **Простота использования:** Более интуитивно понятен в настройке, профили созданы на основе доступа к файлам и сервисам.
- **Интеграция:** Легче интегрируется с существующими системами благодаря возможности автогенерации профилей.
- **Политики:** Профили менее детализированы и чаще всего определяются для конкретных приложений на основе стандартных путей.

4.2 Режимы профилей Enforce и Complain

4.2.1 Режим Enforce

- **Принцип работы:** Политика безопасности применяется строго, и любые действия, нарушающие правила, блокируются.

- **Использование:** Подходит для систем, где безопасность имеет первостепенное значение и политика уже протестирована.
- **Преимущества:** Нарушающие правила будут заблокированы.
- **Недостатки:** Возможны неожиданные блокировки, если политика еще не выверена.

Режим Complain

- **Принцип работы:** Система не блокирует запрещенные действия, но регистрирует их в логах, что позволяет увидеть нарушения без влияния на работу приложений.
- **Использование:** Полезен для тестирования и разработки политик без риска остановки важного процесса.
- **Преимущества:** Позволяет настраивать и оттачивать политику безопасности, минимизируя прерывания работы системы.
- **Недостатки:** Блокировок запрещенных действий не произойдет, будут писаться только логи.

5 Заключение

В ходе выполнения данной лабораторной работы я освоил управление доступом к файлам с помощью *AppArmor*.