

Отчёт по лабораторной работе №6

Мандатное разграничение прав в Linux

Саргсян Арам Грачьевич

Содержание

1	Цель работы	5
2	Теоретическое введение	6
2.1	SELinux	6
2.2	Apache	7
3	Выполнение лабораторной работы	9
4	Выводы	15
	Список литературы	16

Список иллюстраций

3.1	Запуск сервера	9
3.2	Параметр ServerName	9
3.3	Команды getenforce и sestatus	10
3.4	Запуск apache	10
3.5	Контекст безопасности	10
3.6	Состояние переключателей SELinux	11
3.7	Статистика по политике	11
3.8	Типы файлов и поддиректории	12
3.9	Запуск в браузере	12
3.10	Изменение контекста безопасности	13
3.11	Запуск в браузере с ошибкой	13
3.12	Лог файлы	13
3.13	Изменение порта	14
3.14	Настройки	14
3.15	Открытие файла	14

Список таблиц

1 Цель работы

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux. Проверить работу SELinux на практике совместно с веб-сервером Apache.

2 Теоретическое введение

2.1 SELinux

SELinux (Security-Enhanced Linux) - это набор обязательных контролов доступа (MAC), разработанных для улучшения безопасности операционных систем на базе ядра Linux. SELinux предоставляет дополнительные уровни безопасности, которые работают в дополнение к стандартным системам управления доступом, таким как управление правами доступа (DAC - discretionary access control).

Основные особенности SELinux:

1. Принудительный контроль доступа (MAC): SELinux предоставляет механизм, который определяет, к каким ресурсам и операциям пользователи и процессы имеют доступ. В отличие от системы управления доступом на основе прав доступа (DAC), где пользователи могут управлять своими файлами и процессами, SELinux предписывает жесткие правила доступа на уровне ядра.
2. Политики безопасности: SELinux использует политики безопасности, определяющие, какие действия разрешены для различных объектов и субъектов (пользователей и процессов). Политики могут быть настроены и настраиваться в зависимости от потребностей системы.
3. Роли и контексты: В SELinux каждому процессу и ресурсу назначаются контексты безопасности, которые определяют его роль и права доступа. Это помогает изолировать процессы и уменьшает риск распространения атак.

4. Проверка соблюдения политик: SELinux постоянно проверяет соблюдение политик безопасности и блокирует доступ, который нарушает эти политики. Это повышает уровень безопасности, предотвращая многие типичные уязвимости.
5. Гибкость настройки: SELinux позволяет администраторам настраивать политики безопасности под конкретные потребности системы, создавать собственные политики и определять, какие действия разрешены и какие запрещены.
6. Аудит и журналирование: SELinux обеспечивает детализированный аудит и журналирование событий, что позволяет администраторам исследовать инциденты безопасности и выявлять аномалии.

SELinux является мощным инструментом для улучшения безопасности Linux-систем, но его конфигурация может быть сложной и требовать понимания принципов безопасности. В большинстве дистрибутивов Linux SELinux предоставляется как опция, и его активация и настройка зависят от конкретных потребностей системы и уровня безопасности [1].

2.2 Apache

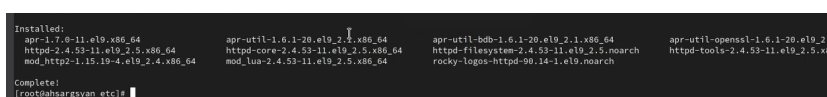
Apache, также известный как Apache HTTP Server, - это свободный и открытый веб-сервер, разработанный Apache Software Foundation. Этот веб-сервер является одним из самых популярных и широко используемых в мире, и он играет ключевую роль в инфраструктуре множества веб-сайтов и приложений. Вот некоторые основные характеристики и функции Apache:

1. Сервер статических и динамических контентов: Apache способен обслуживать как статические веб-страницы, так и динамические, включая страницы, создаваемые с использованием языков программирования, таких как PHP, Python, и Ruby.

2. Модульная архитектура: Apache использует модульную архитектуру, которая позволяет добавлять и настраивать разнообразные функциональные возможности, такие как аутентификация, шифрование, сжатие и многое другое с помощью модулей.
3. Открытое ПО: Apache является свободным программным обеспечением с открытым исходным кодом, что означает, что его исходный код доступен для общественности для просмотра, изменения и распространения в соответствии с лицензией Apache.
4. Поддержка множества протоколов: Apache поддерживает множество сетевых протоколов, включая HTTP, HTTPS (через модуль SSL/TLS), и другие протоколы, что делает его универсальным инструментом для обслуживания разнообразных веб-приложений.
5. Виртуальные хосты: Apache поддерживает конфигурацию виртуальных хостов, что позволяет hostить несколько сайтов на одном сервере с разными доменными именами и настройками.
6. Безопасность: Apache предоставляет множество механизмов для обеспечения безопасности, включая возможность настройки правил доступа, аутентификации и шифрования данных.
7. Логирование: Apache генерирует лог-файлы, которые записывают информацию о запросах, ошибках и активности сервера, что полезно для мониторинга и анализа.
8. Apache используется множеством организаций и индивидуальных разработчиков для развертывания веб-сайтов и веб-приложений на серверах Linux и других операционных системах. Благодаря обширному сообществу и богатой документации, Apache остается одним из наиболее надежных и гибких веб-серверов [2].

3 Выполнение лабораторной работы

1. Установил веб-сервис apache для дальнейшей работы (рис. 3.1).

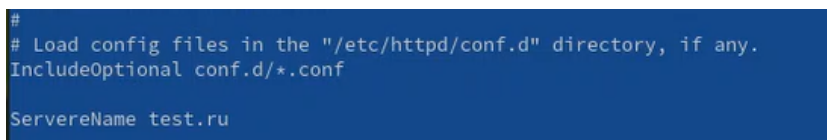


```
Installed:
apr-1.7.0-11.el9.x86_64      apr-util-1.6.1-20.el9_2.1.x86_64      apr-util-bdb-1.6.1-20.el9_2.1.x86_64      apr-util-openssl-1.6.1-20.el9_2.1.
httpd-2.4.53-11.el9.x86_64  httpd-core-2.4.53-11.el9_2.5.x86_64      httpdfilesystem-2.4.53-11.el9_2.5.noarch      httpd-tools-2.4.53-11.el9_2.5.x86_
mod_http2-1.15.19-4.el9_2.4.x86_64  mod_lua-2.4.53-11.el9_2.5.x86_64      rocky-logos-httpd-90.14-1.el9.noarch

Complete!
[root@ahsargyan etc]#
```

Рис. 3.1: Запуск сервера

2. В конфигурационном файле `/etc/httpd/httpd.conf` задал параметр `ServerName` (рис. 3.2).



```
#
# Load config files in the "/etc/httpd/conf.d" directory, if any.
IncludeOptional conf.d/*.conf

ServerName test.ru
```

Рис. 3.2: Параметр ServerName

3. Вошел в систему с полученными учётными данными и убедился, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` и `sestatus`. Запустил веб-сервис (рис. 3.3, 3.4).

```
guest@ahsargsyan:/ root@ahsargsyan:/home/ahsargsyan guest2@ahsargsyan
[ahsargsyan@ahsargsyan ~]$ service httpd status
Redirecting to /bin/systemctl status httpd.service
o httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: inactive (dead)
   Docs: man:httpd.service(8)
[ahsargsyan@ahsargsyan ~]$ service httpd start
Redirecting to /bin/systemctl start httpd.service
Job for httpd.service failed because the control process exited with error code.
See "systemctl status httpd.service" and "journalctl -xeu httpd.service" for details.
[ahsargsyan@ahsargsyan ~]$ getenforce
Enforcing
[ahsargsyan@ahsargsyan ~]$ 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 deny_unknown status: allowed
Memory protection checking: actual (secure)
Max kernel policy version: 33
[ahsargsyan@ahsargsyan ~]$
```

Рис. 3.3: Команды getenforce и sestatus

```
Job for httpd.service failed because the control process exited with error code.
See "systemctl status httpd.service" and "journalctl -xeu httpd.service" for details.
[ahsargsyan@ahsargsyan ~]$ service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ahsargsyan@ahsargsyan ~]$ service httpd status
Redirecting to /bin/systemctl status httpd.service
o httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: active (running) since Sun 2023-10-01 13:05:50 MSK; 6s ago
   Docs: man:httpd.service(8)
   Main PID: 4724 (httpd)
   Status: "Started, listening on: port 80"
   Tasks: 213 (limit: 24611)
   Memory: 41.2M
   CPU: 116ms
   CGroup: /system.slice/httpd.service
           └─4724 /usr/sbin/httpd -DFOREGROUND
             └─4733 /usr/sbin/httpd -DFOREGROUND
               └─4734 /usr/sbin/httpd -DFOREGROUND
                 └─4735 /usr/sbin/httpd -DFOREGROUND
                   └─4736 /usr/sbin/httpd -DFOREGROUND

Oct 01 13:05:50 ahsargsyan.localadmin systemd[1]: Starting The Apache HTTP Server...
Oct 01 13:05:50 ahsargsyan.localadmin httpd[4724]: Server configured, listening on: port 80
Oct 01 13:05:50 ahsargsyan.localadmin systemd[1]: Started The Apache HTTP Server.
[ahsargsyan@ahsargsyan ~]$
```

Рис. 3.4: Запуск apache

4. Определил его контекст безопасности (рис. 3.5)

```
[root@ahsargsyan ahsargsyan]# ps aux | grep httpd
system_u:system_r:httpd_t:s0 root      4724  0.0  0.2 20116 11356 ?        Ss   13:05   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  4733  0.0  0.1 21600 7228 ?        S    13:05   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  4734  0.0  0.4 2521228 19184 ?    Sl   13:05   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  4735  0.0  0.4 2259020 17140 ?    Sl   13:05   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  4736  0.0  0.4 2324556 17140 ?    Sl   13:05   0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-c0:c0:c1023 root    5004  0.0  0.0 221664 2252 pts/2  S+   13:07   0:00 grep --color=auto httpd
[root@ahsargsyan ahsargsyan]# sestatus -bigrep httpd
```

Рис. 3.5: Контекст безопасности

5. Посмотрел текущее состояние переключателей SELinux (рис. 3.6)

```
Without options, show SELinux status.
[root@ahsargsyan ahsargsyan]# sestatus -b | grep httpd
httpd_anon_write                off
httpd_builtin_scripting         on
httpd_can_check_spam             off
httpd_can_connect_ftp            off
httpd_can_connect_ldap           off
httpd_can_connect_mythtv        off
httpd_can_connect_zabbix        off
httpd_can_manage_courier_spool   off
httpd_can_network_connect       off
httpd_can_network_connect_cobbler off
httpd_can_network_connect_db     off
httpd_can_network_memcache      off
httpd_can_network_relay         off
httpd_can_sendmail              off
httpd_dbus_avahi                off
httpd_dbus_sssd                 off
httpd_dontaudit_search_dirs     off
httpd_enable_cgi                on
httpd_enable_ftp_server         off
httpd_enable_homedirs           off
httpd_execmem                   off
httpd_manage_ftp                off
```

Рис. 3.6: Состояние переключателей SELinux

6. Посмотрел статистику по политике с помощью команды seinfo (рис. 3.7)

```
Statistics for policy file: /sys/fs/selinux/policy
Policy Version:          33 (MLS enabled)
Target Policy:           selinux
Handle unknown classes:  allow
Classes:                 135
Sensitivities:           1
Types:                   5100
Users:                   8
Booleans:                353
Allow:                   65008
Auditallow:              170
Type_trans:              265344
Type_member:             35
Role allow:              38
Constraints:             70
MLS Constrains:          72
Permissives:             2
Defaults:                7
Allowxperm:              0
Auditallowxperm:         0
Ibendportcon:            0
Initial SIDs:            27
Genfscon:                109
Netifcon:                0
Permissions:             457
Categories:              1024
Attributes:              258
Roles:                   14
Cond. Expr.:             384
Neverallow:              0
Dontaudit:               8572
Type_change:             87
Range_trans:             6164
Role_trans:              420
Validatetrans:           0
MLS Val. Tran:           0
Polcap:                  6
Typebounds:              0
Neverallowxperm:         0
Dontauditxperm:          0
Ibpkeycon:               0
Fs_use:                  35
Portcon:                 660
Nodecon:                 0
```

Рис. 3.7: Статистика по политике

7. Определил тип файлов и поддиректорий, находящихся в директории /var/www, определил тип файлов, находящихся в директории /var/www/html, определил круг пользователей, которым разрешено создание файлов в директории /var/www/html. Создал от имени суперпользователя html-файл test.html. Проверил контекст созданного вами файла. Занесите в отчёт контекст, присваиваемый по умолчанию вновь созданным файлам в директории /var/www/html. Обратился к файлу через веб-сервер (рис. 3.8, 3.9)

```
[root@ahsargsyan ahsargsyan]# ls -lZ /var/www/html
total 0
[root@ahsargsyan ahsargsyan]# touch /var/www/html/test.html
[root@ahsargsyan ahsargsyan]#
[root@ahsargsyan www]# cat /var/www/html/test.html
<html>
<body>test</body>
</html>[root@ahsargsyan]# ls -lZ /var/www/html
total 4
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 32 Oct  1 13:13 test.ht
[root@ahsargsyan www]#
```

Рис. 3.8: Типы файлов и поддиректории

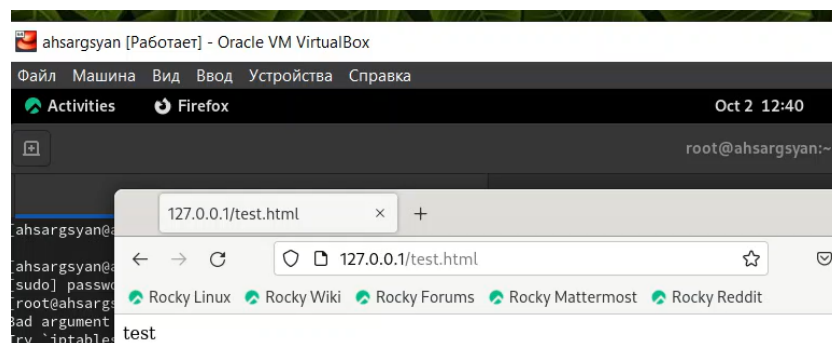


Рис. 3.9: Запуск в браузере

8. Изменил контекст файла /var/www/html/test.html с httpd_sys_content_t на любой другой, к которому процесс httpd не должен иметь доступа, попробовал ещё раз получить доступ к файлу через веб-сервер (рис. 3.10, 3.11).

```
[root@ahsargsyan ~]# ls -Z /var/www/html/test.html
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test.html
[root@ahsargsyan ~]# chcon -t samba_share_t /var/www/html/test.html
[root@ahsargsyan ~]# ls -Z /var/www/html/test.html
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html
[root@ahsargsyan ~]#
```

Рис. 3.10: Изменение контекста безопасности

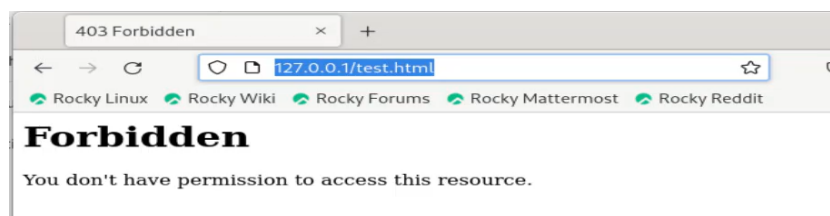


Рис. 3.11: Запуск в браузере с ошибкой

9. Просмотрел log-файлы веб-сервера Apache (рис. 3.12)

```
[root@ahsargsyan ~]# ls -l /var/www/html/test.html
-rw-r--r--. 1 root root 32 Oct  1 13:13 /var/www/html/test.html
[root@ahsargsyan ~]# tail /var/log/messages
Oct  2 12:44:41 ahsargsyan setroubleshoot[3848]: SELinux is preventing /usr/sbin/httpd from getattr access on the file /
on (92.2 confidence) suggests *****#012#012If you want to fix the label. #012/var/www/html/test.htm
n you can run restorecon. The access attempt may have been stopped due to insufficient permissions to access a parent di
mand accordingly.#012Do#012# /sbin/restorecon -v /var/www/html/test.html#012#012***** Plugin public_content (7.83 confi
want to treat test.html as public content#012Then you need to change the label on test.html to public_content_t or publi
blic_content_t '/var/www/html/test.html'#012# restorecon -v '/var/www/html/test.html'#012#012***** Plugin catchall (1.4
12#012If you believe that httpd should be allowed getattr access on the test.html file by default.#012Then you should re
y module to allow this access.#012Do#012allow this access for now by executing:#012# ausearch -c 'httpd' --raw | audit2a
012
Oct  2 12:44:50 ahsargsyan setroubleshoot[3848]: SELinux is preventing /usr/sbin/httpd from getattr access on the file /
run: sealert -l 47453959-fd4c-4ea2-8766-9619e953f32b
Oct  2 12:44:50 ahsargsyan setroubleshoot[3848]: SELinux is preventing /usr/sbin/httpd from getattr access on the file /
on (92.2 confidence) suggests *****#012#012If you want to fix the label. #012/var/www/html/test.htm
n you can run restorecon. The access attempt may have been stopped due to insufficient permissions to access a parent di
mand accordingly.#012Do#012# /sbin/restorecon -v /var/www/html/test.html#012#012***** Plugin public_content (7.83 confi
want to treat test.html as public content#012Then you need to change the label on test.html to public_content_t or publi
blic_content_t '/var/www/html/test.html'#012# restorecon -v '/var/www/html/test.html'#012#012***** Plugin catchall (1.4
12#012If you believe that httpd should be allowed getattr access on the test.html file by default.#012Then you should re
y module to allow this access.#012Do#012allow this access for now by executing:#012# ausearch -c 'httpd' --raw | audit2a
012
Oct  2 12:44:51 ahsargsyan setroubleshoot[3848]: SELinux is preventing /usr/sbin/httpd from getattr access on the file /
run: sealert -l 47453959-fd4c-4ea2-8766-9619e953f32b
Oct  2 12:44:51 ahsargsyan setroubleshoot[3848]: SELinux is preventing /usr/sbin/httpd from getattr access on the file /
on (92.2 confidence) suggests *****#012#012If you want to fix the label. #012/var/www/html/test.htm
n you can run restorecon. The access attempt may have been stopped due to insufficient permissions to access a parent di
mand accordingly.#012Do#012# /sbin/restorecon -v /var/www/html/test.html#012#012***** Plugin public_content (7.83 confi
want to treat test.html as public content#012Then you need to change the label on test.html to public_content_t or publi
blic_content_t '/var/www/html/test.html'#012# restorecon -v '/var/www/html/te
```

Рис. 3.12: Лог файлы

10. Открыл файл через 81 порт (рис. 3.13, 3.14, 3.15)

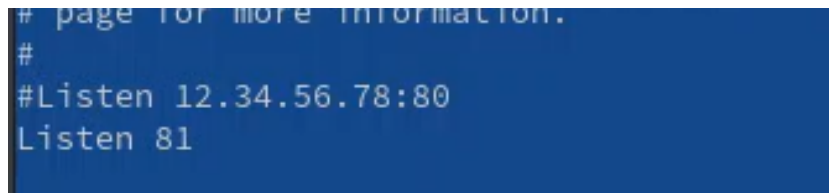


Рис. 3.13: Изменение порта

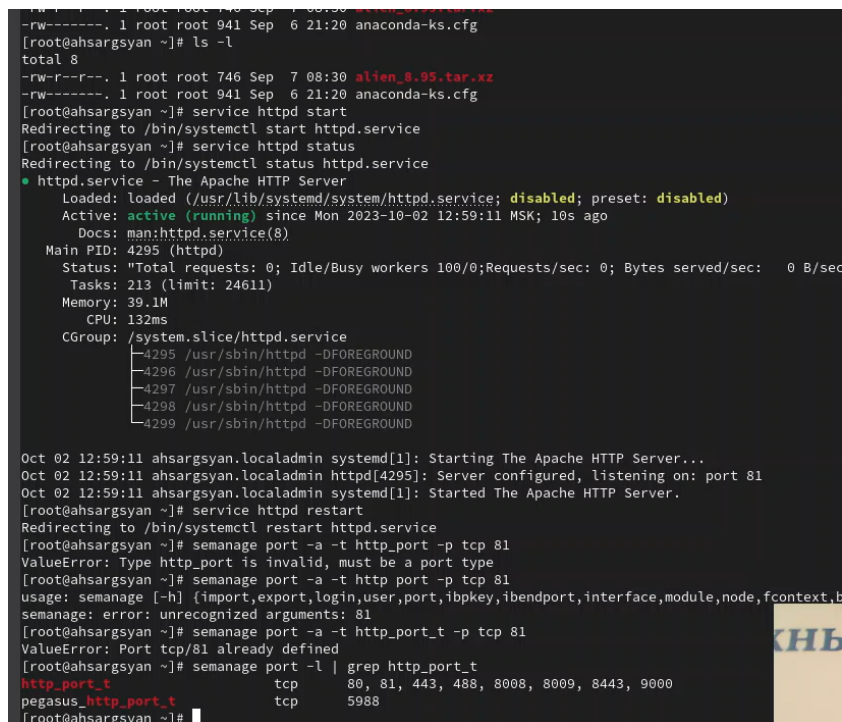


Рис. 3.14: Настройки

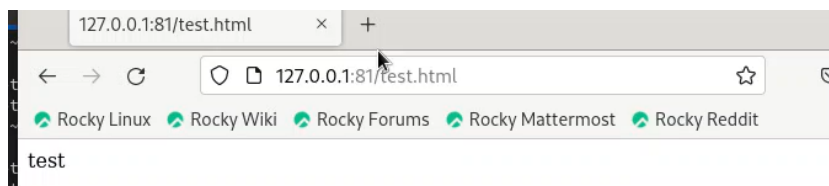


Рис. 3.15: Открытие файла

4 Выводы

Я развил навыки администрирования ОС Linux. Получил первое практическое знакомство с технологией SELinux, а также проверил работу SELinux на практике совместно с веб-сервером Apache.

Список литературы

1. Xu W., Shehab M., Ahn G.-J. Visualization based policy analysis // Proceedings of the 13th ACM symposium on Access control models and technologies. ACM, 2008.
2. Laurie B. Apache. 3rd ed / под ред. Laurie P. Sebastopol: O'Reilly Media, Inc, 2007. 608 с.