

# **Отчёт по лабораторной работе №6**

**Знакомство с SELinux**

Худдыева Дженнет

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>5</b>
2.1	Подготовка . . . . .	5
2.2	Изучение механики SetUID . . . . .	5
<b>3</b>	<b>Выводы</b>	<b>13</b>
	<b>Список литературы</b>	<b>14</b>

# List of Figures

2.1	запуск http . . . . .	6
2.2	контекст безопасности http . . . . .	6
2.3	переключатели SELinux для http . . . . .	7
2.4	создание html-файла и доступ по http . . . . .	8
2.5	ошибка доступа после изменения контекста . . . . .	9
2.6	лог ошибок . . . . .	10
2.7	переключение порта . . . . .	11
2.8	доступ по http на 81 порт . . . . .	12

# 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 Выполнение лабораторной работы

### 2.1 Подготовка

1. Установили httpd
2. Задали имя сервера
3. Открыли порты для работы с протоколом http

### 2.2 Изучение механики SetUID

1. Войдите в систему с полученными учётными данными и убедитесь, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` и `sestatus`.
2. Обратитесь с помощью браузера к веб-серверу, запущенному на вашем компьютере, и убедитесь, что последний работает: `service httpd status` или `/etc/rc.d/init.d/httpd status` Если не работает, запустите его так же, но с параметром `start`.

```

root@jkhuddiheva ~]# sudo systemctl enable httpd
root@jkhuddiheva ~]# service httpd status
Redirecting to /bin/systemctl status httpd.service
* httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Active: active (running) since Mon 2025-09-15 02:14:15 MSK; 6min ago
     Docs: man:httpd.service(8)
    Main PID: 128038 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served: 0; CPU times: 0s; Uptime: 0s"
   Tasks: 177 (limit: 24615)
  Memory: 21.7M
    CPU: 1.881s
   CGroup: /system.slice/httpd.service
            └─128038 /usr/sbin/httpd -DFOREGROUND
               └─128039 /usr/sbin/httpd -DFOREGROUND
                  └─128040 /usr/sbin/httpd -DFOREGROUND
                     └─128041 /usr/sbin/httpd -DFOREGROUND
                        └─128042 /usr/sbin/httpd -DFOREGROUND

Sep 15 02:14:15 jkhuddiheva.localdomain systemd[1]: Starting The Apache HTTP Server:
Sep 15 02:14:15 jkhuddiheva.localdomain httpd[128038]: Server configured, listening on
Sep 15 02:14:15 jkhuddiheva.localdomain systemd[1]: Started The Apache HTTP Server:

```

Figure 2.1: запуск http

3. Найдите веб-сервер Apache в списке процессов, определите его контекст безопасности и занесите эту информацию в отчёт. Например, можно использовать команду `ps auxZ | grep httpd` или `ps -eZ | grep httpd`

```

system_u:system_r:httpd_t:s0 apache 128042 0.1 0.2 1441056 10584 ?
Sl 02:14 0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 root 128430 0.0 0.0 22168
2560 pts/0 S+ 02:22 0:00 grep --color=auto httpd
root@jkhuddiheva ~]# sestatus -bigrep httpd
sestatus: invalid option -- 'i'

Usage: sestatus [OPTION]

  -v Verbose check of process and file contexts.
  -b Display current state of booleans.

Without options, show SELinux status.
root@jkhuddiheva ~]# sestatus -b httpd
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

```

Figure 2.2: контекст безопасности http

4. Посмотрите текущее состояние переключателей SELinux для Apache с помощью команды `sestatus -bigrep httpd`. Обратите внимание, что многие из них находятся в положении «off».

<b>httpd_can_network_connect_cobbler</b>	off
<b>httpd_can_network_connect_db</b>	off
<b>httpd_can_network_memcache</b>	off
<b>httpd_can_network_redis</b>	off
<b>httpd_can_network_relay</b>	off
<b>httpd_can_sendmail</b>	off
<b>httpd_dbus_avahi</b>	off
<b>httpd_dbus_sssd</b>	off
<b>httpd_dontaudit_search_dirs</b>	off
<b>httpd_enable_cgi</b>	on
<b>httpd_enable_ftp_server</b>	off
<b>httpd_enable_homedirs</b>	off
<b>httpd_execmem</b>	off
<b>httpd_graceful_shutdown</b>	off
<b>httpd_manage_ipa</b>	off
<b>httpd_mod_auth_ntlm_winbind</b>	off
<b>httpd_mod_auth_pam</b>	off
<b>httpd_read_user_content</b>	off
<b>httpd_run_ipa</b>	off
<b>httpd_run_preupgrade</b>	off
<b>httpd_run_stickshift</b>	off
<b>httpd_serve_cobbler_files</b>	off
<b>httpd_setrlimit</b>	off
<b>httpd_ssi_exec</b>	off
<b>httpd_sys_script_anon_write</b>	off
<b>httpd_tmp_exec</b>	off
<b>httpd_tty_comm</b>	off
<b>httpd_unified</b>	off
<b>httpd_use_cifs</b>	off
<b>httpd_use_fusefs</b>	off
<b>httpd_use_gpg</b>	off
<b>httpd_use_nfs</b>	off
<b>httpd_use_openssl</b>	off

Figure 2.3: переключатели SELinux для http

5. Посмотрите статистику по политике с помощью команды `seinfo`, также определите множество пользователей, ролей, типов.
6. Определите тип файлов и поддиректорий, находящихся в директории `/var/www`, с помощью команды `ls -lZ /var/www`. В поддиректориях могут располагаться системные скрипты и контент для http.
7. Определите тип файлов, находящихся в директории `/var/www/html`: `ls -lZ /var/www/html`. В директории изначально нет файлов.
8. Определите круг пользователей, которым разрешено создание файлов в директории `/var/www/html`. Создавать файлы может только root.

9. Создайте от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись в директорию) html-файл `/var/www/html/test.html` следующего содержания: `Test`
10. Проверьте контекст созданного вами файла. Занесите в отчёт контекст, присваиваемый по умолчанию вновь созданным файлам в директории `/var/www/html`.
11. Обратитесь к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Убедитесь, что файл был успешно отображён.

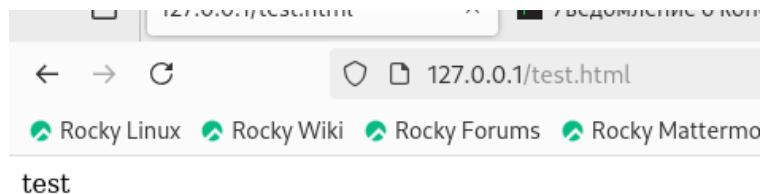


Figure 2.4: создание html-файла и доступ по http

12. Изучите справку `man httpd_selinux` и выясните, какие контексты файлов определены для `httpd`. Сопоставьте их с типом файла `test.html`. Проверить контекст файла можно командой `ls -Z`. `ls -Z /var/www/html/test.html`. Основным контекстом является `httpd_sys_content_t`, его мы и увидели в выводе команды.
13. Измените контекст файла `/var/www/html/test.html` с `httpd_sys_content_t` на любой другой, к которому процесс `httpd` не должен иметь доступа, например, на `samba_share_t`: `chcon -t samba_share_t /var/www/html/test.html` `ls -Z /var/www/html/test.html` После этого проверьте, что контекст поменялся.



14. Попробуйте ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Вы должны получить сообщение об ошибке: `Forbidden You don't have permission to access /test.html on this server`. При изменении контекста файл стал считаться чужим для `http` и программа не может его прочитать.

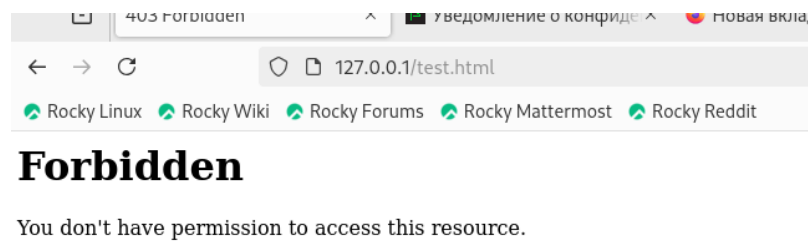


Figure 2.5: ошибка доступа после изменения контекста

15. Проанализируйте ситуацию. Почему файл не был отображён, если права доступа позволяют читать этот файл любому пользователю? `ls -l /var/www/html/test.html` Просмотрите `log`-файлы веб-сервера `Apache`. Также просмотрите системный `лог`-файл: `tail /var/log/messages` Если в системе окажутся запущенными процессы `setroubleshootd` и `audtd`, то вы также сможете увидеть ошибки, аналогичные указанным выше, в файле `/var/log/audit/audit.log`. Проверьте это утверждение самостоятельно.

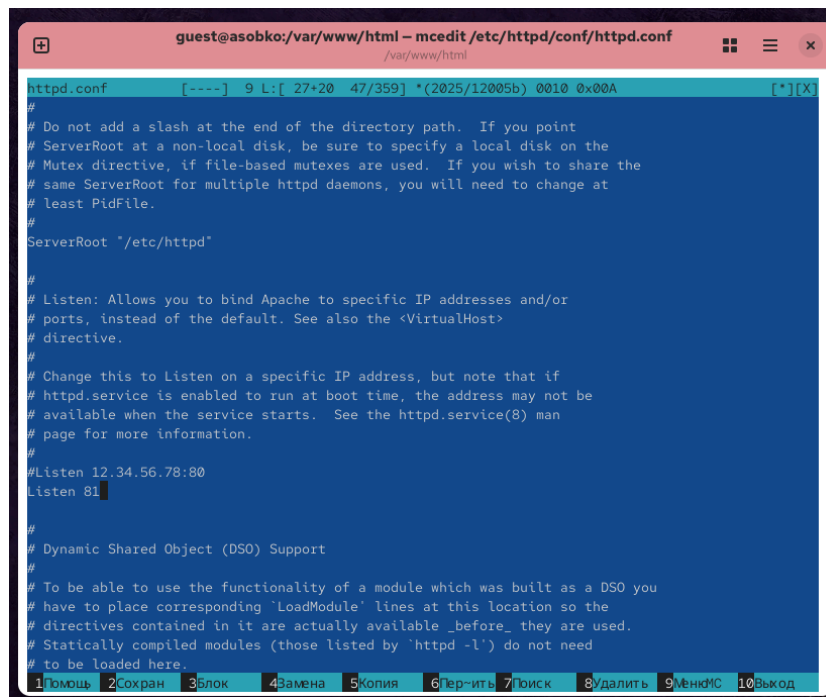
```

ject.SetTroubleshootPrivileged@2 comm="systemd" exe="/usr/lib/systemd/systemd" ho
stname=? addr=? terminal=? res=success'UID="root" AUID="unset"
type=SERVICE_STOP msg=audit(1757893623.438:1395): pid=1 uid=0 auid=4294967295 se
s=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=dbus-1.1-org.fedoraproj
ect.SetTroubleshootPrivileged@2 comm="systemd" exe="/usr/lib/systemd/systemd" hos
tname=? addr=? terminal=? res=success'UID="root" AUID="unset"
type=SERVICE_STOP msg=audit(1757893623.482:1396): pid=1 uid=0 auid=4294967295 se
s=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=setroubleshootd comm="sy
stemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'U
ID="root" AUID="unset"
type=BPF msg=audit(1757893633.979:1397): prog-id=326 op=LOAD
type=BPF msg=audit(1757893633.979:1398): prog-id=327 op=LOAD
type=SERVICE_START msg=audit(1757893634.042:1399): pid=1 uid=0 auid=4294967295 s
s=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=systemd-hostnamed comm=
"systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=succes
s'UID="root" AUID="unset"
type=SERVICE_STOP msg=audit(1757893664.087:1400): pid=1 uid=0 auid=4294967295 se
s=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=systemd-hostnamed comm=
"systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success
'UID="root" AUID="unset"
type=BPF msg=audit(1757893664.115:1401): prog-id=327 op=UNLOAD
type=BPF msg=audit(1757893664.115:1402): prog-id=326 op=UNLOAD

```

Figure 2.6: лог ошибок

16. Попробуйте запустить веб-сервер Apache на прослушивание TCP-порта 81 (а не 80, как рекомендует IANA и прописано в /etc/services). Для этого в файле /etc/httpd/httpd.conf найдите строчку Listen 80 и замените её на Listen 81.



```
httpd.conf [----] 9 L:[ 27+20 47/359] *(2025/12005b) 0010 0x00A [*][X]
#
# Do not add a slash at the end of the directory path. If you point
# ServerRoot at a non-local disk, be sure to specify a local disk on the
# Mutex directive, if file-based mutexes are used. If you wish to share the
# same ServerRoot for multiple httpd daemons, you will need to change at
# least PidFile.
#
ServerRoot "/etc/httpd"
#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on a specific IP address, but note that if
# httpd.service is enabled to run at boot time, the address may not be
# available when the service starts. See the httpd.service(8) man
# page for more information.
#
#Listen 12.34.56.78:80
Listen 81
#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding 'LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by 'httpd -l') do not need
# to be loaded here.
```

Figure 2.7: переключение порта

17. Выполните перезапуск веб-сервера Apache. Произошёл сбой? Поясните почему? Сбой не происходит, порт 81 уже вписан в разрешенные
18. Проанализируйте лог-файлы: `tail -nl /var/log/messages` Просмотрите файлы `/var/log/http/error_log`, `/var/log/http/access_log` и `/var/log/audit/audit.log` и выясните, в каких файлах появились записи.
19. Выполните команду `semanage port -a -t http_port_t -p tcp 81` После этого проверьте список портов командой `semanage port -l | grep http_port_t` Убедитесь, что порт 81 появился в списке.
20. Попробуйте запустить веб-сервер Apache ещё раз.
21. Верните контекст `httpd_sys_content_t` к файлу `/var/www/html/test.html`: `chcon -t httpd_sys_content_t /var/www/html/test.html` После этого попробуйте получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1:81/test.html`. Вы должны увидеть содержимое файла — слово «test».

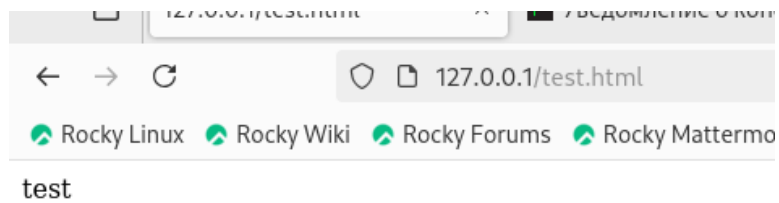


Figure 2.8: доступ по http на 81 порт

22. Исправьте обратно конфигурационный файл `apache`, вернув `Listen 80`.
23. Удалите привязку `http_port_t` к 81 порту: `semanage port -d -t http_port_t -p tcp 81` и проверьте, что порт 81 удалён.
24. Удалите файл `/var/www/html/test.html`: `rm /var/www/html/test.html`

## **3 Выводы**

В процессе выполнения лабораторной работы мною были получены базовые навыки работы с технологией seLinux.

# Список литературы

1. SELinux в CentOS
2. Веб-сервер Apache