

Отчет по лабораторной работе №6

Мандатное разграничение прав в Linux

Смородова Дарья Владимировна

2022 Oct 15th

Содержание

| | | |
|---|--------------------------------|----|
| 1 | Цель работы | 5 |
| 2 | Теоретическое введение | 6 |
| 3 | Выполнение лабораторной работы | 7 |
| 4 | Выводы | 17 |
| 5 | Список литературы | 18 |

List of Tables

List of Figures

| | | |
|------|--|----|
| 3.1 | Задание параметра ServerName | 7 |
| 3.2 | Отключение фильтра | 7 |
| 3.3 | Проверка режима и политики | 8 |
| 3.4 | Проверка статуса | 8 |
| 3.5 | Веб-сервер Apache | 9 |
| 3.6 | Просмотр переключателей SELinux для Apache | 10 |
| 3.7 | Статистика по политике | 11 |
| 3.8 | Определение типов файлов и поддиректорий в директории /var/www | 11 |
| 3.9 | Определение типов файлов в директории /var/www/html | 11 |
| 3.10 | Создание файла | 12 |
| 3.11 | Проверка контекста | 12 |
| 3.12 | Получение доступа к файлу через браузер | 12 |
| 3.13 | Проверка контекста файла | 12 |
| 3.14 | Изменение контекста файла /var/www/html/test.html | 13 |
| 3.15 | Получение доступа к файлу через браузер | 13 |
| 3.16 | Просмотр log-файла веб-сервера Apache | 13 |
| 3.17 | Изменение TCP-порта с 80 на 81 | 14 |
| 3.18 | Анализ и просмотр лог-файлов | 14 |
| 3.19 | Выполнение и проверка списка портов | 15 |
| 3.20 | Возвращение контекста | 15 |
| 3.21 | Получение доступа к файлу через браузер | 15 |
| 3.22 | Исправленный файл apache | 16 |
| 3.23 | Удаление привязки к 81 порту и удаление файла | 16 |

1 Цель работы



Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux¹. Проверить работу SELinx на практике совместно с веб-сервером Apache.

2 Теоретическое введение¹

Apache HTTP-сервер (является искажённым сокращением от англ. a patchy server; среди русских пользователей общепринято переводное апáч) — свободный веб-сервер.

Apache является кроссплатформенным ПО, поддерживает операционные системы Linux, BSD, macOS, Microsoft Windows, Novell NetWare, BeOS.

Основными достоинствами Apache считаются надёжность и гибкость конфигурации. Он позволяет подключать внешние модули для предоставления данных, использовать СУБД для аутентификации пользователей, модифицировать сообщения об ошибках и т. д. Поддерживает IPv4.

Сервер был написан в начале 1995 года и считается, что его имя восходит к шуточному названию «a patchy» (с англ. —«в заплатках»), так как он устранял ошибки популярного тогда сервера Всемирной паутины NCSA HTTPd 1.3. В дальнейшем, с версии 2.x, сервер был переписан заново и теперь не содержит кода NCSA. На данный момент разработка ведётся в ветке 2.4, а в версиях 1.3, 2.0 и 2.2 производятся лишь исправления ошибок безопасности. На текущий момент последняя версия ветки 2.4 — 2.4.46 (5 августа 2020), для первой версии это 1.3.42.

Веб-сервер Apache разрабатывается и поддерживается открытым сообществом разработчиков под эгидой Apache Software Foundation и включён во многие программные продукты, среди которых СУБД Oracle и IBM WebSphere.

¹Wikipedia. Apache HTTP Server

3 Выполнение лабораторной работы¹

1. В конфигурационном файле `/etc/httpd/httpd.conf` зададим параметр `ServerName` (рис. 3.1):

```
[root@smorodovadv ~]# echo "ServerName test.ru" >> /etc/httpd/httpd.conf
[root@smorodovadv ~]# cat /etc/httpd/httpd.conf
ServerName test.ru
[root@smorodovadv ~]#
```

Figure 3.1: Задание параметра `ServerName`

2. Также проследим, чтобы пакетный фильтр был отключён или в своей рабочей конфигурации позволял подключаться к 80-у и 81-у портам протокола `tcp`. Отключим фильтр и добавим разрешающие правила (рис. 3.2):

```
[root@smorodovadv ~]# iptables -F
[root@smorodovadv ~]# iptables -P INPUT ACCEPT iptables -P OUTPUT ACCEPT
Bad argument `iptables'
Try `iptables -h' or `iptables --help' for more information.
[root@smorodovadv ~]# iptables -P INPUT ACCEPT
[root@smorodovadv ~]# iptables -P OUTPUT ACCEPT
[root@smorodovadv ~]# iptables -I INPUT -p tcp --dport 80 -j ACCEPT
[root@smorodovadv ~]# iptables -I INPUT -p tcp --dport 81 -j ACCEPT
[root@smorodovadv ~]# iptables -I OUTPUT -p tcp --sport 80 -j ACCEPT
[root@smorodovadv ~]# iptables -I OUTPUT -p tcp --sport 81 -j ACCEPT
[root@smorodovadv ~]#
```

Figure 3.2: Отключение фильтра

3. Войдем в систему с полученными учётными данными и убедимся, что SELinux работает в режиме `enforcing` политики `targeted` (рис. 3.3):

¹Методические материалы к лабораторной работе

```
[root@smorodovadv ~]# getenforce
Enforcing
[root@smorodovadv ~]# setstatus
bash: setstatus: command not found...
[root@smorodovadv ~]# 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
[root@smorodovadv ~]#
```

Figure 3.3: Проверка режима и политики

4. Обратимся с помощью браузера к веб-серверу, запущенному на компьютере, и убедимся, что последний работает (рис. 3.4):

```
[root@smorodovadv ~]# 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; vendor pre
   Active: active (running) since Sat 2022-10-15 19:48:58 MSK; 1min 0s ago
     Docs: man:httpd.service(8)
   Main PID: 42068 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes
   Tasks: 213 (limit: 12208)
  Memory: 27.2M
     CPU: 143ms
   CGroup: /system.slice/httpd.service
           └─42068 /usr/sbin/httpd -DFOREGROUND
             └─42069 /usr/sbin/httpd -DFOREGROUND
               └─42073 /usr/sbin/httpd -DFOREGROUND
                 └─42074 /usr/sbin/httpd -DFOREGROUND
                   └─42075 /usr/sbin/httpd -DFOREGROUND

окт 15 19:48:57 smorodovadv.localdomain systemd[1]: Starting The Apache HTTP Se
окт 15 19:48:58 smorodovadv.localdomain systemd[1]: Started The Apache HTTP Ser
окт 15 19:48:58 smorodovadv.localdomain httpd[42068]: Server configured, listen
lines 1-19/19 (END)
```

Figure 3.4: Проверка статуса

5. Найдем веб-сервер Apache в списке процессов, определим его контекст безопасности (рис. 3.5):


```

[root@smorodovadv smorodovadv]# ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root 42068 0.0 0.5 20248 11576 ?
Ss 19:48 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 42069 0.0 0.3 21572 7352 ?
S 19:48 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 42073 0.0 0.6 1210512 13120 ?
Sl 19:48 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 42074 0.0 0.5 1079376 11072 ?
Sl 19:48 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 42075 0.0 0.5 1079376 11072 ?
Sl 19:48 0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 root 42945 0.0 0.1 221692
2352 pts/0 S+ 20:03 0:00 grep --color=auto httpd
[root@smorodovadv smorodovadv]# ps -eZ | grep httpd
system_u:system_r:httpd_t:s0 42068 ? 00:00:00 httpd
system_u:system_r:httpd_t:s0 42069 ? 00:00:00 httpd
system_u:system_r:httpd_t:s0 42073 ? 00:00:00 httpd
system_u:system_r:httpd_t:s0 42074 ? 00:00:00 httpd
system_u:system_r:httpd_t:s0 42075 ? 00:00:00 httpd
[root@smorodovadv smorodovadv]#

```

Figure 3.5: Веб-сервер Apache

6. Посмотрим текущее состояние переключателей SELinux для Apache (рис. 3.6):

```

[root@smorodovadv smorodovadv]# 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_avaahi 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_graceful_shutdown off
httpd_manage_ipa off
httpd_mod_auth_ntlm_winbind off
httpd_mod_auth_pam off
httpd_read_user_content off
httpd_run_ipa off
httpd_run_preupgrade off
httpd_run_stickshift off
httpd_serve_cobbler_files off
httpd_setrlimit off
httpd_ssi_exec off
httpd_sys_script_anon_write off
httpd_tmp_exec off
httpd_tty_comm off
httpd_unified off
httpd_use_cifs off
httpd_use_fusefs off
httpd_use_gpg off
httpd_use_nfs off
httpd_use_openscryptoki off
httpd_use_openstack off
httpd_use_sasl off
httpd_verify_dns off
[root@smorodovadv smorodovadv]#

```

Figure 3.6: Просмотр переключателей SELinux для Apache

7. Посмотрим статистику по политике, а также определим множество пользователей(8), ролей(14), типов(5002). (рис. 3.7):

```

[root@smorodovadv smorodovadv]# seinfo
Statistics for policy file: /sys/fs/selinux/policy
Policy Version:          33 (MLS enabled)
Target Policy:           selinux
Handle unknown classes:  allow
Classes:                 133
Sensitivities:           1
Types:                   5002
Users:                   8
Booleans:                347
Allow:                   63996
Auditallow:              168
Type_trans:              258486
Type_member:             35
Role_allow:              38
Constraints:             72
MLS Constrain:           72
Permissives:             0
Defaults:                7
Allowxperm:              0
Auditallowxperm:         0
Ibendportcon:            0
Initial SIDs:            27
Genfscon:                106
Netifcon:                0
Permissions:             454
Categories:             1024
Attributes:              254
Roles:                   14
Cond. Expr.:             381
Neverallow:              0
Dontaudit:               8417
Type_change:             87
Range_trans:             5960
Role_trans:              420
Validatetrans:           0
MLS Val. Tran:           0
Polcap:                  5
Typebounds:              0
Neverallowxperm:         0
Dontauditxperm:          0
Ibpkeycon:               0
Fs_use:                  33
Portcon:                 651
Nodecon:                 0

```

Figure 3.7: Статистика по политике

8. Определим тип файлов и поддиректорий, находящихся в директории /var/www (рис. 3.8):

```

Netifcon:                0
Nodecon:                 0
[root@smorodovadv smorodovadv]# ls -lZ /var/www
итого 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 мая 16 15
:10 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 6 мая 16 15
:10 html
[root@smorodovadv smorodovadv]#

```

Figure 3.8: Определение типов файлов и поддиректорий в директории /var/www

9. Определим тип файлов, находящихся в директории /var/www/html и круг пользователей, которым разрешено создание файлов в директории /var/www/html (рис. 3.9):

```

[root@smorodovadv smorodovadv]# ls -lZ /var/www/html
итого 0
[root@smorodovadv smorodovadv]#

```

Figure 3.9: Определение типов файлов в директории /var/www/html

10. Создадим от имени суперпользователя html-файл /var/www/html/test.html (рис. 3.10):

```
[root@smorodovadv smorodovadv]# touch /var/www/html/test.html
[root@smorodovadv smorodovadv]# ls -lZ /var/www/html
итого 0
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 0 окт 15 20
:10 test.html
[root@smorodovadv smorodovadv]# echo "<html>
> <body>test</body>
> </html>" > /var/www/html/test.html
[root@smorodovadv smorodovadv]# cat /var/www/html/test.html
<html>
<body>test</body>
</html>
[root@smorodovadv smorodovadv]#
```

Figure 3.10: Создание файла

11. Проверим контекст созданного файла: httpd_sys_content_t (рис. 3.11):

```
[root@smorodovadv smorodovadv]# ls -lZ /var/www/html
итого 4
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 33 окт 15 2
0:12 test.html
[root@smorodovadv smorodovadv]#
```

Figure 3.11: Проверка контекста

12. Обратимся к файлу через веб-сервер, введя в браузере адрес <http://127.0.0.1/test.html>. Убедимся, что файл был успешно отображён (рис. 3.12):

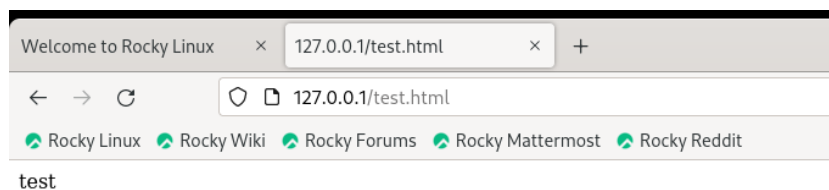


Figure 3.12: Получение доступа к файлу через браузер

13. Проверим контекст файла (рис. 3.13):

```
[root@smorodovadv smorodovadv]# man httpd_selinux
Нет справочной страницы для httpd_selinux
[root@smorodovadv smorodovadv]# ls -lZ /var/www/html/test.html
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test.html
[root@smorodovadv smorodovadv]#
```

Figure 3.13: Проверка контекста файла

14. Изменим контекст файла `/var/www/html/test.html` с `httpd_sys_content_t` на `samba_share_t`. После этого проверим, что контекст поменялся (рис. 3.14):

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

Figure 3.14: Изменение контекста файла /var/www/html/test.html

15. Попробуем ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Получим сообщение об ошибке, так как мы ранее изменили контекст файла(рис. 3.15):

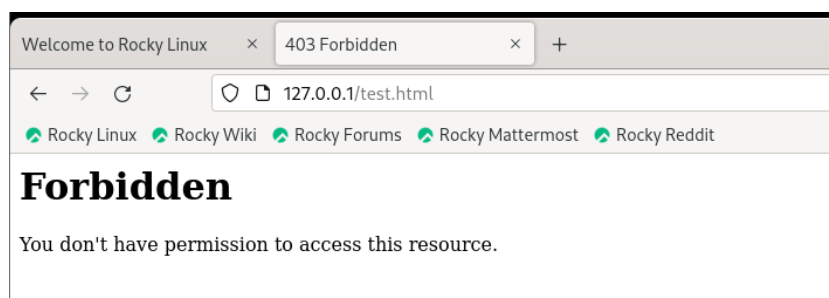


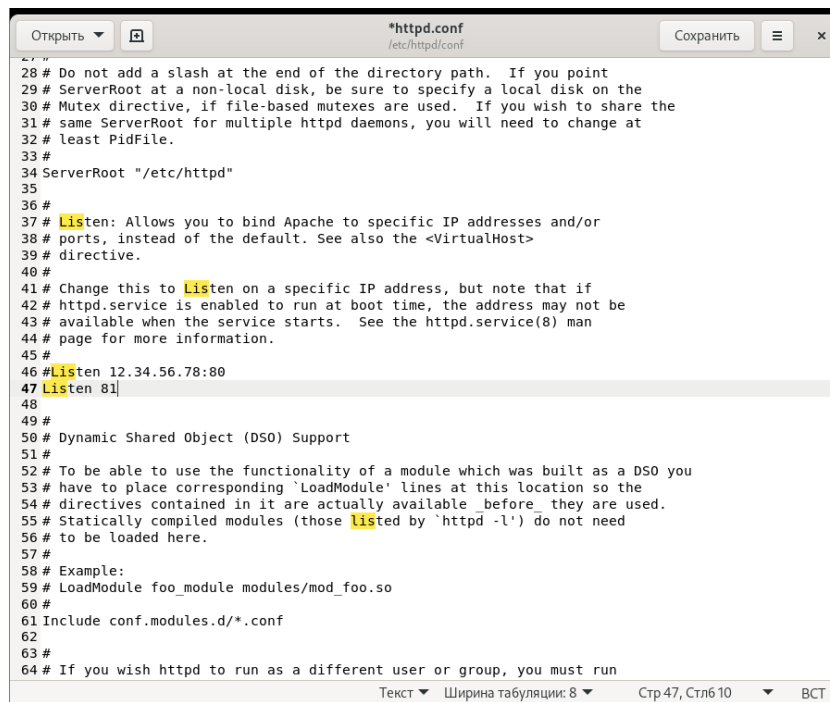
Figure 3.15: Получение доступа к файлу через браузер

16. Просмотрим log-файлы веб-сервера Apache. Также посмотрим системный лог-файл (рис. 3.16):

```
[root@smorodovadv smorodovadv]# ls -l /var/www/html/test.html
-rw-r--r--. 1 root root 33 окт 15 20:12 /var/www/html/test.html
[root@smorodovadv smorodovadv]# tail /var/log/messages
Oct 15 20:20:15 smorodovadv setroubleshoot[43871]: SELinux запрещает /usr/sbin/httpd
доступ getattr к файл /var/www/html/test.html. Для выполнения всех сообщени
й SELinux: sealert -l f0d7cabd-5f1c-4dfa-bf7c-91b269763061
Oct 15 20:20:15 smorodovadv setroubleshoot[43871]: SELinux запрещает /usr/sbin/httpd
доступ getattr к файл /var/www/html/test.html.#012#012***** Модуль restore
con предлагает (точность 92.2) *****#012#012Если вы хотите
исправить метку.$TARGETЗнак PATH по умолчанию должен быть httpd_sys_content_t#0
12То вы можете запустить restorecon. Возможно, попытка доступа была остановлена
из-за недостаточных разрешений для доступа к родительскому каталогу, и в этом сл
учае попытайтесь соответствующим образом изменить следующую команду.#012Сделать#
012# /sbin/restorecon -v /var/www/html/test.html#012#012***** Модуль public_con
tent предлагает (точность 7.83) *****#012#012Если вы хотите леч
ить test.html как общедоступный контент#012То необходимо изменить метку test.htm
l с public_content_t на public_content_rw_t.#012Сделать#012# semanage fcontext -
```

Figure 3.16: Просмотр log-файла веб-сервера Apache

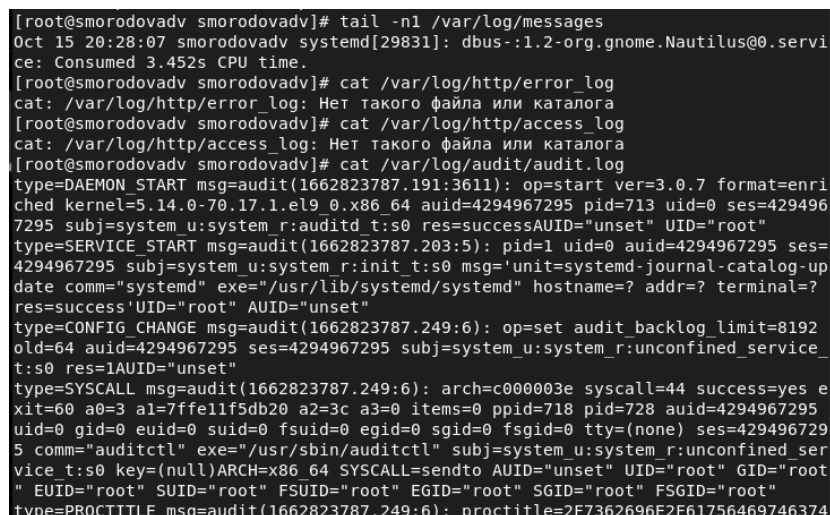
17. Попробуем запустить веб-сервер Apache на прослушивание TCP-порта 81 (а не 80, как рекомендует IANA и прописано в /etc/services). Для этого в файле /etc/httpd/httpd.conf найдем строчку `Listen 80` и заменим её на `Listen 81` (рис. 3.17):



```
28 # Do not add a slash at the end of the directory path. If you point
29 # ServerRoot at a non-local disk, be sure to specify a local disk on the
30 # Mutex directive, if file-based mutexes are used. If you wish to share the
31 # same ServerRoot for multiple httpd daemons, you will need to change at
32 # least PidFile.
33 #
34 ServerRoot "/etc/httpd"
35 #
36 #
37 # Listen: Allows you to bind Apache to specific IP addresses and/or
38 # ports, instead of the default. See also the <VirtualHost>
39 # directive.
40 #
41 # Change this to Listen on a specific IP address, but note that if
42 # httpd.service is enabled to run at boot time, the address may not be
43 # available when the service starts. See the httpd.service(8) man
44 # page for more information.
45 #
46 #Listen 12.34.56.78:80
47 Listen 81
48 #
49 #
50 # Dynamic Shared Object (DSO) Support
51 #
52 # To be able to use the functionality of a module which was built as a DSO you
53 # have to place corresponding 'LoadModule' lines at this location so the
54 # directives contained in it are actually available before they are used.
55 # Statically compiled modules (those listed by 'httpd -l') do not need
56 # to be loaded here.
57 #
58 # Example:
59 # LoadModule foo_module modules/mod_foo.so
60 #
61 Include conf.modules.d/*.conf
62 #
63 #
64 # If you wish httpd to run as a different user or group, you must run
```

Figure 3.17: Изменение TCP-порта с 80 на 81

18. Просмотрим файлы `/var/log/http/error_log` `/var/log/http/access_log` и `/var/log/audit/audit.log` (рис. 3.18):



```
[root@smorodovadv smorodovadv]# tail -n1 /var/log/messages
Oct 15 20:28:07 smorodovadv systemd[29831]: dbus-1.2-org.gnome.Nautilus@0.servi
ce: Consumed 3.452s CPU time.
[root@smorodovadv smorodovadv]# cat /var/log/http/error_log
cat: /var/log/http/error_log: Нет такого файла или каталога
[root@smorodovadv smorodovadv]# cat /var/log/http/access_log
cat: /var/log/http/access_log: Нет такого файла или каталога
[root@smorodovadv smorodovadv]# cat /var/log/audit/audit.log
type=DAEMON_START msg=audit(1662823787.191:3611): op=start ver=3.0.7 format=enri
ched kernel=5.14.0-70.17.1.el9_0.x86_64 auid=4294967295 pid=713 uid=0 ses=429496
7295 subj=system u:system r:auditd t:s0 res=successAUID="unset" UID="root"
type=SERVICE_START msg=audit(1662823787.203:5): pid=1 uid=0 auid=4294967295 ses=
4294967295 subj=system u:system r:init t:s0 msg='unit=systemd-journal-catalog-up
date comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=?
res=success'UID="root" AUID="unset"
type=CONFIG_CHANGE msg=audit(1662823787.249:6): op=set audit_backlog_limit=8192
old=64 auid=4294967295 ses=4294967295 subj=system_u:system_r:unconfined_service_
t:s0 res=1AUID="unset"
type=SYSCALL msg=audit(1662823787.249:6): arch=c000003e syscall=44 success=yes e
xit=60 a0=3 a1=7ffe11f5db20 a2=3c a3=0 items=0 ppid=718 pid=728 auid=4294967295
uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=429496729
5 comm="auditctl" exe="/usr/sbin/auditctl" subj=system_u:system_r:unconfined_ser
vice t:s0 key=(null)ARCH=x86_64 SYSCALL=sendto AUID="unset" UID="root" GID="root"
EUID="root" SUID="root" FSUID="root" EGID="root" SGID="root" FSGID="root"
type=PROCTITLE msg=audit(1662823787.249:6): proctitle=2F7362696E2F61756469746374
```

Figure 3.18: Анализ и просмотр лог-файлов

19. Выполним команду `semanage port -a -t http_port_t -p tcp 81`. После этого

проверим список портов. Убедимся, что порт 81 появился в списке (рис. 3.19):

```
[root@smorodovadv smorodovadv]# semanage port -a -t http_port_t -p tcp 81
ValueError: Порт tcp/81 уже определен
[root@smorodovadv smorodovadv]# semanage port -l | grep http_port_t
semanage port: error: one of the arguments -a/--add -d/--delete -m/--modify -l/--list -E/--extract -D/--deleteall is required
[root@smorodovadv smorodovadv]# semanage port -l | grep http_port_t
http_port_t          tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t  tcp      5988
[root@smorodovadv smorodovadv]#
```

Figure 3.19: Выполнение и проверка списка портов

20. Вернем контекст `httpd_sys_content_t` к файлу `/var/www/html/test.html` (рис. 3.20):

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

Figure 3.20: Возвращение контекста

21. Попробуем получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1:81/test.html`. Снова увидим содержимое файла — слово «test» (рис. 3.21):

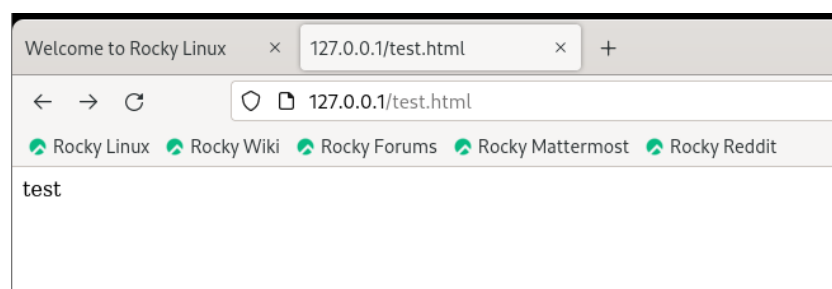


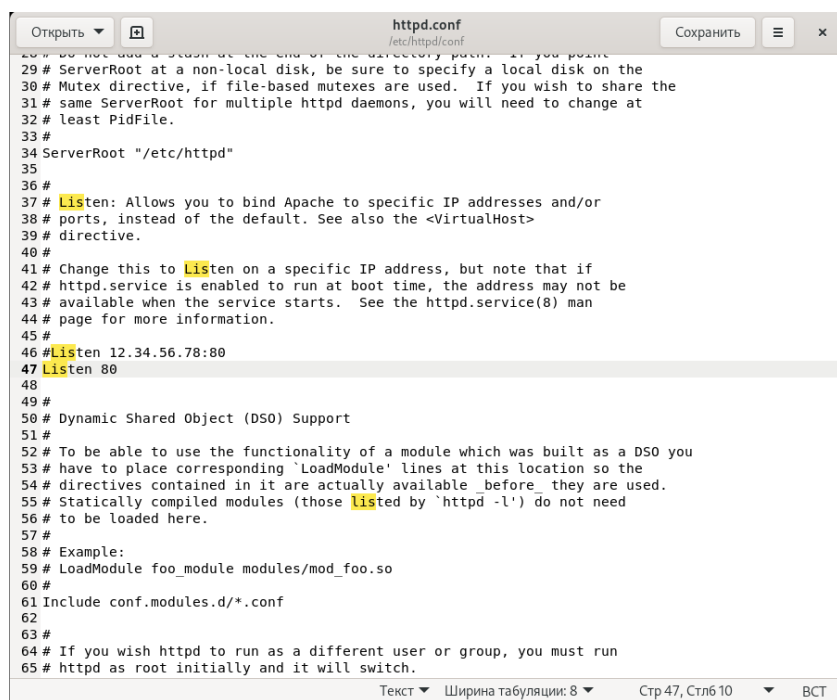
Figure 3.21: Получение доступа к файлу через браузер

22. Исправим обратно конфигурационный файл `apache`, вернув `Listen80` (рис. 3.22):

```
[root@smorodovadv smorodovadv]# semanage port -d -t http_port_t -p tcp 81
ValueError: Порт tcp/81 определен на уровне политики и не может быть удален
[root@smorodovadv smorodovadv]# semanage port -l | grep http_port_t
http_port_t          tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t  tcp      5988
[root@smorodovadv smorodovadv]# rm /var/www/html/test.html
rm: удалить обычный файл '/var/www/html/test.html'? y
[root@smorodovadv smorodovadv]#
```

Figure 3.22: Исправленный файл apache

23. Удалим привязку `http_port_t` к 81 порту и удалим файл `/var/www/html/test.html` (рис. 3.23):



```
29 # ServerRoot at a non-local disk, be sure to specify a local disk on the
30 # Mutex directive, if file-based mutexes are used. If you wish to share the
31 # same ServerRoot for multiple httpd daemons, you will need to change at
32 # least PidFile.
33 #
34 ServerRoot "/etc/httpd"
35 #
36 #
37 # Listen: Allows you to bind Apache to specific IP addresses and/or
38 # ports, instead of the default. See also the <VirtualHost>
39 # directive.
40 #
41 # Change this to Listen on a specific IP address, but note that if
42 # httpd.service is enabled to run at boot time, the address may not be
43 # available when the service starts. See the httpd.service(8) man
44 # page for more information.
45 #
46 #Listen 12.34.56.78:80
47 Listen 80
48 #
49 #
50 # Dynamic Shared Object (DSO) Support
51 #
52 # To be able to use the functionality of a module which was built as a DSO you
53 # have to place corresponding 'LoadModule' lines at this location so the
54 # directives contained in it are actually available before they are used.
55 # Statically compiled modules (those listed by 'httpd -l') do not need
56 # to be loaded here.
57 #
58 # Example:
59 # LoadModule foo_module modules/mod_foo.so
60 #
61 Include conf.modules.d/*.conf
62 #
63 #
64 # If you wish httpd to run as a different user or group, you must run
65 # httpd as root initially and it will switch.
```

Figure 3.23: Удаление привязки к 81 порту и удаление файла

4 Выводы

В ходе выполнения данной лабораторной работы, мы развили навыки администрирования ОС Linux, получили первое практическое знакомство с технологией SELinux и проверили работу SELinux на практике совместно с веб-сервером Apache.

5 Список литературы

1. Методические материалы к лабораторной работе, представленные на сайте “ТУИС РУДН”
2. Wikipedia. Apache HTTP Server