Отчет по лабораторной работе №6

Мандатное разграничение прав в Linux

Бурдина Ксения Павловна

2022 Oct 11th

Содержание

Список иллюстраций

1. Цель работы

Целью данной работы является развитие навыков администрирования ОС Linux, получение практического знакомства с технологией SELinux, а также проверка работы SELinux на практике совместно с веб-сервером Apache.

2. Теоретическое введение

При подготовке стенда нам необходима для работы политика targeted и режим enforcing, которые используются в данном дистрибутиве по умолчанию, т.е. каких-то специальных настроек не требуется. При этом пользователю следует убедиться, что политика и режим включены, особенно когда работа будет проводиться повторно и велика вероятность изменений при предыдущем использовании системы.

При необходимости администратор должен разбираться в работе SELinux и уметь как исправить конфигурационный файл /etc/selinux/config, так и проверить используемый режим и политику. Нам необходимо установить веб-сервер Apache. Причем следует учитывать, что при установке системы в конфигурации «рабочая станция» указанный пакет не ставится.

Перед началом работы в конфигурационном файле /etc/httpd/httpd.conf необходимо задать параметр ServerName, чтобы при запуске веб-сервера не выдавались лишние сообщения об ошибках, не относящихся к лабораторной работе. Также необходимо проследить, чтобы пакетный фильтр был отключён или в своей рабочей конфигурации позволял подключаться к 80-у и 81-у портам протокола tcp [1].

Выполним все действия для подготовки к работе. Установим httpd для работы, после чего проверим наличие необходимых файлов в каталоге и настроим фильтры. Контрольные команды для дальнейшей работы:

```
[root@10 ~]# cd /etc/httpd
[root@10 httpd]# ls
conf  conf.d  conf.modules.d  logs  modules  run  state
[root@10 httpd]# cd conf
[root@10 conf]# cat httpd.conf
ServerName test.ru
[root@10 conf]# cd
[root@10 ~]# iptables -F
[root@10 ~]# iptables -P INPUT ACCEPT
[root@10 ~]# iptables -P OUTPUT ACCEPT
```

Рис. 2.1.: Подготовка к работе

3. Ход выполнения лабораторной работы

1. Войдем в систему с полученными учётными данными и убедимся, что SELinux работает в режиме enforcing политики targeted:

```
[root@10 kpburdina]# getenforce
Enforcing
[root@10 kpburdina]# 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
```

Рис. 3.1.: Вызов команд getenforce и sestatus

Видим, что у нас действительно все работает верно.

2. Обратимся с помощью браузера к веб-серверу, запущенному на нашем компьютере, и убедимся, что последний работает:

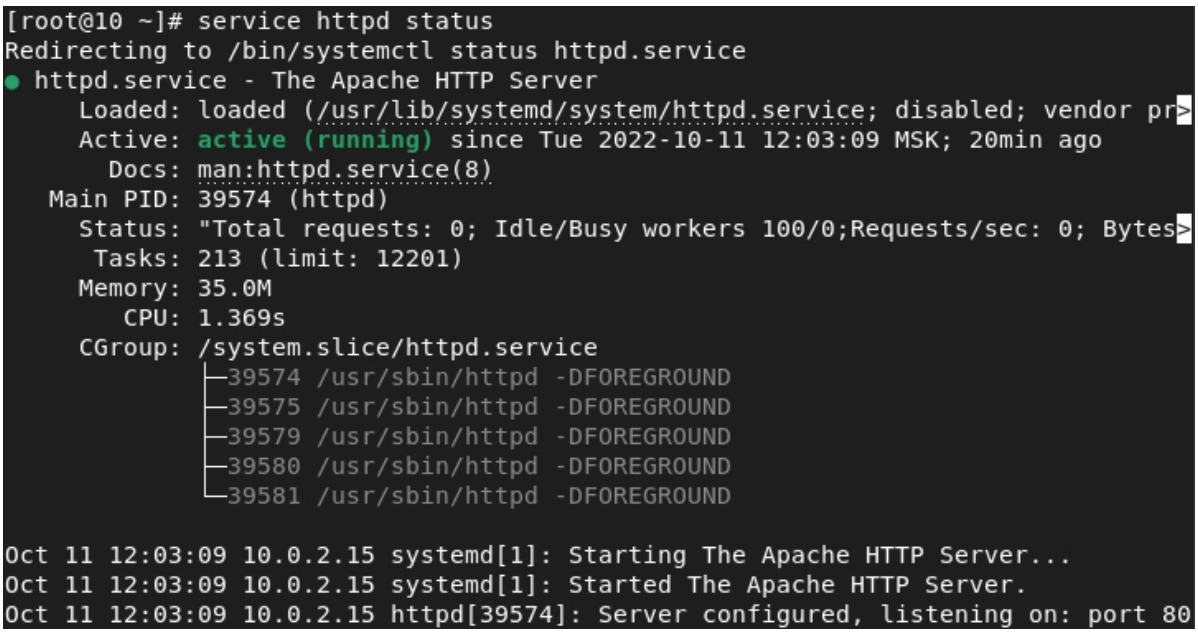

Рис. 3.2.: Обращение к веб-серверу

3. Найдем веб-сервер Apache в списке процессов, определим его контекст безопасности:

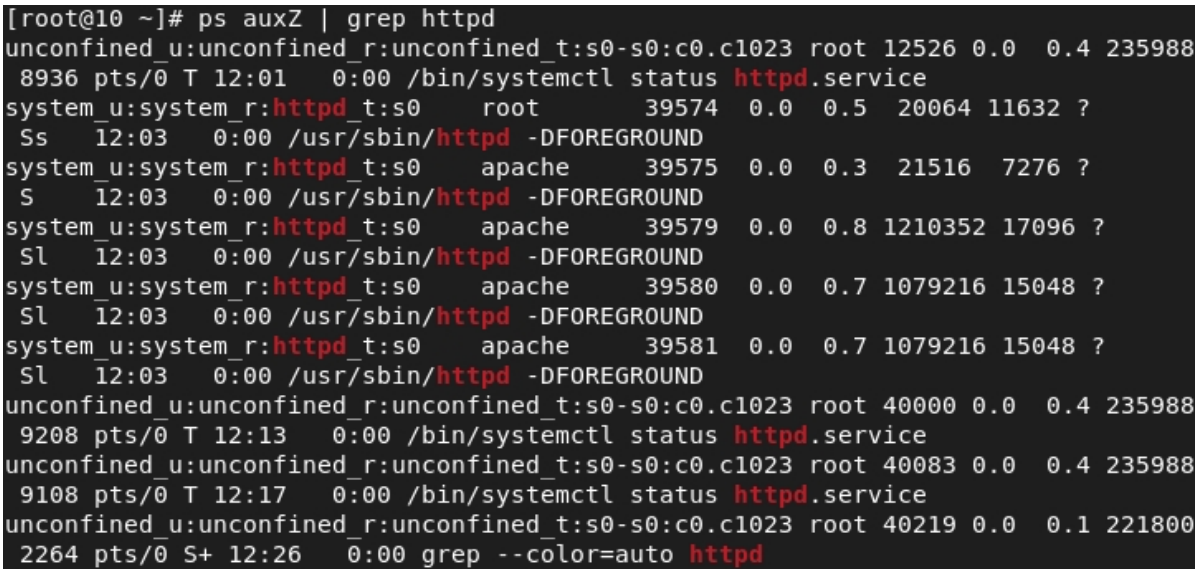

Рис. 3.3.: Нахождение Apache в списке процессоров

Контекстом безопасности будет system_u:system_r.

4. Посмотрим текущее состояние переключателей SELinux для Apache:

```
[root@10 ~]# 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
```

Рис. 3.4.: Состояние переключателей SELinux

```
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_opencryptoki                       off
httpd_use_openstack                          off
httpd_use_sasl                               off
httpd_verify_dns                             off
```

Рис. 3.5.: Состояние переключателей SELinux

Обратим внимание на то, что многие из них находятся в положении “off”.

5. Посмотрим статистику по политике с помощью команды seinfo и определим множество пользователей, ролей и типов:

```
[root@10 ~]# seinfo
Statistics for policy file: /sys/fs/selinux/policy
Policy Version:             33 (MLS enabled)
Target Policy:              selinux
Handle unknown classes:     allow
  Classes:             133    Permissions:         454
  Sensitivities:         1    Categories:         1024
  Types:              5002    Attributes:          254
  Users:                 8    Roles:                14
  Booleans:            347    Cond. Expr.:         381
  Allow:             63996    Neverallow:            0
  Auditallow:          168    Dontaudit:          8417
  Type_trans:       258486    Type_change:          87
  Type_member:          35    Range_trans:        5960
  Role allow:           38    Role_trans:          420
  Constraints:          72    Validatetrans:         0
  MLS Constrain:        72    MLS Val. Tran:         0
  Permissives:           0    Polcap:                5
  Defaults:              7    Typebounds:            0
  Allowxperm:            0    Neverallowxperm:       0
  Auditallowxperm:       0    Dontauditxperm:        0
  Ibendportcon:          0    Ibpkeycon:             0
  Initial SIDs:         27    Fs_use:               33
  Genfscon:            106    Portcon:             651
  Netifcon:              0    Nodecon:               0
```

Рис. 3.6.: Статистика по политике

Видим, что у нас имеется 8 пользователей, 5002 типов и 14 ролей.

6. Определим тип файлов и поддиректорий, находящихся в директории /var/www:

```
[root@10 ~]# ls -lZ /var/www
total 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 May 16 15:10 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0     6 May 16 15:10 html
```

Рис. 3.7.: Просмотр директории /var/www

Можем увидеть контекст файлов: system_u, object_r, httpd_sys_content_t/httpd_sys_script_exec

7. Определим тип файлов, находящихся в директории /var/www/html:

```
[root@10 ~]# ls -lZ /var/www/html
total 0
```

Рис. 3.8.: Просмотр директории /var/www/html

Видим, что у нас в данной директории отсутствуют какие-либо файлы.

8. Определяя круг пользователей, которым разрешено создание файлов в данной директории, можно сказать, что создание файлов разрешено всем пользователям.

9. Создадим от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись в директорию) html-файл /var/www/html/test.html и запишем в него содержание:

```
[root@10 ~]# touch /var/www/html/test.html
```

Рис. 3.9.: Создание файла test.html

```
[root@10 ~]# ls /var/www/html
test.html
```

Рис. 3.10.: Проверка создания файла

```
[root@10 html]# nano test.html
```

Рис. 3.11.: Открытие файла на редактирование

```
<html>
<body>
test
</body>
</html>
```

Рис. 3.12.: Запись содержимого в файл

10. Проверим контекст созданного нами файла:

```
[root@10 html]# cat test.html
<html>
<body>
test
</body>
</html>
```

Рис. 3.13.: Чтение содержимого файла

Контекст, присваиваемый по умолчанию вновь созданным файлам в директории /var/www/html, выглядит следующим образом:

```
[root@10 html]# ls -Z /var/www/html/test.html
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test.html
```

Рис. 3.14.: Контекст созаваемых файлов

Всем создаваемым файлам присваивается контекст: unconfined_u:object_r:httpd_sys_content

11. Обратимся к файлу через веб-браузер, введя в браузере адрес http://127.0.0.1/test.html:

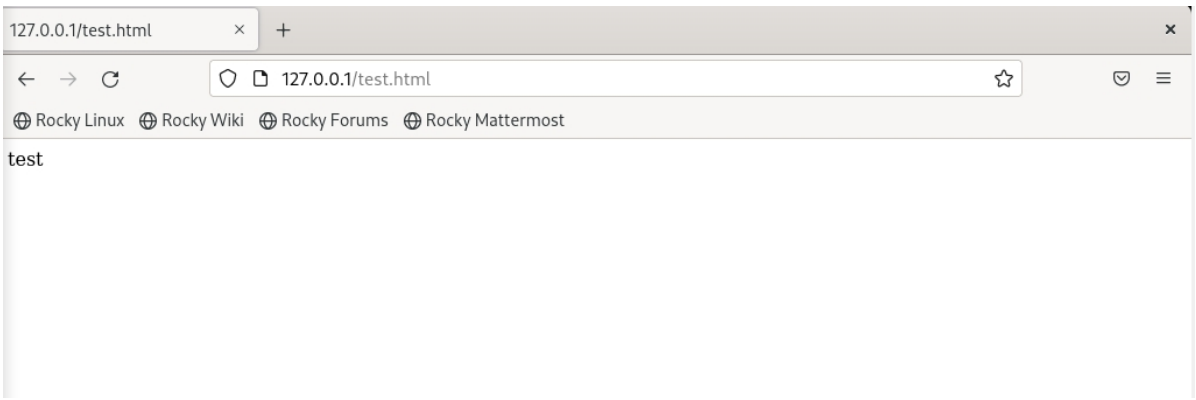

Рис. 3.15.: Открытие файла через браузер

Видим, что наш файл был успешно передан на сервер, поскольку на открывшейся странице отобразился текст нашего файла.

12. Изучим справку по httpd_selinux:

```
[root@10 html]# man httpd_selinux
No manual entry for httpd_selinux
```

Рис. 3.16.: Открытие справки по httpd_selinux

К сожалению, нам невозможно изучить справку, однако посмотреть, какие контексты файлов определены для httpd, мы можем, и они аналогичны тем, что отобразились при просмотре файла test.html.

13. Изменим 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, после чего проверим правильность изменения контекста:

```
[root@10 ~]# chcon -t samba_share_t /var/www/html/test.html
[root@10 ~]# ls -Z /var/www/html/test.html
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html
```

Рис. 3.17.: Изменение контекста файла

14. Попробуем ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1/test.html. На этот раз мы получим следующее сообщение об ошибке:

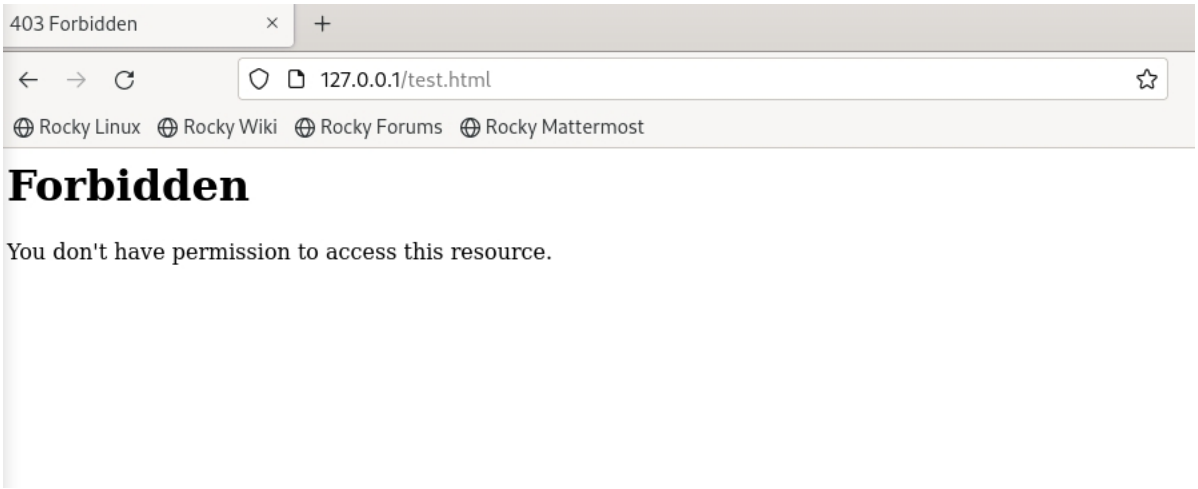

Рис. 3.18.: Сообщение об ошибке доступа

15. Проанализируем ситуацию. Наш файл не был открыт, поскольку на сервер загружены данные по контексту те, которые назначаются всем файлам по умолчанию. Поэтому при попытке перейти на сайт с измененными данными, мы получим отказ в выполнении действия.

Но несмотря на это, права доступа позволяют читать файл любому пользователю:

```
[root@10 ~]# ls -l /var/www/html/test.html
-rw-r--r--. 1 root root 36 Oct 11 13:30 /var/www/html/test.html
```

Рис. 3.19.: Проверка прав на чтение файла

Теперь посмотрим log-файлы веб-сервера Apache, после чего посмотрим системный log-файл:

```
[root@10 ~]# tail /var/log/messages
Oct 11 15:23:42 localhost setroubleshoot[42474]: SELinux is preventing /usr/sbin/httpd from getattr ac
cess on the file /var/www/html/test.html.#012#012*****  Plugin restorecon (92.2 confidence) suggests
 ***********************#012#012If you want to fix the label. #012/var/www/html/test.html default lab
el should be httpd_sys_content_t.#012Then you can run restorecon. The access attempt may have been sto
pped due to insufficient permissions to access a parent directory in which case try to change the foll
owing command accordingly.#012Do#012# /sbin/restorecon -v /var/www/html/test.html#012#012*****  Plugin
 public_content (7.83 confidence) suggests   *******************#012#012If you want to treat test.htm
l as public content#012Then you need to change the label on test.html to public_content_t or public_co
ntent_rw_t.#012Do#012# semanage fcontext -a -t public_content_t '/var/www/html/test.html'#012# restore
con -v '/var/www/html/test.html'#012#012*****  Plugin catchall (1.41 confidence) suggests   **********
*****************#012#012If you believe that httpd should be allowed getattr access on the test.html fi
le by default.#012Then you should report this as a bug.#012You can generate a local policy module to a
llow this access.#012Do#012allow this access for now by executing:#012# ausearch -c 'httpd' --raw | au
dit2allow -M my-httpd#012# semodule -X 300 -i my-httpd.pp#012
Oct 11 15:23:42 localhost setroubleshoot[42474]: failed to retrieve rpm info for /var/www/html/test.ht
ml
Oct 11 15:23:42 localhost setroubleshoot[42474]: SELinux is preventing /usr/sbin/httpd from getattr ac
cess on the file /var/www/html/test.html. For complete SELinux messages run: sealert -l 54560a6e-b61f-
43e3-8a7b-7a75abd6d657
Oct 11 15:23:42 localhost setroubleshoot[42474]: SELinux is preventing /usr/sbin/httpd from getattr ac
cess on the file /var/www/html/test.html.#012#012*****  Plugin restorecon (92.2 confidence) suggests
 ***********************#012#012If you want to fix the label. #012/var/www/html/test.html default lab
el should be httpd_sys_content_t.#012Then you can run restorecon. The access attempt may have been sto
pped due to insufficient permissions to access a parent directory in which case try to change the foll
owing command accordingly.#012Do#012# /sbin/restorecon -v /var/www/html/test.html#012#012*****  Plugin
 public_content (7.83 confidence) suggests   *******************#012#012If you want to treat test.htm
```

Рис. 3.20.: Просмотр log-файла Apache

```
[root@10 ~]# tail /var/log/audit/audit.log
type=SERVICE_STOP msg=audit(1665494203.759:280): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:syste
m_r:init_t:s0 msg='unit=dnf-makecache comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=
? res=success'UID="root" AUID="unset"
type=SERVICE_START msg=audit(1665497818.100:281): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:syst
em_r:init_t:s0 msg='unit=dnf-makecache comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal
=? res=success'UID="root" AUID="unset"
type=SERVICE_STOP msg=audit(1665497818.101:282): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:syste
m_r:init_t:s0 msg='unit=dnf-makecache comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=
? res=success'UID="root" AUID="unset"
type=BPF msg=audit(1665501005.008:283): prog-id=69 op=LOAD
type=SERVICE_START msg=audit(1665501005.162:284): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:syst
em_r:init_t:s0 msg='unit=fprintd comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res
=success'UID="root" AUID="unset"
type=USER_AUTH msg=audit(1665501009.258:285): pid=43817 uid=0 auid=1000 ses=2 subj=system_u:system_r:xdm_t:s0-s
0:c0.c1023 msg='op=PAM:authentication grantors=pam_usertype,pam_localuser,pam_unix,pam_gnome_keyring acct="kpbu
rdina" exe="/usr/libexec/gdm-session-worker" hostname=10.0.2.15 addr=? terminal=/dev/tty1 res=success'UID="root
" AUID="kpburdina"
type=USER_ACCT msg=audit(1665501009.334:286): pid=43817 uid=0 auid=1000 ses=2 subj=system_u:system_r:xdm_t:s0-s
0:c0.c1023 msg='op=PAM:accounting grantors=pam_unix,pam_localuser acct="kpburdina" exe="/usr/libexec/gdm-sessio
n-worker" hostname=10.0.2.15 addr=? terminal=/dev/tty1 res=success'UID="root" AUID="kpburdina"
type=CRED_REFR msg=audit(1665501009.338:287): pid=43817 uid=0 auid=1000 ses=2 subj=system_u:system_r:xdm_t:s0-s
0:c0.c1023 msg='op=PAM:setcred grantors=pam_localuser,pam_unix,pam_gnome_keyring acct="kpburdina" exe="/usr/lib
exec/gdm-session-worker" hostname=10.0.2.15 addr=? terminal=/dev/tty1 res=success'UID="root" AUID="kpburdina"
type=SERVICE_STOP msg=audit(1665501035.788:288): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:syste
m_r:init_t:s0 msg='unit=fprintd comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=
success'UID="root" AUID="unset"
type=BPF msg=audit(1665501035.803:289): prog-id=69 op=UNLOAD
```

Рис. 3.21.: Просмотр системного log-файла

Видим, что в системе есть запущенные процессы setroubleshootd и audtd, поэтому можно увидеть ошибки, аналогичные указанным выше в файле.

16. Теперь попробуем запустить веб-сервер Apache на прослушивание TCP-порта. Откроем файл на просмотр:

```
[root@10 ~]# gedit /etc/httpd/httpd.conf
```

Рис. 3.22.: Открытие файла на редактирование

Однако не сможем изменить данные, поскольку информация из файла нам не была доступна.

17. Если после смены строки с прослушиванием выполнить перезапуск сервера, то он не будет работать, поскольку сервер настроен на обределенную частоту прослушки и не сможет быть подключен с другими параметрами.

18. Проанализируем log-файлы:

```
[root@10 ~]# tail -n1 /var/log/messages
Oct 11 18:25:17 localhost journal[44192]: Set document metadata failed: Unable to set metadata key
[root@10 ~]# tail -n1 /var/log/http/error_log
tail: cannot open '/var/log/http/error_log' for reading: No such file or directory
[root@10 ~]# tail -n1 /var/log/http/access_log
tail: cannot open '/var/log/http/access_log' for reading: No such file or directory
[root@10 ~]# tail -n1 /var/log/audit/audit.log
type=BPF msg=audit(1665501878.648:295): prog-id=70 op=UNLOAD
```

Рис. 3.23.: Анализ log-файла messages

После чего посмотрим еще некоторые файлы, однако в силу невозможности работы при текущих критериях, можем понять, что записи не появились и в одном из файлов.

19. Выполним конанду “semanage port” для проверки списка команд, вводимых портом:

```
[root@10 ~]# semanage port -a -t http_port_t -p tcp 81
ValueError: Port tcp/81 already defined
[root@10 ~]# semanage port -a -t http_port_t -p tcp 80
ValueError: Port tcp/80 already defined
[root@10 ~]# 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
```

Рис. 3.24.: Команда semanage port

Видим, что порт 81 присутстсвует в нашем списке.

20. При попытке запустить веб-сервер Apache ещё раз, все стало работать. То есть на данный момент все добавлено и загружено для возможности пользоваться доступом к файлу.

21. Вернем контекст httpd_sys_content__t к файлу /var/www/html/ test.html:

```
[root@10 ~]# chcon -t httpd_sys_content_t /var/www/html/test.html
```

Рис. 3.25.: Возвращение контекста

При попытке получить доступ к исходному файлу, мы снова видим в окне слово test.

22. Вернем обратно конфигурационный файл Apache, исправив Listen 80.

23. Удалим привязку http_port_t к 81 порту:

```
[root@10 ~]# semanage port -d -t http_port_t -p tcp 81
```

Рис. 3.26.: Удаление привязки к порту 81

24. Удалим файл /var/www/html/test.html для завершения полной работы:

```
[root@10 ~]# rm /var/www/html/test.html
rm: remove regular file '/var/www/html/test.html'? y
[root@10 ~]# cd /var/www/html
[root@10 html]# ls
```

Рис. 3.27.: Удаление файла test.html и проверка содержимого каталога

4. Выводы

В ходе работы мы развили навыки администрирования ОС Linux; получили практическое знакомство с технологией SELinux; проверили работу SELinux на практике совместно с веб-сервером Apache.

5. Список литературы

1. Методические материалы курса [1]