

Основы информационной безопасности

Лабораторная работа № 6. Мандатное разграничение прав в Linux

Демидова Екатерина Алексеевна

Содержание

1	Цель работы	4
2	Теоретические сведения	5
3	Выполнение лабораторной работы	6
4	Выводы	16
	Список литературы	17

Список иллюстраций

3.1	Подготовка лабораторного стенда	6
3.2	Проверка статуса SELinux	7
3.3	Проверка статуса веб-сервера	7
3.4	Просмотр контекста безопасности веб-сервера	8
3.5	Состояние переключателей SELinux для Apache	8
3.6	Статистика по политике	9
3.7	Множества пользователей, ролей, типов	10
3.8	Просмотр типов директорий в /var/www	10
3.9	Содержимое html-файла /var/www/html/test.html	11
3.10	Установка пароля для пользователя с правами администратора .	11
3.11	Открытие html-страницы через браузер	12
3.12	Изменение контекста файла /var/www/html/test.html	12
3.13	Отказ в доступе к html-странице через браузер	12
3.14	Просмотр лог-файлов	13
3.15	Замена прослушиваемого порта	13
3.16	Открытие html-страницы через браузер при прослушивании 81 порта	14
3.17	Просмотр лог-файлов	14
3.18	Просмотр портов с помощью seamnager	14

1 Цель работы

Целью данной работы является приобретение практических навыков администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux. Проверить работу SELinux на практике совместно с веб-сервером Apache.

2 Теоретические сведения

SELinux (SELinux) — это система принудительного контроля доступа, реализованная на уровне ядра[1]. Впервые эта система появилась в четвертой версии CentOS, а в 5 и 6 версии реализация была существенно дополнена и улучшена. Эти улучшения позволили SELinux стать универсальной системой, способной эффективно решать массу актуальных задач. Стоит помнить, что классическая система прав Unix применяется первой, и управление перейдет к SELinux только в том случае, если эта первичная проверка будет успешно пройдена.

Домен – список действий, которые может выполнять процесс. Обычно в качестве домена определяется минимально-возможный набор действий, при помощи которых процесс способен функционировать. Таким образом, если процесс дискредитирован, злоумышленнику не удастся нанести большого вреда.

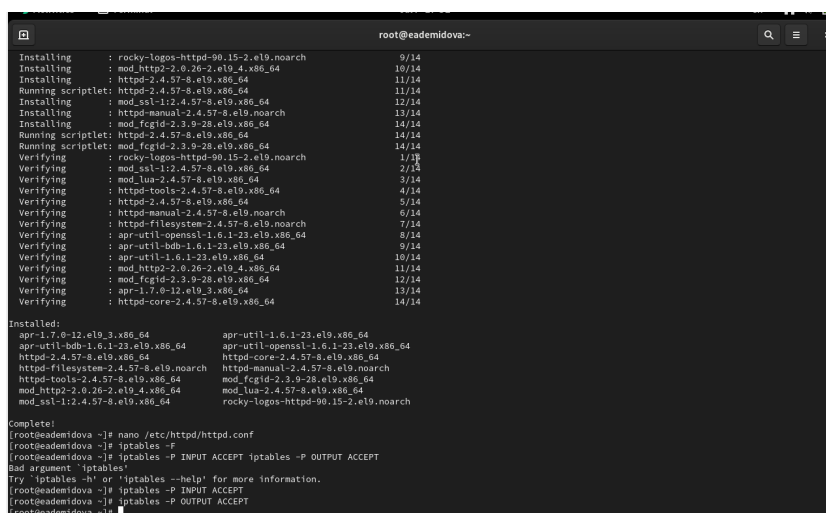
Роль – список доменов, которые могут быть применены. Если какого-то домена нет в списке доменов какой-то роли, то действия из этого домена не могут быть применены.

Тип – набор действий, которые допустимы по отношению к объекту. Тип отличается от домена тем, что он может применяться к пайпам, каталогам и файлам, в то время как домен применяется к процессам.

Контекст безопасности – все атрибуты SELinux — роли, типы и домены.

3 Выполнение лабораторной работы

В конфигурационном файле `/etc/httpd/httpd.conf` зададим параметр `ServerName`. Также необходимо проследить, чтобы пакетный фильтр был отключён или в своей рабочей конфигурации позволял подключаться к 80-у и 81-у портам протокола `tcp`. Отключим фильтр командами(рис. 3.1)



```
root@eademidova:~#
Installing      : rocky-logos-httpd-90.15-2.el9.noarch          9/14
Installing      : mod_http2-2.0.26-2.el9_4.x86_64             10/14
Installing      : httpd-2.4.57-8.el9.x86_64                  11/14
Running scriptlet: httpd-2.4.57-8.el9.x86_64                  11/14
Installing      : mod_ssl-1.2.4.57-8.el9.x86_64                12/14
Installing      : httpd-manual-2.4.57-8.el9.noarch              13/14
Installing      : mod_fcgid-2.3.9-28.el9.x86_64                14/14
Running scriptlet: httpd-2.4.57-8.el9.x86_64                  14/14
Verifying       : rocky-logos-httpd-90.15-2.el9.noarch         1/14
Verifying       : mod_ssl-1.2.4.57-8.el9.x86_64                2/14
Verifying       : mod_lua-2.4.57-8.el9.x86_64                 3/14
Verifying       : httpd-tools-2.4.57-8.el9.x86_64              4/14
Verifying       : httpd-2.4.57-8.el9.x86_64                    5/14
Verifying       : httpd-manual-2.4.57-8.el9.noarch              6/14
Verifying       : httpd-filesystem-2.4.57-8.el9.noarch         7/14
Verifying       : apr-util-openssl-1.6.1-23.el9.x86_64         8/14
Verifying       : apr-util-bdb-1.6.1-23.el9.x86_64             9/14
Verifying       : apr-util-1.6.1-23.el9.x86_64                 10/14
Verifying       : mod_http2-2.0.26-2.el9_4.x86_64              11/14
Verifying       : mod_fcgid-2.3.9-28.el9.x86_64                12/14
Verifying       : apr-1.7.0-12.el9_3.x86_64                    13/14
Verifying       : httpd-core-2.4.57-8.el9.x86_64               14/14

Installed:
apr-1.7.0-12.el9_3.x86_64      apr-util-1.6.1-23.el9.x86_64
apr-util-bdb-1.6.1-23.el9.x86_64  apr-util-openssl-1.6.1-23.el9.x86_64
httpd-2.4.57-8.el9.x86_64      httpd-core-2.4.57-8.el9.x86_64
httpd-filesystem-2.4.57-8.el9.noarch  httpd-manual-2.4.57-8.el9.noarch
httpd-tools-2.4.57-8.el9.x86_64  mod_fcgid-2.3.9-28.el9.x86_64
mod_http2-2.0.26-2.el9_4.x86_64  mod_lua-2.4.57-8.el9.x86_64
mod_ssl-1.2.4.57-8.el9.x86_64  rocky-logos-httpd-90.15-2.el9.noarch

Complete!
[root@eademidova ~]# nano /etc/httpd/httpd.conf
[root@eademidova ~]# iptables -F
[root@eademidova ~]# iptables -P INPUT ACCEPT iptables -P OUTPUT ACCEPT
Bad argument 'iptables'
Try 'iptables -h' or 'iptables --help' for more information.
[root@eademidova ~]# iptables -P INPUT ACCEPT
[root@eademidova ~]# iptables -P OUTPUT ACCEPT
[root@eademidova ~]#
```

Рис. 3.1: Подготовка лабораторного стенда

Войдем в систему с полученными учётными данными и убедимся, что SELinux работает в режиме `enforcing` политики `targeted` с помощью команд `getenforce` и `sestatus`(рис. 3.2).

```
root@eademidova:~  
htpdp-tools-2.4.57-8.el9.x86_64      mod_fcgid-2.3.9-28.el9.x86_64  
mod_http2-2.0.26-2.el9.4.x86_64    mod_lua-2.4.57-8.el9.x86_64  
mod_ssl-1:2.4.57-8.el9.x86_64      rocky-logos-httpd-90.15-2.el9.noarch  
  
Complete!  
[root@eademidova ~]# nano /etc/httpd/httpd.conf  
[root@eademidova ~]# iptables -F  
[root@eademidova ~]# iptables -P INPUT ACCEPT iptables -P OUTPUT ACCEPT  
Bad argument 'iptables'  
Try 'iptables -h' or 'iptables --help' for more information.  
[root@eademidova ~]# iptables -P INPUT ACCEPT  
[root@eademidova ~]# iptables -P OUTPUT ACCEPT  
[root@eademidova ~]# getenforce  
Enforcing  
[root@eademidova ~]# 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. 3.2: Проверка статуса SELinux

Обратимся с помощью браузера к веб-серверу, запущенному на нашем компьютере, и убедитесь, что последний работает(рис. 3.3).

```
[root@eademidova ~]# service httpd status  
Redirecting to /bin/systemctl status httpd.service  
○ httpd.service - The Apache HTTP Server  
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)  
   Active: inactive (dead)  
     Docs: man:httd.service(8)  
[root@eademidova ~]# systemctl enable httpd  
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.  
[root@eademidova ~]# systemctl start httpd  
[root@eademidova ~]# 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 Sun 2024-07-07 18:01:01 MSK; 2s ago  
     Docs: man:httd.service(8)  
  Main PID: 5409 (httpd)  
    Status: "Started, listening on: port 443, port 80"  
    Tasks: 178 (limit: 10966)  
   Memory: 25.1M  
      CPU: 58ms  
   CGroup: /system.slice/httpd.service  
           └─5409 /usr/sbin/httpd -DFOREGROUND  
             └─5411 /usr/sbin/httpd -DFOREGROUND  
               └─5412 /usr/sbin/httpd -DFOREGROUND  
                 └─5413 /usr/sbin/httpd -DFOREGROUND  
                   └─5414 /usr/sbin/httpd -DFOREGROUND  
                     └─5415 /usr/sbin/httpd -DFOREGROUND  
  
Jul 07 18:01:01 eademidova.localdomain systemd[1]: Starting The Apache HTTP Server...  
Jul 07 18:01:01 eademidova.localdomain systemd[1]: Started The Apache HTTP Server.  
Jul 07 18:01:01 eademidova.localdomain httpd[5409]: Server configured, listening on: port 443, port 80  
[root@eademidova ~]#
```

Рис. 3.3: Проверка статуса веб-сервера

Найдите веб-сервер Apache в списке процессов, определим его контекст безопасности(рис. 3.4)

```
[root@eademidova ~]# ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root      5409  0.0  0.9 23544 10588 ?        Ss   18:01   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  5411  0.0  0.4 24248  8204 ?        S    18:01   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  5412  0.0  0.4 25596  7308 ?        S    18:01   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  5413  0.0  0.6 984852 12088 ?       Sl   18:01   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  5414  0.0  0.7 1115988 14536 ?       Sl   18:01   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  5415  0.0  0.6 984852 12076 ?       Sl   18:01   0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-c0:c0.c1923 root  5610  0.0  0.1 221664 2304 pts/0  S+   18:03   0:00 grep --color=auto httpd

[root@eademidova ~]# ps -eZ | grep httpd
system_u:system_r:httpd_t:s0  5409 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0  5411 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0  5412 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0  5413 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0  5414 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0  5415 ?        00:00:00 httpd
[root@eademidova ~]#
```

Рис. 3.4: Просмотр контекста безопасности веб-сервера

Мы можем видеть контекст безопасности SELinux: system_u:system_r:httpd_t.

Посмотрим текущее состояние переключателей SELinux для Apache(рис. 3.5)

```
root@eademidova:~
[root@eademidova ~]# 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_sss             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_openssl          off
```

Рис. 3.5: Состояние переключателей SELinux для Apache

Посмотрим статистику по политике с помощью команды seinfo(рис. 3.6):


```
zarara_domain
[root@eademidova ~]# seinfo
Statistics for policy file: /sys/fs/selinux/policy
Policy Version:          33 (MLS enabled)
Target Policy:           selinux
Handle unknown classes:  allow
Classes:                 135      Permissions:          457
Sensitivities:           1        Categories:          1024
Types:                   5145     Attributes:           259
Users:                   8         Roles:               15
Booleans:                356      Cond. Expr.:         388
Allow:                   65500     Neverallow:          0
Auditallow:              176      Dontaudit:           8682
Type_trans:              271770   Type_change:         94
Type_member:             37        Range_trans:         5931
Role allow:              40        Role_trans:          417
Constraints:             70       Validatetrans:       0
MLS Constrains:          72       MLS Val. Tran:       0
Permissives:             4        Polcap:              6
Defaults:                7        Typebounds:          0
Allowxperm:              0        Neverallowxperm:     0
Auditallowxperm:         0        Dontauditxperm:     0
Ibendportcon:            0        Ibpkeycon:           0
Initial SIDs:            27       Fs_use:              35
Genfscon:                109      Portcon:             665
Netifcon:                0        Nodecon:             0
[root@eademidova ~]# seinfo -u
```

Рис. 3.6: Статистика по политике

Также посмотрим множество пользователей, ролей, типов(рис. 3.7):

```
net11rcon: 0 nodecon: 0
[root@eademidova ~]# seinfo -u

Users: 8
  guest_u
  root
  staff_u
  sysadm_u
  system_u
  unconfined_u
  user_u
  xguest_u
[root@eademidova ~]# seinfo -r

Roles: 15
  auditadm_r
  container_user_r
  dbadm_r
  guest_r
  logadm_r
  nx_server_r
  object_r
  secadm_r
  staff_r
  sysadm_r
  system_r
  unconfined_r
  user_r
  webadm_r
  xguest_r
[root@eademidova ~]# seinfo -t

Types: 5145
  NetworkManager_dispatcher_chronyc_script_t
  NetworkManager_dispatcher_chronyc_t
  NetworkManager_dispatcher_cloud_script_t
  NetworkManager_dispatcher_cloud_t
  NetworkManager_dispatcher_console_script_t
  NetworkManager_dispatcher_console_t
  NetworkManager_dispatcher_console_var_run_t
  NetworkManager_dispatcher_custom_t
  NetworkManager_dispatcher_ddclient_script_t
  NetworkManager_dispatcher_ddclient_t
```

Рис. 3.7: Множества пользователей, ролей, типов

Определив тип файлов и поддиректорий, находящихся в директории /var/www, с помощью команды `ls -lZ /var/www`, увидим, что есть директория, содержащая cgi-скрипты, и директория /var/www/html, содержащая все скрипты httpd(в данный момент пустая)(рис. 3.8):

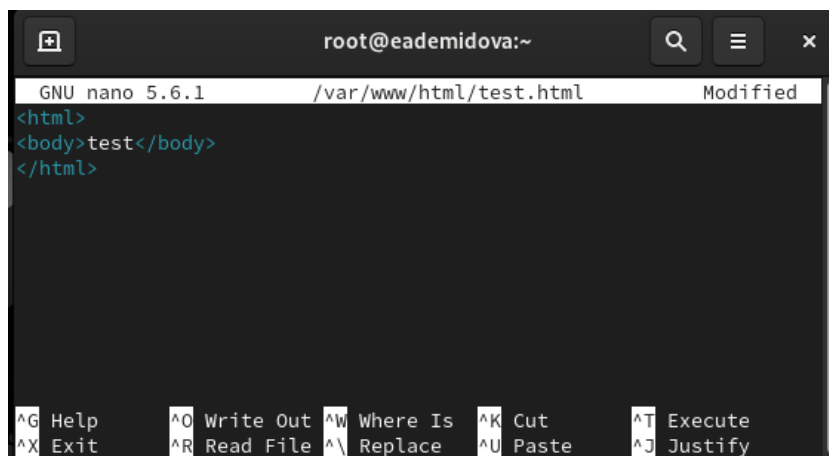
```
[root@eademidova ~]# ls -lZ /var/www/
total 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 Apr 22 04:04 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 6 Apr 22 04:04 html
[root@eademidova ~]# ls -lZ /var/www/html/
total 0
[root@eademidova ~]#
```

Рис. 3.8: Просмотр типов директорий в /var/www

Можно увидеть, что создание файлов в директории /var/www/html разрешено только владельцу – root.

Создадим от имени суперпользователя (так как в дистрибутиве после установки

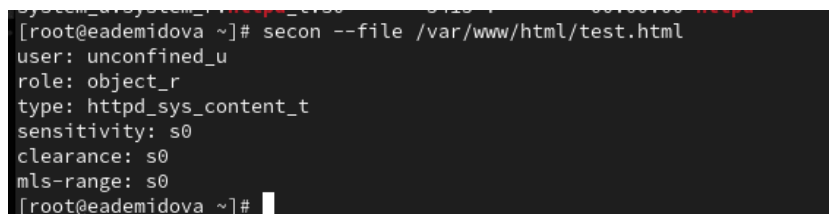
только ему разрешена запись в директорию) html-файл /var/www/html/test.html следующего содержания(рис. 3.9):



```
root@eademidova:~  
GNU nano 5.6.1 /var/www/html/test.html Modified  
<html>  
<body>test</body>  
</html>  
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute  
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

Рис. 3.9: Содержимое html-файла /var/www/html/test.html

Затем посмотрим контекст безопасности, который был задан по умолчанию этому файлу(3.10):



```
[root@eademidova ~]# secon --file /var/www/html/test.html  
user: unconfined_u  
role: object_r  
type: httpd_sys_content_t  
sensitivity: s0  
clearance: s0  
mls-range: s0  
[root@eademidova ~]#
```

Рис. 3.10: Установка пароля для пользователя с правами администратора

Увидим, что файлам по умолчанию сопоставляется свободный пользователь SELinux unconfined_u, указана роль object_r используется по умолчанию для файлов на «постоянных» носителях и на сетевых файловых системах и тип httpd_sys_content_t, который позволяет процессу httpd получить доступ к файлу

Обратимся к файлу через веб-сервер, введя в браузере адрес <http://127.0.0.1/test.html>, убедимся, что файл был успешно отображён.(рис. 3.11):

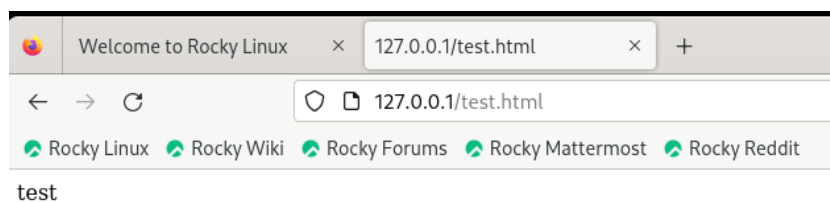


Рис. 3.11: Открытие html-страницы через браузер

Изучив справку `man httpd_selinux`, выясним, какие контексты файлов определены для `httpd`. Сопоставив их с типом файла `test.html` увидим, что его контекст `httpd_sys_content_t` для содержимого, которое должно быть доступно для всех скриптов `httpd` и для самого демона.

Изменим контекст файла `/var/www/html/test.html` с `httpd_sys_content_t` на тот, к которому процесс `httpd` не должен иметь доступа – `samba_share_t` (рис. 3.12):

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

Рис. 3.12: Изменение контекста файла `/var/www/html/test.html`

Теперь снова попробуем получить доступ к файлу через браузер и получим отказ (рис. 3.13):

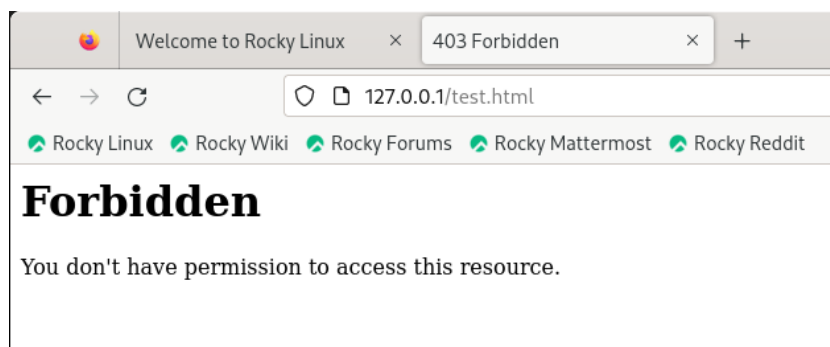


Рис. 3.13: Отказ в доступе к html-странице через браузер

Посмотрим log-файлы веб-сервера Apache и системный лог-файл и увидим,

что отказ происходит, так как доступ запрещен SELinux именно к веб-серверу(на просто просмотр текстовых файлов это не влияет)(рис. 3.14):

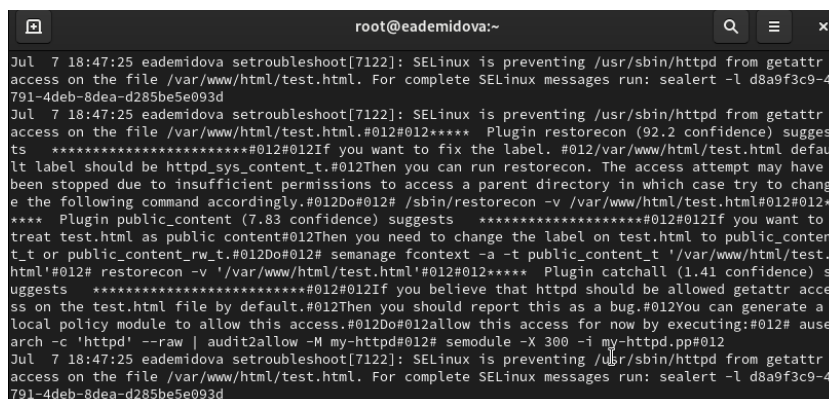


Рис. 3.14: Просмотр лог-файлов

Запустим веб-сервер Apache на прослушивание TCP-порта 81. Для этого в файле /etc/httpd/httpd.conf найдем строчку Listen 80 и заменим её на Listen 81(рис. 3.15):

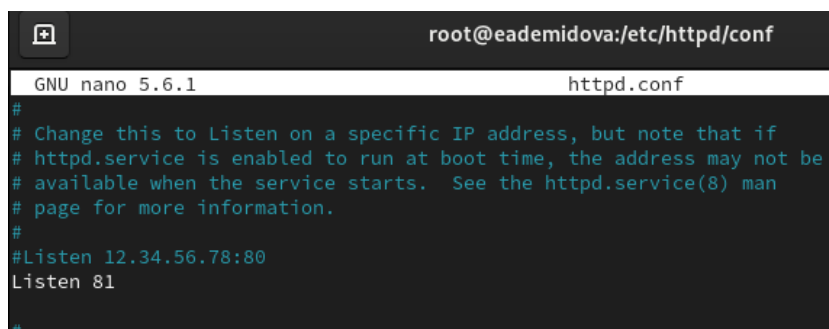


Рис. 3.15: Замена прослушиваемого порта

Выполнив перезапуск веб-сервера Apache и увидим предупреждение безопасности, так как 81 порт не является официальным портом для доступа по TCP(рис. 3.16):

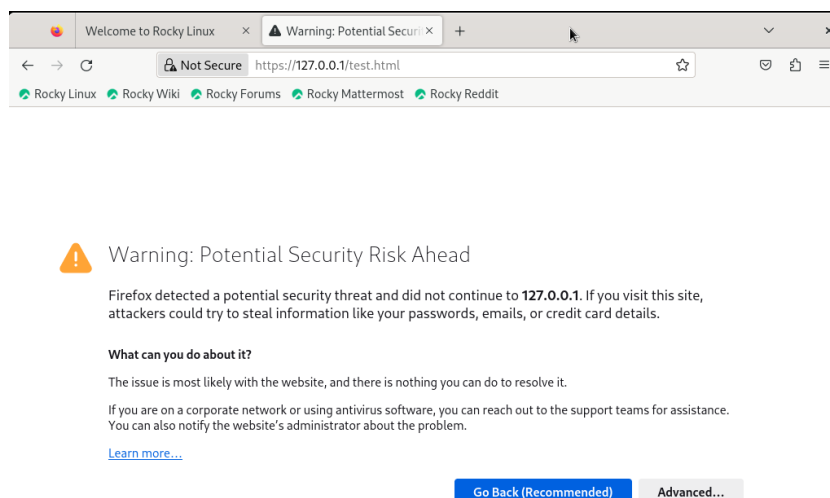


Рис. 3.16: Открытие html-страницы через браузер при прослушивании 81 порта

Просмотрев лог-файлы увидим, что порт для прослушивания был сменен(рис. 3.17):

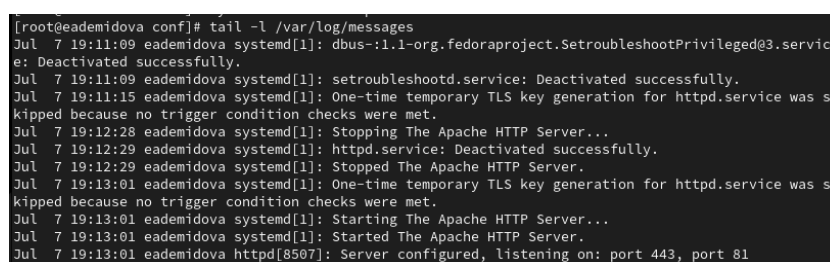


Рис. 3.17: Просмотр лог-файлов

Также этот порт мог быть отключен, тогда мы бы совсем не видели страницу, добавлять порты и просматривать актуальные можно с помощью команды seamanage(рис. ??):



Рис. 3.18: Просмотр портов с помощью seamanage

В конце работы вернем все сделанные изменения в файлах конфигурации

веб-сервера.

4 Выводы

В результате выполнения работы были приобретены практические навыки администрирования ОС Linux. Получено первое практическое знакомство с технологией SELinux. Проверена работа SELinux на практике совместно с веб-сервером Apache.

Список литературы

1. SELinux – описание и особенности работы с системой. Часть 1 [Электронный ресурс]. Habr, 2014. URL: <https://habr.com/ru/companies/kingservers/articles/209644/>.