

Основы информационной безопасности

Лабораторная работа № 6. Мандатное разграничение прав в Linux

Нзита Диатезилуа Катенди

Содержание

ПЦель работы	4
Теоретические сведения	5
Выполнение лабораторной работы	6
Выводы	16
Список литературы	17

Список иллюстраций

1	Подготовка лабораторного стенда	6
2	Проверка статуса SELinux	7
3	Проверка статуса веб-сервера	7
4	Просмотр контекста безопасности веб-сервера	8
5	Состояние переключателей SELinux для Apache	8
6	Статистика по политике	9
7	Множества пользователей, ролей, типов	10
8	Просмотр типов директорий в /var/www	10
9	Содержимое html-файла /var/www/html/test.html	11
10	Установка пароля для пользователя с правами администратора .	11
11	Открытие html-страницы через браузер	12
12	Изменение контекста файла /var/www/html/test.html	12
13	Отказ в доступе к html-странице через браузер	12
14	Просмотр лог-файлов	13
15	Замена прослушиваемого порта	13
16	Открытие html-страницы через браузер при прослушивании 81 порта	14
17	Просмотр лог-файлов	14
18	Просмотр портов с помощью seamnager	15

ПЦель работы

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

Теоретические сведения

SELinux (SELinux) — это система принудительного контроля доступа, реализованная на уровне ядра[@habr_selinux]. Впервые эта система появилась в четвертой версии CentOS, а в версиях 5 и 6 реализация была существенно дополнена и улучшена. Эти улучшения позволили SELinux стать универсальной системой, способной эффективно решать многие современные проблемы. Помните, что сначала применяется классическая система прав Unix, и управление перейдет к SELinux только в случае успешной первоначальной проверки.

Домен — это список действий, которые может выполнять процесс. Обычно домен определяется как минимально возможный набор действий, с помощью которых может функционировать процесс. Таким образом, если процесс дискредитирован, злоумышленник не сможет нанести большого ущерба.

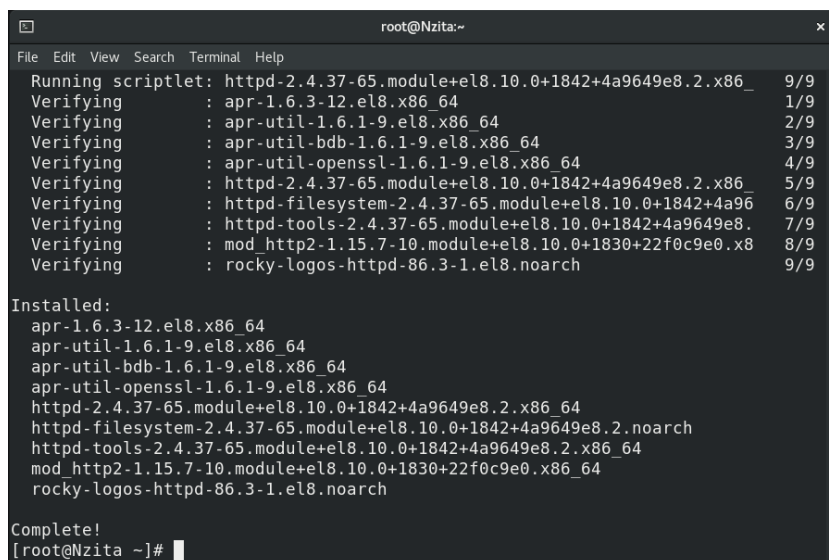
Функция — это список доменов, которые можно применить. Если заданный домен отсутствует в списке доменов роли, действия из этого домена не могут быть применены.

Тип — это набор разрешенных действий по отношению к объекту. Тип отличается от домена тем, что его можно применять к каналам, каталогам и файлам, тогда как домен применяется к процессам.

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

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. @fig:001)



```
root@Nzita:~  
File Edit View Search Terminal Help  
Running scriptlet: httpd-2.4.37-65.module+el8.10.0+1842+4a9649e8.2.x86_ 9/9  
Verifying      : apr-1.6.3-12.el8.x86_64 1/9  
Verifying      : apr-util-1.6.1-9.el8.x86_64 2/9  
Verifying      : apr-util-bdb-1.6.1-9.el8.x86_64 3/9  
Verifying      : apr-util-openssl-1.6.1-9.el8.x86_64 4/9  
Verifying      : httpd-2.4.37-65.module+el8.10.0+1842+4a9649e8.2.x86_ 5/9  
Verifying      : httpd-filesystem-2.4.37-65.module+el8.10.0+1842+4a96 6/9  
Verifying      : httpd-tools-2.4.37-65.module+el8.10.0+1842+4a9649e8. 7/9  
Verifying      : mod_http2-1.15.7-10.module+el8.10.0+1830+22f0c9e0.x8 8/9  
Verifying      : rocky-logos-httpd-86.3-1.el8.noarch 9/9  
  
Installed:  
apr-1.6.3-12.el8.x86_64  
apr-util-1.6.1-9.el8.x86_64  
apr-util-bdb-1.6.1-9.el8.x86_64  
apr-util-openssl-1.6.1-9.el8.x86_64  
httpd-2.4.37-65.module+el8.10.0+1842+4a9649e8.2.x86_64  
httpd-filesystem-2.4.37-65.module+el8.10.0+1842+4a9649e8.2.noarch  
httpd-tools-2.4.37-65.module+el8.10.0+1842+4a9649e8.2.x86_64  
mod_http2-1.15.7-10.module+el8.10.0+1830+22f0c9e0.x86_64  
rocky-logos-httpd-86.3-1.el8.noarch  
  
Complete!  
[root@Nzita ~]#
```

Рис. 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. @fig:002).

```
root@Nzita:~  
File Edit View Search Terminal Help  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@Nzita ~]# sudo usermod -aG wheel admin  
[root@Nzita ~]# sudo adduser user  
[root@Nzita ~]# sudo passwd user  
Changing password for user user.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@Nzita ~]# sudo nano /etc/selinux/config  
[root@Nzita ~]# sudo nano /etc/selinux/config  
[root@Nzita ~]# sudo systemctl restart httpd  
[root@Nzita ~]# iptables -F  
[root@Nzita ~]# iptables -P INPUT ACCEPT iptables -P OUTPUT ACCEPT  
Bad argument 'iptables'  
Try 'iptables -h' or 'iptables --help' for more information.  
[root@Nzita ~]# iptables -I INPUT -p tcp --dport 80 -j ACCEPT  
[root@Nzita ~]# iptables -I INPUT -p tcp --dport 81 -j ACCEPT  
[root@Nzita ~]# iptables -I OUT -p tcp --sport 80 -j ACCEPT  
iptables: No chain/target/match by that name.  
[root@Nzita ~]# iptables -I OUTPUT -p tcp --sport 80 -j ACCEPT  
[root@Nzita ~]# iptables -I OUTPUT -p tcp --sport 81 -j ACCEPT  
[root@Nzita ~]#
```

Рис. 2: Проверка статуса SELinux

Обратимся с помощью браузера к веб-серверу, запущенному на нашем компьютере, и убедитесь, что последний работает(рис. @fig:003).

```
root@Nzita:~  
File Edit View Search Terminal Help  
[root@Nzita ~]# sudo systemctl start httpd  
[root@Nzita ~]# sudo systemctl enable httpd  
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.  
[root@Nzita ~]# sudo systemctl status httpd  
● httpd.service - The Apache HTTP Server  
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor prese  
   Active: active (running) since Fri 2024-10-11 13:54:33 MSK; 1min 26s ago  
     Docs: man:httpd.service(8)  
   Main PID: 41214 (httpd)  
     Status: "Running, listening on: port 80"  
    Tasks: 213 (limit: 12244)  
   Memory: 26.5M  
   CGroup: /system.slice/httpd.service  
           └─41214 /usr/sbin/httpd -DFOREGROUND  
             └─41215 /usr/sbin/httpd -DFOREGROUND  
               └─41216 /usr/sbin/httpd -DFOREGROUND  
                 └─41217 /usr/sbin/httpd -DFOREGROUND  
                   └─41218 /usr/sbin/httpd -DFOREGROUND  
  
Oct 11 13:54:32 Nzita.localdomain systemd[1]: Starting The Apache HTTP Server...  
Oct 11 13:54:33 Nzita.localdomain systemd[1]: Started The Apache HTTP Server.  
Oct 11 13:54:33 Nzita.localdomain httpd[41214]: Server configured, listening on  
lines 1-18/18 (END)
```

Рис. 3: Проверка статуса веб-сервера

Найдите веб-сервер Apache в списке процессов, определим его контекст безопасности(рис. @fig:004)

```
[root@Nzita ~]# ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root 41599 0.0 0.5 265184 11616 ?
Ss 14:00 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 41601 0.0 0.4 269888 8696 ?
S 14:00 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 41602 0.0 0.7 1458808 14552 ?
Sl 14:00 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 41603 0.0 0.6 1327680 12500 ?
Sl 14:00 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 41604 0.0 0.6 1327680 12500 ?
Sl 14:00 0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 root 41994 0.0 0.0 222012
1088 pts/0 R+ 14:13 0:00 grep --color=auto httpd
[root@Nzita ~]# ps -eZ | grep httpd
system_u:system_r:httpd_t:s0 41599 ? 00:00:00 httpd
system_u:system_r:httpd_t:s0 41601 ? 00:00:00 httpd
system_u:system_r:httpd_t:s0 41602 ? 00:00:00 httpd
system_u:system_r:httpd_t:s0 41603 ? 00:00:00 httpd
system_u:system_r:httpd_t:s0 41604 ? 00:00:00 httpd
[root@Nzita ~]#
```

Рис. 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. @fig:005)

```
[root@Nzita ~]# 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_network_connect off
httpd_can_network_connect_cobbler off
httpd_can_network_connect_db off
httpd_can_network_memcache off
httpd_can_network_redis 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
```

Рис. 5: Состояние переключателей SELinux для Apache

Посмотрим статистику по политике с помощью команды seinfo(рис. @fig:006):


```
root@Nzita:~  
File Edit View Search Terminal Help  
Policy Version: 31 (MLS enabled)  
Target Policy: selinux  
Handle unknown classes: allow  
Classes: 132 Permissions: 464  
Sensitivities: 1 Categories: 1024  
Types: 5015 Attributes: 258  
Users: 8 Roles: 15  
Booleans: 349 Cond. Expr.: 399  
Allow: 116257 Neverallow: 0  
Auditallow: 172 Dontaudit: 10529  
Type_trans: 262670 Type_change: 94  
Type_member: 37 Range_trans: 5989  
Role_allow: 40 Role_trans: 421  
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: 34  
Genfscon: 107 Portcon: 649  
Netifcon: 0 Nodecon: 0  
[root@Nzita ~]#
```

Рис. 6: Статистика по политике

Также просмотрим множество пользователей, ролей, типов(рис. @fig:007):

```
root@Nzita:~  
[root@Nzita ~]# seinfo -u  
Users: 8  
  guest_u  
  root  
  staff_u  
  sysadm_u  
  system_u  
  unconfined_u  
  user_u  
  xguest_u  
[root@Nzita ~]# 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@Nzita ~]# seinfo -t  
Types: 5015  
  NetworkManager_etc_rw_t  
  NetworkManager_etc_t  
  NetworkManager_exec_t
```

Рис. 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. @fig:008):

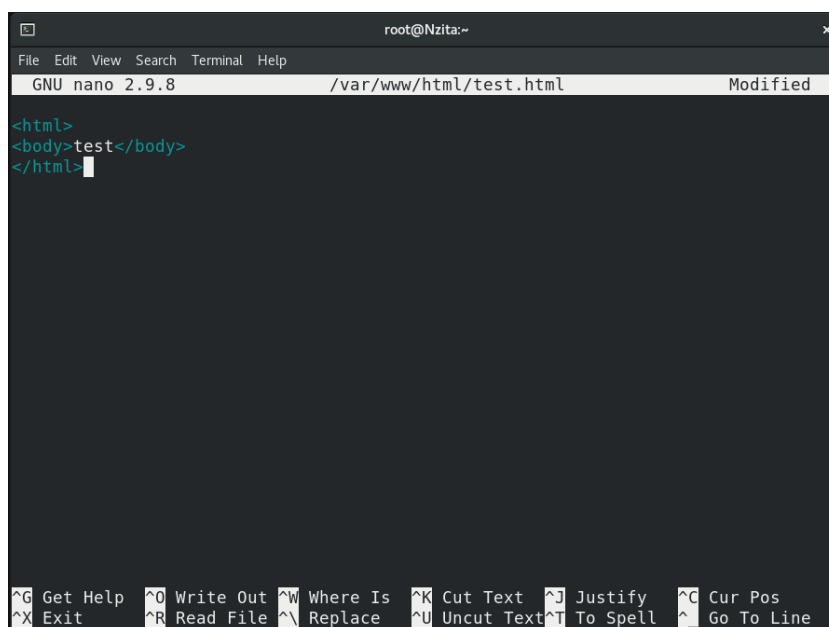
```
root@Nzita:~  
[root@Nzita ~]# ls -lZ /var/www/  
total 0  
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0  6 Aug 12 1  
1:14 cgi-bin  
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0    23 Oct 11 1  
4:23 html  
[root@Nzita ~]# ls -lZ /var/www/html  
total 4  
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 33 Oct 11 1  
4:23 test.html  
[root@Nzita ~]#
```

Рис. 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. @fig:009):



```
root@Nzita:~  
File Edit View Search Terminal Help  
GNU nano 2.9.8 /var/www/html/test.html Modified  
<html>  
<body>test</body>  
</html>  
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos  
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Рис. 9: Содержимое html-файла /var/www/html/test.html

Затем посмотрим контекст безопасности, который был задан по умолчанию этому файлу(@fig:010):



```
[root@Nzita ~]# 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@Nzita ~]#
```

Рис. 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. @fig:011):

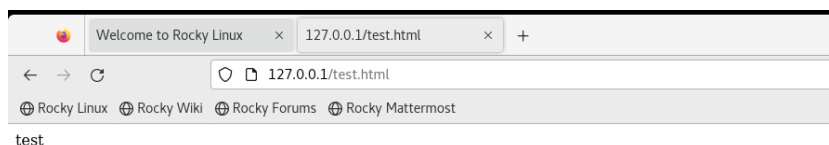


Рис. 11: Открытие html-страницы через браузер

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. @fig:012):



Рис. 12: Изменение контекста файла `/var/www/html/test.html`

Теперь снова попробуем получить доступ к файлу через браузер и получим отказ(рис. @fig:013):

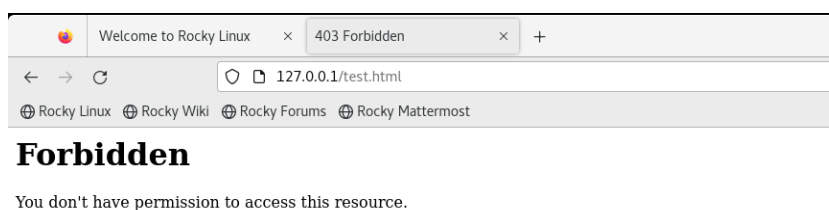
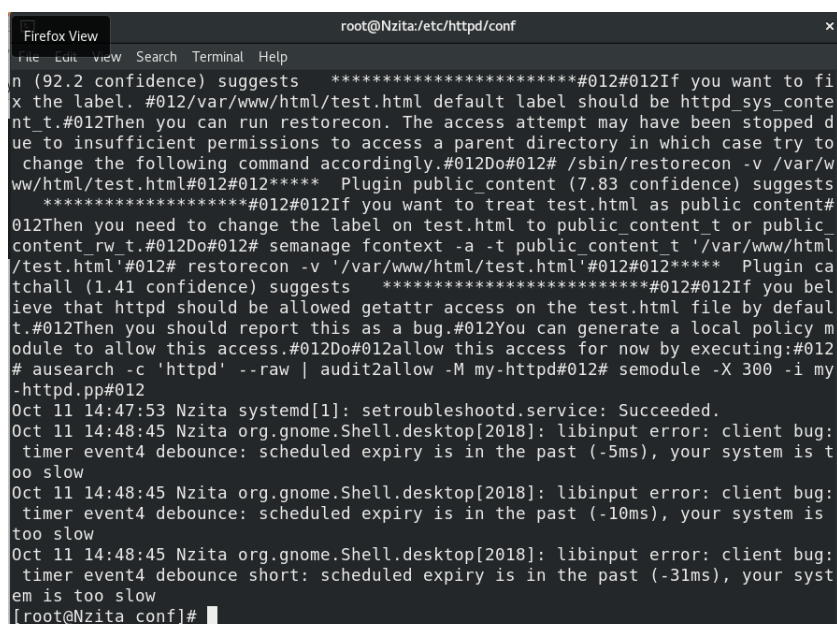


Рис. 13: Отказ в доступе к html-странице через браузер

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. @fig:014):



```
root@Nzita:/etc/httpd/conf#
n (92.2 confidence) suggests *****#012#012If you want to fi
x the label. #012/var/www/html/test.html default label should be httpd_sys_conte
nt_t.#012Then you can run restorecon. The access attempt may have been stopped d
ue to insufficient permissions to access a parent directory in which case try to
change the following command accordingly.#012Do#012# /sbin/restorecon -v /var/w
ww/html/test.html#012#012***** Plugin public_content (7.83 confidence) suggests
*****#012#012If you want to treat test.html as public content#
012Then you need to change the label on test.html to public_content_t or public_
content_rw_t.#012Do#012# semanage fcontext -a -t public_content_t '/var/www/html
/test.html'#012# restorecon -v '/var/www/html/test.html'#012#012***** Plugin ca
tchall (1.41 confidence) suggests *****#012#012If you bel
ieve that httpd should be allowed getattr access on the test.html file by default
t.#012Then you should report this as a bug.#012You can generate a local policy m
odule to allow this access.#012Do#012allow this access for now by executing:#012
# ausearch -c 'httpd' --raw | audit2allow -M my-httpd#012# semodule -X 300 -i my
-httpd.pp#012
Oct 11 14:47:53 Nzita systemd[1]: setroubleshootd.service: Succeeded.
Oct 11 14:48:45 Nzita org.gnome.Shell.desktop[2018]: libinput error: client bug:
timer event4 debounce: scheduled expiry is in the past (-5ms), your system is t
oo slow
Oct 11 14:48:45 Nzita org.gnome.Shell.desktop[2018]: libinput error: client bug:
timer event4 debounce: scheduled expiry is in the past (-10ms), your system is
too slow
Oct 11 14:48:45 Nzita org.gnome.Shell.desktop[2018]: libinput error: client bug:
timer event4 debounce short: scheduled expiry is in the past (-31ms), your syst
em is too slow
[root@Nzita conf]#
```

Рис. 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. @fig:015):



```
root@Nzita:/etc/httpd/conf#
GNU nano 2.9.8 httpd.conf Modified
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 specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 81
```

Рис. 15: Замена прослушиваемого порта

Выполнив перезапуск веб-сервера Apache и увидим предупреждение безопасности, так как 81 порт не является официальным портом для доступа по ТСП(рис. @fig:016):

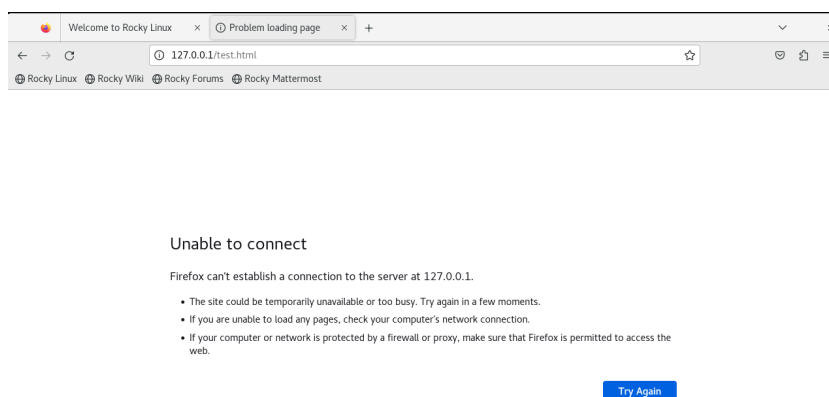


Рис. 16: Открытие html-страницы через браузер при прослушивании 81 порта

Просмотрев лог-файлы увидим, что порт для прослушивания был сменен(рис. @fig:017):

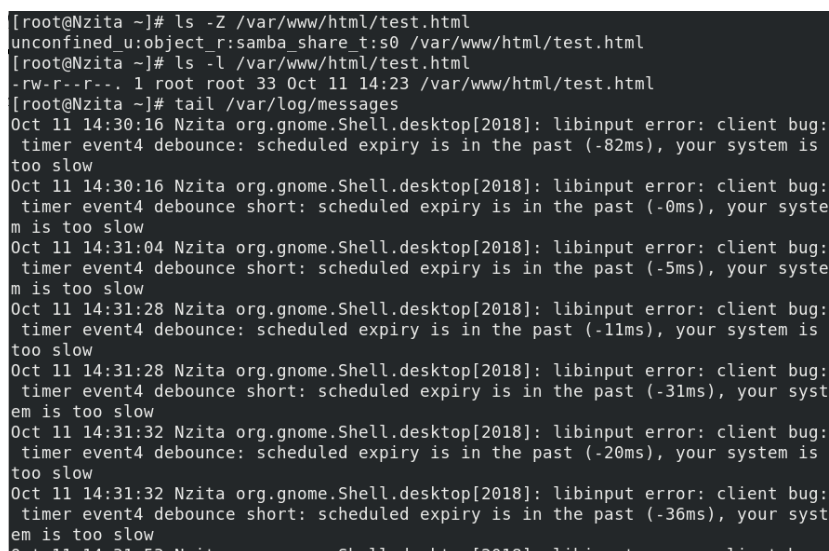
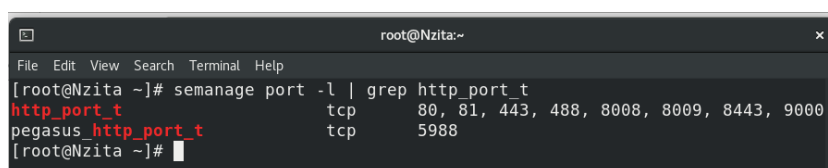


Рис. 17: Просмотр лог-файлов

Также этот порт мог быть отключен, тогда мы бы совсем не видели страницу, добавлять порты и просматривать актуальные можно с помощью команды `seamanager` (рис. @fig:018):



```
root@Nzita:~  
File Edit View Search Terminal Help  
[root@Nzita ~]# seamanager 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@Nzita ~]#
```

Рис. 18: Просмотр портов с помощью `seamanager`

В конце работы вернем все сделанные изменения в файлах конфигурации веб-сервера.

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