

Основы информационной безопасности

Лабораторная работа № 6. Дискреционное разграничение прав в Linux. Основные атрибуты

Подлесный Иван Сергеевич.

14.09.2024

Российский Университет дружбы народов

Информация

- Подлесный Иван Сергеевич
- студент группы НКНбд-01-21
- Российский университет дружбы народов

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.

Выполнение лабораторной работы

В конфигурационном файле `/etc/httpd/httpd.conf` зададим параметр `ServerName`. Отключим фильтр командами(рис. fig. 1)

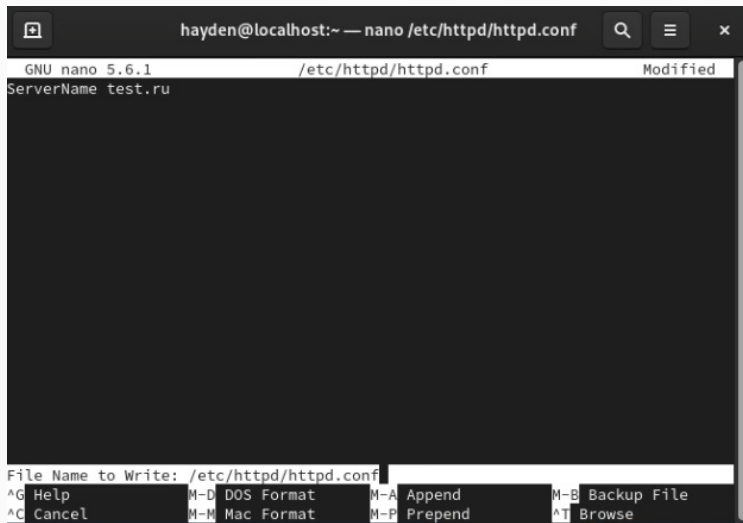
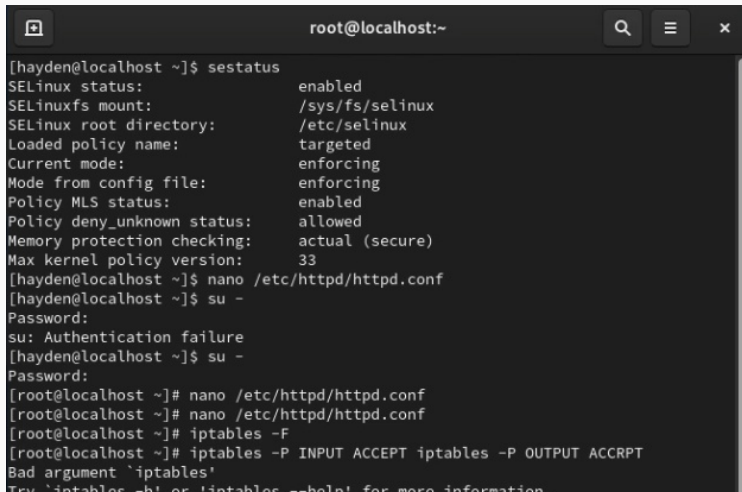


Figure 1: Подготовка лабораторного стенда

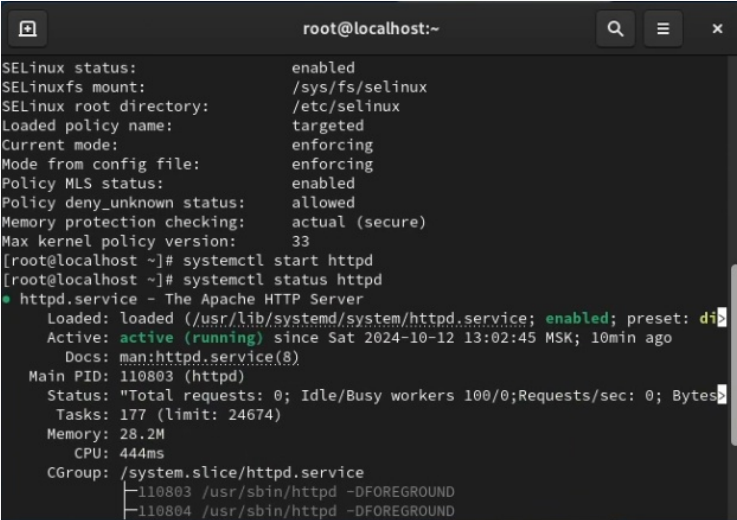
Войдем в систему с полученными учётными данными

убедимся, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` и `sestatus`(рис. fig. 2).

A terminal window titled 'root@localhost:~' with search, menu, and close icons in the title bar. The terminal shows the output of the 'sestatus' command, followed by attempts to edit '/etc/httpd/httpd.conf' using 'nano' and switch to root using 'su -'. The 'su -' attempts fail due to authentication failure. The root user then runs 'iptables -F' and 'iptables -P INPUT ACCEPT iptables -P OUTPUT ACCRPT', which results in a 'Bad argument `iptables`' error.

```
root@localhost:~  
[hayden@localhost ~]$ 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  
[hayden@localhost ~]$ nano /etc/httpd/httpd.conf  
[hayden@localhost ~]$ su -  
Password:  
su: Authentication failure  
[hayden@localhost ~]$ su -  
Password:  
[root@localhost ~]# nano /etc/httpd/httpd.conf  
[root@localhost ~]# nano /etc/httpd/httpd.conf  
[root@localhost ~]# iptables -F  
[root@localhost ~]# iptables -P INPUT ACCEPT iptables -P OUTPUT ACCRPT  
Bad argument `iptables`  
Try `iptables -h' or 'iptables --help' for more information
```


Проверим, что веб-сервер работает (рис. fig. 3).



```
root@localhost:~  
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@localhost ~]# systemctl start httpd  
[root@localhost ~]# systemctl status httpd  
● httpd.service - The Apache HTTP Server  
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)  
   Active: active (running) since Sat 2024-10-12 13:02:45 MSK; 10min ago  
     Docs: man:httpd.service(8)  
  Main PID: 110803 (httpd)  
    Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0; CPU usage: 0%"  
    Tasks: 177 (limit: 24674)  
   Memory: 28.2M  
      CPU: 444ms  
    CGroup: /system.slice/httpd.service  
            └─110803 /usr/sbin/httpd -DFOREGROUND  
              └─110804 /usr/sbin/httpd -DFOREGROUND
```

Figure 3: Проверка статуса веб-сервера

Найдем веб-сервер Apache в списке процессов, определим его контекст безопасности(рис. fig. 4)

```
[root@localhost ~]# ps auxZ | grep httpd
system_u:system_r:httpd_t:s0  root      110803  0.0  0.2  20152 11424 ?        Ss   13:02   0:00 /usr/sbin/httpd
-DFOREGROUND
system_u:system_r:httpd_t:s0  apache  110804  0.0  0.1  22032  7112 ?        S    13:02   0:00 /usr/sbin/httpd
-DFOREGROUND
system_u:system_r:httpd_t:s0  apache  110805  0.0  0.2  1440204 10920 ?      Sl   13:02   0:00 /usr/sbin/httpd
-DFOREGROUND
system_u:system_r:httpd_t:s0  apache  110806  0.0  0.3  1571340 13184 ?      Sl   13:02   0:00 /usr/sbin/httpd
-DFOREGROUND
system_u:system_r:httpd_t:s0  apache  110807  0.0  0.4  1440204 17128 ?      Sl   13:02   0:00 /usr/sbin/httpd
-DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 root    111144  0.0  0.0  221664 2304 pts/0  S+   13:15   0:00 grep -
-color=auto httpd
[root@localhost ~]# 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@localhost ~]#
```

Figure 4: Просмотр контекста безопасности веб-сервера

Посмотрим текущее состояние переключателей SELinux для Apache(рис. fig. 5)

```
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
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@localhost ~]#
```

Figure 5: Состояние переключателей SELinux для Apache

Посмотрим статистику по политике с помощью команды seinfo(рис. fig. 6):

```
* Waiting for authentication...
* Waiting in queue...
* Downloading packages...
* Requesting data...
* Testing changes...
* Installing packages...
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:                    65504   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 Constrain:            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
```

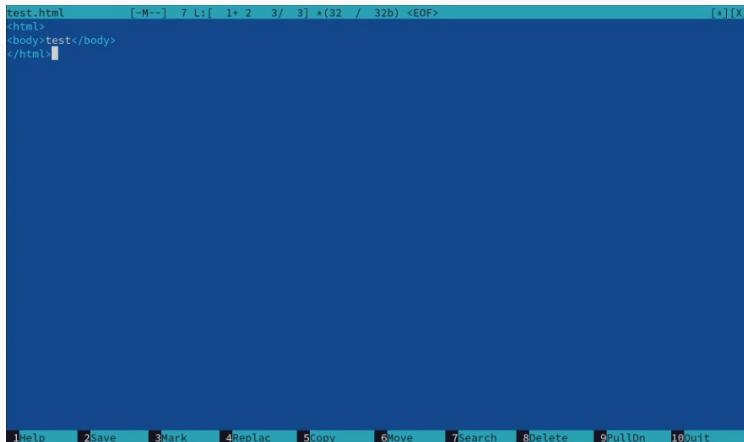
находящихся в директории `/var/www`, с помощью команды `ls -lZ /var/www`, увидим, что есть директория, содержащая cgi-скрипты, и директория `/var/www/html`, содержащая все скрипты httpd(в данный момент пустая)(рис. fig. 7):

```
[root@localhost ~]# ls -lZ /var/www
system_u:object_r:httpd_sys_script_exec_t:s0 cgi-bin      system_u:object_r:httpd_sys_content_t:s0 html
[root@localhost ~]# ls -lZ /var/www/html
[root@localhost ~]# cd /var/www/html
[root@localhost html]# touch test.html
```

Figure 7: Просмотр типов директорий в `/var/www`

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

Создадим от имени суперпользователя html-файл `/var/www/html/test.html` следующего содержания(рис. fig. 8):

A screenshot of a text editor window titled 'test.html'. The editor has a dark blue background and a light blue status bar at the top. The status bar shows the file name 'test.html', a cursor icon, and some statistics: '7 L: [1+ 2 3/ 3] *(32 / 32b) <EOF>'. The main editing area contains the following HTML code:

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

The cursor is positioned at the end of the last line. At the bottom of the window, there is a menu bar with the following items: 1Help, 2Save, 3Mark, 4Replac, 5Copy, 6Move, 7Search, 8Delete, 9PullDn, 10Quit.

Посмотрим контекст безопасности, заданный по умолчанию для html документа(fig. 9):

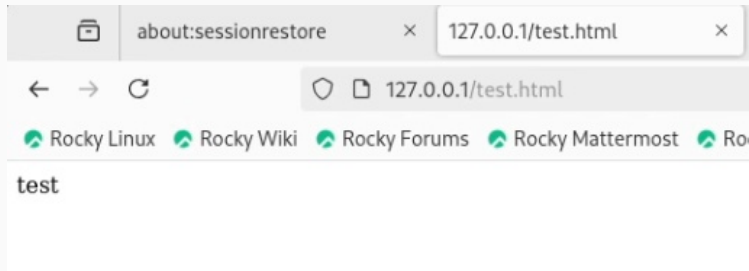
```
[root@localhost html]# secon --file /var/www/html/test.html
user: unconfined_u
role: object_r
type: httpd_sys_content_t
sensitivity: s0
clearance: s0
mls-range: s0
[root@localhost html]#
```

Figure 9: Установка пароля для пользователя с правами администратора

Увидим, что файлам по умолчанию сопоставляется свободный пользователь SELinux `unconfined_u`, указана роль `object_r`

Она 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. 10):



Изучим справку `man httpd_selinux` (через интернет, ибо команда не работает), 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. 11):

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

Figure 11: Изменение контекста файла `/var/www/html/test.html`

Теперь снова попробуем получить доступ к файлу через браузер и получим отказ(рис. fig. 12):

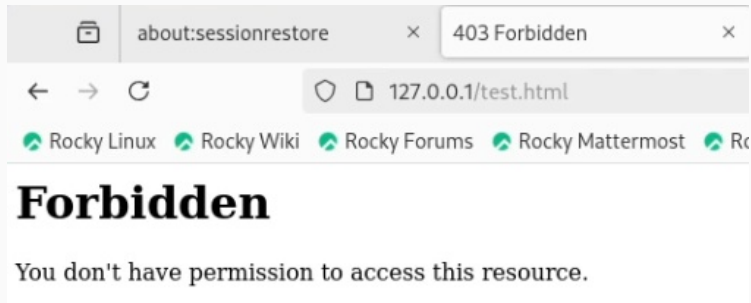
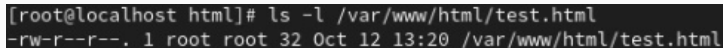


Figure 12: Отказ в доступе к html-странице через браузер

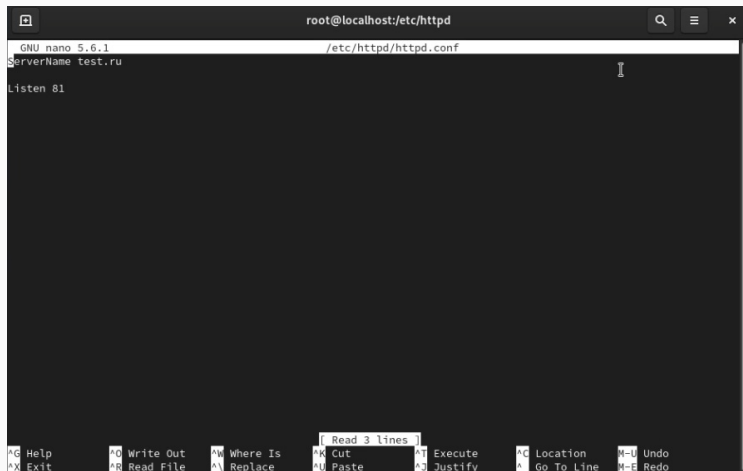
Посмотрим log-файлы веб-сервера Apache и системный лог-файл и увидим, что отказ происходит, так как доступ запрещен SELinux именно к веб-серверу(на просто просмотр текстовых файлов это не влияет)(рис. fig. 13):

A terminal window showing the command 'ls -l /var/www/html/test.html' and its output. The output shows the file is owned by root and has permissions -rw-r--r--.

```
[root@localhost html]# ls -l /var/www/html/test.html
-rw-r--r--. 1 root root 32 Oct 12 13:20 /var/www/html/test.html
```

Figure 13: 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. 14):



```
root@localhost:/etc/httpd
GNU nano 5.6.1 /etc/httpd/httpd.conf
ServerName test.ru
Listen 81

[ Read 3 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   ^U Undo
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^D Justify    ^G Go To Line ^E Redo
```

Figure 14: Замена прослушиваемого порта

Выполним перезапуск веб-сервера Apache и не увидим изменений по не понятным мне причинам, несмотря на то, что 81 порт не является официальным портом для доступа по TCP(рис. fig. 15):

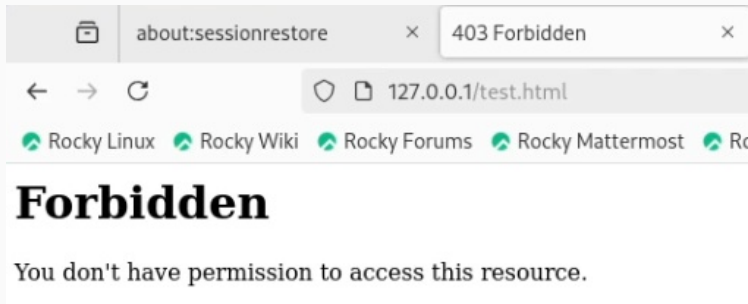


Figure 15: Открытие html-страницы через браузер при прослушивании 81 порта

Просмотрев лог-файлы увидим, что порт для прослушивания был сменен(рис. fig. 16):

```
[root@localhost httpd]# tail -n9 /var/log/messages
Oct 12 13:42:23 localhost systemd[1]: Started dbus-1.1-org.fedoraproject.SetroubleshootPrivileged5.service.
Oct 12 13:42:24 localhost setroubleshoot[113282]: SELinux is preventing /usr/sbin/httpd from getattr access on the file /var/www/html/test.html. For complete SEL
nux messages run: sealert -l 92ba58a9-bd6f-4b2d-8fdc-8968766caaa7
Oct 12 13:42:24 localhost setroubleshoot[113282]: SELinux is preventing /usr/sbin/httpd from getattr access on the file /var/www/html/test.html.#012#012***** Pl
gin restorecon (92.2 confidence) suggests *****#012Do#012# /sbin/restorecon -v /var/www/html/test.html#012#012***** Plugin public_content (7.83 confidence)
d_sys_content_t.#012Then you can run restorecon. The access attempt may have been stopped due to insufficient permissions to access a parent directory in which d
se try to change the following 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.html as public content#012Then you need to change the label on test.html to public_content_t or p
blic_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***** Plugi
catchall (1.41 confidence) suggests *****#012#012If you believe that httpd should be allowed getattr access on the test.html file by defa
lt.#012Then you should report this as a bug.#012You can generate a local policy module 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 12 13:42:24 localhost setroubleshoot[113282]: SELinux is preventing /usr/sbin/httpd from getattr access on the file /var/www/html/test.html. For complete SEL
nux messages run: sealert -l 92ba58a9-bd6f-4b2d-8fdc-8968766caaa7
Oct 12 13:42:24 localhost setroubleshoot[113282]: SELinux is preventing /usr/sbin/httpd from getattr access on the file /var/www/html/test.html.#012#012***** Pl
gin restorecon (92.2 confidence) suggests *****#012#012If you want to fix the label. #012/var/www/html/test.html default label should be htt
d_sys_content_t.#012Then you can run restorecon. The access attempt may have been stopped due to insufficient permissions to access a parent directory in which d
se try to change the following 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.html as public content#012Then you need to change the label on test.html to public_content_t or p
blic_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***** Plugi
catchall (1.41 confidence) suggests *****#012#012If you believe that httpd should be allowed getattr access on the test.html file by defa
lt.#012Then you should report this as a bug.#012You can generate a local policy module 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 12 13:42:34 localhost systemd[1]: dbus-1.1-org.fedoraproject.SetroubleshootPrivileged5.service: Deactivated successfully.
Oct 12 13:42:34 localhost systemd[1]: dbus-1.1-org.fedoraproject.SetroubleshootPrivileged5.service: Consumed 1.363s CPU time.
Oct 12 13:42:34 localhost systemd[1]: setroubleshootd.service: Deactivated successfully.
Oct 12 13:42:35 localhost systemd[1]: packagekit.service: Deactivated successfully.
[root@localhost httpd]# tail -n9 /var/log/httpd/access_log
tail: cannot open '/var/log/httpd/access_log' for reading: No such file or directory
[root@localhost httpd]# tail -n9 /var/log/httpd/access_log
```

Figure 16: 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 `seamange`(рис. fig. 17):

```
[root@localhost httpd]# 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
```

Figure 17: Просмотр портов с помощью `seamnage`

Выводы

В результате выполнения работы были получены базовые навыки работы с технологией SELinux. Проверена работа SELinux на практике совместно с веб-сервером Apache.