

Отчёт по лабораторной работе №6

Информационная безопасность

Мандатное разграничение прав в Linux

Выполнил: Прасолов Валерий Сергеевич
НПИбд-02-21, 1032212968

Содержание

| | | |
|---|---------------------------------------|----|
| 1 | Цель работы | 1 |
| 2 | Теоретическое введение..... | 1 |
| 3 | Выполнение лабораторной работы..... | 2 |
| 4 | Вывод..... | 10 |
| 5 | Список литературы. Библиография | 10 |

1 Цель работы

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

2 Теоретическое введение

1. **SELinux (Security-Enhanced Linux)** обеспечивает усиление защиты путем внесения изменений как на уровне ядра, так и на уровне пространства пользователя, что превращает ее в действительно «непробиваемую» операционную систему. Впервые эта система появилась в четвертой версии CentOS, а в 5 и 6 версии реализация была существенно дополнена и улучшена.

SELinux имеет три основных режим работы:

- Enforcing: режим по умолчанию. При выборе этого режима все действия, которые каким-то образом нарушают текущую политику безопасности, будут блокироваться, а попытка нарушения будет зафиксирована в журнале.
- Permissive: в случае использования этого режима, информация о всех действиях, которые нарушают текущую политику безопасности, будут зафиксированы в журнале, но сами действия не будут заблокированы.
- Disabled: полное отключение системы принудительного контроля доступа.

Политика SELinux определяет доступ пользователей к ролям, доступ ролей к доменам и доступ доменов к типам. Контекст безопасности — все атрибуты SELinux — роли, типы и домены. Более подробно см. в [1].

2. **Apache** — это свободное программное обеспечение, с помощью которого можно создать веб-сервер. Данный продукт возник как доработанная версия другого HTTP-клиента от национального центра суперкомпьютерных приложений (NCSA).

Для чего нужен Apache сервер:

- чтобы открывать динамические PHP-страницы,
- для распределения поступающей на сервер нагрузки,
- для обеспечения отказоустойчивости сервера,
- чтобы потренироваться в настройке сервера и запуске PHP-скриптов.

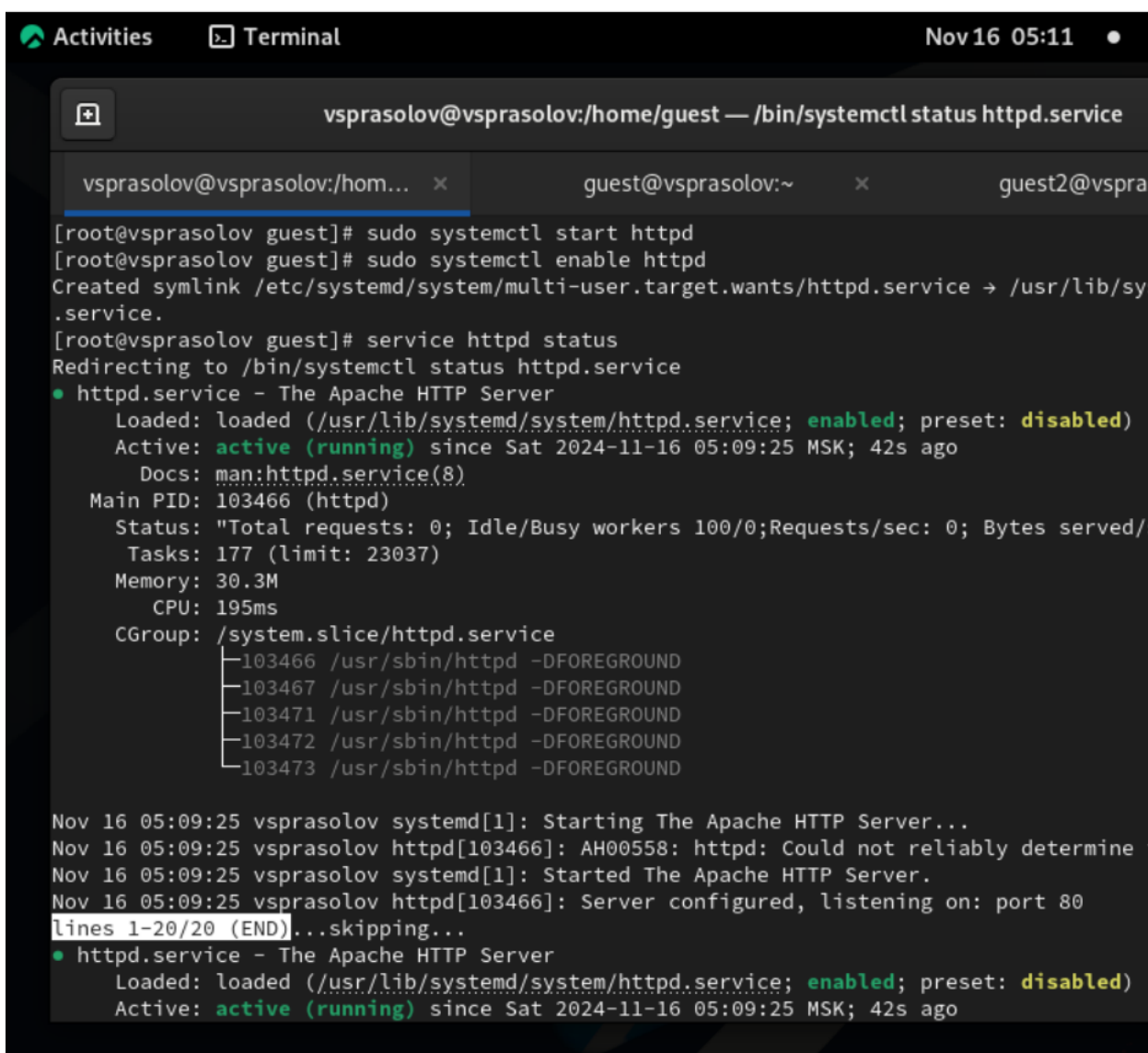
Apache является кроссплатформенным ПО и поддерживает такие операционные системы, как Linux, BSD, MacOS, Microsoft, BeOS и другие.

Более подробно см. в [2].

3 Выполнение лабораторной работы

Вошли в систему под своей учетной записью и убедились, что SELinux работает в режиме enforcing политики targeted с помощью команд “getenforce” и “sestatus”

Обратились с помощью браузера к веб-серверу, запущенному на компьютере, и убедились, что последний работает с помощью команды “service httpd status”



```
vsprsolov@vsprsolov:/home/guest — /bin/systemctl status httpd.service

vsprsolov@vsprsolov:/hom... x      guest@vsprsolov:~ x      guest2@vspra

[root@vsprsolov guest]# sudo systemctl start httpd
[root@vsprsolov guest]# sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/sy
.service.
[root@vsprsolov guest]# 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 Sat 2024-11-16 05:09:25 MSK; 42s ago
  Docs: man:httpd.service(8)
  Main PID: 103466 (httpd)
  Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/
  Tasks: 177 (limit: 23037)
  Memory: 30.3M
  CPU: 195ms
  CGroup: /system.slice/httpd.service
          └─103466 /usr/sbin/httpd -DFOREGROUND
             └─103467 /usr/sbin/httpd -DFOREGROUND
                └─103471 /usr/sbin/httpd -DFOREGROUND
                   └─103472 /usr/sbin/httpd -DFOREGROUND
                      └─103473 /usr/sbin/httpd -DFOREGROUND

Nov 16 05:09:25 vsprsolov systemd[1]: Starting The Apache HTTP Server...
Nov 16 05:09:25 vsprsolov httpd[103466]: AH00558: httpd: Could not reliably determine
Nov 16 05:09:25 vsprsolov systemd[1]: Started The Apache HTTP Server.
Nov 16 05:09:25 vsprsolov httpd[103466]: Server configured, listening on: port 80
lines 1-20/20 (END)...skipping...
• httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
  Active: active (running) since Sat 2024-11-16 05:09:25 MSK; 42s ago
```

(рис. 2. Проверка работы веб-сервера)

С помощью команды “ps auxZ | grep httpd” определили контекст безопасности веб-сервера Apache - httpd_t

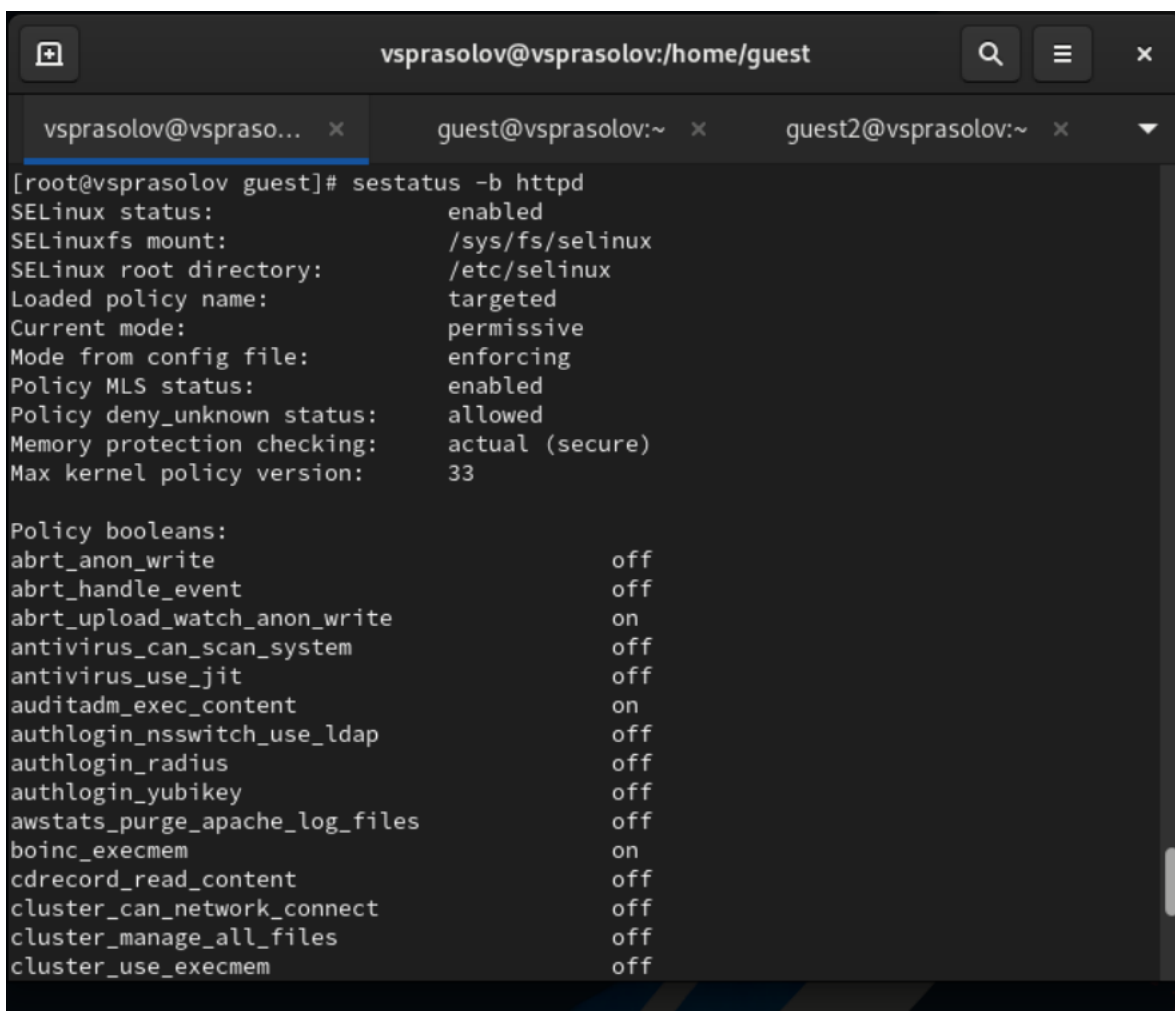
```

[root@vsprasolov guest]# ps auxZ | grep httpd
system_u:system_r:httpd_t:s0      root      103466  0.0  0.3  20364 11504 ?        Ss
05:09  0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0      apache    103467  0.0  0.1  22096  7120 ?        S
05:09  0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0      apache    103471  0.0  0.4 1440272 15288 ?        Sl
05:09  0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0      apache    103472  0.0  0.4 1440272 15260 ?        Sl
05:09  0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0      apache    103473  0.0  0.3 1571408 13280 ?        Sl
05:09  0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 root 103695 0.0 0.0 221664 23
04 pts/0 S+ 05:13  0:00 grep --color=auto httpd
[root@vsprasolov guest]# ps -eZ | grep httpd
system_u:system_r:httpd_t:s0      103466 ?                00:00:00 httpd
system_u:system_r:httpd_t:s0      103467 ?                00:00:00 httpd
system_u:system_r:httpd_t:s0      103471 ?                00:00:00 httpd
system_u:system_r:httpd_t:s0      103472 ?                00:00:00 httpd
system_u:system_r:httpd_t:s0      103473 ?                00:00:00 httpd
[root@vsprasolov guest]#

```

(рис. 3. Контекст безопасности веб-сервера Apache)

Посмотрели текущее состояние переключателей SELinux для Apache с помощью команды “sestatus -bigrep httpd”, многие из переключателей находятся в положении “off”



The image shows a terminal window with a dark theme. The title bar at the top reads 'vsprasolov@vsprasolov:/home/guest'. Below the title bar, there are three tabs: 'vsprasolov@vspraso...', 'guest@vsprasolov:~', and 'guest2@vsprasolov:~'. The active tab is 'vsprasolov@vspraso...'. The terminal content shows the command 'sestatus -b httpd' being executed. The output is as follows:

```
[root@vsprasolov guest]# sestatus -b httpd
SELinux status:                enabled
SELinuxfs mount:              /sys/fs/selinux
SELinux root directory:      /etc/selinux
Loaded policy name:           targeted
Current mode:                 permissive
Mode from config file:       enforcing
Policy MLS status:           enabled
Policy deny_unknown status:   allowed
Memory protection checking:   actual (secure)
Max kernel policy version:    33

Policy booleans:
abrt_anon_write                off
abrt_handle_event              off
abrt_upload_watch_anon_write   on
antivirus_can_scan_system      off
antivirus_use_jit              off
auditadm_exec_content          on
authlogin_nsswitch_use_ldap    off
authlogin_radius               off
authlogin_yubikey              off
awstats_purge_apache_log_files off
boinc_execmem                  on
cdrecord_read_content           off
cluster_can_network_connect    off
cluster_manage_all_files       off
cluster_use_execmem            off
```

(рис. 4. Текущее состояние переключателей SELinux)

Посмотрели статистику по политике с помощью команды "seinfo". Множество пользователей - 8, ролей - 14, типов 5100

```
[root@vsprasolov guest]# 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:                   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
[root@vsprasolov guest]#
```

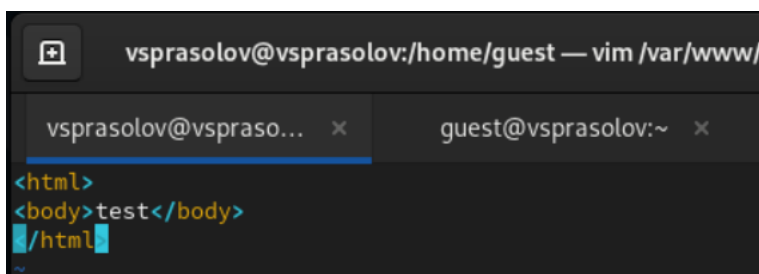
(рис. 5. Статистика по политике)

С помощью команды “ls -lZ /var/www” посмотрели файлы и поддиректории, находящиеся в директории /var/www. Используя команду “ls -lZ /var/www/html”, определили, что в данной директории файлов нет. Только владелец/суперпользователь может создавать файлы в директории /var/www/html

```
[root@vsprasolov guest]# ls -lZ /var/www/html
total 0
[root@vsprasolov guest]# ls -lZ /var/www
total 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 Aug  8 19:30
cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0      6 Aug  8 19:30
html
[root@vsprasolov guest]#
```

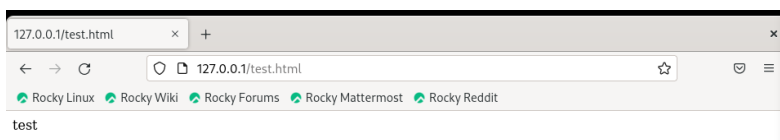
(рис. 6. Просмотр файлов и поддиректорий в директории /var/www)

От имени суперпользователя создали html-файл /var/www/html/test.html. Контекст созданного файла - httpd_sys_content_t



(рис. 7. Создание файла /var/www/html/test.html)

Обратились к файлу через веб-сервер, введя в браузере адрес “http://127.0.0.1/test.html”. Файл был успешно отображен



(рис. 8. Обращение к файлу через веб-сервер)

Изучив справку man httpd_selinux, выяснили, что для httpd определены следующие контексты файлов:

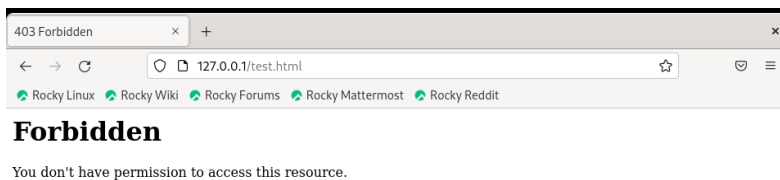
httpd_sys_content_t, httpd_sys_script_exec_t,

httpd_sys_script_ro_t, httpd_sys_script_rw_t,

httpd_sys_script_ra_t, httpd_unconfined_script_exec_t.

Контекст моего файла - httpd_sys_content_t (в таком случае содержимое должно быть доступно для всех скриптов httpd и для самого демона). Изменили контекст файла на samba_share_t командой “sudo chcon -t samba_share_t /var/www/html/test.html” и проверили, что контекст поменялся

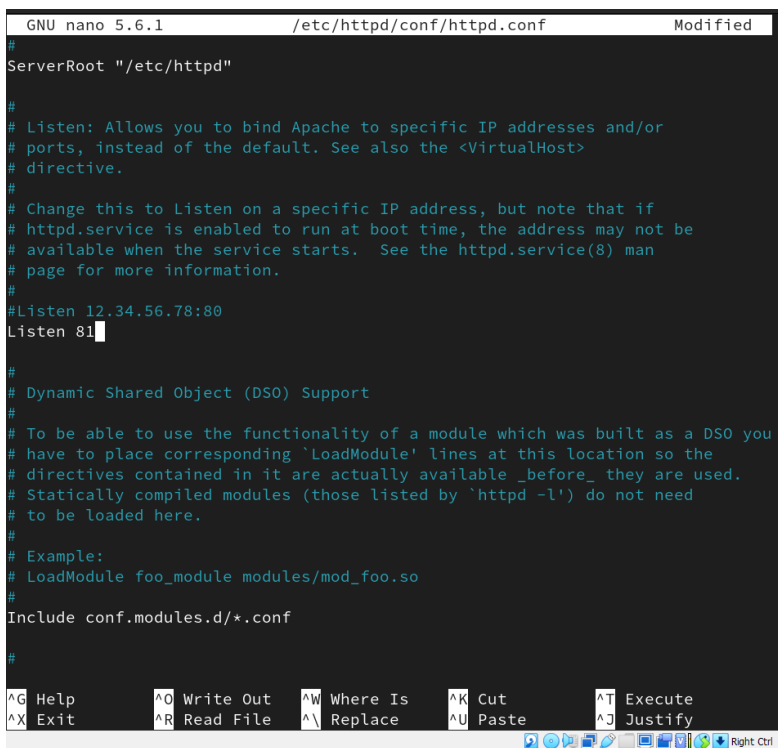
Попробовали еще раз получить доступ к файлу через веб-сервер, введя в браузере адрес “http://127.0.0.1/test.html” и получили сообщение об ошибке (т.к. к установленному ранее контексту процесс httpd не имеет доступа)



(рис. 10. Обращение к файлу через веб-сервер)

Командой “ls -l /var/www/html/test.html” убедились, что читать данный файл может любой пользователь. Просмотрели системный лог-файл веб-сервера Apache командой “sudo tail /var/log/messages”, отображающий ошибки

В файле /etc/httpd/conf/httpd.conf заменили строчку “Listen 80” на “Listen 81”, чтобы установить веб-сервер Apache на прослушивание TCP-порта 81



```
GNU nano 5.6.1 /etc/httpd/conf/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 a specific IP address, but note that if
# httpd.service is enabled to run at boot time, the address may not be
# available when the service starts. See the httpd.service(8) man
# page for more information.
#
#Listen 12.34.56.78:80
Listen 81

#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding 'LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by 'httpd -l') do not need
# to be loaded here.
#
# Example:
# LoadModule foo_module modules/mod_foo.so
#
Include conf.modules.d/*.conf

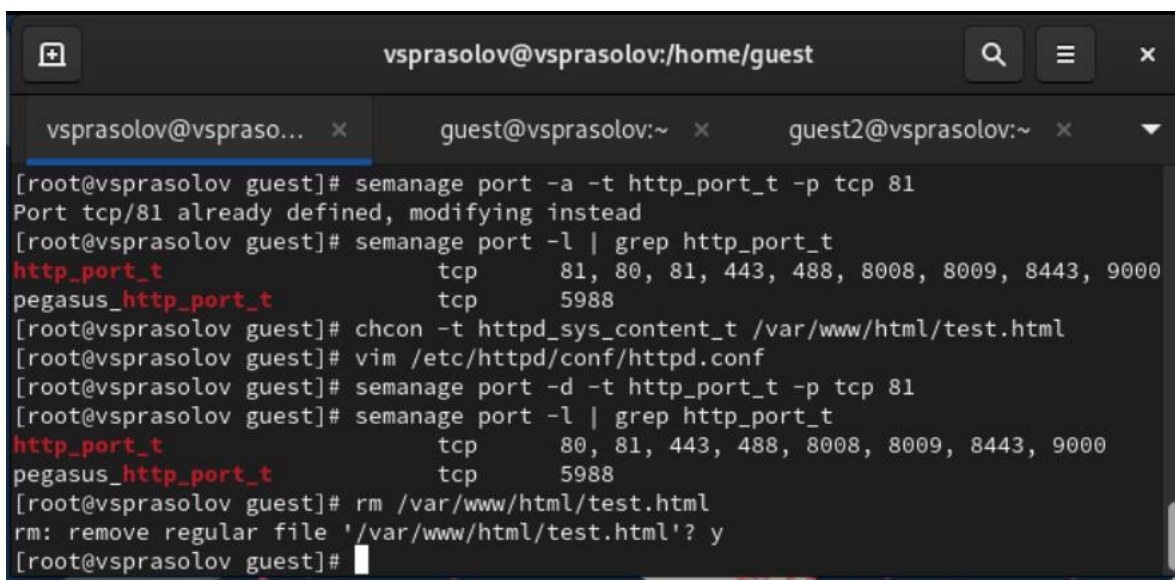
#
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify
```

(рис. 12. Установка веб-сервера Apache на прослушивание TCP-порта 81)

Перезапускаем веб-сервер Apache и анализируем лог-файлы командой “tail -nl /var/log/messages”

Просмотрели файлы “var/log/http/error_log”, “/var/log/http/access_log” и “/var/log/audit/audit.log” и выяснили, что запись появилась в последнем файле

Выполнили команду “semanage port -a -t http_port_t -p tcp 81” и убедились, что порт TCP-81 установлен. Проверили список портов командой “semanage port -l | grep http_port_t”, убедились, что порт 81 есть в списке и запускаем веб-сервер Apache снова



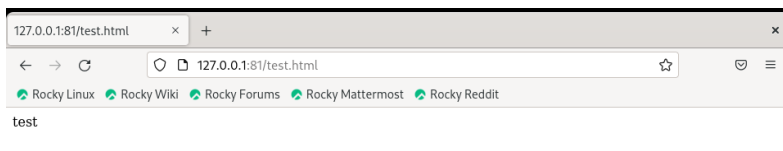
```
vsprasolov@vsprasolov:/home/guest
[root@vsprasolov guest]# semanage port -a -t http_port_t -p tcp 81
Port tcp/81 already defined, modifying instead
[root@vsprasolov guest]# semanage port -l | grep http_port_t
http_port_t          tcp      81, 80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t  tcp      5988
[root@vsprasolov guest]# chcon -t httpd_sys_content_t /var/www/html/test.html
[root@vsprasolov guest]# vim /etc/httpd/conf/httpd.conf
[root@vsprasolov guest]# semanage port -d -t http_port_t -p tcp 81
[root@vsprasolov guest]# 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@vsprasolov guest]# rm /var/www/html/test.html
rm: remove regular file '/var/www/html/test.html'? y
[root@vsprasolov guest]#
```

(рис. 15. Проверка установки порта 81)

Вернули контекст “httpd_sys_content_t” файлу “/var/www/html/test.html” командой “chcon -t httpd_sys_content_t /var/www/html/test.html” и после этого попробовали получить доступ к файлу через веб-сервер, введя адрес “http://127.0.0.1:81/test.html”, в результате чего увидели содержимое файла - слово “test”

```
[mvmalashenko@mvmalashenko ~]$ sudo chcon -t httpd_sys_content_t /var/www/html/test.html
[sudo] password for mvmalashenko:
[mvmalashenko@mvmalashenko ~]$ ls -Z /var/www/html/test.html
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test.html
```

(рис. 16. Возвращение исходного контекста файлу)



(рис. 17. Обращение к файлу через веб-сервер)

Исправили обратно конфигурационный файл apache, вернув “Listen 80”. Попытались удалить привязку http_port к 81 порту командой “semanage port -d -t http_port_t -p tcp 81”, но этот порт определен на уровне политики, поэтому его нельзя удалить

```

[mvmalashenko@mvmalashenko ~]$ nano /etc/httpd/conf/httpd.conf
[mvmalashenko@mvmalashenko ~]$ sudo semanage port -d -t http_port_t -p tcp 81
ValueError: Port tcp/81 is defined in policy, cannot be deleted
[mvmalashenko@mvmalashenko ~]$ sudo 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
[mvmalashenko@mvmalashenko ~]$ cat /etc/httpd/conf/httpd.conf
#
# This is the main Apache HTTP server configuration file. It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs/2.4/> for detailed information.
# In particular, see
# <URL:http://httpd.apache.org/docs/2.4/mod/directives.html>
# for a discussion of each configuration directive.
#
# See the httpd.conf(5) man page for more information on this configuration,
# and httpd.service(8) on using and configuring the httpd service.
#
# Do NOT simply read the instructions in here without understanding
# what they do. They're here only as hints or reminders. If you are unsure
# consult the online docs. You have been warned.
#
# Configuration and logfile names: If the filenames you specify for many
# of the server's control files begin with "/" (or "drive:/" for Win32), the
# server will use that explicit path. If the filenames do *not* begin
# with "/", the value of ServerRoot is prepended -- so 'log/access_log'
# with ServerRoot set to '/www' will be interpreted by the
# server as '/www/log/access_log', whereas '/log/access_log' will be
# interpreted as '/log/access_log'.
#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# Do not add a slash at the end of the directory path. If you point
# ServerRoot at a non-local disk, be sure to specify a local disk on the
# Mutex directive, if file-based mutexes are used. If you wish to share the
# same ServerRoot for multiple httpd daemons, you will need to change at
# least PidFile.
#
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 a specific IP address, but note that if
# httpd.service is enabled to run at boot time, the address may not be
# available when the service starts. See the httpd.service(8) man
# page for more information.
#
#Listen 12.34.56.78:80
Listen 80

```

(рис. 18. Возвращение Listen 80 и попытка удалить порт 81)

Удалили файл “/var/www/html/test.html” командой “rm /var/www/html/test.html”

```

[mvmalashenko@mvmalashenko ~]$ sudo rm /var/www/html/test.html
[mvmalashenko@mvmalashenko ~]$ ls /var/www/html/test.html
ls: cannot access '/var/www/html/test.html': No such file or directory
[mvmalashenko@mvmalashenko ~]$ ls /var/www/html

```

(рис. 19. Удаление файла test.html)

4 Вывод

В ходе выполнения данной лабораторной работы были развиты навыки администрирования ОС Linux, получено первое практическое знакомство с технологией SELinux и проверена работа SELinux на практике совместно с веб-сервером Apache.

5 Список литературы. Библиография

[0] Методические материалы курса

[1] SELinux: <https://habr.com/ru/companies/kingservers/articles/209644/>

[2] Apache: <https://2domains.ru/support/vps-i-servery/shto-takoye-apache>