
Front matter

lang: ru-RU title: "Лабораторная работа №6" subtitle: "Дисциплина: Основы информационной безопасности" author: "Георгес Геден"

Formatting

toc-title: "Содержание" toc: true # Table of contents toc_depth: 2 lof: true # Список рисунков lot: true # Список таблиц fontsize: 12pt linestretch: 1.5 papersize: a4paper documentclass: scrreprt polyglossia-lang: russian polyglossia-otherlangs: english mainfont: PT Serif romanfont: PT Serif sansfont: PT Sans monofont: PT Mono mainfontoptions: Ligatures=TeX romanfontoptions: Ligatures=TeX sansfontoptions: Ligatures=TeX,Scale=MatchLowercase monofontoptions: Scale=MatchLowercase indent: true pdf-engine: lualatex header-includes:

- `\linepenalty=10` # the penalty added to the badness of each line within a paragraph (no associated penalty node) Increasing the value makes tex try to have fewer lines in the paragraph.
- `\interlinepenalty=0` # value of the penalty (node) added after each line of a paragraph.
- `\hyphenpenalty=50` # the penalty for line breaking at an automatically inserted hyphen
- `\exhyphenpenalty=50` # the penalty for line breaking at an explicit hyphen
- `\binoppenalty=700` # the penalty for breaking a line at a binary operator
- `\relpenalty=500` # the penalty for breaking a line at a relation
- `\clubpenalty=150` # extra penalty for breaking after first line of a paragraph
- `\widowpenalty=150` # extra penalty for breaking before last line of a paragraph
- `\displaywidowpenalty=50` # extra penalty for breaking before last line before a display math
- `\brokenpenalty=100` # extra penalty for page breaking after a hyphenated line
- `\predisplaypenalty=10000` # penalty for breaking before a display
- `\postdisplaypenalty=0` # penalty for breaking after a display

- `\floatingpenalty = 20000` # penalty for splitting an insertion (can only be split footnote in standard LaTeX)
- `\raggedbottom` # or `\flushbottom`
- `\usepackage{float}` # keep figures where there are in the text
- `\floatplacement{figure}{H}` # keep figures where there are in the text

Цель работы

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

Теоретическое введение

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].

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:

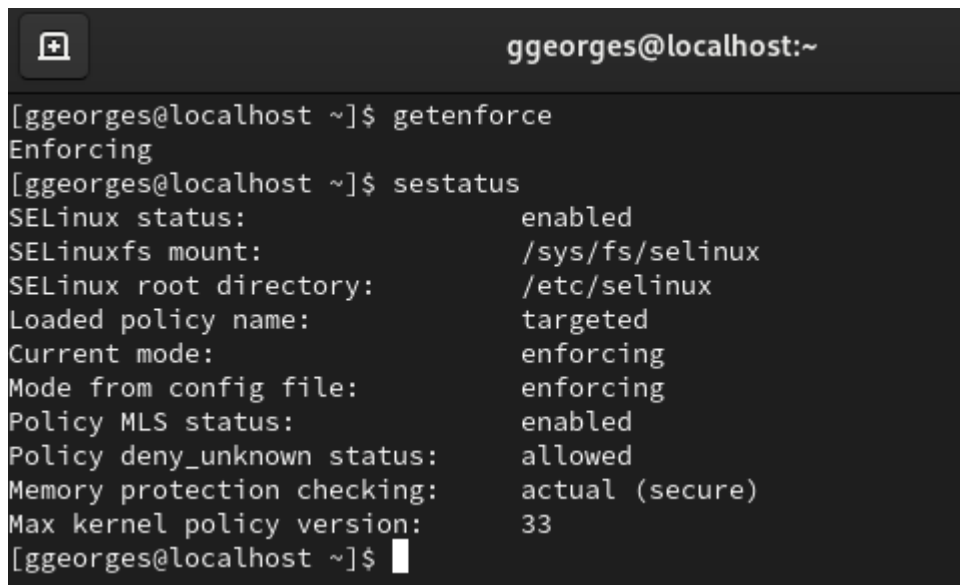
- чтобы открывать динамические PHP-страницы,
- для распределения поступающей на сервер

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].

Выполнение лабораторной работы

1)Входим в систему под своей учетной записью и убеждаемся, что SELinux работает в режиме enforcing политики targeted с помощью команд "getenforce" и "sestatus"(Рисунок 3.1).



```
ggeorges@localhost:~  
[ggeorges@localhost ~]$ getenforce  
Enforcing  
[ggeorges@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  
[ggeorges@localhost ~]$
```

width=70% }

2)Обращаемся с помощью браузера к веб-серверу, запущенному на моем компьютере, и убеждаемся, что последний работает с помощью команды "service httpd status" (Рисунок 3.2).

```
ggeorges@localhost:~ — /bin/systemctl status httpd.service

[ggeorges@localhost ~]$ service httpd start
Redirecting to /bin/systemctl start httpd.service
[ggeorges@localhost ~]$ 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)
   Drop-In: /usr/lib/systemd/system/httpd.service.d
            └─php-fpm.conf
   Active: active (running) since Tue 2024-10-08 20:47:26 MSK; 1min 3s ago
     Docs: man:httpd.service(8)
  Main PID: 1102 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0.00;CPU user 0.0%"
    Tasks: 177 (limit: 23032)
   Memory: 38.9M
      CPU: 392ms
    CGroup: /system.slice/httpd.service
            └─1102 /usr/sbin/httpd -DFOREGROUND
               1207 /usr/sbin/httpd -DFOREGROUND
               1208 /usr/sbin/httpd -DFOREGROUND
               1209 /usr/sbin/httpd -DFOREGROUND
               1210 /usr/sbin/httpd -DFOREGROUND

oct. 08 20:47:26 localhost.localdomain systemd[1]: Starting The Apache HTTP Server: httpd.service.
oct. 08 20:47:26 localhost.localdomain httpd[1102]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1 instead. Please set the 'ServerName' directive globally to suppress this message
oct. 08 20:47:26 localhost.localdomain httpd[1102]: Server configured, listening on: http://127.0.0.1:80
oct. 08 20:47:26 localhost.localdomain systemd[1]: Started The Apache HTTP Server: httpd.service.
lines 1-22/22 (END)...skipping...
```

{ width=70% }

3)С помощью команды “ps auxZ | grep httpd” определяем контекст безопасности веб-сервера Apache - httpd_t(Рисунок 3.3).

```
ggeorges@localhost:~

[ggeorges@localhost ~]$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root 1102 0.0 0.2 20152 10684 ?
Ss 20:47 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 root 1105 0.0 0.7 182340 26744 ?
Ss 20:47 0:00 php-fpm: master process (/etc/php-fpm.conf)
system_u:system_r:httpd_t:s0 apache 1201 0.0 0.3 184564 12912 ?
S 20:47 0:00 php-fpm: pool www
system_u:system_r:httpd_t:s0 apache 1203 0.0 0.3 184564 12912 ?
S 20:47 0:00 php-fpm: pool www
system_u:system_r:httpd_t:s0 apache 1204 0.0 0.3 184564 13040 ?
S 20:47 0:00 php-fpm: pool www
system_u:system_r:httpd_t:s0 apache 1205 0.0 0.3 184564 12912 ?
S 20:47 0:00 php-fpm: pool www
system_u:system_r:httpd_t:s0 apache 1206 0.0 0.3 184564 13040 ?
S 20:47 0:00 php-fpm: pool www
system_u:system_r:httpd_t:s0 apache 1207 0.0 0.1 22036 7116 ?
S 20:47 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 1208 0.1 0.5 2423316 19232 ?
Sl 20:47 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 1209 0.1 0.2 2161108 10832 ?
Sl 20:47 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 1210 0.1 0.4 2161108 15084 ?
Sl 20:47 0:00 /usr/sbin/httpd -DFOREGROUND
```

{ width=70% }

4)Посмотрим текущее состояние переключателей SELinux для Apache с помощью команды “sestatus -bigrep httpd”, многие из переключателей находятся в положении “off”(Рисунок 3.4).

```
ggeorges@localhost:~  
[ggeorges@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.  
[ggeorges@localhost ~]$ sestatus -b httpd  
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  
  
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
```

{ width=70% }

5)Посмотрим статистику по политике с помощью команды “seinfo”.
Множество пользователей - 8, ролей - 14, типов 5100 (Рисунок 3.5).

```
ggeorges@localhost:~  
[ggeorges@localhost ~]$ 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: 65508 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  
[ggeorges@localhost ~]$
```

{ width=70% }

6)С помощью команды "ls -lZ /var/www" посмотрим файлы и поддиректории, находящиеся в директории /var/www. Используя команду "ls -lZ /var/www/html", определяем, что в данной директории файлов нет. Только владелец или суперпользователь может создавать файлы в директории /var/www/html(Рисунок 3.6).

```
ggeorges@localhost:~  
[ggeorges@localhost ~]$ ls -lZ /var/www/  
total 4  
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6  
8 août 19:30 cgi-bin  
drwxr-xr-x. 12 apache apache system_u:object_r:httpd_sys_content_t:s0 4096 1  
1 sept. 17:40 html  
[ggeorges@localhost ~]$
```

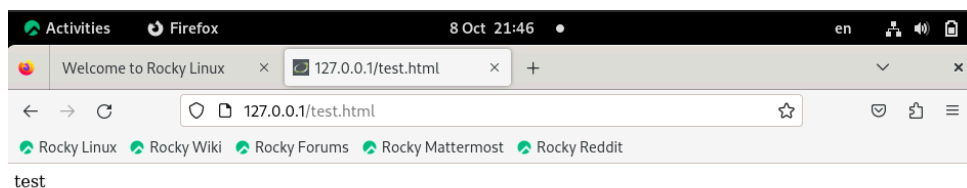
{ width=70% }

7)От имени суперпользователя создаём html-файл /var/www/html/test.html. Контекст созданного файла - httpd_sys_content_t (Рисунок 3.7).

```
ggeorges@localhost:~  
[ggeorges@localhost ~]$ su -  
Password:  
[root@localhost ~]# touch /var/www/html/test.html  
[root@localhost ~]# emacs test.html&  
[1] 49510  
[root@localhost ~]# bash: emacs: commande inconnue...  
Les paquets fournissant ces fichiers sont :  
'emacs-nox'  
'emacs-lucid'  
'emacs'  
  
[1]+  Termine 127          emacs test.html  
[root@localhost ~]# touch /var/www/html/test.html  
[root@localhost ~]# emacs test.html&  
[1] 49522  
[root@localhost ~]# bash: emacs: commande inconnue...  
Les paquets fournissant ces fichiers sont :  
'emacs-nox'  
'emacs-lucid'  
'emacs'  
  
[1]+  Termine 127          emacs test.html  
[root@localhost ~]# cat /var/www/html/test.html  
[root@localhost ~]# nano /var/www/html/test.html  
[root@localhost ~]# cat /var/www/html/test.html  
<html>  
<body>test</body>  
</html>  
[root@localhost ~]# exit  
déconnexion  
[ggeorges@localhost ~]$
```

{ width=70% }

8)Обращаемся к файлу через веб-сервер, введя в браузере адрес "<http://127.0.0.1/test.html>". Файл был успешно отображен (Рисунок 3.8).



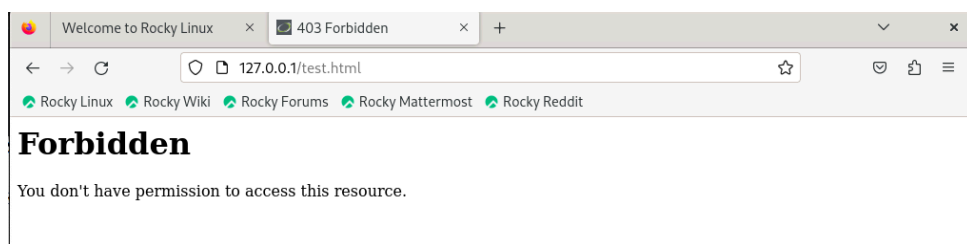
{ width=70% }

9)Изучив справку man httpd_selinux, выясняем, 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). Изменяем контекст файла на samba_share_t командой "sudo chcon -t samba_share_t /var/www/html/test.html" и проверяем, что контекст поменялся(Рисунок 3.9).

```
ggeorges@localhost:~  
[ggeorges@localhost ~]$ ls -Z /var/www/html/test.html  
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test.html  
[ggeorges@localhost ~]$ chcon -t samba_share_t /var/www/html/test.html  
chcon: failed to change context of '/var/www/html/test.html' to 'unconfined_u:object_r:samba_share_t:s0': Operation not permitted  
[ggeorges@localhost ~]$ sudo chcon -t samba_share_t /var/www/html/test.html  
[sudo] password for ggeorges:  
[ggeorges@localhost ~]$ ls -Z /var/www/html/test.html  
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html  
[ggeorges@localhost ~]$
```

{ width=70% }

10) Попробуем еще раз получить доступ к файлу через веб-сервер, введя в браузере адрес "<http://127.0.0.1/test.html>" и получаем сообщение об ошибке (т.к. к установленному ранее контексту процесс httpd не имеет доступа) (Рисунок 3.10).



{ width=70% }

11) Командой "`ls -l /var/www/html/test.html`" убеждаемся, что читать данный файл может любой пользователь. Просматриваем системный лог-файл веб-сервера Apache командой "`sudo tail /var/log/messages`", отображающий ошибки (Рисунок 3.11).


```
ggeorges@localhost:~  
[ggeorges@localhost ~]$ ls -l /var/www/html/test.html  
-rw-r--r--. 1 root root 34 Oct  8 21:40 /var/www/html/test.html  
[ggeorges@localhost ~]$ sudo tail /var/log/messages  
Oct  9 01:39:51 localhost systemd[1]: Starting SETroubleshoot daemon for process  
ing new SELinux denial logs...  
Oct  9 01:39:51 localhost systemd[1]: Started SETroubleshoot daemon for processi  
ng new SELinux denial logs.  
Oct  9 01:39:51 localhost setroubleshoot[3992]: failed to retrieve rpm info for  
path '/var/www/html/test.html':  
Oct  9 01:39:51 localhost systemd[1]: Started dbus-:1.1-org.fedoraproject.Setrou  
bleshootPrivileged@1.service.  
Oct  9 01:39:52 localhost setroubleshoot[3992]: SELinux interdit à /usr/sbin/htt  
pd d'utiliser l'accès getattr sur le fichier /var/www/html/test.html. Pour des m  
essages SELinux exhaustifs, lancez sealert -l 5aabe59f-ffa0-450d-9289-1d47d88fe3  
13  
Oct  9 01:39:52 localhost setroubleshoot[3992]: SELinux interdit à /usr/sbin/htt  
pd d'utiliser l'accès getattr sur le fichier /var/www/html/test.html.#012#012***  
** Le greffon restorecon (92.2 de confiance) suggère *****#012#  
012Si vous souhaitez corriger l'étiquette. #012L'étiquette par défaut de /var/ww  
w/html/test.html devrait être httpd_sys_content_t.#012Alors vous pouvez lancer r  
estorecon. La tentative d'accès pourrait avoir été stoppée due à des permissions  
insuffisantes d'accès au dossier parent, auquel cas essayez de changer la comma  
nde suivante en conséquence.#012Faire#012# /sbin/restorecon -v /var/www/html/tes  
t.html#012#012***** Le greffon public_content (7.83 de confiance) suggère ***  
*****#012#012Si vous souhaitez considérer test.html comme contenu public#  
012Alors vous devez modifier l'étiquette de test.html en public_content_t ou pub  
lic_content_rw_t.#012Faire#012# semanage fcontext -a -t public_content_t '/var/w  
ww/html/test.html'#012# restorecon -v '/var/www/html/test.html'#012#012***** Le  
greffon catchall (1.41 de confiance) suggère *****#012#012Si  
vous pensez que httpd devrait être autorisé à accéder getattr sur test.html file  
par défaut.#012Alors vous devriez rapporter ceci en tant qu'anomalie.#012Vous p  
ouvez générer un module de stratégie local pour autoriser cet accès.#012Faire#01
```

{ width=70% }

12)В файле /etc/httpd/conf/httpd.conf заменяем строчку "Listen 80" на "Listen 81", чтобы установить веб-сервер Apache на прослушивание TCP-порта 81 (Рисунок 3.12).

```
GNU nano 5.6.1 /etc/httpd/conf/httpd.conf Изменён  
# 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 n>  
# 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  
#
```

{ width=70% }

13)Перезапускаем веб-сервер Apache и анализируем лог-файлы командой "tail -nl /var/log/messages" (Рисунок 3.13).

```
root@localhost:~  
[ggeorges@localhost ~]$ su -  
Password:  
[root@localhost ~]# service httpd restart  
Redirecting to /bin/systemctl restart httpd.service  
[root@localhost ~]# tail -nl /var/www/log messages  
tail: nombre de lignes incorrect: « 1 »  
[root@localhost ~]# tail -n1 /var/www/log messages  
tail: impossible d'ouvrir '/var/www/log' en lecture: Aucun fichier ou dossier de ce type  
tail: impossible d'ouvrir 'messages' en lecture: Aucun fichier ou dossier de ce type  
[root@localhost ~]# tail -n1 /var/log/messages  
Oct 9 01:51:38 localhost systemd[1]: systemd-hostnamed.service: Deactivated successfully.  
[root@localhost ~]# tail -n3 /var/log/messages  
Oct 9 01:51:37 localhost systemd[1]: Started The Apache HTTP Server.  
Oct 9 01:51:37 localhost httpd[4142]: Server configured, listening on: port 80  
Oct 9 01:51:38 localhost systemd[1]: systemd-hostnamed.service: Deactivated successfully.  
[root@localhost ~]#
```

{ width=70% }

14) Просматриваем файлы “var/log/http/error_log”,
“/var/log/http/access_log” и “/var/log/audit/audit.log” и выясняем, что запись появилась в последнем файле(Рисунок 3.14).

```
root@localhost:~  
type=SERVICE_START msg=audit(1727543211.143:1562): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=systemd-hostnamed comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success' UID="root" AUID="unset"  
type=SERVICE_STOP msg=audit(1727543241.188:1563): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=systemd-hostnamed comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success' UID="root" AUID="unset"  
type=BPF msg=audit(1727543241.249:1564): prog-id=266 op=UNLOAD  
type=BPF msg=audit(1727543241.249:1565): prog-id=265 op=UNLOAD  
type=BPF msg=audit(1727543250.469:1566): prog-id=267 op=LOAD  
type=BPF msg=audit(1727543250.469:1567): prog-id=268 op=LOAD  
type=SERVICE_START msg=audit(1727543250.702:1568): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=systemd-hostnamed comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success' UID="root" AUID="unset"  
type=SERVICE_STOP msg=audit(1727543299.879:1569): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=systemd-hostnamed comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success' UID="root" AUID="unset"  
type=BPF msg=audit(1727543299.935:1570): prog-id=268 op=UNLOAD  
type=BPF msg=audit(1727543299.935:1571): prog-id=267 op=UNLOAD  
type=SERVICE_START msg=audit(1727546380.649:1572): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_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(1727546380.649:1573): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_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(1727777295.589:1574): prog-id=269 op=LOAD  
type=BPF msg=audit(1727777295.627:1575): prog-id=270 op=LOAD
```

{ width=70% }

15)Выполняем команду "semanage port -a -t http_port_t -p tcp 81" и убеждаемся, что порт TCP-81 установлен. Проверяем список портов командой "semanage port -l | grep http_port_t", убеждаемся, что порт 81 есть в списке и запускаем веб-сервер Apache снова (Рисунок 3.15).

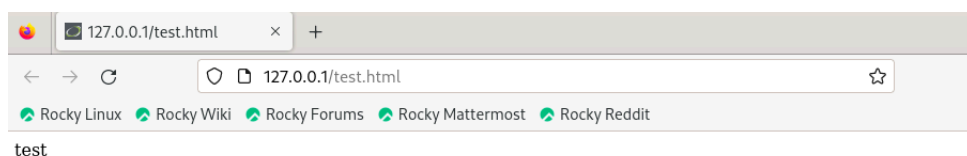
```
root@localhost:~  
[root@localhost ~]# semanage port -a -t http_port_t -p tcp 81  
Port tcp/81 already defined, modifying instead  
[root@localhost ~]# semanage port -l | grep http_port_t  
http_port_t          tcp      81, 80, 81, 443, 488, 8008, 8009, 8443,  
9000  
pegasus_http_port_t  tcp      5988  
[root@localhost ~]# service httpd restart  
Redirecting to /bin/systemctl restart httpd.service  
[root@localhost ~]#
```

{ width=70% }

16)Вернём контекст "httpd_sys_content_t" файлу
"/var/www/html/test.html" командой "chcon -t httpd_sys_content_t
/var/www/html/test.html" (Рисунок 3.16) и после этого пробуем
получить доступ к файлу через веб-сервер, введя адрес
"<http://127.0.0.1:81/test.html>", в результате чего увидим содежимое
файла - слово "test" (Рисунок 3.17).

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

{ width=70% }



{ width=70% }

17)Исправим обратно конфигурационный файл apache, вернув "Listen 80". Попытаемся 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(Рисунок 3.18).

```
root@localhost:~  
[root@localhost ~]# nano /etc/httpd/conf/httpd.conf  
[root@localhost ~]# semanage port -d -t http_port_t -p tcp 81  
[root@localhost ~]# 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@localhost ~]#
```

{ width=70% }

18) Удаляем файл `"/var/www/html/test.html"` командой `"rm /var/www/html/test.html"` (Рисунок 3.19).

```
root@localhost:~  
[root@localhost ~]# rm -R /var/www/html/test.html  
rm : supprimer '/var/www/html/test.html' du type fichier ? y  
[root@localhost ~]# ls /var/www/html/test.html  
ls: impossible d'accéder à '/var/www/html/test.html': Aucun fichier ou dossier d  
e ce type  
[root@localhost ~]#
```

{ width=70% }

Выводы

- В ходе выполнения данной лабораторной работы я развил навыки администрирования ОС Linux, получил первое практическое знакомство с технологией SELinux и проверил работу SELinux на практике совместно с веб-сервером Apache.