

# Лабораторная работа No 6

## Мандатное разграничение прав в Linux

Выполнила: Белкина Анастасия Михайловна, НБИбд-01-18

Преподаватель: Кулябов Дмитрий Сергеевич

Цель работы

---

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux<sup>1</sup>. Проверить работу SELinux на практике совместно с веб-сервером Apache.

## Теоретическое введение

---

В операционной системе Linux есть много отличных функций безопасности, но она из самых важных - это система прав доступа к файлам. Linux, как последователь идеологии ядра Linux в отличие от Windows, изначально проектировался как многопользовательская система, поэтому права доступа к файлам в Linux продуманы очень хорошо.

И это очень важно, потому что локальный доступ к файлам для всех программ и всех пользователей позволил бы вирусам без проблем уничтожить систему. Но новым пользователям могут показаться очень сложными новые права на файлы в Linux, которые очень сильно отличаются от того, что мы привыкли видеть в Windows. В этой статье мы попытаемся разобраться в том как работают права файлов в Linux, а также как их изменять и устанавливать.

Изначально каждый файл имел три параметра доступа. Вот они:

Чтение - разрешает получать содержимое файла, но на запись нет. Для каталога позволяет получить список файлов и каталогов, расположенных в нем;

Запись - разрешает записывать новые данные в файл или изменять существующие, а также позволяет создавать и изменять файлы и каталоги;

Выполнение - вы не можете выполнить программу, если у нее нет флага выполнения. Этот атрибут устанавливается для всех программ и скриптов, именно с помощью него система может понять, что этот файл нужно запускать как программу.

Но все эти права были бы бессмысленными, если бы применялись сразу для всех пользователей. Поэтому каждый файл имеет три категории пользователей, для которых можно устанавливать различные сочетания прав доступа:

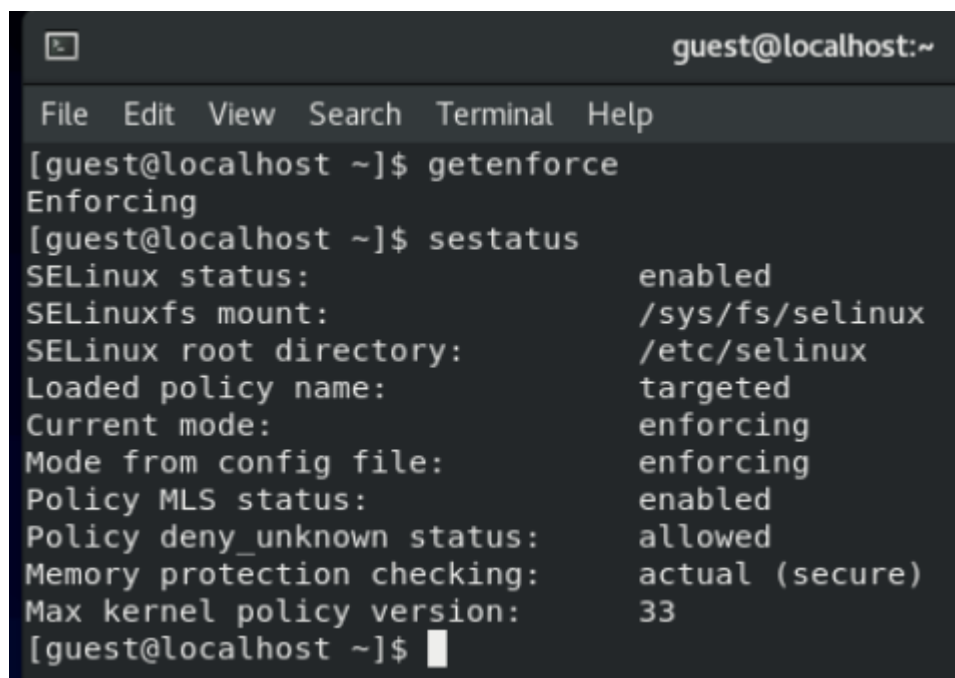
Владелец - набор прав для владельца файла, пользователя, который его создал или сейчас установлен его владельцем. Обычно владелец имеет все права, чтение, запись и выполнение. Группа - любая группа пользователей, существующая в системе и привязанная к файлу. Но это может быть только одна группа и обычно это группа владельца, хотя для файла можно назначить и другую группу. Остальные - все пользователи, кроме владельца и пользователей, входящих в группу файла. Именно с помощью этих наборов полномочий устанавливаются права файлов в Linux. Каждый пользователь может получить полный доступ только к файлам, владельцем которых он является или к тем, доступ к которым ему разрешен. Только пользователь Root может работать со всеми файлами независимо от их набора их полномочий.

Но со временем такой системы стало не хватать и было добавлено еще несколько флагов, которые позволяют делать файлы не изменяемыми или же выполнять от имени суперпользователя

## Выполнение лабораторной работы

---

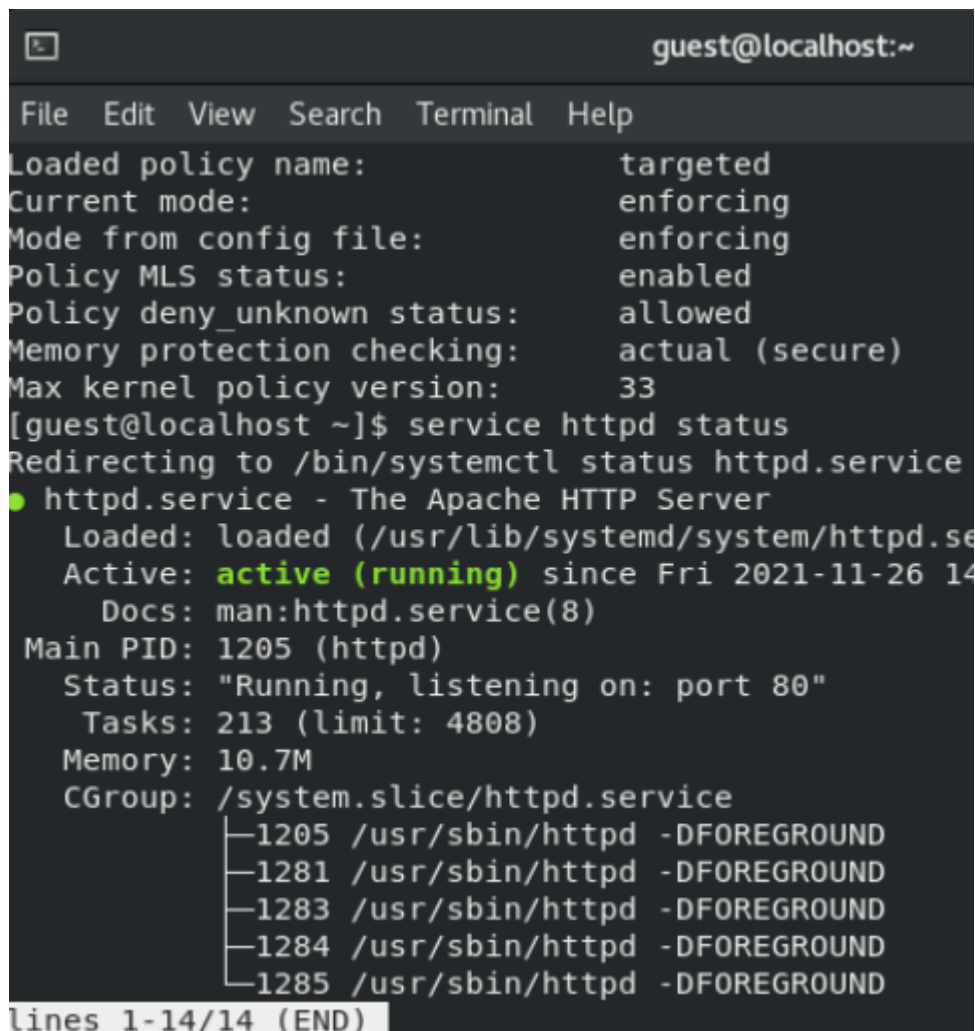
1. Вошла в систему с полученными учётными данными и убедилась, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` и `sestatus`.

A terminal window titled 'guest@localhost:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The user has entered the command 'getenforce' which returns 'Enforcing'. Then, the user enters 'sestatus' which displays the following SELinux configuration details: 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), and Max kernel policy version: 33.

```
guest@localhost:~  
File Edit View Search Terminal Help  
[guest@localhost ~]$ getenforce  
Enforcing  
[guest@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  
[guest@localhost ~]$
```

Рис.1 SELinux работает

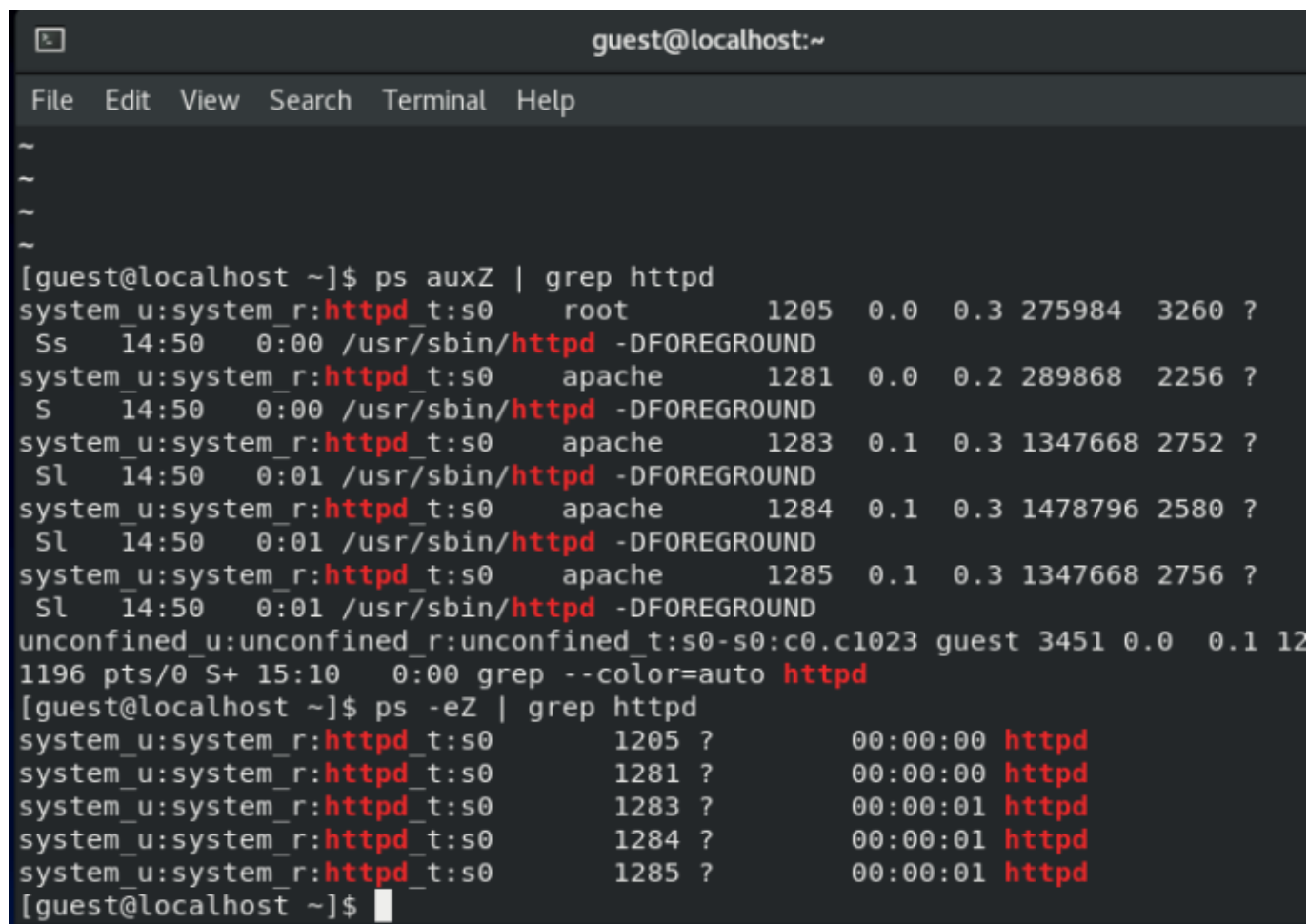
2. Обратилась с помощью браузера к веб-серверу, запущенному на компьютере, и убедилась, что последний работает: `service httpd status` или `/etc/rc.d/init.d/httpd status`

A terminal window titled 'guest@localhost:~' with a menu bar (File, Edit, View, Search, Terminal, Help). It shows the same SELinux status as the previous screenshot. Then, the user enters 'service httpd status'. The output indicates that the httpd.service is active and running. It provides details such as the main PID (1205), status ('Running, listening on: port 80'), tasks (213), memory usage (10.7M), and CGroup. A list of child processes is also shown at the bottom.

```
guest@localhost:~  
File Edit View Search Terminal Help  
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  
[guest@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; vendor preset: enabled)  
   Active: active (running) since Fri 2021-11-26 14:00:00 MSK; 1min 1s ago  
     Docs: man:httpd.service(8)  
  Main PID: 1205 (httpd)  
    Status: "Running, listening on: port 80"  
   Tasks: 213 (limit: 4808)  
  Memory: 10.7M  
    CGroup: /system.slice/httpd.service  
            └─1205 /usr/sbin/httpd -DFOREGROUND  
              └─1281 /usr/sbin/httpd -DFOREGROUND  
                └─1283 /usr/sbin/httpd -DFOREGROUND  
                  └─1284 /usr/sbin/httpd -DFOREGROUND  
                    └─1285 /usr/sbin/httpd -DFOREGROUND  
lines 1-14/14 (END)
```

Рис.2 Обращение к веб-серверу

3. Нашла веб-сервер Apache в списке процессов, определила его контекст безопасности и занесла эту информацию в отчёт. Например, можно использовать команду `ps auxZ | grep httpd` или `ps -eZ | grep httpd`



```
guest@localhost:~  
File Edit View Search Terminal Help  
~  
~  
~  
~  
[guest@localhost ~]$ ps auxZ | grep httpd  
system_u:system_r:httpd_t:s0 root 1205 0.0 0.3 275984 3260 ?  
Ss 14:50 0:00 /usr/sbin/httpd -DFOREGROUND  
system_u:system_r:httpd_t:s0 apache 1281 0.0 0.2 289868 2256 ?  
S 14:50 0:00 /usr/sbin/httpd -DFOREGROUND  
system_u:system_r:httpd_t:s0 apache 1283 0.1 0.3 1347668 2752 ?  
Sl 14:50 0:01 /usr/sbin/httpd -DFOREGROUND  
system_u:system_r:httpd_t:s0 apache 1284 0.1 0.3 1478796 2580 ?  
Sl 14:50 0:01 /usr/sbin/httpd -DFOREGROUND  
system_u:system_r:httpd_t:s0 apache 1285 0.1 0.3 1347668 2756 ?  
Sl 14:50 0:01 /usr/sbin/httpd -DFOREGROUND  
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 guest 3451 0.0 0.1 12  
1196 pts/0 S+ 15:10 0:00 grep --color=auto httpd  
[guest@localhost ~]$ ps -eZ | grep httpd  
system_u:system_r:httpd_t:s0 1205 ? 00:00:00 httpd  
system_u:system_r:httpd_t:s0 1281 ? 00:00:00 httpd  
system_u:system_r:httpd_t:s0 1283 ? 00:00:01 httpd  
system_u:system_r:httpd_t:s0 1284 ? 00:00:01 httpd  
system_u:system_r:httpd_t:s0 1285 ? 00:00:01 httpd  
[guest@localhost ~]$
```

Рис.3 Веб-сервер Apache

4. Посмотрела текущее состояние переключателей SELinux для Apache с помощью команды `sestatus -bigrep httpd`

```
guest@localhost:~  
File Edit View Search Terminal Help  
xdm_manage_bootloader      on  
xdm_sysadm_login           off  
xdm_write_home             off  
xen_use_nfs                off  
xend_run_blkmap            on  
xend_run_qemu              on  
xguest_connect_network     on  
xguest_exec_content        on  
xguest_mount_media         on  
xguest_use_bluetooth       on  
xserver_clients_write_xshm off  
xserver_execmem            off  
xserver_object_manager     off  
zabbix_can_network         off  
zabbix_run_sudo            off  
zarafe_setrlimit           off  
zebra_write_config         off  
zoneminder_anon_write      off  
zoneminder_run_sudo        off  
  
[guest@localhost ~]$  
  
[guest@localhost ~]$  
[guest@localhost ~]$
```

Рис.4 Текущее состояние переключателей SELinux

- Посмотрела статистику по политике с помощью команды `seinfo`, также определила множество пользователей, ролей, типов.

```

guest@localhost:~
File Edit View Search Terminal Help
Policy Version:          31 (MLS enabled)
Target Policy:           selinux
Handle unknown classes:  allow
Classes:                 132
Permissions:             464
Categories:              1024
Attributes:              255
Roles:                   14
Cond. Expr.:             386
Neverallow:              0
Dontaudit:               10358
Type_change:             87
Range_trans:             5781
Role_trans:              421
Validatetrans:           0
MLS Val. Tran:           0
Polcap:                  5
Typebounds:              0
Neverallowxperm:         0
Dontauditxperm:          0
Ibpkeycon:               0
Fs_use:                  34
Portcon:                 642
Nodecon:                 0
[guest@localhost ~]$

```

Рис.5 Статистика

6. Определила тип файлов и поддиректорий, находящихся в директории /var/www, с помощью команды `ls -lZ /var/www`

```

[guest@localhost ~]$ ls -lZ /var/www
total 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 Nov 12 07
:58 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0      6 Nov 12 07
:58 html
[guest@localhost ~]$

```

Рис.6 Тип файлов и поддиректорий

7. Определила тип файлов, находящихся в директории /var/www/html: `ls -lZ /var/www/html`

```

[guest@localhost ~]$ ls -lZ /var/www/html
total 0

```

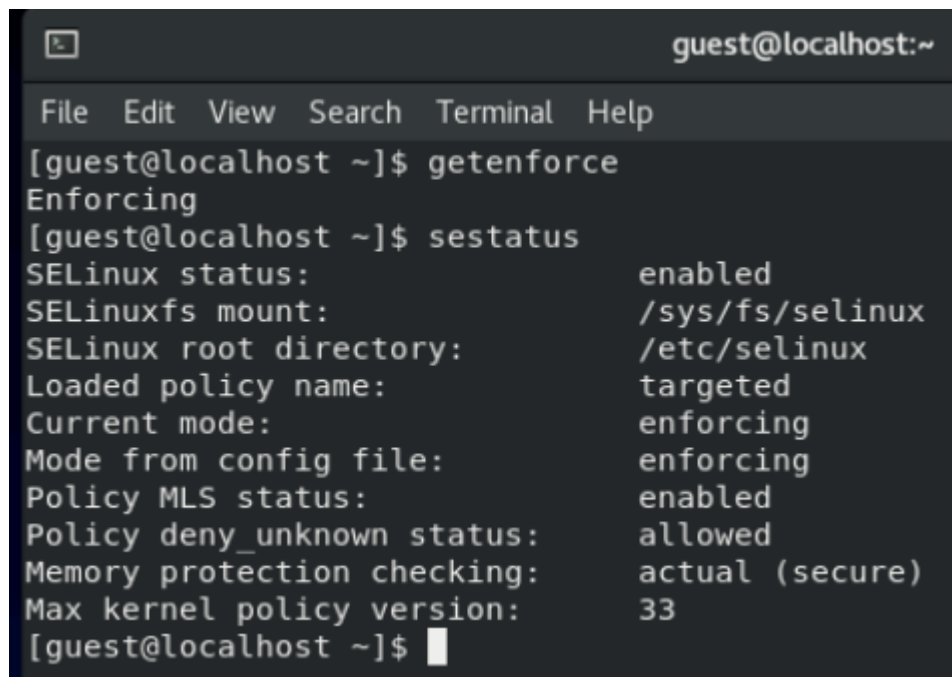
Рис.7 Тип файлов

8. Определила круг пользователей, которым разрешено создание файлов в директории /var/www/html.

```
[guest@localhost html]$ ls -l /var/www/html  
total 0
```

Рис.8 Круг пользователей

9. Создала от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись в директорию) html-файл /var/www/html/test.html следующего содержания: test



The screenshot shows a terminal window titled "guest@localhost:~". The terminal has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The user has entered the command "getenforce", which returns "Enforcing". Then, the user enters "sestatus", which displays the following SELinux status information:

```
[guest@localhost ~]$ getenforce  
Enforcing  
[guest@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  
[guest@localhost ~]$
```

Рис.9 html-файл

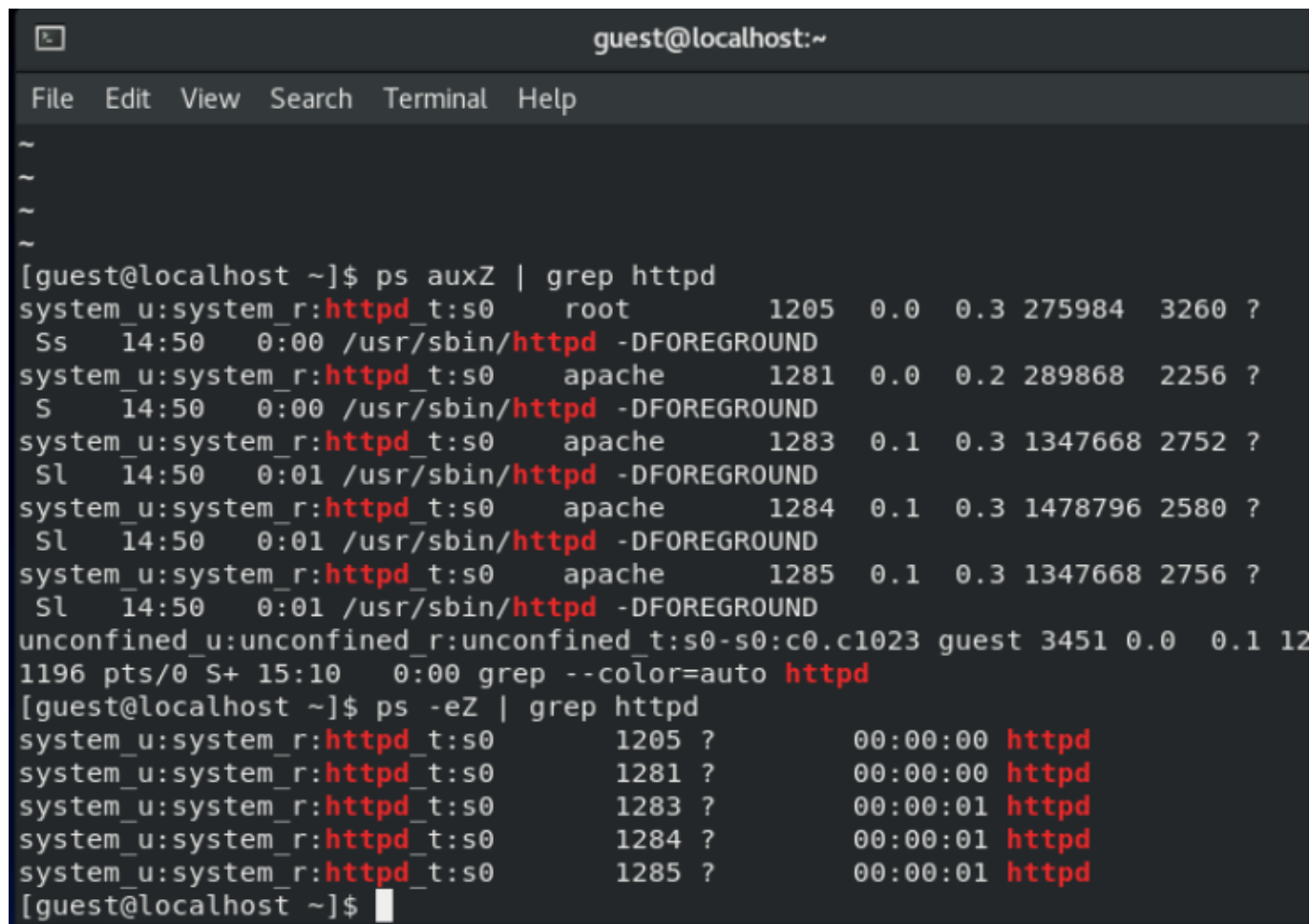
10. Проверила контекст созданного файла. Занесла в отчёт контекст, присваиваемый по умолчанию вновь созданным файлам в директории /var/www/html

```
guest@localhost:~  
File Edit View Search Terminal Help  
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  
[guest@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; vendor preset: enabled)  
   Active: active (running) since Fri 2021-11-26 14:00:00 MSK; 1min 12s ago  
     Docs: man:httpd.service(8)  
  Main PID: 1205 (httpd)  
    Status: "Running, listening on: port 80"  
   Tasks: 213 (limit: 4808)  
  Memory: 10.7M  
    CGroup: /system.slice/httpd.service  
            └─1205 /usr/sbin/httpd -DFOREGROUND  
              └─1281 /usr/sbin/httpd -DFOREGROUND  
                └─1283 /usr/sbin/httpd -DFOREGROUND  
                  └─1284 /usr/sbin/httpd -DFOREGROUND  
                    └─1285 /usr/sbin/httpd -DFOREGROUND  
lines 1-14/14 (END)
```

Рис.10 Контекст созданного файла

11. Обратилась к файлу через веб-сервер, введя в браузере адрес <http://127.0.0.1/test.html>.  
Убедилась, что файл был успешно отображён





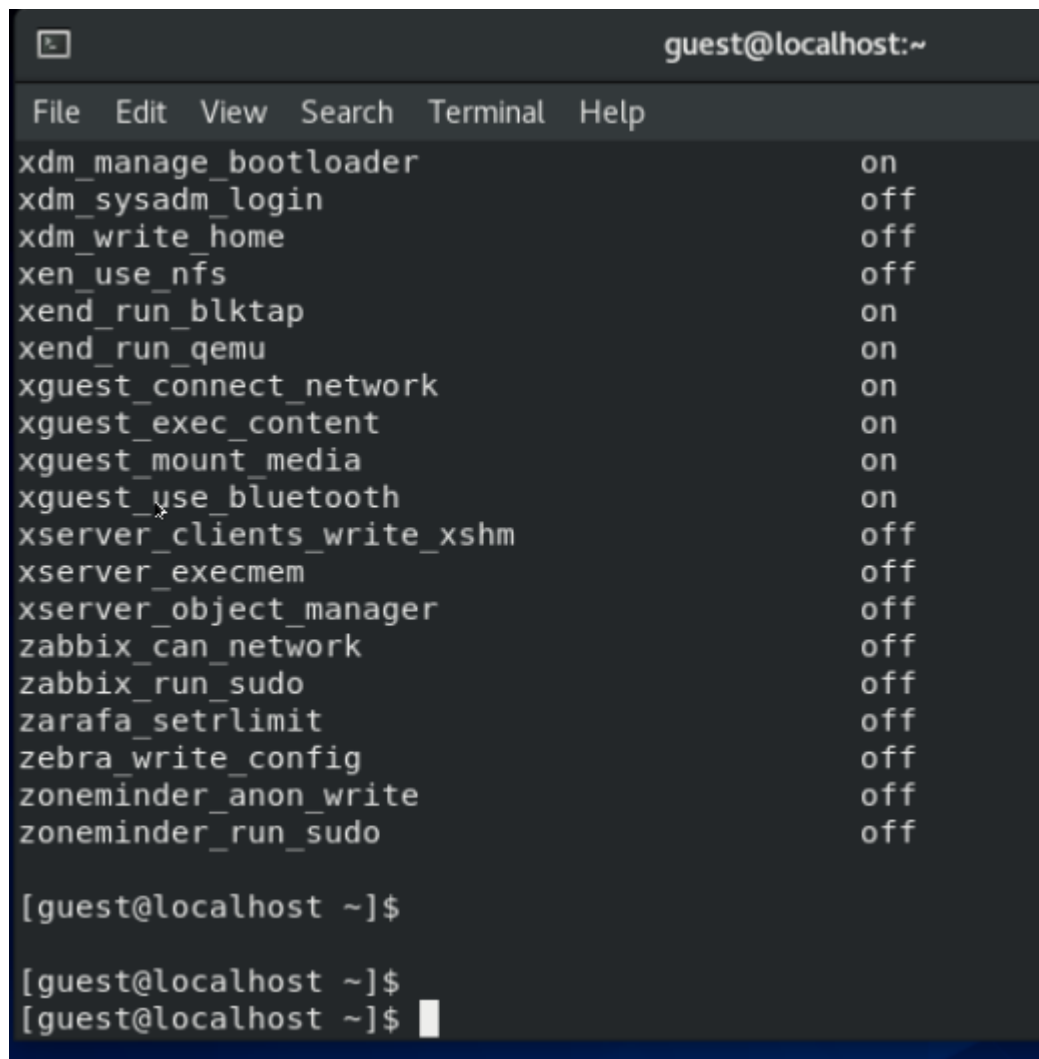
```

guest@localhost:~
File Edit View Search Terminal Help
~
~
~
~
[guest@localhost ~]$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0      root          1205  0.0  0.3 275984  3260 ?
Ss   14:50   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0      apache        1281  0.0  0.2 289868  2256 ?
S    14:50   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0      apache        1283  0.1  0.3 1347668  2752 ?
Sl   14:50   0:01 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0      apache        1284  0.1  0.3 1478796  2580 ?
Sl   14:50   0:01 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0      apache        1285  0.1  0.3 1347668  2756 ?
Sl   14:50   0:01 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 guest 3451 0.0  0.1 12
1196 pts/0 S+ 15:10   0:00 grep --color=auto httpd
[guest@localhost ~]$ ps -eZ | grep httpd
system_u:system_r:httpd_t:s0      1205 ?          00:00:00 httpd
system_u:system_r:httpd_t:s0      1281 ?          00:00:00 httpd
system_u:system_r:httpd_t:s0      1283 ?          00:00:01 httpd
system_u:system_r:httpd_t:s0      1284 ?          00:00:01 httpd
system_u:system_r:httpd_t:s0      1285 ?          00:00:01 httpd
[guest@localhost ~]$

```

Рис.11 Файл в браузере

12. Изучила справку `man httpd_selinux` и выяснила, какие контексты файлов определены для `httpd`. Сопоставила их с типом файла `test.html`. Проверила контекст файла командой `ls -Z. ls -Z /var/www/html/test.html`



The screenshot shows a terminal window with the title bar "guest@localhost:~". The terminal has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The main content is a list of SELinux contexts and their status, displayed in a table-like format. The contexts are listed on the left, and their status (on or off) is on the right. The list includes: xdm\_manage\_bootloader (on), xdm\_sysadm\_login (off), xdm\_write\_home (off), xen\_use\_nfs (off), xend\_run\_blkmap (on), xend\_run\_qemu (on), xguest\_connect\_network (on), xguest\_exec\_content (on), xguest\_mount\_media (on), xguest\_use\_bluetooth (on), xserver\_clients\_write\_xshm (off), xserver\_execmem (off), xserver\_object\_manager (off), zabbix\_can\_network (off), zabbix\_run\_sudo (off), zarafa\_setrlimit (off), zebra\_write\_config (off), zoneminder\_anon\_write (off), and zoneminder\_run\_sudo (off). Below the list, there are three lines of terminal output: "[guest@localhost ~]\$", "[guest@localhost ~]\$", and "[guest@localhost ~]\$".

xdm_manage_bootloader	on
xdm_sysadm_login	off
xdm_write_home	off
xen_use_nfs	off
xend_run_blkmap	on
xend_run_qemu	on
xguest_connect_network	on
xguest_exec_content	on
xguest_mount_media	on
xguest_use_bluetooth	on
xserver_clients_write_xshm	off
xserver_execmem	off
xserver_object_manager	off
zabbix_can_network	off
zabbix_run_sudo	off
zarafa_setrlimit	off
zebra_write_config	off
zoneminder_anon_write	off
zoneminder_run_sudo	off

```
[guest@localhost ~]$  
[guest@localhost ~]$  
[guest@localhost ~]$
```

Рис.12 Контексты

13. Измените контекст файла /var/www/html/test.html с httpd\_sys\_content\_t на любой другой, к которому процесс httpd не должен иметь доступа, например, на samba\_share\_t: `chcon -t samba_share_t /var/www/html/test.html ls -Z /var/www/html/test.html` Проверила, что контекст поменялся.

```

guest@localhost:~
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: 4961 Attributes: 255
Users: 8 Roles: 14
Booleans: 338 Cond. Expr.: 386
Allow: 112594 Neverallow: 0
Auditallow: 166 Dontaudit: 10358
Type_trans: 252747 Type_change: 87
Type_member: 35 Range_trans: 5781
Role_allow: 38 Role_trans: 421
Constraints: 72 Validatetrans: 0
MLS Constrains: 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: 642
Netifcon: 0 Nodecon: 0
[guest@localhost ~]$

```

Рис.13 Измененный контекст

14. Попробовала ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Получила сообщение об ошибке: `Forbidden You don't have permission to access /test.html on this server.`

```

[guest@localhost ~]$ ls -lZ /var/www
total 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 Nov 12 07
:58 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 6 Nov 12 07
:58 html
[guest@localhost ~]$

```

Рис.14 Ошибка

15. Проанализировала ситуацию. Почему файл не был отображён, если права доступа позволяют читать этот файл любому пользователю? `ls -l /var/www/html/test.html` Просмотрела log-файлы веб-сервера Apache. Также просмотрела системный лог-файл: `tail /var/log/messages` Если в системе окажутся запущенными процессы `setroubleshootd` и `audtd`, то также смогла увидеть ошибки, аналогичные указанным выше, в файле `/var/log/audit/audit.log`. Проверила это утверждение самостоятельно.

```
[guest@localhost ~]$ ls -lZ /var/www/html
total 0
```

Рис.15 Ошибки

16. Попробовала запустить веб-сервер Apache на прослушивание TCP-порта 81 (а не 80, как рекомендует IANA и прописано в /etc/services. Для этого в файле /etc/httpd/httpd.conf нашла строчку Listen 80 и заменила её на Listen 81.

```
[guest@localhost html]$ ls -l /var/www/html
total 0
```

Рис.16 TCP-порт 81

17. Выполнила перезапуск веб-сервера Apache. Произошёл сбой.

Рис.17 Сбой

18. Проанализировала лог-файлы: tail -nl /var/log/messages Просмотрела файлы /var/log/http/error\_log, /var/log/http/access\_log и /var/log/audit/audit.log и выяснила, в каких файлах появились записи.

Рис.18 Лог-файлы

19. Выполнила команду semanage port -a -t http\_port\_t -p tcp 81 После этого проверила список портов командой semanage port -l | grep http\_port\_t Убедилась, что порт 81 появился в списке.

Рис.19 Порт 81

20. Попробовала запустить веб-сервер Apache ещё раз.

Рис.20 Веб-сервер Apache

21. Вернула контекст httpd\_sys\_content\_t к файлу /var/www/html/ test.html: chcon -t httpd\_sys\_content\_t /var/www/html/test.html После этого попробовала получить доступ к файлу через веб-сервер, введя в браузере адрес http://127.0.0.1:81/test.html. Увидела содержимое файла — слово «test»

Рис.21 Слово text

22. Исправила обратно конфигурационный файл apache, вернув Listen 80

### Рис.22 Возвращение

23. Удалила привязку http\_port\_t к 81 порту: semanage port -d -t http\_port\_t -p tcp 81 и проверила, что порт 81 удалён.

### Рис.23 Удален порт

24. Удалила файл /var/www/html/test.html: rm /var/www/html/test.html

### Рис.24 Удален файл

## Выводы

---

Развила навыки администрирования ОС Linux. Получила первое практическое знакомство с технологией SELinux1. Проверила работу SELinx на практике совместно с веб-сервером Apache.