

Отчёт по лабораторной работе 6

Мандатное разграничение прав в Linux

Румянцева Александра Сергеевна

Содержание

Цель работы	5
Задание	6
Теория	7
SELinux	7
Режимы работы SELinux	7
Контроль доступа в SELinux	8
Выполнение лабораторной работы	10
Подготовка к выполнению	10
Выполнение основной части лабораторной работы	11
Библиография	21
Выводы	22

Список иллюстраций

0.1	рис.1. Установка веб-сервер Apache. Добавле разрешающих правил.	10
0.2	рис.2. Файл /etc/httpd/httpd.conf	11
0.3	рис.3. Команды getenforce и sestatus.	12
0.4	рис.4. service httpd status	12
0.5	рис.5. Команда ps auxZ grep httpd.	12
0.6	рис.6. Команда sestatus -b grep httpd.	13
0.7	рис.7. Команда seinfo.	13
0.8	рис.8. Программа readfile.c	14
0.9	рис.9. Определение пользователей, которым разрешено создание файлов в директории /var/www/html.	14
0.10	рис.10. html-файл /var/www/html/test.html.	15
0.11	рис.11. Контекст созданного файла.	15
0.12	рис.12. http://127.0.0.1/test.html.	15
0.13	рис.13. Работа с контекстом файла /var/www/html/test.html.	16
0.14	рис.14. http://127.0.0.1/test.html при контексте samba_share_t.	16
0.15	рис.15. Просмотр log-файлы веб-сервера Apache.	17
0.16	рис.16. Просмотр log-файлы веб-сервера Apache.	17
0.17	рис.17. Перезапуск веб-сервера Apache.	18
0.18	рис.18. http://127.0.0.1/test.html при Listen 81.	18
0.19	рис.19. Порт 81.	19
0.20	рис.20. Restart и возврат контекста.	19
0.21	рис.21. http://127.0.0.1:81/test.html.	20
0.22	рис.22. Удаление привязки http_port_t к 81 порту.	20
0.23	рис.23. Удаление файла /var/www/html/test.html.	20

Список таблиц

Цель работы

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

Задание

Лабораторная работа подразумевает изучение технологий SELinux и веб-сервера Apache опытным путём.

Теория

SELinux

SELinux (SELinux) — это система принудительного контроля доступа, реализованная на уровне ядра. Впервые эта система появилась в четвертой версии CentOS, а в 5 и 6 версии реализация была существенно дополнена и улучшена. Эти улучшения позволили SELinux стать универсальной системой, способной эффективно решать массу актуальных задач. Стоит помнить, что классическая система прав Unix применяется первой, и управление перейдет к SELinux только в том случае, если эта первичная проверка будет успешно пройдена.

Режимы работы SELinux

SELinux имеет три основных режим работы, при этом по умолчанию установлен режим Enforcing. Это довольно жесткий режим, и в случае необходимости он может быть изменен на более удобный для конечного пользователя.

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.

Контроль доступа в SELinux

SELinux предоставляет следующие модели управления доступом:

Type Enforcement (TE): основной механизм контроля доступа, используемый в целевых политиках. Позволяет детально, на самом низком уровне управлять разрешениями. Самый гибкий, но и самый трудоемкий для системного администратора механизм.

Role-Based Access Control (RBAC): в этой модели права доступа реализуются в качестве ролей. Ролью называется разрешения на выполнение определенных действий одним или несколькими элементами системы над другими частями системы. По-сути, RBAC является дальнейшим развитием TE.

Multi-Level Security (MLS): многоуровневая модель безопасности, в которой всем объектам системы присваивается определенный уровень доступа. Разрешение или запрет доступа определяется только соотношением этих уровней.

Все процессы и файлы в рамках SELinux имеют контекст безопасности. Давайте посмотрим на контекст на практике, подробно рассмотрев стартовую страницу веб-сервера Apache, находящуюся по адресу `/var/www/html/index.html`

```
$ ls -Z /var/www/html/index.html
```

```
-rw-r--r-- username username system_u:object_r:httpd_sys_content_t /var/www/html/index.html
```

В дополнение к стандартным правам доступа к файлу, мы можем видеть контекст безопасности SELinux: `system_u: object_r: httpd_sys_content_t`.

Контекст базируется на `user:role:type:mls`, но поля `user:role:type` отображаются, в то время как поле `mls` скрыто. Также мы можем видеть целевую политику, в данном случае `httpd_sys_content_t`.

Теперь рассмотрим контекст безопасности SELinux для процесса 'httpd' (веб-сервер Apache):

```
$ ps axZ | grep httpd
```

```
system_u:system_r:httpd_t 3234 ? Ss 0:00 /usr/sbin/httpd
```

Как мы видим, этот процесс запущен на домене `httpd_t`.

Ну а теперь давайте посмотрим на контекст безопасности файла в нашем

домашнем каталоге:

```
$ ls -Z /home/username/myfile.txt
```

```
-rw-r--r-- username username user_u:object_r:user_home_t /home/username/myfile.txt
```

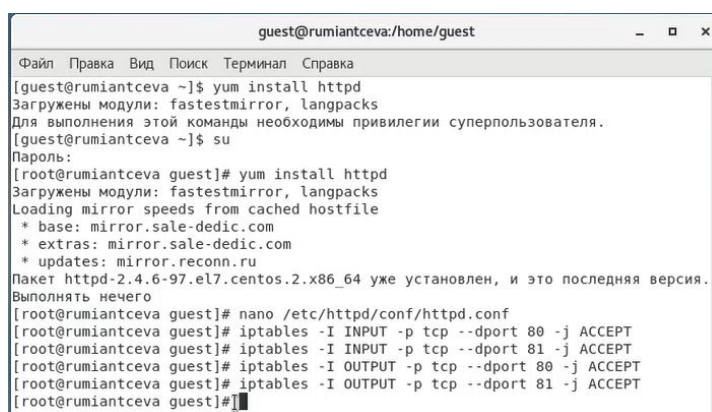
Мы видим, что файл имеет тип `user_home_t`, этот тип присваивается по умолчанию всем файлам в домашнем каталоге.

Доступ разрешен только между элементами с одинаковым типом, именно поэтому веб-сервер Apache может без проблем читать файл `/var/www/html/index.html`, который имеет тип `httpd_sys_content_t`. В то же самое время, так как Apache запущен на домене `httpd_t` и не имеет заполненных полей `userid:username`, он не может получить доступ к файлу `home/username/myfile.txt`, хотя этот файл доступен для чтения процессам, для которых не определена целевая политика. Таким образом, если веб-сервер Apache будет взломан, то злоумышленник не сможет получить доступ к файлам или запускать процессы, которые не находятся в домене `httpd_t`.

Выполнение лабораторной работы

Подготовка к выполнению

1. Установила от имени суперпользователя веб-сервер Apache с помощью команды `yum install httpd`. В моём случае оказалось, что он уже установлен (рис. 1)



```
guest@rumiantceva:/home/guest
Файл  Правка  Вид  Поиск  Терминал  Справка
[guest@rumiantceva ~]$ yum install httpd
Загружены модули: fastestmirror, langpacks
Для выполнения этой команды необходимы привилегии суперпользователя.
[guest@rumiantceva ~]$ su
Пароль:
[root@rumiantceva guest]# yum install httpd
Загружены модули: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirror.sale-dedic.com
 * extras: mirror.sale-dedic.com
 * updates: mirror.reconn.ru
Пакет httpd-2.4.6-97.el7.centos.2.x86_64 уже установлен, и это последняя версия.
Выполнять нечего
[root@rumiantceva guest]# nano /etc/httpd/conf/httpd.conf
[root@rumiantceva guest]# iptables -I INPUT -p tcp --dport 80 -j ACCEPT
[root@rumiantceva guest]# iptables -I INPUT -p tcp --dport 81 -j ACCEPT
[root@rumiantceva guest]# iptables -I OUTPUT -p tcp --dport 80 -j ACCEPT
[root@rumiantceva guest]# iptables -I OUTPUT -p tcp --dport 81 -j ACCEPT
[root@rumiantceva guest]#
```

Рис. 0.1: рис.1. Установка веб-сервер Apache. Добавле разрешающих правил.

2. В конфигурационном файле `/etc/httpd/httpd.conf` задала параметр `ServerName: ServerName test.ru` чтобы при запуске веб-сервера не выдавались лишние сообщения об ошибках, не относящихся к лабораторной работе (рис. 2).

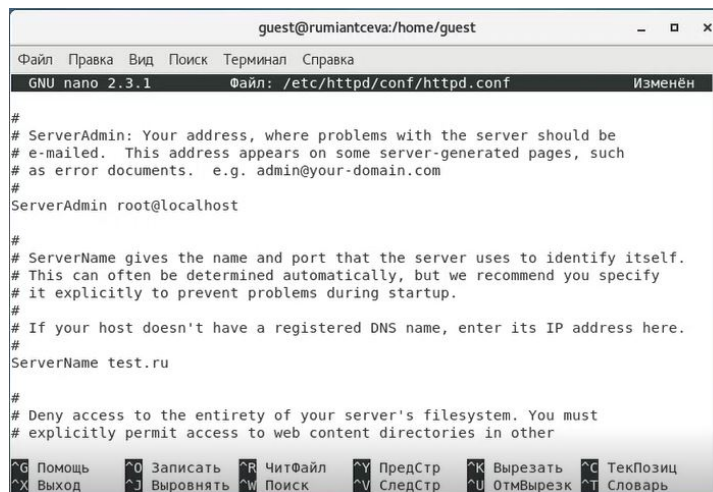


Рис. 0.2: рис.2. Файл /etc/httpd/httpd.conf

3. Отключила пакетный фильтр, точнее сделала так, что он в своей рабочей конфигурации позволял подключаться к 80-му и 81-му портам протокола tcp, добавив разрешающие правила с помощью команд (рис. 1):

```

iptables -I INPUT -p tcp --dport 80 -j ACCEPT
iptables -I INPUT -p tcp --dport 81 -j ACCEPT
iptables -I OUTPUT -p tcp --sport 80 -j ACCEPT
iptables -I OUTPUT -p tcp --sport 81 -j ACCEPT

```

Можно было бы также отключить фильтр командами:

```

iptables -F
iptables -P INPUT ACCEPT iptables -P OUTPUT ACCEPT

```

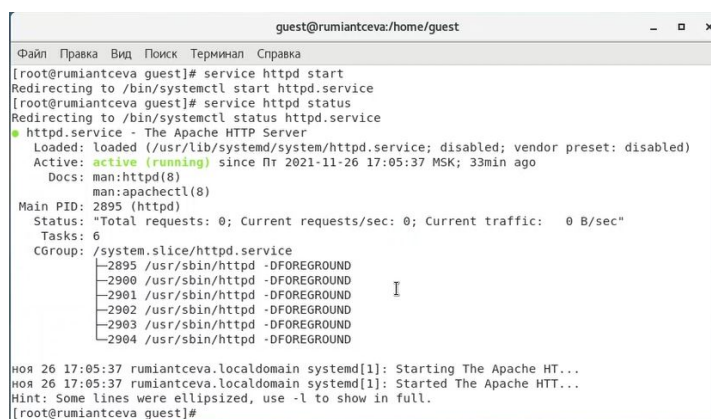
Выполнение основной части лабораторной работы

1. Вошла в систему с полученными учётными данными и убедилась, что SELinux работает в режиме enforcing политики targeted с помощью команд getenforce и sestatus (рис.3).

```
[root@rumiantceva guest]# getenforce
Enforcing
[root@rumiantceva guest]# 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
Max kernel policy version:      31
[root@rumiantceva guest]#
```

Рис. 0.3: рис.3. Команды getenforce и sestatus.

2. Обратилась к веб-серверу, запущенному на нашем компьютере, и убедимся, что последний работает: `service httpd status` (рис. 4).



```
guest@rumiantceva:/home/guest
Файл Правка Вид Поиск Терминал Справка
[root@rumiantceva guest]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@rumiantceva 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; disabled; vendor preset: disabled)
   Active: active (running) since Fri 2021-11-26 17:05:37 MSK; 33min ago
     Docs: man:httpd(8)
           man:apachectl(8)
  Main PID: 2895 (httpd)
    Status: "Total requests: 0; Current requests/sec: 0; Current traffic: 0 B/sec"
     Tasks: 6
    CGroup: /system.slice/httpd.service
            └─2895 /usr/sbin/httpd -DFOREGROUND
              └─2900 /usr/sbin/httpd -DFOREGROUND
                └─2901 /usr/sbin/httpd -DFOREGROUND
                  └─2902 /usr/sbin/httpd -DFOREGROUND
                    └─2903 /usr/sbin/httpd -DFOREGROUND
                      └─2904 /usr/sbin/httpd -DFOREGROUND

ноя 26 17:05:37 rumiantceva.localdomain systemd[1]: Starting The Apache HT...
ноя 26 17:05:37 rumiantceva.localdomain systemd[1]: Started The Apache HT...
Hint: Some lines were ellipsized, use -l to show in full.
[root@rumiantceva guest]#
```

Рис. 0.4: рис.4. `service httpd status`

3. Нашла веб-сервер Apache в списке процессов, определила его контекст безопасности, используя команду `ps auxZ | grep httpd` (рис. 5).

```
[root@rumiantceva guest]# ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root 2895 0.0 0.4 224884 5012 ? Ss 17:05 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 2900 0.0 0.3 226168 3088 ? S 17:05 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 2901 0.0 0.3 226168 3088 ? S 17:05 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 2902 0.0 0.3 226168 3088 ? S 17:05 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 2903 0.0 0.3 226168 3088 ? S 17:05 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 2904 0.0 0.3 226168 3088 ? S 17:05 0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-c0:c1023 root 3617 0.0 0.0 112832 976 pts/0 R+ 17:42 0:00 grep --color=auto httpd
[root@rumiantceva guest]#
```

Рис. 0.5: рис.5. Команда `ps auxZ | grep httpd`.

В моём случае процесс запущен на домене `httpd_t`.

4. Посмотрела текущее состояние переключателей SELinux для Apache с помощью команды `sestatus -b | grep httpd` (рис. 6).

```
guest@rumiantceva:/home/guest
Файл Правка Вид Поиск Терминал Справка
[root@rumiantceva guest]# 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_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 on
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
```

Рис. 0.6: рис.6. Команда sestatus -b | grep httpd.

Наглядно видно, что многие из переключателей находятся в положении «off».

5. Посмотрела статистику по политике с помощью команды seinfo, также определила множество пользователей, ролей и типов. (рис. 7)

Замечу, что для выполнения команды пришлось выполнить установку setools-console

```
guest@rumiantceva:/home/guest
Файл Правка Вид Поиск Терминал Справка
Проверка : setools-console-3.3.8-4.el7.x86_64

Установлено:
setools-console.x86_64 0:3.3.8-4.el7

Выполнено!
[root@rumiantceva guest]# seinfo
Statistics for policy file: /sys/fs/selinux/policy
Policy Version & Type: v.31 (binary, mls)

Classes: 130 Permissions: 272
Sensitivities: 1 Categories: 1024
Types: 4793 Attributes: 253
Users: 8 Roles: 14
Booleans: 316 Cond. Expr.: 362
Allow: 107834 Neverallow: 0
Auditallow: 158 Dontaudit: 10022
Type_trans: 10153 Type_change: 74
Type_member: 35 Role_allow: 37
Role_trans: 414 Range_trans: 5899
Constraints: 143 Validatetrans: 0
Initial SIDs: 27 Fs_use: 32
Genfscon: 103 Portcon: 614
Netifcon: 0 Nodecon: 0
Permissives: 0 Polcap: 5

[root@rumiantceva guest]#
```

Рис. 0.7: рис.7. Команда seinfo.

Из рисунка наглядно видно, что пользователей: 8, ролей: 14, типов: 4793.

6. Определила тип файлов и поддиректорий, находящихся в директории /var/www с помощью команды `ls -lZ /var/www` (рис. 8).
7. Определила тип файлов, находящихся в директории /var/www/html с помощью команды `ls -lZ /var/www/html` (рис. 8). Директория оказалась пустой

```
[root@rumiantceva guest]# ls -lZ /var/www
drwxr-xr-x. root root system_u:object_r:httpd_sys_script_exec_t:s0 cgi-bin
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 html
[root@rumiantceva guest]# ls -lZ /var/www/html
[root@rumiantceva guest]#
```

Рис. 0.8: рис.8. Программа readfile.c

8. Определила круг пользователей, которым разрешено создание файлов в директории /var/www/html (рис. 9).

Я выполнила команду `touch`, команду создания файла, для каждого пользователя. Таким образом я опытным путём определила, что только суперпользователь может создать файл в данной директории.

Файл	Правка	Вид	Поиск	Терминал	Справка
Permissions:		0	Polcap:		5

```
[root@rumiantceva guest]# ls -lZ /var/www
drwxr-xr-x. root root system_u:object_r:httpd_sys_script_exec_t:s0 cgi-bin
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 html
[root@rumiantceva guest]# ls -lZ /var/www/html
[root@rumiantceva guest]# exit
exit
[guest@rumiantceva ~]$ touch /var/www/html/file.txt
touch: невозможно выполнить touch для «/var/www/html/file.txt»: Отказано в доступе
[guest@rumiantceva ~]$ su guest2
Пароль:
[guest2@rumiantceva guest]$ touch /var/www/html/file.txt
touch: невозможно выполнить touch для «/var/www/html/file.txt»: Отказано в доступе
[guest2@rumiantceva guest]$ exit
exit
[guest@rumiantceva ~]$ su asrumiantceva
su: user asrumiantceva does not exist
[guest@rumiantceva ~]$ su rumiantceva
Пароль:
[rumiantceva@rumiantceva guest]$ touch /var/www/html/file.txt
touch: невозможно выполнить touch для «/var/www/html/file.txt»: Отказано в доступе
[rumiantceva@rumiantceva guest]$ exit
exit
[guest@rumiantceva ~]$ su
Пароль:
[root@rumiantceva guest]# touch /var/www/html/file.txt
[root@rumiantceva guest]#
```

Рис. 0.9: рис.9. Определение пользователей, которым разрешено создание файлов в директории /var/www/html.

9. Создала от имени суперпользователя html-файл /var/www/html/test.html следующего содержания (рис. 10):

test



Рис. 0.10: рис.10. html-файл /var/www/html/test.html.

10. Проверила контекст созданного файла (рис. 11).

Контекст, присваиваемый по умолчанию вновь созданным файлам в директории /var/www/html: unconfined_u:object_r:httpd_sys_content_t

```
[root@rumiantceva guest]# nano /var/www/html/test.html
[root@rumiantceva guest]# ls -lZ /var/www/html
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 file.txt
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 test.html
[root@rumiantceva guest]# ls -l /var/www/html
итого 4
-rw-r--r--. 1 root root 0 ноя 26 18:01 file.txt
-rw-r--r--. 1 root root 33 ноя 26 18:03 test.html
[root@rumiantceva guest]#
```

Рис. 0.11: рис.11. Контекст созданного файла.

11. Обратилась к файлу через веб-сервер, введя в браузере firefox адрес: http://127.0.0.1/test.html. Убедилась, что файл был успешно отображен (рис. 12).

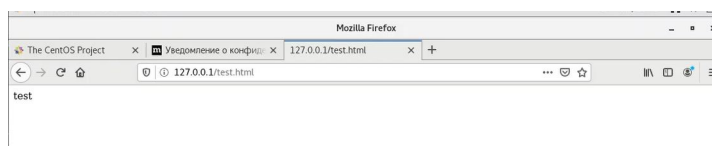


Рис. 0.12: рис.12. http://127.0.0.1/test.html.

12. Изучила справку `man httpd_selinux` и выяснила, какие контексты файлов определены для `httpd` и сопоставим их с типом файла `test.html`. Проверила контекст файла командой `ls -Z /var/www/html/test.html` (рис. 13).

Т.к. по умолчанию пользователи CentOS являются свободными (`unconfined`) от типа, созданному нами файлу `test.html` был сопоставлен SELinux, пользователь `unconfined_u`. Это первая часть контекста.

Далее политика ролевого разделения доступа RBAC используется процессами, но не файлами, поэтому роли не имеют никакого значения для файлов. Роль `object_r` используется по умолчанию для файлов на «постоянных» носителях и на сетевых файловых системах.

Тип `httpd_sys_content_t` позволяет процессу `httpd` получить доступ к файлу. Благодаря наличию последнего типа мы получили доступ к файлу при обращении к нему через браузер.

13. Изменила контекст файла `/var/www/html/test.html` с `httpd_sys_content_t` на другой, к которому процесс `httpd` не должен иметь доступа, в нашем случае, на `samba_share_t` и проверила изменения (рис. 13).

```
[root@rumiantceva guest]# ls -Z /var/www/html/test.html
-rw-r--r-- root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test.html
[root@rumiantceva guest]# chcon -t samba_share_t /var/www/html/test.html
[root@rumiantceva guest]# ls -Z /var/www/html/test.html
-rw-r--r-- root root unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html
[root@rumiantceva guest]#
```

Рис. 0.13: рис.13. Работа с контекстом файла `/var/www/html/test.html`.

14. Попробовала еще раз получить доступ к файлу через веб-сервер, введя в браузере `firefox` адрес `http://127.0.0.1/test.html`. Но получила сообщение об ошибке (рис. 14).



Рис. 0.14: рис.14. `http://127.0.0.1/test.html` при контексте `samba_share_t`.

15. Проанализировала ситуацию, просмотрев log-файлы веб-сервера Apache, системный log-файл и audit.log при условии уже запущенных процессов setroubleshootd и audtd (рис. 15).

```
[root@rumiantceva guest]# ls -l /var/www/html/test.html
-rw-r--r-- 1 root root 33 ноя 26 18:03 /var/www/html/test.html
[root@rumiantceva guest]# tail /var/log/messages
Nov 26 18:10:01 rumiantceva systemd: Started Session 9 of user root.
Nov 26 18:10:01 rumiantceva systemd: Removed slice User Slice of root.
Nov 26 18:13:59 rumiantceva dbus[689]: [system] Activating service name='org.fedoraproject.Setroubleshootd' (using
servicehelper)
Nov 26 18:14:00 rumiantceva dbus[689]: [system] Successfully activated service 'org.fedoraproject.Setroubleshootd'
Nov 26 18:14:01 rumiantceva setroubleshoot: failed to retrieve rpm info for /var/www/html/test.html
Nov 26 18:14:01 rumiantceva setroubleshoot: SELinux is preventing /usr/sbin/httpd from getattr access on the file
/var/www/html/test.html. For complete SELinux messages run: sealert -l ea79d0b3-0157-4913-a30a-7ae9efefef84
Nov 26 18:14:01 rumiantceva python: SELinux is preventing /usr/sbin/httpd from getattr access on the file /var/www
/html/test.html.#012#012***** Plugin restorecon (92.2 confidence) suggests *****#012#012If y
ou want to fix the label, #012/var/www/html/test.html default label should be httpd_sys_content_t.#012Then you can
run restorecon. The access attempt may have been stopped due to insufficient permissions to access a parent direc
tory in which case try to change the following command accordingly.#012Doe#012# /sbin/restorecon -v /var/www/html/t
est.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 publ
ic_content_t.#012*****
...
[root@rumiantceva guest]# tail /var/log/audit/audit.log
type=USER_START msg=audit(1637939401.360:292): pid=4767 uid=0 auid=0 ses=9 subj=system_u:system_r:cron_d_t:s0-s0:c0
-c102# msg="op=PAM:session open grantors=pam_loginuid,pam_keyinit,pam_limits,pam_systemd acct="root" exe="/usr/sbl
n/crond" hostname=? addr=? terminal=cron res=success"
type=CRED_REFR msg=audit(1637939401.364:293): pid=4767 uid=0 auid=0 ses=9 subj=system_u:system_r:cron_d_t:s0-s0:c0
-c1023 msg="op=PAM:setcred grantors=pam_env,pam_fprintd acct="root" exe="/usr/sbin/crond" hostname=? addr=? termina
l=cron res=success"
type=CRED_DISP msg=audit(1637939401.426:294): pid=4767 uid=0 auid=0 ses=9 subj=system_u:system_r:cron_d_t:s0-s0:c0
-c1023 msg="op=PAM:setcred grantors=pam_env,pam_fprintd acct="root" exe="/usr/sbin/crond" hostname=? addr=? termina
l=cron res=success"
type=USER_END msg=audit(1637939401.439:295): pid=4767 uid=0 auid=0 ses=9 subj=system_u:system_r:cron_d_t:s0-s0:c0
-c1023 msg="op=PAM:session close grantors=pam_loginuid,pam_keyinit,pam_limits,pam_systemd acct="root" exe="/usr/sbl
n/crond" hostname=? addr=? terminal=cron res=success"
```

Рис. 0.15: рис.15. Просмотр log-файлы веб-сервера Apache.

Исходя из log-файлов, мы можем заметить, что проблема в измененном контексте на шаге 13, т.к. процесс httpd не имеет доступа на samba_share_t. В системе оказались запущены процессы setroubleshootd и audtd, поэтому ошибки, связанные с измененным контекстом, также есть в файле /var/log/audit/audit.log.

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

```
guest@rumiantceva:~/home/guest
Файл Правка Вид Поиск Терминал Справка
GNU nano 2.3.1 Файл: /etc/httpd/conf/httpd.conf Изменен

# 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 specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#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
```

Рис. 0.16: рис.16. Просмотр log-файлы веб-сервера Apache.

17. Перезапустила веб-сервер Apache и попробовала обратиться к файлу через веб-сервер, введя в браузере firefox адрес `http://127.0.0.1/test.html` (рис. 17, 18).

```
[root@rumiantceva guest]# service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[root@rumiantceva 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; disabled; vendor preset: disabled)
   Active: active (running) since Fri 2021-11-26 18:23:48 MSK; 14s ago
     Docs: man:httpd(8)
    Main PID: 5082 (httpd)
   Process: 5078 ExecStop=/bin/kill -WINCH $(MAINPID) (code=exited, status=0/SUCCESS)
   Status: "Total requests: 0; Current requests/sec: 0; Current traffic:  0 B/sec"
    Tasks: 6
   CGroup: /system.slice/httpd.service
           └─5082 /usr/sbin/httpd -DFOREGROUND
             └─5083 /usr/sbin/httpd -DFOREGROUND
               └─5084 /usr/sbin/httpd -DFOREGROUND
                 └─5085 /usr/sbin/httpd -DFOREGROUND
                   └─5087 /usr/sbin/httpd -DFOREGROUND
                     └─5088 /usr/sbin/httpd -DFOREGROUND

ноя 26 18:23:48 rumiantceva.localdomain systemd[1]: Starting The Apache HTTP Server...
ноя 26 18:23:48 rumiantceva.localdomain systemd[1]: Started The Apache HTTP Server.
[root@rumiantceva guest]#
```

Рис. 0.17: рис.17. Перезапуск веб-сервера Apache.

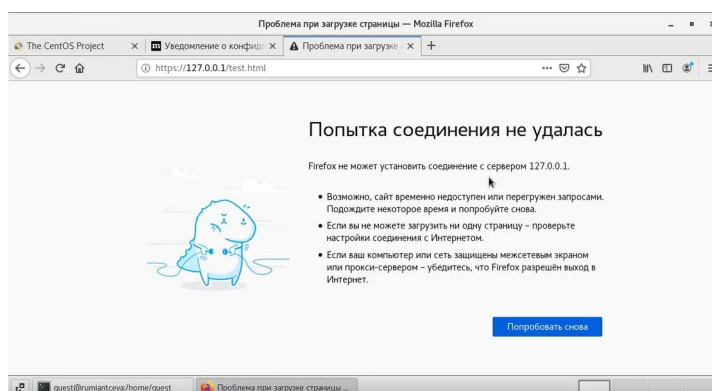


Рис. 0.18: рис.18. `http://127.0.0.1/test.html` при Listen 81.

Из того, что при запуске файла через браузер появилась ошибка, можно сделать предположение, что в списках портов, работающих с веб-сервером Apache, отсутствует порт 81.

18. Проанализировала log-файлы: `tail -n1 /var/log/messages` и просмотрела файлы `/var/log/http/error_log`, `/var/log/http/access_log` и `/var/log/audit/audit.log`. Во всех log-файлах появились записи, кроме `/var/log/messages`.

19. Выполнила команду `semanage port -a -t http_port_t -p tcp 81` и после этого проверила список портов командой `semanage port -l | grep http_port_t` (рис. 19).

```
n/cronq-: nostname=/ addr=/ terminal=cron res=success`
type=SERVICE_STOP msg=audit(1637940228.193:305): pid=1 uid=0 auid=4294967295 ses=4294967295
:init_t:s0 msg='unit=httpd comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=?
type=SERVICE_START msg=audit(1637940228.309:306): pid=1 uid=0 auid=4294967295 ses=4294967295
r:init_t:s0 msg='unit=httpd comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addi
[root@rumiantceva guest]# semanage port -a -t http_port_t -p tcp 81
ValueError: Порт tcp/81 уже определён
[root@rumiantceva 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@rumiantceva guest]#
```

Рис. 0.19: рис.19. Порт 81.

Заметим, что порт 81 уже определён, и он действительно есть в списке портов, так как данный порт определён на уровне политики..

20. Попробовала запустить веб-сервер Apache еще раз. Он успешно запустился.
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` (рис. 20).

```
[root@rumiantceva guest]# service httpd restart
Redirecting to /bin/systemctl restart httpd.service
^[[A[root@rumiantceva 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; disabled; vendor preset: disabled)
   Active: active (running) since Пн 2021-11-26 18:30:04 MSK; 16s ago
     Docs: man:httpd(8)
           man:apachectl(8)
  Process: 5271 ExecStop=/bin/kill -WINCH ${MAINPID} (code=exited, status=0/SUCCESS)
 Main PID: 5278 (httpd)
    Status: "Total requests: 0; Current requests/sec: 0; Current traffic:  0 B/sec"
       Tasks: 6
   CGroup: /system.slice/httpd.service
           └─5278 /usr/sbin/httpd -DFOREGROUND
             └─5279 /usr/sbin/httpd -DFOREGROUND
               └─5280 /usr/sbin/httpd -DFOREGROUND
                 └─5281 /usr/sbin/httpd -DFOREGROUND
                   └─5282 /usr/sbin/httpd -DFOREGROUND
                     └─5284 /usr/sbin/httpd -DFOREGROUND

ноя 26 18:30:04 rumiantceva.localdomain systemd[1]: Stopped The Apache HTTP Server.
ноя 26 18:30:04 rumiantceva.localdomain systemd[1]: Starting The Apache HTTP Server...
ноя 26 18:30:04 rumiantceva.localdomain systemd[1]: Started The Apache HTTP Server.
[root@rumiantceva guest]# chcon -t httpd_sys_content_t /var/www/html/test.html
[root@rumiantceva guest]#
```

Рис. 0.20: рис.20. Restart и возврат контекста.

После вновь попробовала получить доступ к файлу через веб-сервер, введя в браузере firefox адрес `http://127.0.0.1:81/test.html` (рис. 21). Увидели слово содержимое файла - слово «test».

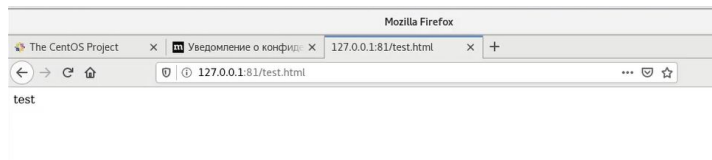


Рис. 0.21: рис.21. <http://127.0.0.1:81/test.html>.

22. Исправила обратно конфигурационный файл apache, вернув Listen 80.
23. Попробовала удалить привязку `http_port_t` к 81 порту: `semanage port -d -t http_port_t -p tcp 81`. Данную команду выполнить невозможно на моей версии CentOS, так как порт 81 определён на уровне политики (рис. 22).

```
[root@rumiantceva guest]# nano /etc/httpd/conf/httpd.conf
[root@rumiantceva guest]# semanage port -d -t http_port_t -p tcp 81
ValueError: Порт tcp/81 определен на уровне политики и не может быть удален
[root@rumiantceva 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
```

Рис. 0.22: рис.22. Удаление привязки `http_port_t` к 81 порту.

24. Удалила файл `/var/www/html/test.html`: `rm /var/www/html/test.html` (рис. 23)

```
[root@rumiantceva guest]# rm /var/www/html/test.html
rm: удалить обычный файл «/var/www/html/test.html»? y
[root@rumiantceva guest]#
```

Рис. 0.23: рис.23. Удаление файла `/var/www/html/test.html`.

Библиография

1. ТУИС РУДН
2. Статья “SELinux – описание и особенности работы с системой.” на сайте [habr.com](https://habr.com/ru/company/kingservers/blog/209644/) <https://habr.com/ru/company/kingservers/blog/209644/>

Выводы

Я развила навыки администрирования ОС Linux. Получила первое практическое знакомство с технологией SELinux. Проверила работу SELinux на практике совместно с веб-сервером Apache.