

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6**

*Дисциплина: Основы информационной безопасности*

*Название работы: Мандатное разграничение прав в Linux*

Студент: Невзоров Дмитрий

**МОСКВА**

2021 г.

## Цель работы

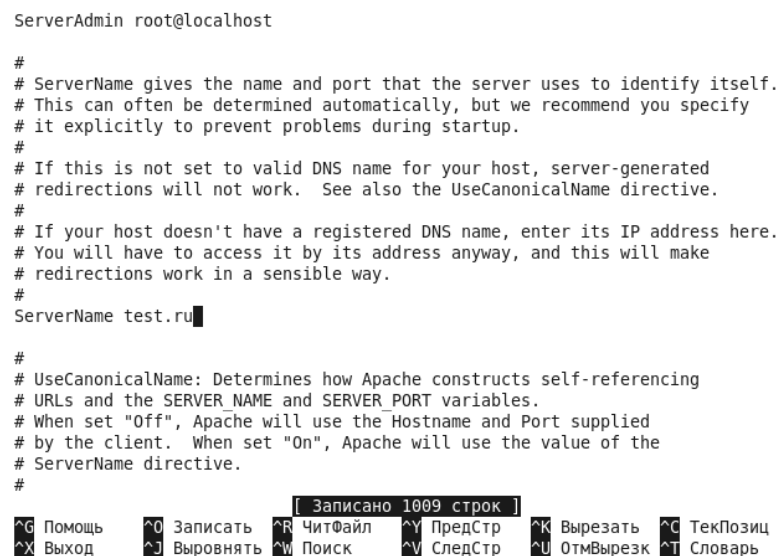
Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux. Проверить работу SELinux на практике совместно с веб-сервером Apache.

## Подготовка лабораторного стенда

1. Установим/обновим (за суперпользователя) веб-сервер Apache с помощью команды `yum install httpd`

```
Загружены модули: fastestmirror, refresh-packagekit, security
Подготовка к установке
Loading mirror speeds from cached hostfile
* base: mirror.corbina.net
* extras: mirror.corbina.net
* updates: mirror.awanti.com
base | 3.7 kB | 00:00
extras | 3.3 kB | 00:00
updates | 3.4 kB | 00:00
Пакет httpd-2.2.15-69.el6.centos.i686 уже установлен, и это последняя версия.
Выполнять нечего
```

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



```
ServerAdmin root@localhost

#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If this is not set to valid DNS name for your host, server-generated
# redirections will not work. See also the UseCanonicalName directive.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
# You will have to access it by its address anyway, and this will make
# redirections work in a sensible way.
#
ServerName test.ru

#
# UseCanonicalName: Determines how Apache constructs self-referencing
# URLs and the SERVER_NAME and SERVER_PORT variables.
# When set "Off", Apache will use the Hostname and Port supplied
# by the client. When set "On", Apache will use the value of the
# ServerName directive.
#
```

3. Также необходимо проследить, чтобы пакетный фильтр был отключен или в своей рабочей конфигурации позволял подключаться к 80-му и 81-му портам протокола tcp. Добавим разрешающие правила с помощью команд:

```
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 также отключить фильтр командами:

```
iptables -F
```

```
iptables -P INPUT ACCEPT iptables -P OUTPUT ACCEPT
```

## Порядок выполнения работы

1. Войдем в систему с полученными учётными данными и убедимся, что SELinux работает в режиме enforcing политики targeted с помощью команд *getenforce* и *sestatus*

```
SELinux status:          enabled
SELinuxfs mount:        /selinux
Current mode:           enforcing
Mode from config file:  enforcing
Policy version:         24
Policy from config file: targeted
```

2. Обратимся к веб-серверу, запущенному на нашем компьютере, и убедимся, что последний работает: *service httpd status*

```
[root@Nevzorov Nevzorov]# service httpd status
httpd (pid 8834) выполняется...
[root@Nevzorov Nevzorov]# █
```

3. Найдем веб-сервер Apache в списке процессов, определим его контекст безопасности, используем команду *ps auxZ | grep httpd*

```
[root@Nevzorov Nevzorov]# ps auxZ | grep httpd
unconfined_u:system_r:httpd_t:s0 root      8834  0.0  0.3 11668 3508 ?        Ss
06:49   0:00 /usr/sbin/httpd
unconfined_u:system_r:httpd_t:s0 apache  8837  0.0  0.2 11804 2896 ?        Ss
06:49   0:00 /usr/sbin/httpd
unconfined_u:system_r:httpd_t:s0 apache  8838  0.0  0.2 11804 2960 ?        Ss
06:49   0:00 /usr/sbin/httpd
unconfined_u:system_r:httpd_t:s0 apache  8839  0.0  0.2 11804 2888 ?        Ss
06:49   0:00 /usr/sbin/httpd
unconfined_u:system_r:httpd_t:s0 apache  8840  0.0  0.2 11804 2888 ?        Ss
06:49   0:00 /usr/sbin/httpd
unconfined_u:system_r:httpd_t:s0 apache  8841  0.0  0.2 11804 2888 ?        Ss
06:49   0:00 /usr/sbin/httpd
unconfined_u:system_r:httpd_t:s0 apache  8842  0.0  0.2 11804 2888 ?        Ss
06:49   0:00 /usr/sbin/httpd
unconfined_u:system_r:httpd_t:s0 apache  8843  0.0  0.2 11804 2888 ?        Ss
06:49   0:00 /usr/sbin/httpd
unconfined_u:system_r:httpd_t:s0 apache  8844  0.0  0.2 11668 2208 ?        Ss
06:49   0:00 /usr/sbin/httpd
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 root 9208  0.0  0.0 4444 82
0 pts/3 S+ 07:43   0:00 grep httpd
[root@Nevzorov Nevzorov]# █
```

В нашем случае контекст безопасности `unconfined_u:system_r:httpd_t`

4. Посмотрим текущее состояние переключателей SELinux для Apache с помощью команды *sestatus -b | grep httpd*

```
[root@Nevzorov Nevzorov1]# sestatus -b | grep httpd
allow_httpd_anon_write           off
allow_httpd_mod_auth_ntlm_winbind off
allow_httpd_mod_auth_pam         off
allow_httpd_sys_script_anon_write off
httpd_builtin_scripting          on
httpd_can_check_spam              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_avahi                 on
httpd_dbus_sssd                  off
httpd_enable_cgi                 on
httpd_enable_ftp_server          off
httpd_enable_homedirs            off
httpd_execmem                    off
httpd_manage_ipa                 off
httpd_read_user_content          off
httpd_run_preupgrade             off
httpd_run_stickshift             off
httpd_serve_cobbler_files        off
httpd_setrlimit                  off
httpd_ssi_exec                   off
httpd_tmp_exec                   off

httpd_tty_comm                   on
httpd_unified                    on
httpd_use_cifs                   off
httpd_use_fusefs                 off
httpd_use_gpg                    off
httpd_use_nfs                    off
httpd_use_openstack              off
httpd_verify_dns                 off
```

Многие из переключателей находятся в положении «off».

- Посмотрим статистику по политике с помощью команды *seinfo*, также определим множество пользователей, ролей и типов.

```
[root@Nevzorov Nevzorov1]# seinfo

Statistics for policy file: /etc/selinux/targeted/policy/policy.24
Policy Version & Type: v.24 (binary, mls)

Classes:      81      Permissions:    238
Sensitivities: 1      Categories:    1824
Types:        3920    Attributes:    295
Users:        9      Roles:         12
Booleans:     237    Cond. Expr.:  277
Allow:        323336  Neverallow:    0
Auditallow:   141    Dontaudit:     274738
Type_trans:   42431  Type_change:   38
Type_member:  48     Role_allow:    19
Role_trans:   386    Range_trans:   6258
Constraints:  90     Validatetrans: 0
Initial SIDs: 27     Fs_use:        23
Genfscon:     84     Portcon:       474
Netifcon:     0      Nodecon:       0
Permissives:  90     Polcap:        2
```

Пользователей: 9, ролей: 12, типов: 3920.

- Определим тип файлов и поддиректорий, находящихся в директории */var/www* с помощью команды *ls -lZ /var/www*

```
[Nevzorov1@Nevzorov ~]$ 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 error
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 html
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 icons
```

- Определим тип файлов, находящихся в директории */var/www/html* с помощью команды *ls -lZ /var/www/html*

```
[Nevzorov1@Nevzorov ~]$ ls -lZ /var/www/html
[Nevzorov1@Nevzorov ~]$
```

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

```
[Nevzorov1@Nevzorov ~]$ echo "test" > /var/www/html/test.txt
bash: /var/www/html/test.txt: Отказано в доступе
[Nevzorov1@Nevzorov ~]$ su
Пароль:
[root@Nevzorov Nevzorov1]# echo "test" > /var/www/html/test.txt
[root@Nevzorov Nevzorov1]# su guest
[guest@Nevzorov Nevzorov1]$ echo "test" > /var/www/html/test1.txt
bash: /var/www/html/test1.txt: Отказано в доступе
[guest@Nevzorov Nevzorov1]$ su guest2
Пароль:
[guest2@Nevzorov Nevzorov1]$ echo "test" > /var/www/html/test1.txt
bash: /var/www/html/test1.txt: Отказано в доступе
[guest2@Nevzorov Nevzorov1]$ exit
exit
[guest@Nevzorov Nevzorov1]$ exit
exit
[root@Nevzorov Nevzorov1]#
```

Видно, что только суперпользователь может создать файл в данной директории.

9. В следствие этого создадим от имени суперпользователя html-файл `/var/www/html/test.html` следующего содержания:

`<html>`

`<body>test</body>`

`</html>`

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

]

Записано 3 строки

10. Проверим контекст созданного файла.

```
[root@Nevzorov Nevzorov1]# ls -lZ /var/www/html
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 test.html
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 test.txt
[root@Nevzorov Nevzorov1]# ls -l /var/www/html
итого 8
-rw-r--r--. 1 root root 33 Окт 25 09:49 test.html
-rw-r--r--. 1 root root 5 Окт 25 09:28 test.txt
[root@Nevzorov Nevzorov1]#
```

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

11. Обратимся к файлу через веб-сервер, введя в браузере firefox адрес

<http://127.0.0.1/test.html>

Убедимся, что файл был успешно отображен.



12. Изучим справку *man httpd\_selinux* и выясним, какие контексты файлов определены для *httpd* и сопоставим их с типом файла *test.html*. Проверим контекст файла командой `ls -Z /var/www/html/test.htm`

```
[root@Nevzorov Nevzorov1]# 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@Nevzorov Nevzorov1]#
```

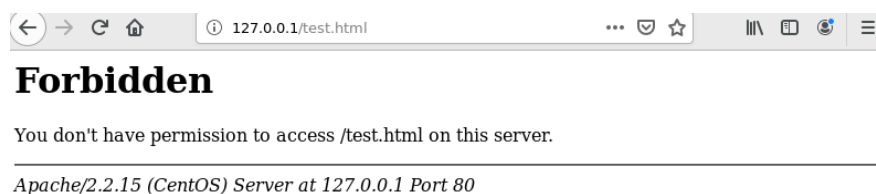
Т.к. по умолчанию пользователи CentOS являются свободными (unconfined) от типа, созданному нами файлу *test.html* был сопоставлен SELinux, пользователь *unconfined\_u*. Это первая часть контекста. Далее политика ролевого разделения доступа RBAC используется процессами, но не файлами, поэтому роли не 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. Изменим 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*:

```
chcon -t samba_share_t /var/www/html/test.html
```

```
ls -Z /var/www/html/test.html
```

14. Попробуем еще раз получить доступ к файлу через веб-сервер, введя в браузере firefox адрес <http://127.0.0.1/test.html>



Мы получили сообщение об ошибке.



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

```
type=USER_START msg=audit(1603680001.944:2287): user pid=15617 uid=0 auid=0 ses=288 subj=system_u:system_r:crond_t:s0-s0:c0.c1023 msg='op=PAM:session_open acct="root" exe="/usr/sbin/crond" hostname=? addr=? terminal=cron res=success'
type=CRED_DISP msg=audit(1603680001.989:2288): user pid=15617 uid=0 auid=0 ses=288 subj=system_u:system_r:crond_t:s0-s0:c0.c1023 msg='op=PAM:setcred acct="root" exe="/usr/sbin/crond" hostname=? addr=? terminal=cron res=success'
type=USER_END msg=audit(1603680001.989:2289): user pid=15617 uid=0 auid=0 ses=288 subj=system_u:system_r:crond_t:s0-s0:c0.c1023 msg='op=PAM:session_close acct="root" exe="/usr/sbin/crond" hostname=? addr=? terminal=cron res=success'
type=USER_AUTH msg=audit(1603680517.512:2290): user pid=15689 uid=500 auid=500 ses=124 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 msg='op=PAM:unix_chkpwd acct="eakhityaev" exe="/sbin/unix_chkpwd" hostname=? addr=? terminal=? res=success'
type=USER_ACCT msg=audit(1603680602.043:2291): user pid=15699 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:crond_t:s0-s0:c0.c1023 msg='op=PAM:accounting acct="root" exe="/usr/sbin/crond" hostname=? addr=? terminal=cron res=success'
type=CRED_ACQ msg=audit(1603680602.044:2292): user pid=15699 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:crond_t:s0-s0:c0.c1023 msg='op=PAM:setcred acct="root" exe="/usr/sbin/crond" hostname=? addr=? terminal=cron res=success'
type=LOGIN msg=audit(1603680602.048:2293): pid=15699 uid=0 subj=system_u:system_r:crond_t:s0-s0:c0.c1023 old auid=4294967295 new auid=0 old ses=4294967295 new ses=289
type=USER_START msg=audit(1603680602.052:2294): user pid=15699 uid=0 auid=0 ses=289 subj=system_u:system_r:crond_t:s0-s0:c0.c1023 msg='op=PAM:session_open acct="root" exe="/usr/sbin/crond" hostname=? addr=? terminal=cron res=success'
type=CRED_DISP msg=audit(1603680602.094:2295): user pid=15699 uid=0 auid=0 ses=289 subj=system_u:system_r:crond_t:s0-s0:c0.c1023 msg='op=PAM:setcred acct="root" exe="/usr/sbin/crond" hostname=? addr=? terminal=cron res=success'
type=USER_END msg=audit(1603680602.094:2296): user pid=15699 uid=0 auid=0 ses=289 subj=system_u:system_r:crond_t:s0-s0:c0.c1023 msg='op=PAM:session_close acct="root" exe="/usr/sbin/crond" hostname=? addr=? terminal=cron res=success'
type=USER_ACCT msg=audit(1603681201.119:2297): user pid=15794 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:crond_t:s0-s0:c0.c1023 msg='op=PAM:accounting acct="root" exe="/usr/sbin/crond" hostname=? addr=? terminal=cron res=success'
type=CRED_ACQ msg=audit(1603681201.120:2298): user pid=15794 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:crond_t:s0-s0:c0.c1023 msg='op=PAM:setcred acct="root" exe="/usr/sbin/crond" hostname=? addr=? terminal=cron res=success'
type=LOGIN msg=audit(1603681201.121:2299): pid=15794 uid=0 subj=system_u:system_r:crond_t:s0-s0:c0.c1023 old auid=4294967295 new auid=0 old ses=4294967295 new ses=290
type=USER_START msg=audit(1603681201.121:2300): user pid=15794 uid=0 auid=0 ses=290 subj=system_u:system_r:crond_t:s0-s0:c0.c1023 msg='op=PAM:session_open acct="root" exe="/usr/sbin/crond" hostname=? addr=? terminal=cron res=success'
type=CRED_DISP msg=audit(1603681201.197:2301): user pid=15794 uid=0 auid=0 ses=290 subj=system_u:system_r:crond_t:s0-s0:c0.c1023 msg='op=PAM:setcred acct="root" exe="/usr/sbin/crond" hostname=? addr=? terminal=cron res=success'
type=USER_END msg=audit(1603681201.197:2302): user pid=15794 uid=0 auid=0 ses=290 subj=system_u:system_r:crond_t:s0-s0:c0.c1023 msg='op=PAM:session_close acct="root" exe="/usr/sbin/crond" hostname=? addr=? terminal=cron res=success'
```

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. Попробуем 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.

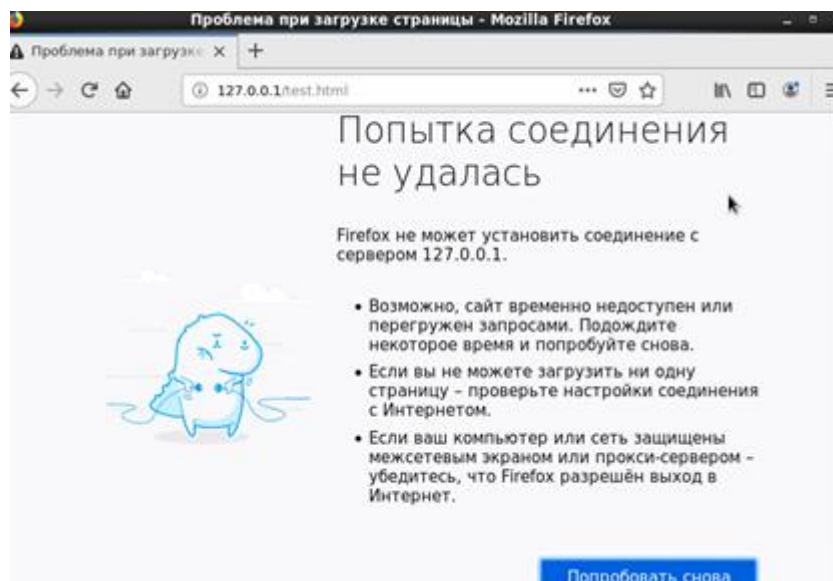
```
GNU nano 2.0.9      Файл: /etc/httpd/conf/httpd.conf

# MaxRequestsPerChild: maximum number of requests a server process serves
<IfModule worker.c>
StartServers      4
MaxClients        300
MinSpareThreads   25
MaxSpareThreads   75
ThreadsPerChild   25
MaxRequestsPerChild 0
</IfModule>

#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, in addition to 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 (0.0.0.0)
#
#Listen 12.34.56.78:80
Listen 81

#
# Dynamic Shared Object (DSO) Support
[ Записано 1009 строк ]
^G Помощь  ^O Записать ^R ЧитФайл ^Y ПредСтр ^K Вырезать ^C ТекПозиц
^X Выход   ^J Выводить ^W Поиск   ^V СледСтр ^U ОтмВырезк ^T Словарь
```

17. Перезапустим веб-сервер Apache и попробуем обратиться к файлу через веб-сервер, введя в браузере firefox адрес <http://127.0.0.1/test.html>



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. Подтвердим свои догадки, проанализировав log-файлы: `tail -n1 /var/log/messages` и просмотрев файлы `/var/log/http/error_log`, `/var/log/http/access_log` и `/var/log/audit/audit.log`



```

127.0.0.1 - - [25/Oct/2020:06:53:16 +0300] "GET /favicon.ico HTTP/1.1" 404 284 "-"
"Mozilla/5.0 (X11; Linux i686; rv:68.0) Gecko/20100101 Firefox/68.0"
127.0.0.1 - - [25/Oct/2020:10:13:37 +0300] "GET /test.html HTTP/1.1" 200 33 "-"
"Mozilla/5.0 (X11; Linux i686; rv:68.0) Gecko/20100101 Firefox/68.0"
127.0.0.1 - - [25/Oct/2020:10:13:37 +0300] "GET /favicon.ico HTTP/1.1" 404 284 "-"
"Mozilla/5.0 (X11; Linux i686; rv:68.0) Gecko/20100101 Firefox/68.0"
127.0.0.1 - - [25/Oct/2020:10:34:06 +0300] "GET /test.html HTTP/1.1" 403 286 "-"
"Mozilla/5.0 (X11; Linux i686; rv:68.0) Gecko/20100101 Firefox/68.0"
127.0.0.1 - - [26/Oct/2020:05:32:01 +0300] "GET /test.html HTTP/1.1" 200 33 "-"
"Mozilla/5.0 (X11; Linux i686; rv:68.0) Gecko/20100101 Firefox/68.0"
127.0.0.1 - - [26/Oct/2020:05:32:01 +0300] "GET /favicon.ico HTTP/1.1" 404 284 "-"
"Mozilla/5.0 (X11; Linux i686; rv:68.0) Gecko/20100101 Firefox/68.0"
127.0.0.1 - - [26/Oct/2020:05:33:53 +0300] "GET /test.html HTTP/1.1" 403 286 "-"
"Mozilla/5.0 (X11; Linux i686; rv:68.0) Gecko/20100101 Firefox/68.0"

type=USER_START msg=audit(1603681261.215:2306): user pid=15802 uid=0 auid=0 ses=
91 subj=system_u:system_r:cron_t:s0-s0:c0.c1023 msg='op=PAM:session_open acct="
oot" exe="/usr/sbin/crond" hostname=? addr=? terminal=cron res=success'
type=CRED_DISP msg=audit(1603681261.312:2307): user pid=15802 uid=0 auid=0 ses=2
1 subj=system_u:system_r:cron_t:s0-s0:c0.c1023 msg='op=PAM:setcred acct="root"
xe="/usr/sbin/crond" hostname=? addr=? terminal=cron res=success'
type=USER_END msg=audit(1603681261.312:2308): user pid=15802 uid=0 auid=0 ses=29
subj=system_u:system_r:cron_t:s0-s0:c0.c1023 msg='op=PAM:session_close acct="r
ot" exe="/usr/sbin/crond" hostname=? addr=? terminal=cron res=success'
type=USER_ACCT msg=audit(1603681801.331:2309): user pid=15891 uid=0 auid=4294967
95 ses=4294967295 subj=system_u:system_r:cron_t:s0-s0:c0.c1023 msg='op=PAM:acco
unting acct="root" exe="/usr/sbin/crond" hostname=? addr=? terminal=cron res=succ
ss'
type=CRED_ACQ msg=audit(1603681801.331:2310): user pid=15891 uid=0 auid=42949672
5 ses=4294967295 subj=system_u:system_r:cron_t:s0-s0:c0.c1023 msg='op=PAM:setcr
d acct="root" exe="/usr/sbin/crond" hostname=? addr=? terminal=cron res=success'
type=LOGIN msg=audit(1603681801.339:2311): pid=15891 uid=0 subj=system_u:system_
:cron_t:s0-s0:c0.c1023 old auid=4294967295 new auid=0 old ses=4294967295 new se
s=292
type=USER_START msg=audit(1603681801.339:2312): user pid=15891 uid=0 auid=0 ses=
92 subj=system_u:system_r:cron_t:s0-s0:c0.c1023 msg='op=PAM:session_open acct="
oot" exe="/usr/sbin/crond" hostname=? addr=? terminal=cron res=success'
type=CRED_DISP msg=audit(1603681801.382:2313): user pid=15891 uid=0 auid=0 ses=2
2 subj=system_u:system_r:cron_t:s0-s0:c0.c1023 msg='op=PAM:setcred acct="root" e
xe="/usr/sbin/crond" hostname=? addr=? terminal=cron res=success'
type=USER_END msg=audit(1603681801.382:2314): user pid=15891 uid=0 auid=0 ses=292
subj=system_u:system_r:cron_t:s0-s0:c0.c1023 msg='op=PAM:session_close acct="ro
ot" exe="/usr/sbin/crond" hostname=? addr=? terminal=cron res=success'

```

Во всех log-файлах появились записи, кроме /var/log/messages.

## 19. Выполним команду `semanage port -a -t http_port_t -p tcp 81`

После этого проверим список портов командой `semanage port -l | grep http_port_t`

```

[root@Nevzorov Nevzorov1]# semanage port -a -t http_port_t -p tcp 81
/usr/sbin/semanage: Порт tcp/81 уже определен
[root@Nevzorov Nevzorov1]# semanage port -l | grep http_port_t
tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000 st.html
pegasus http port t      tcp      5988

```

Убедились, что порт 81 присутствует в списке.

## 20. Попробуем теперь запустить веб-сервер Apache еще раз.

```

[root@Nevzorov Nevzorov1]# service httpd restart
Останавливается httpd:
Запускается httpd:
[root@Nevzorov Nevzorov1]# service httpd status
httpd (pid 15944) выполняется...
[root@Nevzorov Nevzorov1]#

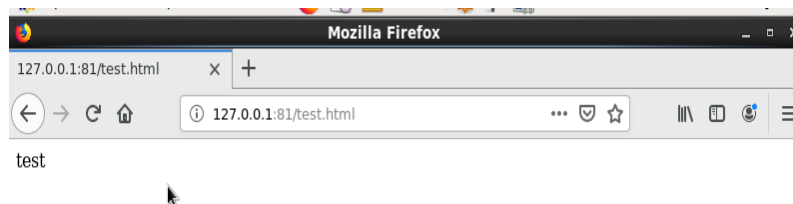
```

## 21. Вернем контекст `httpd_sys_content_t` к файлу `/var/www/html/test.html`:

```
chcon -t httpd_sys_content_t /var/www/html/test.html
```

```
[root@Nevzorov Nevzorov1]# chcon -t httpd_sys_content_t /var/www/html/test.html
[root@Nevzorov Nevzorov1]#
```

После этого вновь попробуем получить доступ к файлу через веб-сервер, введя в браузере firefox адрес <http://127.0.0.1:81/test.html>



Увидели слово содержимое файла - слово «test».

22. Исправим обратно конфигурационный файл apache, вернув Listen 80.

```
# MaxRequestsPerChild: maximum number of requests a server process serves
<IfModule worker.c>
StartServers      4
MaxClients        300
MinSpareThreads   25
MaxSpareThreads   75
ThreadsPerChild   25
MaxRequestsPerChild 0
</IfModule>

#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, in addition to 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 (0.0.0.0)
#
#Listen 12.34.56.78:80
Listen 80

#
# Dynamic Shared Object (DSO) Support
#
```

Записано 1009 строк

Помощь    Записать    ЧитФайл    ПредСтр    Вырезать    ТекПозиц  
Выход    Выровнять    Поиск    СледСтр    ОтмВырезк    Словарь

23. Удалим 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, поэтому получаем ошибку.

```
[root@Nevzorov Nevzorov1]# semanage port -d -t http_port_t -p tcp 81
/usr/sbin/semanage: Порт tcp/81 определен на уровне политики и не может быть удален
[root@Nevzorov Nevzorov1]#
```

24. Удалим файл `/var/www/html/test.html`: `rm /var/www/html/test.html`

```
[root@Nevzorov Nevzorov1]# rm /var/www/html/test.html
rm: удалить обычный файл «/var/www/html/test.html»? y
[root@Nevzorov Nevzorov1]#
```

## Вывод

Я развил навыки администрирования ОС Linux. Получил первое практическое знакомство с технологией SELinux. Проверил работу SELinux на практике совместно с веб-сервером Apache.