

Лабораторная работа No 6.

Сагдеров Камал, НФИбд-04-22

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Подготовка лабораторного стенда	6
2.2	Выполнение	7
3	Выводы	14

Список иллюстраций

2.1	ServerName	6
2.2	ServerName	7
2.3	iptables	7
2.4	sestatus	8
2.5	httpd	8
2.6	Контекст безопасности	9
2.7	Статистика по политике	9
2.8	Статистика по политике	9
2.9	/var/www/html	10
2.10	Статистика по политике	10
2.11	test.html	10
2.12	test.html	11
2.13	Проверка	11
2.14	test.html	11
2.15	Проверка	12
2.16	/var/log/messages	12
2.17	/var/log/audit/audit.log	12
2.18	Порт 81	12
2.19	Перезапуск	13
2.20	/var/log/messages	13
2.21	Добавление	13
2.22	Проверка	13

Список таблиц

1 Цель работы

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux.

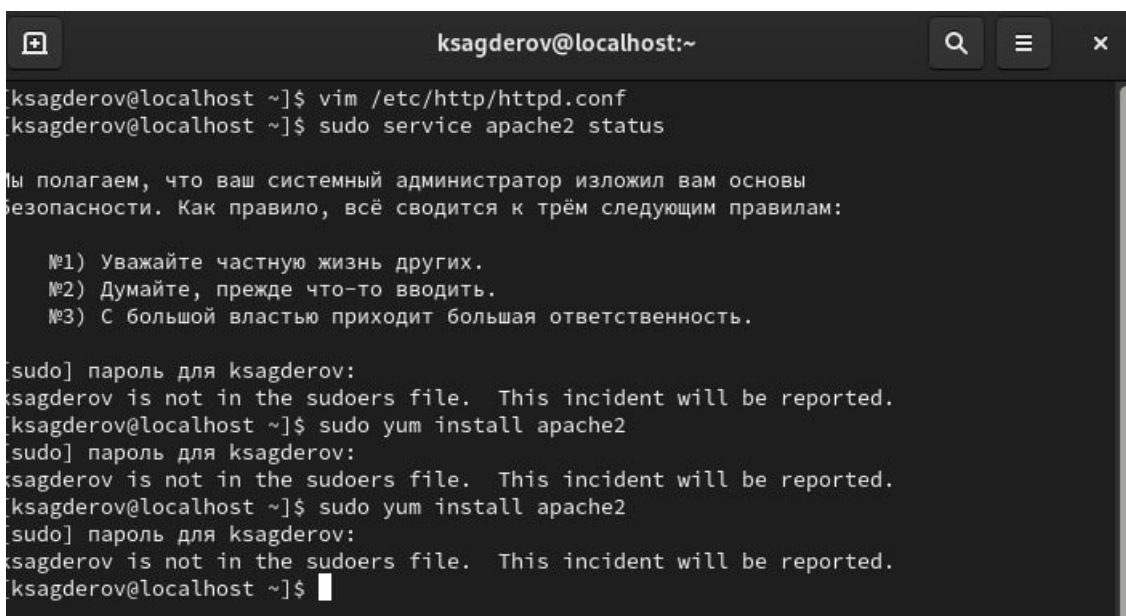
2 Выполнение лабораторной работы

2.1 Подготовка лабораторного стенда

1. Установить Apache2 при помощи dnf.

```
dnf install httpd
```

2. В конфигурационном файле httpd.conf прописать параметр ServerName (2.1).



```
ksagderov@localhost:~  
ksagderov@localhost ~]$ vim /etc/http/httpd.conf  
ksagderov@localhost ~]$ sudo service apache2 status  
Мы полагаем, что ваш системный администратор изложил вам основы  
безопасности. Как правило, всё сводится к трём следующим правилам:  
  
№1) Уважайте частную жизнь других.  
№2) Думайте, прежде что-то вводить.  
№3) С большой властью приходит большая ответственность.  
  
[sudo] пароль для ksagderov:  
ksagderov is not in the sudoers file. This incident will be reported.  
ksagderov@localhost ~]$ sudo yum install apache2  
[sudo] пароль для ksagderov:  
ksagderov is not in the sudoers file. This incident will be reported.  
ksagderov@localhost ~]$ sudo yum install apache2  
[sudo] пароль для ksagderov:  
ksagderov is not in the sudoers file. This incident will be reported.  
ksagderov@localhost ~]$
```

Рис. 2.1: ServerName

```
[root@localhost ksagderov]# vi sudo
[root@localhost ksagderov]# sudo yum install apache2
Rocky Linux 9 - BaseOS          9.9 kB/s | 4.1 kB      00:00
Rocky Linux 9 - BaseOS          1.1 MB/s | 2.2 MB      00:02
Rocky Linux 9 - AppStream        13 kB/s | 4.5 kB      00:00
Rocky Linux 9 - AppStream        4.1 MB/s | 7.4 MB      00:01
Rocky Linux 9 - Extras           5.4 kB/s | 2.9 kB      00:00
Нет соответствия аргументу: apache2
Ошибка: Совпадений не найдено: apache2
[root@localhost ksagderov]#
```

Рис. 2.2: ServerName

3. Отключить пакетный фильтр при помощи iptables (2.2).

```
Выполнено!
[root@localhost ksagderov]# sudo firewall-cmd --permanent --add-service=https
success
[root@localhost ksagderov]# sudo firewall-cmd --reload
success
[root@localhost ksagderov]#
```

Рис. 2.3: iptables

ServerName

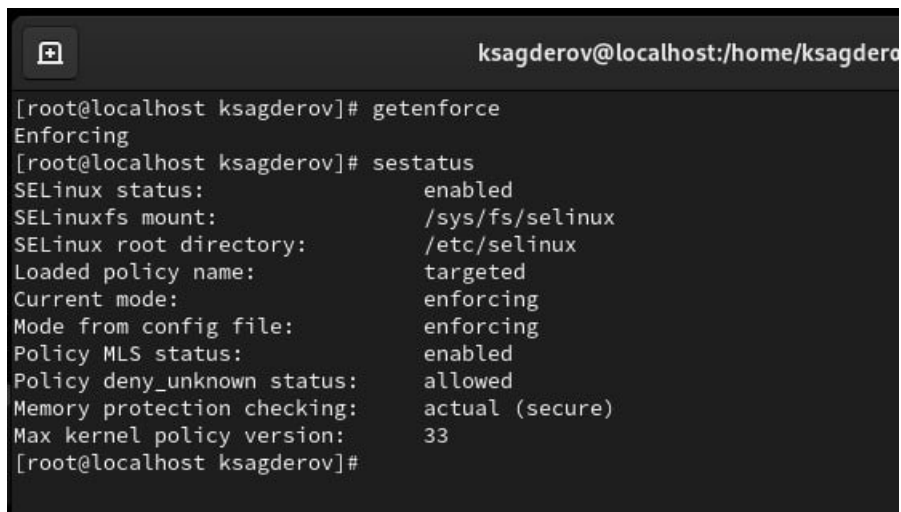
ServerName

ServerName

ServerName

2.2 Выполнение

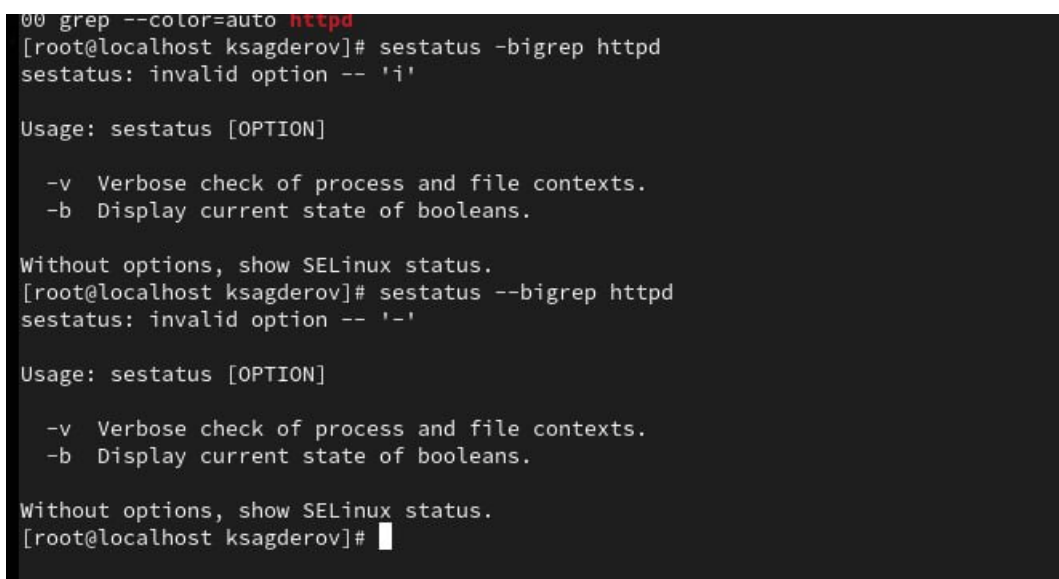
1. Проверим правильность работы SELinux. Должен быть выставлен режим enforcing политики targeted (2.3).



```
ksagderov@localhost:/home/ksagderov
[root@localhost ksagderov]# getenforce
Enforcing
[root@localhost ksagderov]# 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
[root@localhost ksagderov]#
```

Рис. 2.4: sestatus

2. Запустим Apache веб-сервер (??).



```
00 grep --color=auto httpd
[root@localhost ksagderov]# 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.
[root@localhost ksagderov]# sestatus --bigrep httpd
sestatus: invalid option -- '-'

Usage: sestatus [OPTION]

  -v  Verbose check of process and file contexts.
  -b  Display current state of booleans.

Without options, show SELinux status.
[root@localhost ksagderov]#
```

Рис. 2.5: httpd

3. В списке процессов найдем httpd (??). На этот процесс выставлен следующий контекст безопасности (первый столбец изображения **[gentooSELinuxTutorialsLinuxServices?]**).


```

[root@localhost ksagderov]# ps auxZ | grep httpd
system_u:system_r:httpd_t:s0      root      41637  0.0  0.6 20128 11556 ?        Ss   17:41   0:00 /usr/s
bin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0      apache    41638  0.0  0.4 21612  7436 ?        S    17:41   0:00 /usr/s
bin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0      apache    41643  0.0  0.8 1669272 15208 ?       Sl   17:41   0:00 /usr/s
bin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0      apache    41644  0.0  0.6 1538136 11112 ?       Sl   17:41   0:00 /usr/s
bin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0      apache    41645  0.0  0.6 1538136 11112 ?       Sl   17:41   0:00 /usr/s
bin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 root 42056 0.0  0.1 221820 2392 pts/0 R+ 17:55  0:
00 grep --color=auto httpd
[root@localhost ksagderov]#

```

Рис. 2.6: Контекст безопасности

4. Посмотрим текущее состояние переключателей SELinux для Apache2 (??).
5. Также посмотрим текущую статистику по политике (2.4).

```

[root@localhost ksagderov]# sudo vi /var/www/html/trst.html
[root@localhost ksagderov]# ls -lZ /var/www/html
итого 4
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 34 anp 27 18:03 trst.html
[root@localhost ksagderov]#

```

Рис. 2.7: Статистика по политике

6. Посмотрим текущий контекст безопасности для файлов и поддиректорий в директории /var/www (2.5).
 - Установлен контекст httpd_sys_script_exec_t для cgi-скриптов, чтобы был разрешен им доступ ко всем sys-типам.
 - Установлен контекст httpd_sys_content_t для содержимого, которое должно быть доступно для всех скриптов httpd и для самого демона.

Статистика по политике

Рис. 2.8: Статистика по политике

7. В директории /var/www/html пусто.

```

[root@localhost ksagderov]# sudo vi /var/www/html/trst.html
[root@localhost ksagderov]# ls -lZ /var/www/html
итого 4
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 34 anp 27 18:03 trst.html
[root@localhost ksagderov]#

```

Рис. 2.9: /var/www/html

8. В директории /var/www/html создавать папки может только root (право w есть только у него).

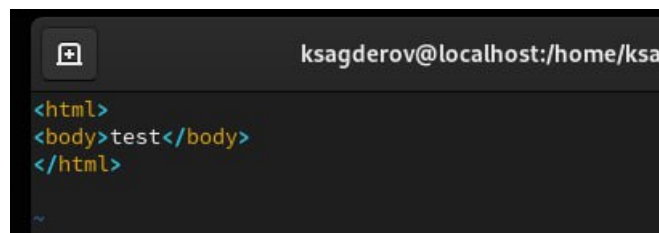
```

[root@localhost ksagderov]# ls -Z /var/www/html/test.html
ls: невозможно получить доступ к '/var/www/html/test.html': Нет такого файла или каталога
[root@localhost ksagderov]#

```

Рис. 2.10: Статистика по политике

9. Создадим файл /var/www/html/test.html (2.9).



The image shows a code editor window with a dark background. The title bar at the top reads 'ksagderov@localhost:/home/ksa'. The editor contains the following HTML code:

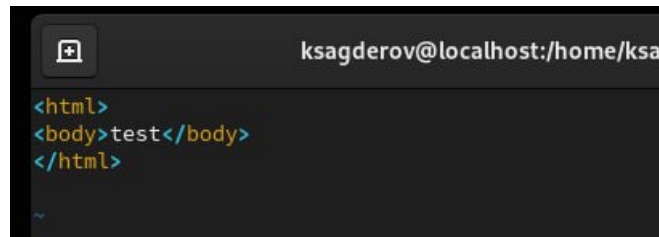
```

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

```

Рис. 2.11: test.html

10. Проверим контекст созданного нами файла (2.12).

A screenshot of a code editor window. The title bar shows the user 'ksagderov@localhost:/home/ksa'. The editor contains the following HTML code:

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

Рис. 2.12: test.html

11. Перейдем в браузер и в нем проверим доступность данного файла (2.10).

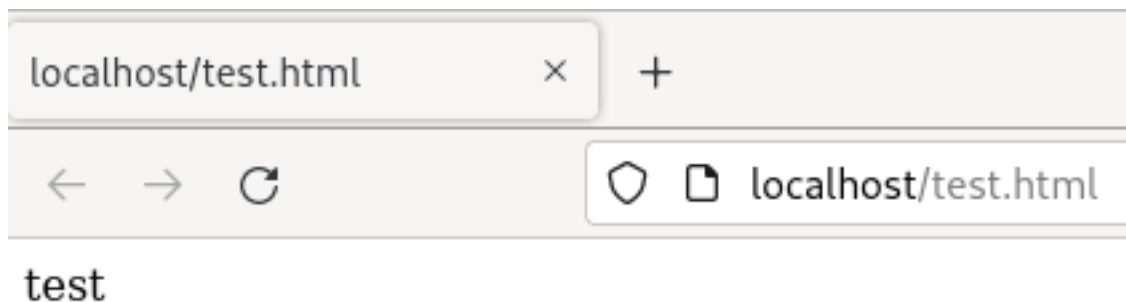
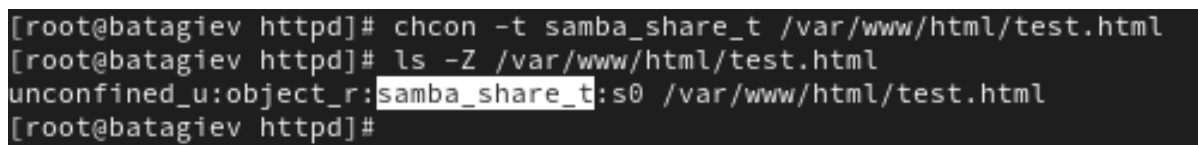


Рис. 2.13: Проверка

12. Изменим контекст файла, чтобы Apache не смог получить доступ (??).

A screenshot of a terminal window. The user is at the root of a system named 'batagiev' and is in the 'httpd' process. The following commands and output are shown:

```
[root@batagiev httpd]# chcon -t samba_share_t /var/www/html/test.html
[root@batagiev httpd]# ls -Z /var/www/html/test.html
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html
[root@batagiev httpd]#
```

Рис. 2.14: test.html

13. Проверим, что доступ к файлу стал не доступен (2.8).

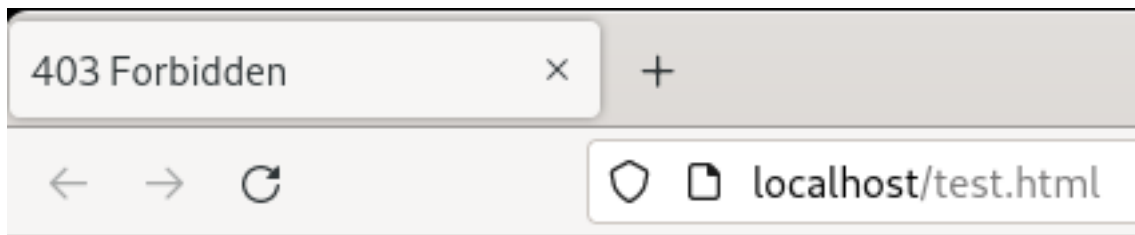


Рис. 2.15: Проверка

14. Посмотрим логи от веб-сервера Apache (??).

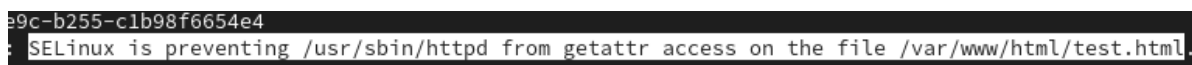


Рис. 2.16: /var/log/messages

Также проверим audit.log (??).

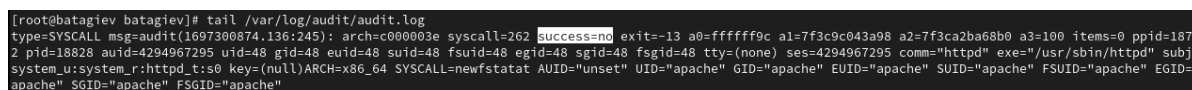


Рис. 2.17: /var/log/audit/audit.log

15. Поменяем порт, на котором работает Apache.

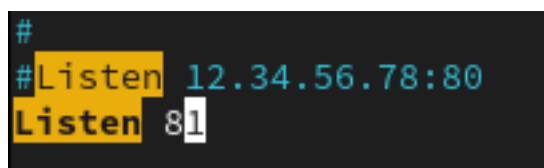


Рис. 2.18: Порт 81

16. Перезапустим веб-сервер (успешно).

```
[root@batagiev httpd]# systemctl restart httpd.service
[root@batagiev httpd]# systemctl status httpd.service
● httpd.service - The Apache HTTP Server
```

Рис. 2.19: Перезапуск

17. В логах наблюдаем запуск сервера на 81 порту.

```
Oct 14 19:32:03 batagiev systemd[1]: Starting The Apache HTTP Server...
Oct 14 19:32:03 batagiev httpd[42657]: Server configured, listening on: port 81
Oct 14 19:32:03 batagiev systemd[1]: Started The Apache HTTP Server.
[root@batagiev batagiev]#
```

Рис. 2.20: /var/log/messages

18. Добавим порт в semanage для http_port_t и проверим его добавление

```
[root@batagiev httpd]# semanage port -a -t http_port_t -p tcp 81
ValueError: Port tcp/81 already defined
[root@batagiev httpd]#
```

Рис. 2.21: Добавление

```
[root@batagiev httpd]# 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. 2.22: Проверка

19. Ввернем контекст файлу test.html.

20. Удалим привязку порта.

21. Удалим файл test.html.

3 Выводы

В результате выполнения работы я выполнил цели работы.