

Отчет по лабораторной работе 6

Даниил Генералов, 1032212280

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	9

Список иллюстраций

2.1	SELinux	6
2.2	SELinux	7
2.3	SELinux	7
2.4	SELinux	8

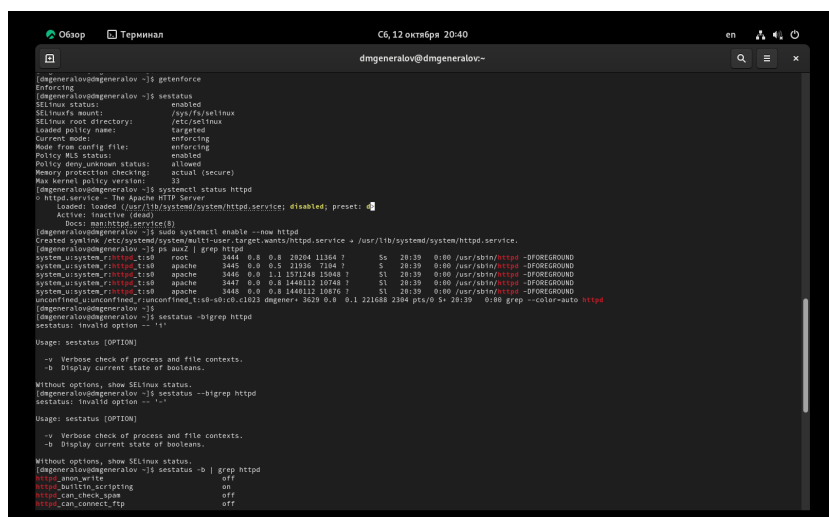
Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

Сначала нужно установить в нашу лабораторную виртуальную машину Apache и отключить пакетный фильтр. После этого мы проверяем, что SELinux работает в режиме enforcing targeted, что Apache запущен (и исправляем это), а также что Apache запущен под своим собственным контекстом httpd_t, и что есть много SELinux-boolean про httpd fig. 2.1.



```
[dmgeneralov@dmgeneralov ~]$ getenforce
Enforcing
[dmgeneralov@dmgeneralov ~]$ sestatus
SELinux status: enabled
SELinuxfs mount: /var/lib/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: 31
[dmgeneralov@dmgeneralov ~]$ systemctl status httpd
○ httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: inactive (dead)
     Docs: man:httpd.service(8)
[dmgeneralov@dmgeneralov ~]$ sudo systemctl enable --now httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[dmgeneralov@dmgeneralov ~]$ ps aux2 | grep httpd
system_u:system_r:httpd_t:s0 root      3444  0.0  0.8 28204 11364 ?        Ss   20:39   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  3445  0.0  0.5 21936 7184 ?        S    20:39   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  3446  0.0  1.1 52124 15448 ?        S    20:39   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  3447  0.0  0.8 144812 10748 ?       S    20:39   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  3448  0.0  0.8 144812 10876 ?       S    20:39   0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-c0.c023 dmgeneralov 3629  0.0  0.1 221088 2304 pts/0  Ss   20:39   0:00 grep --color=auto httpd
[dmgeneralov@dmgeneralov ~]$ sestatus -b|grep httpd
sestatus: invalid option -- 'b'
Usage: sestatus [OPTION]
  -v Verbose check of process and file contexts.
  -b Display current state of booleans.
Without options, show SELinux status.
[dmgeneralov@dmgeneralov ~]$ 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.
[dmgeneralov@dmgeneralov ~]$ sestatus -b | grep httpd
httpd_allow_write          on
httpd_builtin_crypting     on
httpd_can_check_ssh        off
httpd_can_connect_ftp      off
```

Рис. 2.1: SELinux

После этого мы видим, что есть 135 классов и 5145 типов в SELinux. Затем мы видим, что в папке /var/www/html и дочерних папках контекст system_u:object_r:httpd_sys_content_t, а в /var/www/cgi-bin system_u:object_r:httpd_sys_sc. Если суперпользователь создаст файл в /var/www/html, то ему будет выдан контекст unconfined_u:object_r:httpd_sys_content_t: теперь httpd имеет к нему доступ (рис. fig. 2.2).

```
Обзор Терминал C6, 12 октября 20:45 dmgeneralov@dmgeneralov-

[dmgeneralov@dmgeneralov ~]$ setseif
Statistics for policy file: /sys/fs/selinux/policy
Policy Version: 31 (MLS enabled)
Target Policy: selinux
Handle unknown classes: allow
Classes: 135 Permutations: 457
Dontauditrules: 1 Categories: 1024
Types: 5145 Attributes: 259
Donts: 8 Roles: 15
Booleans: 356 Cond. Expr.: 368
Dontauditrules: 83044 Denialallow: 8662
Auditallow: 176 Dontaudit: 8662
Type-trans: 27170 Type-change: 84
Type-member: 37 Range-trans: 5931
Role allow: 40 Role-trans: 417
Constraints: 70 Validatetrans: 8
MLS Constrains: 72 MLS Val. Tran: 8
Permissives: 4 Policies: 8
Defaults: 7 Typebounds: 8
Allowrules: 8 Memberallowrules: 8
Auditallowrules: 0 Dontauditperm: 8
Dontauditperm: 0 Dapolicycon: 8
Initiatl SIDs: 27 Fa_user: 35
Dontcon: 189 Ports: 665
Netifcon: 0 Nodecon: 8

[dmgeneralov@dmgeneralov ~]$ ls -l2 /var/www
ntro 8
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 аэр 8 19:30 ci-b/vh
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 6 аэр 8 19:30 hml
[dmgeneralov@dmgeneralov ~]$ ls -l2 /var/www/html
ntro 8
[dmgeneralov@dmgeneralov ~]$ ls -l2 /var/www/html
ntro 8
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 6 аэр 8 19:30
drwxr-xr-x. 4 root root system_u:object_r:httpd_sys_content_t:s0 33 авт 12 20:37
[dmgeneralov@dmgeneralov ~]$ cat | sudo tee /var/www/html/test.html
<html>
<body>test</body>
</html>
[dmgeneralov@dmgeneralov ~]$ ls -l2 /var/www/html
ntro 4
drwxr-xr-x. 2 root root unconfined_u:object_r:httpd_sys_content_t:s0 33 авт 12 20:44 test.html
[dmgeneralov@dmgeneralov ~]$
```

Рис. 2.2: SELinux

Теперь можно проверить, что сервер имеет к нему доступ, попробовав скачать этот файл с него. После этого мы меняем контекст этого файла, и теперь сервер не имеет к нему доступа SeTroubleshoot подсказывает, что ошибка доступа произошла именно из-за SELinux, и предлагает, как ее можно исправить (рис. fig. 2.3).

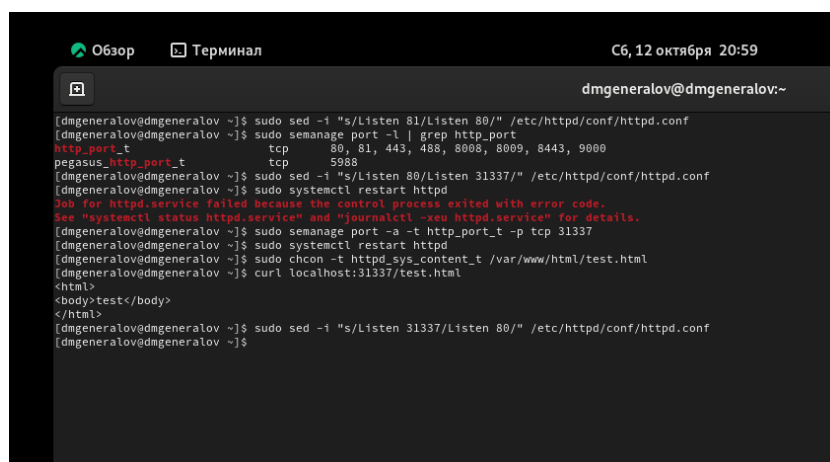
```
Обзор Терминал C6, 12 октября 20:52 dmgeneralov@dmgeneralov-

[dmgeneralov@dmgeneralov ~]$ curl localhost/test.html
<html>
<body>test</body>
</html>
[dmgeneralov@dmgeneralov ~]$ sudo chcon -t samba_share_t /var/www/html/test.html
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html
[dmgeneralov@dmgeneralov ~]$ curl localhost/test.html
<DOCType HTML PUBLIC "-//W3C//DTD HTML 3.0//EN">
<html><head>
<title>test</title>
</head><body>
<div>test</div></body>
</html>
[dmgeneralov@dmgeneralov ~]$ ls -l2 /var/www/html
ntro 4
drwxr-xr-x. 2 root root unconfined_u:object_r:httpd_sys_content_t:s0 33 авт 12 20:44 test.html
[dmgeneralov@dmgeneralov ~]$
```

Рис. 2.3: SELinux

Если мы настроим, чтобы Apache пытался слушать порт 81, а не порт 80, то он смог запуститься, потому что оказывается в настройках по умолчанию порт

81 уже разрешен для типа http_port_t. Чтобы продемонстрировать ситуацию, когда это не разрешено, я вместо этого выбрал порт 31337: теперь запуск httpd блокируется. Чтобы это было разрешено, нужно добавить этот порт в список разрешенных для типа http_port_t: после этого запуск будет успешен, и, вернув исходный контекст файлу, можно будет скачать его (рис. fig. 2.4).



```
[dmgeneralov@dmgeneralov ~]$ sudo sed -i "s/Listen 81/Listen 80/" /etc/httpd/conf/httpd.conf
[dmgeneralov@dmgeneralov ~]$ sudo semanage port -l | grep http_port
http_port_t                tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t        tcp      5988
[dmgeneralov@dmgeneralov ~]$ sudo sed -i "s/Listen 80/Listen 31337/" /etc/httpd/conf/httpd.conf
[dmgeneralov@dmgeneralov ~]$ sudo systemctl restart httpd
Job for httpd.service failed because the control process exited with error code.
See "systemctl status httpd.service" and "journalctl -xeu httpd.service" for details.
[dmgeneralov@dmgeneralov ~]$ sudo semanage port -a -t http_port_t -p tcp 31337
[dmgeneralov@dmgeneralov ~]$ sudo systemctl restart httpd
[dmgeneralov@dmgeneralov ~]$ sudo chcon -t httpd_sys_content_t /var/www/html/test.html
[dmgeneralov@dmgeneralov ~]$ curl localhost:31337/test.html
<html>
<body>test</body>
</html>
[dmgeneralov@dmgeneralov ~]$ sudo sed -i "s/Listen 31337/Listen 80/" /etc/httpd/conf/httpd.conf
[dmgeneralov@dmgeneralov ~]$
```

Рис. 2.4: SELinux

3 Выводы

В этой лабораторной работе мы рассмотрели, как использовать SELinux для ограничения возможностей процессов, на примере Apache.