

Лабораторная работа №6

Информационная безопасность

Николаев Дмитрий Иванович

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	7
4	Выводы	19
	Список литературы	20

Список иллюстраций

3.1	Режим работы SELinux	7
3.2	Обращение к веб-серверу	8
3.3	Проверка контекста безопасности процессов	8
3.4	Состояние переключателей SELinux	9
3.5	Статистика по политике	10
3.6	Информация о файлах и поддиректориях директории /var/www	10
3.7	Создание html-файла	11
3.8	Проверка контекста созданного файла	11
3.9	Обращение к файлу через веб-сервер	12
3.10	Вызов справки и проверка контекста созданного файла	12
3.11	Смена контекста файла	12
3.12	Попытка получения доступа к файлу через веб-сервер	13
3.13	Последние сообщения log-файла	13
3.14	Смена порта прослушивания с 80 на 81	14
3.15	Перезапуск веб-сервера Apache	14
3.16	Попытка подключения к веб-серверу через браузер	15
3.17	Анализ сообщений лог-файлов	15
3.18	Установка 81 порта и проверка списка всех портов	16
3.19	Повторный запуск веб-сервера Apache	16
3.20	Возвращение контекста httpd_sys_content_t нашему файлу	16
3.21	Получение доступа к файлу через браузер	17
3.22	Смена порта прослушивания с 81 на 80	17
3.23	Попытка удаления привязки к 81 порту	17
3.24	Удаление файла test.html	18

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 Теоретическое введение

Домен — список действий, которые может выполнять процесс. Обычно в качестве домена определяется минимально возможный набор действий, при помощи которых процесс способен функционировать. Таким образом, если процесс дискредитирован, злоумышленнику не удастся нанести большого вреда.

Роль — список доменов, которые могут быть применены. Если какого-то домена нет в списке доменов какой-то роли, то действия из этого домена не могут быть применены.

Тип — набор действий, которые допустимы по отношению к объекту. Тип отличается от домена тем, что он может применяться к пайпам, каталогам и файлам, в то время как домен применяется к процессам.

Контекст безопасности — все атрибуты SELinux — роли, типы и домены [1].

3 Выполнение лабораторной работы

Установим веб-сервер Apache с помощью команды `yum install httpd`, после чего следуем согласно [2].

1. Войдем в систему и убедимся, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` и `sestatus`. ([3.1]).

```
[dinikolaev@dinikolaev ~]$ getenforce
Enforcing
[dinikolaev@dinikolaev ~]$ 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
[dinikolaev@dinikolaev ~]$
```

Рис. 3.1: Режим работы SELinux

2. Обратимся с помощью браузера к веб-серверу, запущенному на нашем компьютере, и убедимся, что последний работает: `service httpd status` ([3.2]).

```
[dinikolaev@dinikolaev ~]$ 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; preset: disabled)
   Active: active (running) since Sat 2023-10-07 22:32:51 MSK; 2min 12s ago
     Docs: man:httpd.service(8)
  Main PID: 42745 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0 B/sec"
    Tasks: 213 (limit: 5689)
  Memory: 38.2M
    CPU: 134ms
  CGroup: /system.slice/httpd.service
          └─42745 /usr/sbin/httpd -DFOREGROUND
            └─42753 /usr/sbin/httpd -DFOREGROUND
              └─42754 /usr/sbin/httpd -DFOREGROUND
                └─42755 /usr/sbin/httpd -DFOREGROUND
                  └─42756 /usr/sbin/httpd -DFOREGROUND

окт 07 22:32:51 dinikolaev systemd[1]: Starting The Apache HTTP Server...
окт 07 22:32:51 dinikolaev systemd[1]: Started The Apache HTTP Server.
окт 07 22:32:51 dinikolaev httpd[42745]: Server configured, listening on: port 80
[dinikolaev@dinikolaev ~]$
```

Рис. 3.2: Обращение к веб-серверу

- Найдем веб-сервер Apache в списке процессов, определим его контекст безопасности командами `ps auxZ | grep httpd` и `ps -eZ | grep httpd`. Имеем следующий контекст: `system_u:system_r:httpd_t` (пользователь:роль:тип) ([3.3]).

```
[dinikolaev@dinikolaev ~]$ ps auxZ | grep httpd
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 dinikol+ 42684 0.0 0.9 236232 8808 pts/0 T 22:31
0:00 /bin/systemctl status httpd.service
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 dinikol+ 42699 0.0 0.9 236232 8864 pts/0 T 22:32
0:00 /bin/systemctl status httpd.service
system_u:system_r:httpd_t:s0 root 42745 0.0 1.0 20128 9748 ? Ss 22:32 0:00 /usr
/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 42753 0.0 0.5 21612 5772 ? S 22:32 0:00 /usr
/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 42754 0.0 1.4 1079384 13824 ? Sl 22:32 0:00 /usr
/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 42755 0.0 1.6 1210520 15772 ? Sl 22:32 0:00 /usr
/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 42756 0.0 1.4 1079384 13912 ? Sl 22:32 0:00 /usr
/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 dinikol+ 43031 0.0 0.9 236232 9080 pts/0 T 22:33
0:00 /bin/systemctl status httpd.service start
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 dinikol+ 43140 0.0 0.2 221688 2332 pts/0 S+ 22:3
6 0:00 grep --color=auto httpd
[dinikolaev@dinikolaev ~]$ ps -eZ | grep httpd
system_u:system_r:httpd_t:s0 42745 ? 00:00:00 httpd
system_u:system_r:httpd_t:s0 42753 ? 00:00:00 httpd
system_u:system_r:httpd_t:s0 42754 ? 00:00:00 httpd
system_u:system_r:httpd_t:s0 42755 ? 00:00:00 httpd
system_u:system_r:httpd_t:s0 42756 ? 00:00:00 httpd
[dinikolaev@dinikolaev ~]$
```

Рис. 3.3: Проверка контекста безопасности процессов

- Посмотрим текущее состояние переключателей SELinux для Apache с помощью команды `sestatus -b | grep httpd` ([3.4]):


```
[dinikolaev@dinikolaev ~]$ 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_manage_courier_spool               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                     off
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
httpd_serve_cobbler_files                   off
httpd_setrlimit                            off
httpd_ssl_exec                             off
httpd_sys_script_anon_write                 off
httpd_tmp_exec                             off
httpd_tty_comm                             off
httpd_unified                              off
httpd_use_cifs                             off
httpd_use_fusefs                            off
httpd_use_gpg                              off
httpd_use_nfs                              off
httpd_use_opencryptoki                      off
httpd_use_openstack                        off
httpd_use_sasl                             off
httpd_verify_dns                           off
[dinikolaev@dinikolaev ~]$
```

Рис. 3.4: Состояние переключателей SELinux

5. Посмотрим статистику по политике с помощью команды `seinfo`. Как видим, имеется 8 пользователей, 15 ролей и 5135 типов ([3.5]).

```
[dinikolaev@dinikolaev ~]$ seinfo
Statistics for policy file: /sys/fs/selinux/policy
Policy Version:          33 (MLS enabled)
Target Policy:           selinux
Handle unknown classes:  allow

Classes:          135      Permissions:        457
Sensitivities:    1        Categories:         1024
Types:            5135     Attributes:         259
Users:            8        Roles:              15
Booleans:         357     Cond. Expr.:       390
Allow:            65380    Neverallow:         0
Auditallow:       172     Dontaudit:          8647
Type_trans:       267809   Type_change:        94
Type_member:      37       Range_trans:        6164
Role allow:       39       Role_trans:         419
Constraints:      70       Validatetrans:      0
MLS Constrain:    72       MLS Val. Tran:      0
Permissives:      2        Polcap:             6
Defaults:         7        Typebounds:         0
Allowxperm:       0        Neverallowxperm:    0
Auditallowxperm:  0        Dontauditxperm:     0
Ibendportcon:     0        Ibpkeycon:          0
Initial SIDs:     27       Fs_use:             35
Genfscon:         109     Portcon:            665
Netifcon:         0        Nodecon:            0

[dinikolaev@dinikolaev ~]$
```

Рис. 3.5: Статистика по политике

6. Определим тип файлов и поддиректорий, находящихся в директории /var/www с помощью команды `ls -lZ /var/www` — в ней находится две поддиректории. Определим тип файлов, находящихся в директории /var/www/html командой `ls -lZ /var/www/html` — директория пуста. Определим круг пользователей, которым разрешено создание файлов в директории /var/www/html — разрешено только владельцу директории ([3.6]).

```
[dinikolaev@dinikolaev ~]$ ls -lZ /var/www
итого 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0  6 июл 20 11:44 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0      6 июл 20 11:44 html
[dinikolaev@dinikolaev ~]$ ls -lZ /var/www/html
итого 0
[dinikolaev@dinikolaev ~]$
```

Рис. 3.6: Информация о файлах и поддиректориях директории /var/www

7. Создадим от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись в директорию) html-файл /var/www/html/test.html следующего содержания ([3.7]):

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

```
[dinikolaev@dinikolaev ~]$ su
Пароль:
[root@dinikolaev dinikolaev]# touch /var/www/html/test.html
[root@dinikolaev dinikolaev]# vim /var/www/html/test.html
[root@dinikolaev dinikolaev]# cat /var/www/html/test.html
<html>
    <body>test</body>
</html>
[root@dinikolaev dinikolaev]#
```

Рис. 3.7: Создание html-файла

8. Проверим контекст созданного нами файла. По умолчанию устанавливается следующий контекст: пользователь — unconfined_u (несвязанный), роль — object_r, тип — httpd_sys_content_t ([3.8]).

```
[dinikolaev@dinikolaev ~]$ ps auxZ | grep html
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 dinikol+ 43656 0.0  0.2 221688 2332 pts/0 S+ 22:4
8  0:00 grep --color=auto html
[dinikolaev@dinikolaev ~]$ ps auxZ | grep test.html
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 dinikol+ 43666 0.0  0.2 221824 2316 pts/0 S+ 22:4
9  0:00 grep --color=auto test.html
[dinikolaev@dinikolaev ~]$ ps auxZ | grep www
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 dinikol+ 43672 0.0  0.2 221688 2332 pts/0 S+ 22:4
9  0:00 grep --color=auto www
[dinikolaev@dinikolaev ~]$ ps -eZ | grep html
[dinikolaev@dinikolaev ~]$ ls -lZ /var/www/html
итого 4
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 34 окт  7 22:46 test.html
[dinikolaev@dinikolaev ~]$
```

Рис. 3.8: Проверка контекста созданного файла

9. Обратимся к файлу через веб-сервер, введя в браузере адрес <http://127.0.0.1/test.html>. Убедимся, что файл был успешно отображён ([3.9]).

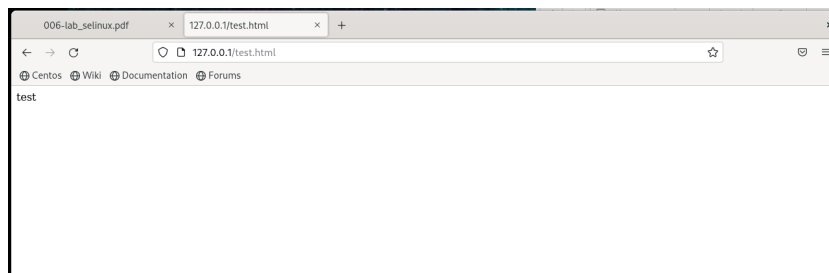


Рис. 3.9: Обращение к файлу через веб-сервер

10. Изучим справку `man httpd_selinux`. Справки по `httpd_selinux` нет, а справки по `httpd` и `selinux` не содержат информации о возможных контекстах. Проверим контекст нашего файла командой `ls -Z /var/www/html/test.html` ([3.10]).

```
[dinikolaev@dinikolaev ~]$ man selinux
[dinikolaev@dinikolaev ~]$ man httpd
[dinikolaev@dinikolaev ~]$ man httpd_selinux
Нет справочной страницы для httpd_selinux
[dinikolaev@dinikolaev ~]$ ls -Z /var/www/html/test.html
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test.html
```

Рис. 3.10: Вызов справки и проверка контекста созданного файла

11. Изменим 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` ([3.11]):

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

```
[dinikolaev@dinikolaev ~]$ su
Пароль:
[root@dinikolaev dinikolaev]# chcon -t samba_share_t /var/www/html/test.html
[root@dinikolaev dinikolaev]# ls -Z /var/www/html/test.html
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html
[root@dinikolaev dinikolaev]#
```

Рис. 3.11: Смена контекста файла

12. Попробуем ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Получили сообщение об ошибке ([3.12]):

Forbidden You don't have permission to access /test.html on this server

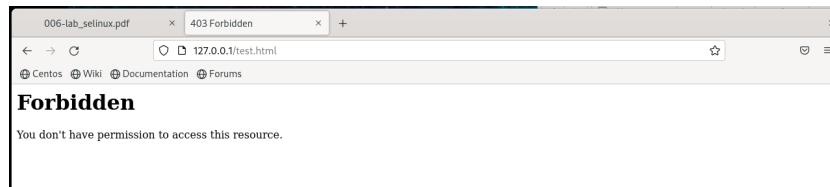


Рис. 3.12: Попытка получения доступа к файлу через веб-сервер

13. Проанализируем ситуацию командой `ls -l /var/www/html/test.html`. Посмотрим log-файлы веб-сервера Apache. Также посмотрим системный лог-файл: `tail /var/log/messages` ([3.13]).

```
[dinikolaev@dinikolaev ~]$ ls -l /var/www/html/test.html
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html
[dinikolaev@dinikolaev ~]$ ls -l /var/www/html/test.html
-rw-r--r--. 1 root root 34 окт  7 22:46 /var/www/html/test.html
[dinikolaev@dinikolaev ~]$ tail /var/log/messages
tail: невозможно открыть '/var/log/messages' для чтения: Отказано в доступе
[dinikolaev@dinikolaev ~]$ sudo tail /var/log/messages
[sudo] пароль для dinikolaev:
Oct  7 23:11:58 dinikolaev packagekitd[2066]: Failed to get cache filename for webkit2gtk3-jsc
Oct  7 23:11:58 dinikolaev packagekitd[2066]: Failed to get cache filename for kernel
Oct  7 23:11:58 dinikolaev packagekitd[2066]: Failed to get cache filename for kernel-core
Oct  7 23:11:58 dinikolaev packagekitd[2066]: Failed to get cache filename for kernel-modules
Oct  7 23:11:58 dinikolaev packagekitd[2066]: Failed to get cache filename for kernel-modules-core
Oct  7 23:11:58 dinikolaev packagekitd[2066]: Failed to get cache filename for kernel-devel
Oct  7 23:12:07 dinikolaev systemd[1]: dbus-1.1-0.fedoraproject.SetroubleshootPrivileged@0.service:
Deactivated successfully.
Oct  7 23:12:07 dinikolaev systemd[1]: setroubleshootd.service: Deactivated successfully.
Oct  7 23:14:32 dinikolaev systemd[1]: Starting Fingerprint Authentication Daemon...
Oct  7 23:14:32 dinikolaev systemd[1]: Started Fingerprint Authentication Daemon.
```

Рис. 3.13: Последние сообщения log-файла

Не смогли получить доступ к кэшу и сообщение о деактивации от `setroubleshootd.service`.

14. Попробуем 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/conf/httpd.conf` найдем строчку `Listen 80` и заменим её на `Listen 81` ([3.14]).

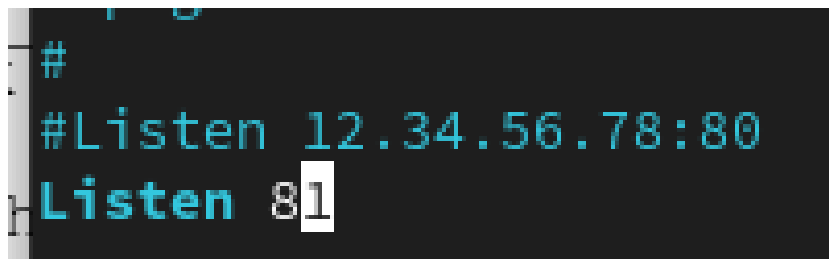


Рис. 3.14: Смена порта прослушивания с 80 на 81

15. Выполним перезапуск веб-сервера Apache ([3.15]).

```
[dinikolaev@dinikolaev ~]$ sudo systemctl restart httpd
[sudo] пароль для dinikolaev:
[dinikolaev@dinikolaev ~]$ sudo netstat -tlnp | grep httpd
tcp6      0      0  ::::81               :::*                   LISTEN     8814/httpd

[dinikolaev@dinikolaev ~]$ sudo ss -tlnp | grep httpd
LISTEN 0      511      *:81                  :::*                   users:((("httpd",pid=8821,fd=4),("httpd",pid=8820,fd=4),("httpd",pid=8816,fd=4),("httpd",pid=8814,fd=4))
[dinikolaev@dinikolaev ~]$ 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; preset: disabled)
   active: active (running) since Sun 2023-10-08 01:33:23 MSK; 52s ago
     docs: man:httpd.service(8)
   main pid: 8814 (httpd)
   status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0"
     tasks: 213 (limit: 4460)
    memory: 33.3M
       cpu: 67ms
   cgroup: /system.slice/httpd.service
           └─8814 /usr/sbin/httpd -DFOREGROUND
             └─8815 /usr/sbin/httpd -DFOREGROUND
               └─8816 /usr/sbin/httpd -DFOREGROUND
                 └─8820 /usr/sbin/httpd -DFOREGROUND
                   └─8821 /usr/sbin/httpd -DFOREGROUND

окт 08 01:33:23 dinikolaev systemd[1]: Starting The Apache HTTP Server...
окт 08 01:33:23 dinikolaev systemd[1]: Started The Apache HTTP Server.
окт 08 01:33:23 dinikolaev httpd[8814]: Server configured, listening on: port 81
lines 1-19/19 (END)
```

Рис. 3.15: Перезапуск веб-сервера Apache

Снова попытавшись получить доступ через веб-сервер, введя адрес в браузере увидим следующую картину ([3.16]).

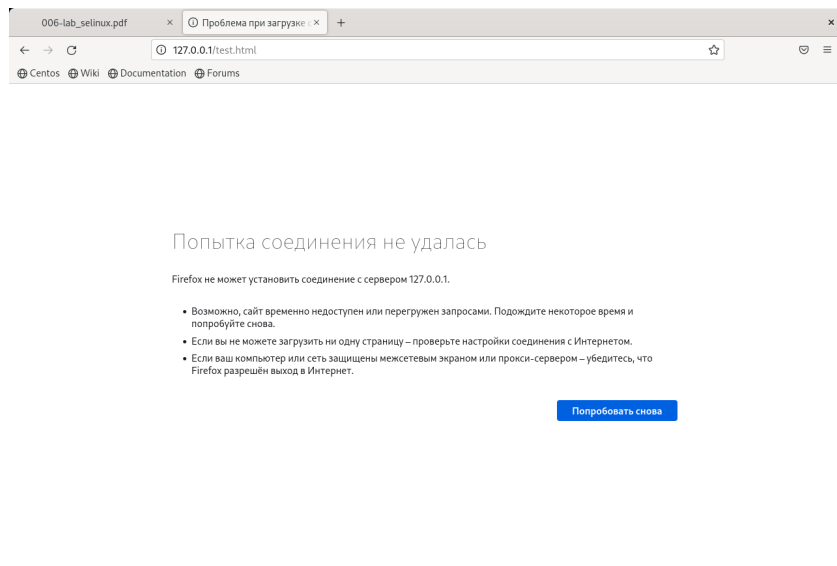


Рис. 3.16: Попытка подключения к веб-серверу через браузер

16. Проанализируем лог-файлы: `tail -n10 /var/log/messages`. Также посмотрим файлы `/var/log/httpd/error_log`, `/var/log/httpd/access_log` и `/var/log/audit/audit.log`/ Новые записи появились в файлах `/var/log/messages` и `/var/log/httpd/error_log`. ([3.17]).

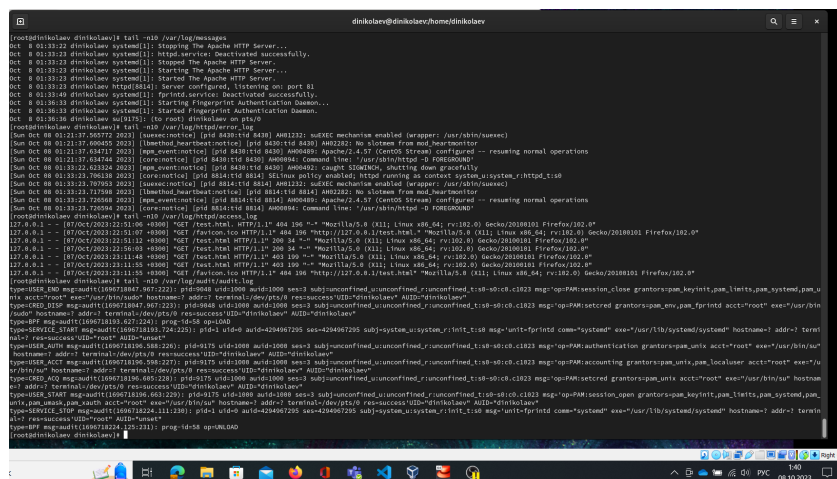


Рис. 3.17: Анализ сообщений лог-файлов

17. Выполним команду `semanage port -a -t http_port_t -p tcp 81`. После прове-

рим список портов командой `semanage port -l | grep http_port_t`. Убедимся, что порт 81 появился в списке ([3.18]).

```
[dinikolaev@dinikolaev ~]$ su
Пароль:
[root@dinikolaev dinikolaev]# semanage port -a -t http_port_t -p tcp 81
ValueError: Port tcp/81 already defined
[root@dinikolaev dinikolaev]# 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@dinikolaev dinikolaev]#
```

Рис. 3.18: Установка 81 порта и проверка списка всех портов

18. Попробуем запустить веб-сервер Apache ещё раз ([3.19]).

```
[root@dinikolaev dinikolaev]# systemctl restart httpd
[root@dinikolaev dinikolaev]# netstat -tlnp | grep httpd
tcp6      0      0 :::81          :::*           LISTEN     9325/httpd
[root@dinikolaev dinikolaev]# ss -tlnp | grep httpd
LISTEN 0      511      *:81           *:~             users:((("httpd",pid=9333,fd=4),("httpd",pid=9332,fd=4),("h
tcpd",pid=9330,fd=4),("httpd",pid=9325,fd=4))
[root@dinikolaev dinikolaev]# exit
exit
[dinikolaev@dinikolaev ~]$ 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; preset: disabled)
   Active: active (running) since Sun 2023-10-08 01:43:58 MSK; 39s ago
     Docs: man:httpd.service(8)
  Main PID: 9325 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0 B/sec"
    Tasks: 213 (limit: 4460)
   Memory: 23.5M
      CPU: 61ms
  CGroup: /system.slice/httpd.service
          └─9325 /usr/sbin/httpd -DFOREGROUND
            └─9326 /usr/sbin/httpd -DFOREGROUND
              └─9330 /usr/sbin/httpd -DFOREGROUND
                └─9332 /usr/sbin/httpd -DFOREGROUND
                  └─9333 /usr/sbin/httpd -DFOREGROUND

окт 08 01:43:58 dinikolaev systemd[1]: Starting The Apache HTTP Server...
окт 08 01:43:58 dinikolaev systemd[1]: Started The Apache HTTP Server.
окт 08 01:43:58 dinikolaev httpd[9325]: Server configured, listening on: port 81
[dinikolaev@dinikolaev ~]$
```

Рис. 3.19: Повторный запуск веб-сервера Apache

19. Вернем контекст `httpd_sys_content_t` к файлу `/var/www/html/test.html`: `chcon -t httpd_sys_content_t /var/www/html/test.html` ([3.20]). После этого попробуем получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1:81/test.html`. Увидим содержимое файла — слово «test» ([3.21]).

```
[dinikolaev@dinikolaev ~]$ su
Пароль:
[root@dinikolaev dinikolaev]# chcon -t httpd_sys_content_t /var/www/html/test.html
[root@dinikolaev dinikolaev]#
```

Рис. 3.20: Возвращение контекста `httpd_sys_content_t` нашему файлу

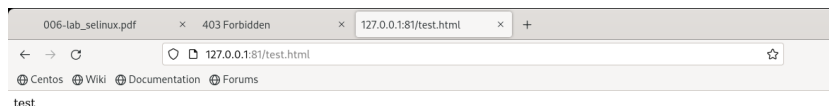


Рис. 3.21: Получение доступа к файлу через браузер

20. Исправим обратно конфигурационный файл apache, вернув Listen 80 ([3.22]).

```
#
#Listen 12.34.56.78:80
Listen 80
#
```

Рис. 3.22: Смена порта прослушивания с 81 на 80

21. Удалим привязку http_port_t к 81 порту: semanage port -d -t http_port_t -p tcp 81. Порт 81 не был удален, так как в начале 80 и 81 порты были настроены по умолчанию ([3.23]).

```
[root@dinikolaev dinikolaev]# vim /etc/httpd/conf/httpd.conf
[root@dinikolaev dinikolaev]# semanage port -d -t http_port_t -p tcp 81
ValueError: Port tcp/81 is defined in policy, cannot be deleted
[root@dinikolaev dinikolaev]# systemctl restart httpd
[root@dinikolaev dinikolaev]# netstat -tlnp | grep httpd
tcp6      0      0 :::80          :::*           LISTEN     10024/httpd
[root@dinikolaev dinikolaev]# ss -tlnp | grep httpd
LISTEN 0      511      *:80          *:80          users:((("httpd",pid=10031,fid=4),("httpd",pid=10030,fid=4),
("httpd",pid=10029,fid=4),("httpd",pid=10024,fid=4))
[root@dinikolaev dinikolaev]# semanage port -d -t http_port_t -p tcp 81
ValueError: Port tcp/81 is defined in policy, cannot be deleted
[root@dinikolaev dinikolaev]# 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@dinikolaev dinikolaev]#
```

Рис. 3.23: Попытка удаления привязки к 81 порту

22. Удалим файл /var/www/html/test.html: rm /var/www/html/test.html ([3.24]).

```
[dinikolaev@dinikolaev ~]$ sudo rm /var/www/html/test.html  
[sudo] пароль для dinikolaev:  
[dinikolaev@dinikolaev ~]$
```

Рис. 3.24: Удаление файла test.html

4 Выводы

В ходе выполнения лабораторной работы я развил навыки администрирования ОС Linux, получил первое практическое знакомство с технологией SELinux и проверил работу SELinux на практике совместно с веб-сервером Apache.

Список литературы

1. SELinux [Электронный ресурс]. URL: <https://habr.com/ru/companies/kingsewers/articles/209644/>.
2. Кулябов Д. С., Королькова А. В., Геворкян М. Н. Лабораторная работа №6 [Электронный ресурс]. RUDN, 2023. URL: https://esystem.rudn.ru/pluginfile.php/2090210/mod_resource/content/2/006-lab_selinux.pdf.