# Лабораторная работа № 6. Мандатное разграничение прав в Linux

## 6.1. Цели работы

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux[1].

Проверить работу SELinx на практике совместно с веб-сервером Apache.

## 6.2. Организация и описание лабораторного стенда

Для проведения указанной лабораторной работы на одно рабочее место требуется компьютер с установленной операционной системой Linux, поддерживающей технологию SELinux.

Предполагается использовать стандартный дистрибутив Linux CentOS с включённой политикой SELinux targeted и режимом enforcing. Для выполнения заданий требуется наличие учётной записи администратора (root) и учётной записи обычного пользователя. Постоянно работать от учётной записи root неправильно с точки зрения безопасности.

## 6.3. Подготовка лабораторного стенда и методические рекомендации

1. При подготовке стенда обратите внимание, что необходимая для работы и указанная выше политика targeted и режим enforcing используются в данном дистрибутиве по умолчанию, т.е. каких-то специальных настроек не требуется. При этом следует убедиться, что политика и режим включены, особенно когда работа будет проводиться повторно и велика вероятность изменений при предыдущем использовании системы.
2. При необходимости администратор должен разбираться в работе SELinux и уметь как исправить конфигурационный файл `/etc/selinux/config`, так и проверить используемый режим и политику.
3. Необходимо, чтобы был установлен веб-сервер Apache. При установке системы в конфигурации «рабочая станция» указанный пакет не ставится.
4. В конфигурационном файле `/etc/httpd/httpd.conf` необходимо задать параметр ServerName:

[1] При составлении работы использовались материалы [5; 6].

```
ServerName test.ru
```
чтобы при запуске веб-сервера не выдавались лишние сообщения об ошибках, не относящихся к лабораторной работе.

5. Также необходимо проследить, чтобы пакетный фильтр был отключён или в своей рабочей конфигурации позволял подключаться к 80-у и 81-у портам протокола tcp.
   Отключить фильтр можно командами
```
iptables -F
iptables -P INPUT ACCEPT iptables -P OUTPUT ACCEPT
```
   либо добавить разрешающие правила:
```
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
```
6. Обратите внимание, что данные правила не являются «точными» и рекомендуемыми на все случаи жизни, они лишь позволяют правильно организовать работу стенда.
7. В работе специально не делается акцент, каким браузером (или какой консольной программой) будет производиться подключение к веб-серверу. По желанию могут использоваться разные программы, такие как консольные `links`, `lynx`, `wget` и графические `konqueror`, `opera`, `firefox` или др.

## 6.4. Порядок выполнения работы

1. Войдите в систему с полученными учётными данными и убедитесь, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` и `sestatus`.
2. Обратитесь с помощью браузера к веб-серверу, запущенному на вашем компьютере, и убедитесь, что последний работает:
```
service httpd status
```
   или
```
/etc/rc.d/init.d/httpd status
```
   Если не работает, запустите его так же, но с параметром `start`.
3. Найдите веб-сервер Apache в списке процессов, определите его контекст безопасности и занесите эту информацию в отчёт. Например, можно использовать команду
```
ps auxZ | grep httpd
```
   или
```
ps -eZ | grep httpd
```
4. Посмотрите текущее состояние переключателей SELinux для Apache с помощью команды
```
sestatus -bigrep httpd
```
   Обратите внимание, что многие из них находятся в положении «off».

5. Посмотрите статистику по политике с помощью команды `seinfo`, также определите множество пользователей, ролей, типов.
6. Определите тип файлов и поддиректорий, находящихся в директории `/var/www`, с помощью команды
   ```
   ls -lZ /var/www
   ```
7. Определите тип файлов, находящихся в директории `/var/www/html`:
   ```
   ls -lZ /var/www/html
   ```
8. Определите круг пользователей, которым разрешено создание файлов в директории `/var/www/html`.
9. Создайте от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись в директорию) html-файл `/var/www/html/test.html` следующего содержания:
   ```
   <html>
   <body>test</body>
   </html>
   ```
10. Проверьте контекст созданного вами файла. Занесите в отчёт контекст, присваиваемый по умолчанию вновь созданным файлам в директории `/var/www/html`.
11. Обратитесь к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Убедитесь, что файл был успешно отображён.
12. Изучите справку `man httpd_selinux` и выясните, какие контексты файлов определены для `httpd`. Сопоставьте их с типом файла `test.html`. Проверить контекст файла можно командой `ls -Z`.
    ```
    ls -Z /var/www/html/test.html
    ```
    Рассмотрим полученный контекст детально. Обратите внимание, что так как по умолчанию пользователи CentOS являются свободными от типа (unconfined в переводе с англ. означает свободный), 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. (В директории `/proc` файлы, относящиеся к процессам, могут иметь роль system_r. Если активна политика MLS, то могут использоваться и другие роли, например, secadm_r. Данный случай мы рассматривать не будем, как и предназначение :s0).
    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. Измените 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:
    ```
    chcon -t samba_share_t /var/www/html/test.html
    ls -Z /var/www/html/test.html
    ```

После этого проверьте, что контекст поменялся.

14. Попробуйте ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Вы должны получить сообщение об ошибке:
```
Forbidden
You don't have permission to access /test.html on this server.
```
15. Проанализируйте ситуацию. Почему файл не был отображён, если права доступа позволяют читать этот файл любому пользователю?
```
ls -l /var/www/html/test.html
```
Просмотрите log-файлы веб-сервера Apache. Также просмотрите системный лог-файл:
```
tail /var/log/messages
```
Если в системе окажутся запущенными процессы setroubleshootd и audtd, то вы также сможете увидеть ошибки, аналогичные указанным выше, в файле `/var/log/audit/audit.log`. Проверьте это утверждение самостоятельно.
16. Попробуйте 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/httpd.conf` найдите строчку `Listen 80` и замените её на `Listen 81`.
17. Выполните перезапуск веб-сервера Apache. Произошёл сбой? Поясните почему?
18. Проанализируйте лог-файлы:
```
tail -nl /var/log/messages
```
Просмотрите файлы `/var/log/http/error_log`, `/var/log/http/access_log` и `/var/log/audit/audit.log` и выясните, в каких файлах появились записи.
19. Выполните команду
```
semanage port -a -t http_port_t -p tcp 81
```
После этого проверьте список портов командой
```
semanage port -l | grep http_port_t
```
Убедитесь, что порт 81 появился в списке.
20. Попробуйте запустить веб-сервер Apache ещё раз. Поняли ли вы, почему он сейчас запустился, а в предыдущем случае не смог?
21. Верните контекст httpd_sys_content__t к файлу /var/www/html/ test.html:
```
chcon -t httpd_sys_content_t /var/www/html/test.html
```
После этого попробуйте получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1:81/test.html`.
Вы должны увидеть содержимое файла — слово «test».
22. Исправьте обратно конфигурационный файл apache, вернув `Listen 80`.
23. Удалите привязку http_port_t к 81 порту:
```
semanage port -d -t http_port_t -p tcp 81
```
и проверьте, что порт 81 удалён.
24. Удалите файл `/var/www/html/test.html`:
```
rm /var/www/html/test.html
```

## 6.5. Содержание отчёта

Отчёт должен включать:

1. титульный лист;
2. формулировку цели работы;
3. описание процесса выполнения задания. Для каждого действия, производимого в командной строке, в отчёт следует включить:
   - краткое описание действия;
   - вводимая команда или команды;
   - результаты выполнения команд (снимок экрана);
4. выводы, согласованные с целью работы.