

Глава 9. Введение в SELinux

9.1. Методы управления доступом

Методы управления доступом

- В традиционной системе безопасности используется механизм discretionary access control — DAC
- SELinux может реализовывать другие модели безопасности в дополнение к DAC:
 - MAC
 - RBAC

1. В большинстве операционных систем имеются средства управления доступом, которые определяют, может ли определенный субъект (пользователь или программа) получить доступ к определенному объекту (ресурсу).
2. В системах UNIX® применяется разграничительный контроль доступа (discretionary access control, DAC). Этот метод позволяет ограничить доступ к объектам на основе групп, к которым они принадлежат.

Например, в GNU/Linux для каждого файла определены владелец, группа, а также указаны права доступа к этому файлу. Правами доступа определяется, кто может получить доступ к файлу, кто может открыть его для чтения, кто может внести в него изменения, кто может запустить этот файл на выполнение.

3. Такое разграничение прав доступа может привести к возникновению ряда проблем из-за того, что программа, в которой может быть обнаружена уязвимость, наследует все права доступа пользователя. Следовательно, она может выполнять действия с тем же уровнем привилегий, какой есть у пользователя (что нежелательно).
4. Вместо того чтобы определять ограничения подобным образом, более безопасно использовать принцип наименьшего уровня привилегий (principle of least privilege), согласно которому программы могут делать только то, что им необходимо для выполнения своих задач, и не более того.

5. Другой тип контроля называется принудительным управлением доступом (mandatory access control, MAC).

Например, если у вас есть программа, задача которой состоит в приеме запросов через сокеты, при этом ей не нужно иметь доступ к файловой системе, то такая программа будет иметь возможность только прослушивать определенный сокет и не будет иметь доступа к файловой системе. Таким образом, даже если в программе будет обнаружена уязвимость, то возможности доступа данной программы будут жестко ограничены.

6. Еще одним методом управления доступом является управление доступом на основе ролей (role-based access control, RBAC).
7. При использовании RBAC права доступа предоставляются на основе ролей, выдаваемых системой безопасности. Отличие концепции ролей от традиционных групп состоит в том, что группа представляет одного или нескольких пользователей, в то время как роль, хотя она также может быть применена к нескольким пользователям, представляет совокупность полномочий на выполнение определенных действий.
8. SELinux добавляет в операционную систему GNU/Linux поддержку как MAC, так и RBAC.

9.2. Основные понятия SELinux

Что такое SELinux

- Каждый файл, каталог или устройство это объект
- Каждый процесс — субъект
- У объектов и субъектов имеются метки
- Политика устанавливает правила взаимодействия объектов и субъектов на основе меток

1. При использовании SELinux, файлы, включая директории и устройства являются объектами. Процессы, такие как, выполнение команды пользователем или приложение Mozilla® Firefox®, являются субъектами.
2. Основное назначение архитектуры MAC - это возможность принудительного назначения административно-установленной политики безопасности над всеми процессами и файлами системы, при этом решение основывается на метках, содержащих множество значимой информации по безопасности. Когда механизм SELinux реализован, он переводит систему в состоянии достаточной защищенности и предоставляет критичную поддержку приложениям, защищая приложения от взлома или обхода безопасности.
3. MAC предоставляет строгое разделение приложений и позволяет безопасное исполнение не доверенных приложений. Обладая способностью ограничивать привилегии, связанные с исполнением процессов, MAC ограничивает рамки потенциальной угрозы, таким образом ограничивая взлом уязвимостей в приложениях и системных службах. MAC включает защиту информации от пользователей корректно авторизованных в системе с ограниченными правами также как и от авторизованных пользователей, которые неосознанно исполняют вредоносный код.

Что такое SELinux

- Пользователи Linux сопоставляются (маппируются) с пользователями SELinux
- Пользователи SELinux - это часть политики SELinux

4. В операционных системах Linux с запущенным SELinux, существуют пользователи Linux и пользователи SELinux. Пользователи SELinux - это часть политики SELinux. Пользователи Linux сопоставляются (маппируются) с пользователями SELinux. Для того, чтобы избежать путаницы, в данном руководстве используются два термина "пользователь Linux" и "пользователь SELinux" для различия двух разных понятий.

Модели управления доступом SELinux

- **Type Enforcement (TE):** Основной механизм ограничения, используемый в целевых политиках
- **Role-Based Access Control (RBAC):** в этой модели права доступа реализуются в качестве ролей
- **Multi-Level Security (MLS):** многоуровневая модель безопасности, в которой всем объектам системы присваивается определенный уровень доступа
- **Multi-Category Security(MCS):** Расширение MLS, используется в целевой политике для ограничения виртуальных машин и контейнеров через sVirt

5. В дополнение к DAC SELinux (Security Enhanced Linux) предлагает несколько вспомогательных моделей управления доступом:

- **Type Enforcement (TE):** Основной механизм ограничения, используемый в целевых политиках. Позволяет детально, на самом низком уровне управлять разрешениями. Самый гибкий, но и самый трудоемкий для системного администратора механизм.
- **Role-Based Access Control (RBAC):** в этой модели права доступа реализуются в качестве ролей. Ролью называется разрешения на выполнение определенных действий одним или несколькими элементами системы над другими частями системы. По-сути, RBAC является дальнейшим развитием TE.
- **Multi-Level Security (MLS):** многоуровневая модель безопасности, в которой всем объектам системы присваивается определенный уровень доступа. Разрешение или запрет доступа определяется только соотношением этих уровней.
- **Multi-Category Security(MCS):** Расширение MLS, используется в целевой политике для ограничения виртуальных машин и контейнеров через sVirt.

Что НЕ МОЖЕТ SELinux

- Не антивирус
- Не замена паролям, брандмауэрам, или разрешениям на доступ и т.д.
- Не система типа все-в-одном

6. SELinux не является:

- антивирусным программным обеспечением.
- заменой паролям, межсетевым экранам или другим системам безопасности.
- решением безопасности "всё в одном".

7. SELinux разработан, для усовершенствования существующих решений по безопасности. Даже с запущенным SELinux, необходимо использование практик по безопасности, таких как обновление программного обеспечения последними обновлениями, использование сложных паролей, межсетевых экранов и прочего

Терминология SELinux

- **Сущность (identity)** - этот термин схож с понятием "пользователь" в классической схеме доступа. Сущность может иметь такое же название, как и логин пользователя, но в отличие от логина, сущность не меняется после выполнения команды su.
 - **Домен (domain)** - это список того, что может делать отдельный процесс. Фактически домен - это действия, минимально необходимые одному процессу для выполнения его задачи.
 - **Роль (role)** - это список доменов, которые могут быть использованы. Если некоего домена нет в списке, то роль не может выполнить действия из этого домена.
 - **Тип (type)** - это набор действий (операция) применительно к объекту. Важно понять отличие от домена. Домен относится к процессам, а тип - к объектам.
8. **Сущность (identity)** - этот термин схож с понятием "пользователь" в классической схеме доступа. Сущность может иметь такое же название, как и логин пользователя, но в отличие от логина, сущность не меняется после выполнения команды su. Если провести аналогию, то сущность - это конкретный человек, Вася Пупкин, Петя Смирнов и т.д.
 9. **Домен (domain)** - это список того, что может делать отдельный процесс. Фактически домен - это действия, минимально необходимые одному процессу для выполнения его задачи. По аналогии из реальной жизни, доменом можно назвать набор действий для совершения какой-либо операции.
 10. **Роль (role)** - это список доменов, которые могут быть использованы. Если некоего домена нет в списке, то роль не может выполнить действия из этого домена. В данном случае можно провести аналогию с должностью. То есть роль - это фактически должность (или должностная инструкция), которая может выполнять определённые наборы операций, или, в понятии SELinux, домены.
 11. **Тип (type)** - это набор действий (операция) применительно к объекту. Важно понять отличие от домена. Домен относится к процессам, а тип - к объектам, таким как файлы, каталоги, пайпы(pipes), сокеты и т.д.

Терминология SELinux

- **Уровень (level)** - состоит из чувствительности и категории. Используется в системах MLS/MCS.
 - **Контекст безопасности (context)** - это набор всех атрибутов, связанных с объектами и субъектами. Контекст безопасности для субъектов (процессов) состоит из сущности, роли, домена, чувствительности и категории `user:role:type:sensitivity:category`.
 - **Переход (transition)** - это смена контекста безопасности. Есть два основных типа переходов:
 - Переход домена процесса - процесс меняет контекст;
 - Переход типа файла - создание файлов в определённых подкаталогах.
 - **Политика (policy)** - это набор правил, контролирующих взаимодействие ролей, доменов, типов и т.д.
12. **Уровень (level)** — это атрибут многоуровневого управления доступом MLS и MCS. Пространство MLS - это пара уровней, записанных в виде `lowlevel-highlevel`, если уровни в данной паре отличаются или, если не отличаются, то просто `lowlevel`. То есть (`s0-s0` то же самое, что и `s0`). Если дополнительно определены категории, то уровень записывается как `sensitivity:category-set`. Если категории не определены, то запись выглядит как `sensitivity`.
13. **Контекст безопасности (context)** - это набор всех атрибутов, связанных с объектами и субъектами. Контекст безопасности для субъектов (процессов) состоит из сущности, роли, домена, чувствительности и категории. Обычно используется только сущность-роль-домен(или тип), а, например, целевая политика от Fedora использует только домены и типы.
14. **Переход (transition)** - это смена контекста безопасности. Есть два основных типа переходов:
1. Переход домена процесса - процесс меняет контекст; Например, запускается из-под пользователя некий демон. Selinux, на основе метки исполняемого файла, меняет его контекст.
 2. Переход типа файла - создание файлов в определённых подкаталогах. Например, пользователь создаёт html-страничку в каталоге WEB-сервера. Чтобы WEB-сервер получил доступ к этой страничке, необходимо сменить контекст безопасности файла (WEB-сервер не имеет доступа к контексту пользователя).

- 15. **Политика (policy)** - это набор правил, контролирующих взаимодействие ролей, доменов, типов и т.д.
- 9. Политики работают на уровне системных вызовов и обрабатываются ядром, но можно реализовать и на уровне приложения. Политики описываются при помощи специального языка описания правил доступа.

Политики SELinux

- Целевая (targeted)
- Минимальная (minimum)
- Многоуровневая (MLS)
- Строгая (strict)

10. В настоящий момент уже разработано несколько готовых политик безопасности, которые можно использовать по умолчанию на серверах и на домашних компьютерах. Всё, что требуется от системного администратора - выбрать используемую политику и перезагрузить компьютер с включённым SELinux.

11. В среднем, политика безопасности SELinux для всей системы содержит более ста тысяч правил, так что её создание и отладка занимает значительное время.

12. Наиболее распространены следующие три политики:

1. **Целевая (targeted).** Эта политика разработана компанией Red Hat и является наиболее используемой;
2. **Минимальная (minimum).** Является модификацией целевой политики, в которой только выбранные процессы защищаются.
3. **Многоуровневая (MLS).** Позволяет обеспечивать уровни безопасности и может использоваться госструктурами для хранения информации различных уровней секретности;
4. **Строгая (strict).** Этот вариант политики подразумевает правило "Что не разрешено, то запрещено".

9.3. Базовая настройка SELinux

Режимы работы SELinux

- Три режима работы:
 1. Enforcing
 2. Permissive
 3. Disabled
 - Команда `sestatus` — показывает состояние SELinux
 - Команда `getenforce` — режим работы
 - Команда `setenforce` — переключение режима работы
1. Включение или выключение SELinux производится в файле: `/etc/selinux/config`
 2. Параметр `SELINUX=` - определяет режим работы SELinux:
 1. `enforcing` - SELinux включен и применяет политики.
 2. `permissive` - SELinux включен, но не ограничивает пользователей.
 3. `disabled` - SELinux выключен/
 3. Параметр `SELINUXTYPE=` определяет политику SELinux.
 4. После переключения режимов работы необходимо перезагрузить компьютер.
 5. Основные утилиты:
 1. `secon` – показывает контекст пользователя.
 2. `semanage` - Утилита управления политикой SELinux.
 3. `audit2allow` - сканирует журналы аудита для того, чтобы сгенерировать политики SELinux.
 4. `audit2why` - Эта утилита обрабатывает сообщения аудита SELinux, принятые со стандартного ввода, и сообщает, какой компонент политики вызвал каждый из запретов.
 5. `chcat` - изменяет категорию безопасности SELinux для файла.
 6. `fixfiles` - восстанавливает контекст безопасности SELinux для файла.

Глава 9. Введение в SELinux

7. `genhomedircon` - создает спецификации с контекстом SELinux для файлов домашних директорий пользователей.
 8. `load_policy` - утилита, использующаяся для загрузки/замены политики в ядре.
 9. `run_init` - запускает скрипт `init` в правильном контексте SELinux.
 10. `open_init_pty` - запустить программу в псевдо-терминал. Используется в `run_init` для того, чтобы запустить программу после установки правильного контекста.
 11. `restorecon` - восстанавливает заданный по умолчанию контекст безопасности SELinux для файла(файлов).
 12. `restorecond` - демон, который отслеживает создание файлов, и выставляет для них заданный по умолчанию контекст SELinux.
 13. `semodule` - утилита управления пакетами модулей политики SELinux.
 14. `semodule_deps` - показывает зависимости между пакетами модулей политики SELinux.
 15. `semodule_expand` - утилита добавления пакета модуля политики SELinux.
 16. `semodule_link` - это утилита разработчика, предназначенная для связывания нескольких пакетов модулей политик в единый пакет модулей политик.
 17. `semodule_package` - это утилита используемая для создания пакета модуля политики SELinux из бинарного модуля политики и других источников данных, таких как файл контекстов.
 18. `sestatus` - та утилита предназначена для просмотра статуса системы, использующей SELinux.
 19. `setfiles` - устанавливает контекст безопасности SELinux для файла.
 20. `setsebool` - устанавливает значение переключателя (boolean) SELinux
6. Имеется также удобный графический интерфейс этих утилит - пакет `policycoreutils-gui`.

Ограниченные и неограниченные процессы

- В целевой политике все процессы делятся на две категории
 - Ограниченные т. е. защищаются (ограничиваются) SELinux
 - Неограниченные те что используют DAC механизм в своей работе

7. Когда используется целевая политика targeted, процессы, которые являются целевыми, запускаются в ограниченном домене, остальные процессы запускаются в неограниченном домене. Например, по умолчанию пользователи, прошедшие авторизацию, работают в домене `unconfined_t` и системные процессы запущенные `init`-ом запускаются в домене `initrc_t` - оба домена неограниченные.
8. Неограниченные домены (так же, как и ограниченные) - это субъекты для операций выполнения и записи в память. По умолчанию, субъекты запущенные в неограниченном домене не могут выделить память для записи и запустить ее. Это уменьшает степень угрозы атаки переполнения буфера `buffer overflow attacks`. Эти проверки памяти отключаются установкой Булевых переключателей, что позволяет изменять политику SELinux "на ходу". Настройка Булевых значений рассматривается позже.
9. Почти каждая сетевая служба ограничена. Также большинство процессов, которые запускаются в Linux с привилегиями пользователя `root` и выполняют задачи для пользователей, такие как приложение `passwd`, ограничены. Когда процесс ограничен, он запускается в своём собственном домене, например процесс `httpd` запускается в домене `httpd_t`. Если ограниченный процесс скомпрометирован атакующим, в зависимости от конфигурации SELinux, доступ атакующего к ресурсам и вред, который он может нанести ограничен.
10. Неограниченные (`unconfined`) процессы выполняются в неограниченных (`unconfined`) доменах, программы запускаемые `init` выполняются в неограниченном `unconfined initrc_t` домене, неограниченные процессы ядра запускаются в домене `kernel_t`. Для неограниченных процессов правила политики SELinux также применяются, но правила политики существуют для разрешения практически всех доступов для процессов, запущенных в неограниченных доменах. Процессы запущенные

Глава 9. Введение в SELinux

в неограниченных доменах откатываются к использованию только правил DAC. Если неограниченный процесс скомпрометирован, SELinux не ограничивает атакующего от получения доступа к системным ресурсам и информации, но, конечно, правила DAC всё равно используются. SELinux это улучшение механизмов безопасности над DAC, но SELinux не заменяет его.

Определение контекста

- Опция `-Z` позволяет определить контекст объекта или субъекта
 - `- ls -Z`
 - `- ps -Z`
- В целевых политиках для предоставления доступа к ресурсам домен субъекта должен иметь права доступа к типу объекта
- Правила доступа описываются в политике
- Для стандартных сервисов уже имеются политики, разработанные майтейнерами дистрибутивов

11. В целевых (targeted) политиках предоставление доступа основано на анализе меток. Политика проверяет, может ли домен процесса (субъекта) получить доступ к ресурсу (объекту). SELinux перехватывает системные вызовы и разрешает доступ, если политика это позволяет.

12. Информация о метках находится в контексте.

13. Основная опция получения информации о контекстах `-Z`.

Пример: Мы проверим контекст файлов, которые использует демон apache (httpd). И контекст процесса веб сервера. В политиках имеется разрешение для домена `httpd_t` получать доступ к типам `httpd_sys_content_t`. Чтобы процесс получил домен `httpd_t` программа имеет тип `httpd_exec_t`.

```
[root@sl0 devel]# ls -Z /var/www/html/index.html
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0
/var/www/html/index.html

[root@sl0 devel]# ls -Z /usr/sbin/httpd
-rwxr-xr-x. root root system_u:object_r:httpd_exec_t:s0
/usr/sbin/httpd

[root@sl0 devel]# ps -eZ | grep httpd | head -1
system_u:system_r:httpd_t:s0      1100 ?          00:00:00 httpd
```


14. В примере выше мы не видим самой политики. Политика представляет из себя бинарный файл, который во время старта системы загружается в ядро.

Булевы значения (переключатели)

- Политика может предусматривать настройку разрешения или запрета на выполнение каких-либо особых действий
- Такие разрешения могут находиться в двух состояниях да или нет, поэтому называются булевыми
- Команда `semanage boolean -l` или `getsebool -a` выводят список всех булевых значений
- Команде `setsebool` устанавливает нужное значение

15. Переключатели позволяют изменять части политики SELinux во время работы (без перезапуска и остановки), не обладая глубоким пониманием создания политики SELinux. Это позволяет вносить изменения, такие как: разрешение доступа службам к файловым системам NFS, без перезагрузки или recompilation политики SELinux.

16. Для получения списка переключателей, объяснения, за что отвечает каждый переключатель, включен или выключен, необходимо выполнить команду `semanage boolean -l` от имени пользователя root.

Пример:

```
[root@sl0 devel]# semanage boolean -l | egrep '(SELi|
httpd.*connect_db) '
```

SELinux boolean	State	Default	Description
httpd_can_network_connect_db	(off , off)	off	Allow HTTPD scripts and modules to connect to databases over the network.

17. Команда `getsebool -a` выводит список переключателей, показывает выключены они или нет, но не даёт описания, за что они отвечают.

Пример:

```
[root@sl0 devel]# getsebool -a | grep httpd.*connect_db
httpd_can_network_connect_db --> off
```

18. Для получения статуса одного конкретного Булева значения (переключателя) `boolean-`

Глава 9. Введение в SELinux

name используется команда `getsebool boolean-name`

Пример:

```
[root@sl0 devel]# getsebool httpd_can_network_connect_db
httpd_can_network_connect_db --> off
```

19. Команда `setsebool boolean-name x` переводит переключатели в состояние включено или выключено, где `boolean-name` - название переключателя, а `x` - `on` для включения или `off` для выключения.

Пример:

```
[root@sl0 devel]# setsebool httpd_can_network_connect_db on
[root@sl0 devel]# getsebool httpd_can_network_connect_db
httpd_can_network_connect_db --> on
```

Изменения контекста файлов

- Для предоставления доступа к файлам необходимо иметь правильный контекст
- Временное назначение контекста выполняется командой `chcon`
- Восстановление контекста — `restorecon`
- Постоянный контекст управляется командой `semanage fcontext`

20. Чтобы некоторый ограниченный процесс получил доступ к файлу, последний, в свою очередь, должен иметь нужный тип.

21. Команда `chcon` устанавливает временный контекст.

Пример: Создаем собственный ресурс и получаем запрет доступа. После установки правильного контекста доступ появляется.

```
[root@sl0 myweb]# pwd
/myweb
[root@sl0 myweb]# ls -Z
-rw-r--r--. root root unconfined_u:object_r:default_t:s0
index.html
[root@sl0 myweb]# cat /etc/httpd/conf.d/mypage.conf
Alias /mypage /myweb
<Directory /myweb>
    AllowOverride None
    Require all granted
</Directory>

[root@sl0 myweb]# wget http://127.0.0.1/mypage/index.html -O
/dev/null
--2017-08-01 21:01:26--  http://127.0.0.1/mypage/index.html
```

Глава 9. Введение в SELinux

```
Connecting to 127.0.0.1:80... connected.
```

```
HTTP request sent, awaiting response... 403 Forbidden
```

```
2017-08-01 21:01:26 ERROR 403: Forbidden.
```

```
[root@sl0 myweb]# chcon -R -t httpd_sys_content_t /myweb
```

```
[root@sl0 myweb]# wget http://127.0.0.1/mypage/index.html -O /dev/null
```

```
--2017-08-01 21:03:13-- http://127.0.0.1/mypage/index.html
```

```
Connecting to 127.0.0.1:80... connected.
```

```
HTTP request sent, awaiting response... 200 OK
```

```
Length: 115877 (113K) [text/html]
```

```
Saving to: '/dev/null'
```

```
100%[=====>] 115,877 --.-K/s  
in 0s
```

```
2017-08-01 21:03:13 (609 MB/s) - '/dev/null' saved [115877/115877]
```

22. Команда `restorecon` восстанавливает контекст по умолчанию.

Пример: Восстановление контекста приводит к запрету доступа.

```
[root@sl0 myweb]# restorecon -R /myweb
```

```
[root@sl0 myweb]# wget http://127.0.0.1/mypage/index.html -O /dev/null
```

```
--2017-08-01 21:07:12-- http://127.0.0.1/mypage/index.html
```

```
Connecting to 127.0.0.1:80... connected.
```

```
HTTP request sent, awaiting response... 403 Forbidden
```

```
2017-08-01 21:07:12 ERROR 403: Forbidden.
```

```
[root@sl0 myweb]# ls -Z
```

```
-rw-r--r--. root root unconfined_u:object_r:default_t:s0  
index.html
```

23. Команда `semanage fcontext` изменяет контекст SELinux для файлов. При использовании целевой политики `targeted`, изменения вносимые данной командой, добавляются в файл `/etc/selinux/targeted/contexts/files/file_contexts`, если изменения вносятся для существующих файлов, то они добавляются в файл `file_contexts`, или

добавляются файл `file_contexts.local` для новых файлов и каталогов, например при создании каталога `/web/.setfiles`, использующаяся при маркировке файловой системы и `restorecon`, использующаяся для восстановления контекста SELinux по умолчанию, читают эти файлы. Это значит, что изменения вносимые командой `semanage fcontext` постоянно, даже если файловая система будет перемаркирована. Политика SELinux контролирует возможность пользователей изменять контекст файлов.

24. Для внесения изменений в контекст SELinux изменений, которые сохранятся при перемаркировании файловой системы надо:

1. Выполнить команду `semanage fcontext -a options file-name|directory-name` Помните, что необходимо использовать полные пути к файлам и каталогам.
2. Выполнить команду `restorecon -v file-name|directory-name` для применения изменений контекста.

Пример: Применение контекста к каталогу.

```
[root@sl0 myweb]# ls -dZ /myweb
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 /myweb
[root@sl0 myweb]# ls -Z /myweb
-rw-r--r--. root root unconfined_u:object_r:default_t:s0
index.html
[root@sl0 myweb]# semanage fcontext -a -t httpd_sys_content_t
/myweb
[root@sl0 myweb]# ls -Z /myweb
-rw-r--r--. root root unconfined_u:object_r:default_t:s0
index.html
[root@sl0 myweb]# ls -dZ /myweb
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 /myweb
[root@sl0 myweb]# restorecon -Rv /myweb/
restorecon reset /myweb context
unconfined_u:object_r:default_t:s0-
>unconfined_u:object_r:httpd_sys_content_t:s0
[root@sl0 myweb]# ls -dZ /myweb
drwxr-xr-x. root root unconfined_u:object_r:httpd_sys_content_t:s0
/myweb
[root@sl0 myweb]# ls -Z /myweb
-rw-r--r--. root root unconfined_u:object_r:default_t:s0
index.html
[root@sl0 myweb]# cat
/etc/selinux/targeted/contexts/files/file_contexts.local
# This file is auto-generated by libsemanage
```

Глава 9. Введение в SELinux

```
# Do not edit directly.
```

```
/myweb      system_u:object_r:httpd_sys_content_t:s0
[root@sl0 myweb]# semanage fcontext -a -t httpd_sys_content_t
'/myweb(/.*)?'
[root@sl0 myweb]# restorecon -Rv /myweb/
restorecon reset /myweb/index.html context
unconfined_u:object_r:default_t:s0-
>unconfined_u:object_r:httpd_sys_content_t:s0
[root@sl0 myweb]# cat
/etc/selinux/targeted/contexts/files/file_contexts.local
# This file is auto-generated by libsemanage
# Do not edit directly.

/myweb      system_u:object_r:httpd_sys_content_t:s0
/myweb(/.*)? system_u:object_r:httpd_sys_content_t:s0
[root@sl0 myweb]# ls -Z /myweb/
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0
index.html
[root@sl0 myweb]# wget http://127.0.0.1/mypage/index.html -O
/dev/null
--2017-08-01 21:25:44--  http://127.0.0.1/mypage/index.html
Connecting to 127.0.0.1:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 115877 (113K) [text/html]
Saving to: '/dev/null'

100%[=====>] 115,877      --.-K/s
in 0s

2017-08-01 21:25:44 (1.33 GB/s) - '/dev/null' saved
[115877/115877]
```

25. Удаление перманентной маркировки файлов производится командой `semanage fcontext -d options file-name|directory-name`

Пример: Удаление лишнего правила маркировки.

```
[root@sl0 myweb]# semanage fcontext -d -t httpd_sys_content_t
/myweb
```



```
[root@sl0 myweb]# cat
/etc/selinux/targeted/contexts/files/file_contexts.local
# This file is auto-generated by libsemanage
# Do not edit directly.

/myweb(/.*)?      system_u:object_r:httpd_sys_content_t:s0
[root@sl0 myweb]# restorecon -Rv /myweb
[root@sl0 myweb]# ls -Zd /myweb
drwxr-xr-x. root root unconfined_u:object_r:httpd_sys_content_t:s0
/myweb
```

9.3.1.1. Задание.

1. Проверьте состояние SELinux. Если оно выключено, то включите его или переведите в режим enforcing.
2. Установите веб-сервер apache и проанализируйте как промаркированы файлы этой службы.
3. Создайте файл /var/www/html/index.html и проверьте какой контекст установлен на этот файл. Убедитесь, что веб-сервер показывает стартовую страницу.
4. Создайте алиас для каталога /mywebpage, поместите в этот каталог файл index.html и проверьте доступна ли эта страница. На данном этапе страница не должна быть доступна.
5. Сделайте временную перемаркировку файлов для веб-сервера и проверьте доступ к странице алиаса.
6. Восстановите маркировку по умолчанию и создайте постоянное правило маркировки. Примените эти правила к каталогу /mywebpage. Убедитесь, что правила маркировки отработали правильно и доступ к алиасу восстановился.
7. Разрешите сетевое подключение веб-сервера к базам данных.