На данной лекции вы услышали про основные механизмы безопасности в Linux. В Linux защитные механизмы включают определение ролей и привилегий для пользователей. Когда пользователь пытается получить доступ к файлу, ядро операционной системы проверяет соответствие его роли и запрашиваемых привилегий. Если у пользователя недостаточно прав, ему отказывается доступ

Однако важно отметить, что на лекции не сказали ни слова о SELinux — мощному инструменту для повышения безопасности Linux. SELinux позволяет еще более детально определять права доступа к файлам и ресурсам. В моем докладе мы и сфокусируемся на нем.

**Введение**

SELinux (SELinux) — это система принудительного контроля доступа, реализованная на уровне ядра.

Была разработана для того, чтобы исправить недостатки традиционной системы безопасности UNIX (DAC)

Обычно необходимо его устанавливать самостоятельно, однако в нескольких дистрибутивах SELinux устанавливается "из коробки" - это RHEL, CentOS, Fedora.

SELinux работает "после" DAC (хотя и работает независимо от него). То есть операции, запрещенные в DAC не могут быть разрешены в SELinux.

**Основные термины**

**Домен**— список действий, которые может выполнять процесс. Обычно в качестве домена определяется минимально-возможный набор действий, при помощи которых процесс способен функционировать.

**Роль**— список доменов, которые могут быть применены.

**Тип**— набор действий, которые допустимы по отношения к объекту. Тип отличается от домена тем, что он может применяться к каталогам и файлам (т.е. для данных), в то время как домен применяется к процессам.

**Контекст безопасности** — все атрибуты SELinux — роли, типы и домены.

**Субъект** – программа, процесс

**Объект** – файлы данных

**Решение проблем традиционной модели безопасности.**

SELinux следует модели минимально необходимых привилегий для каждого сервиса, пользователя и программы. По умолчанию установлен «запретительный режим», когда каждый элемент системы имеет только те права, которые жизненно необходимы ему для функционирования. Если же пользователь, программа или сервис пытаются получить доступ к ресурсу, который не является необходимым в данный момент, то им будет просто отказано в доступе, а такая попытка будет зарегистрирована в журнале.



SELinux реализована на уровне ядра, так что прикладные приложения могут совсем ничего не знать о версии этой системы принудительного контроля доступа, особенностях её работы и т.д. В случае грамотной настройки, SELinux никак не повлияет на функционирование сторонних программ и сервисов. Однако, если приложение, к которому попытаются получить доступ, способно перехватывать сообщения об ошибках SELinux, удобство пользования таки приложением существенно возрастает. Ведь в случае попытки доступа к защищенному ресурсу или файлу, SELinux передает в основное приложение ошибку из семейства «access denied».

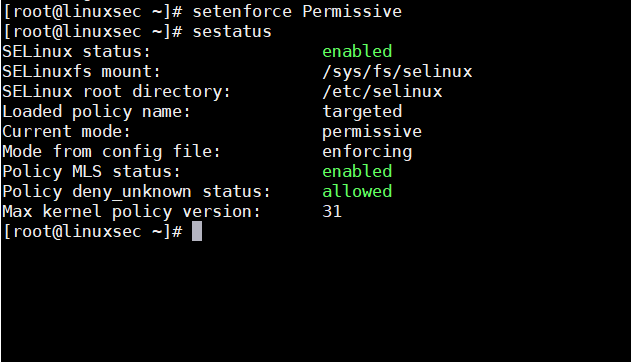
**Режимы работы SELinux**

SELinux имеет три основных режим работы, при этом по умолчанию установлен режим Enforcing. Это довольно жесткий режим, и в случае необходимости он может быть изменен на более удобный для конечного пользователя.

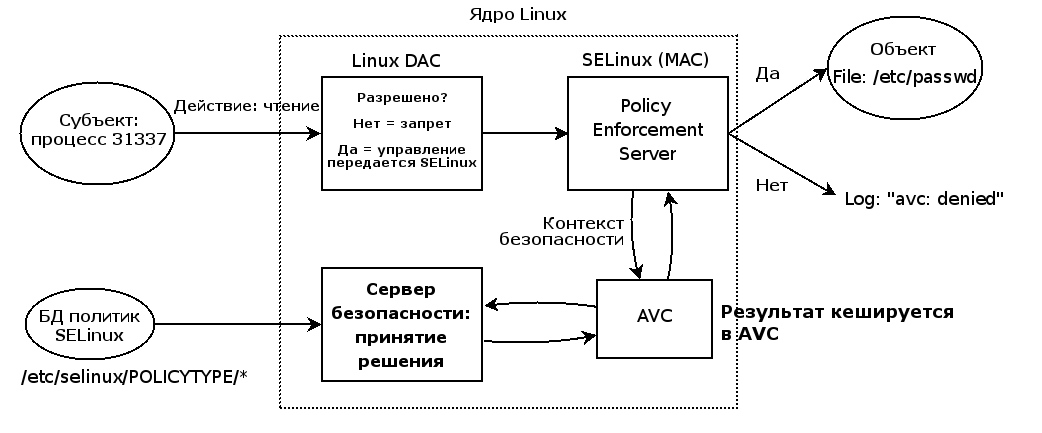
**Enforcing**: Режим по-умолчанию. При выборе этого режима все действия, которые каким-то образом нарушают текущую политику безопасности, будут блокироваться, а попытка нарушения будет зафиксирована в журнале.

**Permissive**: информация о всех действиях, которые нарушают текущую политику безопасности, будут зафиксированы в журнале, но сами действия не будут заблокированы.

**Disabled**: Полное отключение системы принудительного контроля доступа.



**Политики SELinux**



SELinux предоставляет высокий уровень контроля доступа и безопасности в операционных системах Linux. Режим работы "Enforcing" автоматически блокирует все действия, кроме разрешенных, обеспечивая максимальную информационную безопасность. Однако, настройка SELinux в этом режиме может быть сложной и привести к проблемам для конечных пользователей.

Например, пользователи столкнутся с ограничением доступа, если захотят использовать систему хоть сколько-нибудь нетривиальным образом.

Администраторы тратят дополнительное время на конфигурации SELinux для работы с различными приложениями.

Для решения этих проблем предлагаются различные подходы. В частности, заранее сконфигурированные целевые политики для ключевых приложений, таких как веб-сервер (httpd), DNS-сервер (named), DHCP-сервер (dhcpd), сервер баз данных MySQL (mysqld) позволяют добиться приемлемого уровня безопасности, минимизируя проблемы для пользователей.

Помимо этого, SELinux поддерживает два типа политик: targeted и strict .

Политика **targeted** - все процессы не внесенные в специальные ограниченные домены, работают в неограниченном домене unconfined\_t. Домен unconfined\_t представляет собой менее ограниченное окружение, где SELinux не применяет строгих политик безопасности

Политика **strict** - все процессы работают в специальных ограниченных доменах, в неограниченном домене unconfined\_t никто не работает.

**Отличие политики безопасности SELinux от DAC**

В традиционной системе DAC, доступ основан на правах владельца файла или группы. Например, если у файла установлены права чтения для всех, любой пользователь может читать файл, независимо от контекста запроса.

SELinux вводит дополнительный уровень абстракции и безопасности. Каждый процесс и объект в системе обладает контекстом безопасности.

**Синтаксис назначения контекста безопасности:**

Контекст безопасности определяет тип объекта и его свойства.

chcon -t <type> <object>

Пример назначения контекста безопасности:

chcon -t httpd\_sys\_content\_t index.html

**Синтаксис разрешения политики безопасности SELinux:**

allow <subject> <object> : <class> { permissions };

<subject>: субъект (пользователь, роль или процесс)

<object>: объект (файл, директория и т.д.)

<class>: класс объекта (file, dir, и так далее)

{ permissions }: разрешенные действия (read, write, execute и т.д.)

Пример разрешения политики безопасности SELinux:

Правило SELinux позволяет веб-серверу читать файлы с определенным контекстом безопасности.

allow httpd\_t httpd\_sys\_content\_t : file { read };

В данном контексте разрешается процессу с типом httpd\_t читать файлы с типом httpd\_sys\_content\_t.

**Почему httpd не сможет открыть файлы, которые ему не назначены:**

В SELinux, доступ к ресурсам определяется не только правами владельца, но и контекстом безопасности. Если файл не имеет соответствующего контекста, то даже если у пользователя httpd\_t есть права на чтение, SELinux может заблокировать доступ.

Это создает дополнительный уровень гибкости и безопасности, позволяя более точно определять, какие процессы могут взаимодействовать с какими ресурсами, что существенно повышает безопасность системы.