SELinux (Security-Enhaced Linux)

Лапшин Никита Б01-814

December 5, 2021

1 Предисловие

В этой статье будут рассмотрен SELinux и его возможности для усиления защиты конфиденциальности данных. Эта статья является обзорной, поэтому здесь не будут рассмотрены подробности, связанные с приницпами работы SELinux, лишь основы его работы.

2 Общее описание

SELinux это система с принудительным контролем доступа, была разработана в 2000 году в американском университете NSA. Изначально являлась серией патчей на уже имеющееся ядро Linux-а, позже была добавлена в сам Linux. Впервые появилась на дистрибутиве Centos 4 компании Red Hat, дальнейшей разработкой и улучшениями также занимался Red Hat. Сейчас в составе некоторых дистрибутивов есть заранее подготовленные правила (политики) для работы SELinux.

SELinux позволяет контролировать ресурсы в операционной системе такие как например файлы. В стандартном Linux доступ к файлу описывается в виде "-rwxrwxrwx", где первые три бита "rwx" описывают доступ процесса, создавшего файл, вторые три – процессов имеющих тот же групповой идентификатор процесса, что и у создателя, третьи три – для всех остальных процессов. В SELinux правила доступа к файлу могут быть расширены и заданы при загрузке операционной системы.

Однако стоит уточнить, что SELinux не является антивирусным программным обеспечениям, так же как и не является заменой файерволу и паролям. Он лишь усиливает уже имеющиеся решения для защиты и не может их заменить самостоятельно.

3 Политики (policies) SELinux

SELinux является модулем ядра Linux. В ядре же SELinux использует правила, которые были заданы администратором при загрузке операционной системы. Правила же могут определять доступ пользователей и процессов, созданных этими пользователями, к различным файлам, устройствам и даже сети. Такие правила помогают избежать случайных или злонамеренных действий со стороны пользователя. Любое действие проверяется по правилам, и в случае запрета операция блокируется и возвращает ошибку.

Политики работаю на уровне системных вызовов, то есть применяются самим ядром, но также можно реализовать политики и на пользовательском уровне. SELinux не заменяет классическую модель безопасности Linux, проверки выполняемые SELinux идут после классических. Таким образом невозможно дать доступ к объекту субъекту, который с точки зрения Linux не имеет к нему доступа.

Примером того как используются политики может послужить сервер. Так разработчик точно знает какие файлы серверу нужны для корректной работы, он может указать в политике эти файлы, как единственные доступные серверу. Таким образом сервер не сможет обращаться к другим файлам и не сможет нанести вред остальной системе в случае его взлома злоумышленниками.

Необязательно, что пользователь и программы, которые были запущены пользователем, будут иметь одинаковые права доступа, что также может быть указано в правилах. Пример работы такого правила: shell в котором работает пользователь имеют доступ к его домашнему каталогу,

тогда как запущенный пользователем мессенджер не будет иметь доступ к частям домашнего каталога, как например к директории содержащей ssh ключи пользователя.

Также стоит упомянуть, что существует кэш решений относительно действий процессов. Это помогает существенно ускорить повторный ответ на действия процесса, в документации этот кэш называется AVC (acess vector cache).

4 Контекст (context) в SELinux

Для расширения возможностей в SELinux введено такое понятие как контекст.Процессы и файлы а SELinux имеют контекст, который помогает составлять правила в SELinux. Описание контекста выглядит следующим образом: SELinux user:role:type:level. Перед дальнейшим рассмотрением дадим определение всем полям контекста, а также важным понятиям домена и точки входа в домен.

- Домен (domain) – определяет набор действий доступный процессу, обычно это минимальный набор действий, необходимый для работы этого процесса.
- Пользователь SELinux (SELinux user) – используются в SELinux для определения прав доступа пользователей Linux. Каждому пользователю в Linux определяется пользователь в SELinux, что также накладывает все ограничения определенные для пользователя в SELinux.
- Роль (role) – атрибут ролевой системы контроля доступа (Role-Based Access Control) использующейся в SELinux. Пользователю SELinux выдаются “роли”, которые определяют доступ пользователя к объектам операционной системы.
- Тип (type) – с помощью этого атрибута система может определять доступ субъекта к объекту, например, процесса к файлу.
- Точка входа (entrypoint) в домен – тип файла, необходимый для входа процесса в домен и получения всех прав доступа этого домена.
- Уровень (level) – атрибут представляется в виде s0:c1, где s – чувствительность (sensetive) и c – категория. Чувствительность может принимать значения от s0 до s15, а категория от c0 до c1023, также категорий может быть несколько, так например, s0:c1, c4 также является уровнем. Если мы хотим, чтобы в уровне с чувствительностью s1 были категории от c1 до c10, то можно использовать запись s1:c1.c10. Данный атрибут используется в MLS политике, которая будет рассмотрена позже, также при рассмотрении MLS мы дадим более подробное описание этого атрибута.

Для лучшего понимания введенных терминов, рассмотрим пример.

Пример 1.1

1) Пользователь хочет сменить пароль для чего он запускает passwd утилиту. В свою очередь вызывающийся файл /usr/bin/passwd имеет следующее описание:
system_u:object_r:passwd_exec_t:s0. Этот файл имеет тип passd_exec_t.

2) passwd попытается открыть файл /etc/shadow, его описание: system_u:object_r:shadow_t:s0. Он имеет тип shadow_t. shadow_t используется для файлов, доступ к которым необходим для смены пароля.

3) Правила SELinux разрешают открытие на чтение и запись процессам в домене passwd_t к файлам типа shadow_t. В свою очередь у домена есть точка входа доступная для типа passwd_exec_t.

4) Таким образом когда пользователь вызовет passwd, запущенный процесс сможет перейти в домен passwd_t. Что в свою очередь дает доступ этому процессу к файлам имеющим тип shadow_t.

Пример 1.1 является достаточно простым объяснением работы типов и доменов внутри SELinux. Теперь давайте рассмотрим работу этих правил чуть подробней для лучшего понимания.

Пример 1.2

SELinux следит за тем чтобы:

1) В passwd_t домен можно попасть только при условии использования приложений, имеющих тип passwd_exec_t. Эти приложения в свою очередь могут быть запущены только авторизированными библиотеками общего пользования, как например с типом lib_t.

2) В отличии от обычного Linux, даже процесс с с правами супер пользователя (SU) не может получить доступ к файлам типа shadow_t, до тех пор пока он не находится в домене passwd_t. Что позволяет уменьшить уязвимость при доступе злоумышленника к SU правам.

3) Только процесс из авторизированного домена имеет право переходить в домен passwd_t. Так например процесс в домене sendemail_t не должен пытаться запустить passwd, поэтому также он не должен попадать в домен passwd_t. Такая попытка в SELinux сразу приведет к ошибке.

4) Также процесс работающий в passwd_t домене может обращаться только к файлам с определенным типом, как например, etc_t и shadow_t. Такое ограничение не позволит записать данные в произвольный файл.

5 Типы политик

Теперь же поговорим о типах политики. Мы рассмотрим политики с разными функционалами всего будет рассмотрено 3 типа политик: целевые политики, строгие политики и mls. Конечно, существуют другие типы политик, такие как стандартная и минимальная, но в этой статье мы рассмотрим только основные.

5.1 “Целевые” (targeted) политики

Этот тип политики был разработан для проекта Fedora. В рамках политики описано более 200 процессов, которые могут выполняться в операционной системе. Процессы, описанные в “целевой” политики, выполняются в домене confined_t, остальные в unconfined_t. Процессы, выполняющиеся в unconfined_t не защищаются в SELinux. Поэтому любые сторонние приложения смогут работать без проблем со стороны SELinux.

В ограниченный(confined) домен входят почти все приложения, которые ожидают внешних подключений по сети, как например, sshd и httpd. Также большинство процессов имеющих root-права попадают в ограниченный домен. Это помогает снизить ущерб от атак со стороны злоумышленника, так как все процессы, имеющие большое количество прав доступа, все еще будут проходить проверки со стороны SELinux.

Теперь обсудим неограниченный (unconfined) домен.Как уже было сказано ранее процессы не описанные в “целевой” политике работают в неограниченном домене. Процессы работающие в этом домене регулируются только DAC правилами. SELinux не будет отслеживать действия в этом домене, поэтому любой процесс созданный злоумышленником будет ограничен только DAC.

5.2 Строгие политики

Основная идея такого типа: “Запрещено все, если правилами не указано обратного” (Также называется принцип наименьших прав). То есть все правила будут являться разрешениями, а не запретами. Эта политика основана на Reference Policy компании Tresys.

5.3 Многоуровневая модель безопасности (Multy-Level Security)

MLS использует модель Белла-Лападуллы. Это модель контроля и управления доступом, основанная на мандатной модели управления доступом. В ней процессы и пользователи называются субъектами, а файлы, устройства называются объектами. Объектам и субъектам присваиваются уровни секретности и безопасности соответственно, в англоязычной литературе все это объединено в

понятие security level. Для простоты понимания сразу перенесем все описание работы модели на реалии SELinux – субъект будем называть процессом, а объект файлом.

Основная идея модели Белла-Лападуллы заключается в двух правилах:
1) Процесс может читать файлы с уровнем доступа ниже или равному его уровню безопасности. В английской литературе используется термин No Read Up (NRU).
2) Процесс может писать только в файлы с уровнем выше или равном его уровню безопасности. No Write Down (NWD).

Для применения этой модели в SELinux используется атрибут контекста "уровень" о котором мы говорили ранее. Чтобы можно было применить правила NWD и NRU необходимо определить "операцию" сравнения уровней:
1) Уровень l1 превосходит уровень l2 – если чувствительность s1 больше либо равна s2 и множество категорий l1 включает в себя множество категорий l2 (в том числе они могут быть равны).
2) Уровень l2 уступает уровню l2 – если чувствительность s2 меньше либо равна s1 и категории l1 являются подмножеством категорий l2.
3) Уровень l1 равен уровню l2 – если чувствительность s1 равна s2 и множества категорий l1 и l2 совпадают.
4) Уровень l1 и l2 не сравнимы – если ни один из уровней не превосходит или равняется другому.

Пример: s1:c1, c2 превосходит уровни s0:c1, c2 и s1:c1 s0:c1 уступает s1:c1, c2 s1:c1 равен s1:c1 s2:c1, c3 не сравним с s3:c2, c3

Таким образом в SELinux сравниваются уровни доступа и определяются правила NWD и NRU:
1) Если уровень доступа процесса превосходит или равен уровню доступа файла, то процесс может открыть файл на чтение.
2) Если уровень доступа процесса уступает или равен уровню доступа файла, то файл может быть открыт на запись.

6 Заключение

На этом обзорная статья заканчивается. Из всего вышесказанного, SELinux дает возможность администраторам задавать свои правила разграничения доступа поверх уже имеющейся системы контроля доступа Linux. Еще раз заметим, что SELinux не дает полного контроля над системой доступа, она лишь может добавить к уже имеющимся ограничениям правила, заданные администратором. Но даже такой подход позволяет решить множество задач по улучшению безопасности данных.

Введенный в SELinux контекст дает широкие возможности по наложению ограничений на процессы, исполняющиеся внутри системы. Также SELinux предоставляет заранее подготовленные типы политик, что позволяет не "изобретать велосипед", а выбрать что-то подходящее для нужд пользователя.

7 Источники

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/selinux_users_and_administrators_
selinux
http://selinuxproject.org/page/Main_Page
https://ru.wikipedia.org/wiki/SELinux