Доклад

SELinux

Тарутина Кристина Олеговна

Содержание

Список иллюстраций

Список таблиц

1 Введение

SELinux — самый популярный модуль безопасности Linux, используемый для изоляции и защиты компонентов системы друг от друга. Одной из ключевых особенностей SELinux является то, что он позволяет системным администраторам блокировать несанкционированный доступ к системным ресурсам. Эта архитектура безопасности обеспечивает разделение привилегий между пользователями системы и процессами, позволяя администраторам более эффективно управлять доступом к ресурсам.

Однако, несмотря на то, что во многих дистрибутивах Linux модули безопасности Linux включены по умолчанию, общие знания SELinux среди разработчиков и системных администраторов не так распространены, как можно было бы ожидать, что переодически приводит к эксцессам и ошибкам.

2 Основная часть

2.0.1 Что такое SELinux?

Security-Enhanced Linux (SELinux) — это архитектура безопасности, созданная Агентством национальной безопасности США (АНБ) и Red Hat. Этот модуль безопасности доступен для большинства дистрибутивов Linux, но в основном используется в RHEL и Fedora.

SELinux применяет политики обязательного контроля доступа (MAC). MAC — это тип системы безопасности, который позволяет системным администраторам устанавливать централизованную политику безопасного доступа и указывать, какие пользователи и процессы имеют доступ к определенным ресурсам. Система также работает по модели наименьших привилегий, по умолчанию блокируя доступ к ресурсам. [1]

В SELinux системные администраторы отличают пользователя от приложений, которые он запускает. Например, оболочка пользователя имеет полный доступ к домашнему каталогу. Однако если пользователь запускает почтовый клиент, SELinux блокирует доступ клиента к определенным частям домашнего каталога.

SELinux также повышает уровень защиты за счет разделения политик безопасности и применения решений безопасности внутри ядра. Такое разделение дает системным администраторам больший контроль над общей безопасностью системы. Дополнительная защита также обеспечивается тем, что SELinux интегрирован в ядро Linux, что означает, что он всегда работает, и пользователи и процессы не могут его отключить.

2.0.2 Архитектура SELinux

Архитектуру SELinux можно разделить на четыре основных компонента:

1. Субъект должен запросить доступ, чтобы выполнить действие. В большинстве случаев
2. диспетчер объектов (OM), который контролирует доступ субъекта. Он отправит запрос
3. Сервер безопасности – сервер безопасности принимает решения на основе Политики бе
4. Access Vector Cache (AVC) – это кэш, в котором хранятся решения сервера безопаснос

В документации SELinux представлена краткая диаграмма, показывающая взаимодействие между этими четырьмя компонентами.(рис. 2.1).

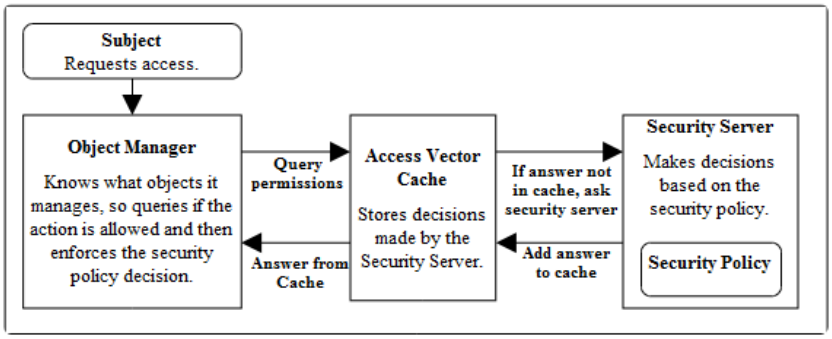


Рис. 2.1: Диаграма архитектуры

Сервер безопасности может быть только один, расположенный в ядре. Однако AVC и OM могут находиться как в ядре, так и в пользовательском пространстве. В пространстве ядра платформа модулей безопасности Linux представляет собой OM, поскольку она решает, какие службы ядра могут быть ограничены структурой безопасности, такой как SELinux. AVC обычно реализуются как хеш-карта, используемая для кэширования решений в реализациях ядра или пользовательской среды. В пространстве пользователя одно приложение может быть как OM, так и AVC для ресурсов, которыми оно управляет. Это возможно с помощью API SELinux для запроса сервера безопасности.[3]

2.0.3 Режимы SELinux

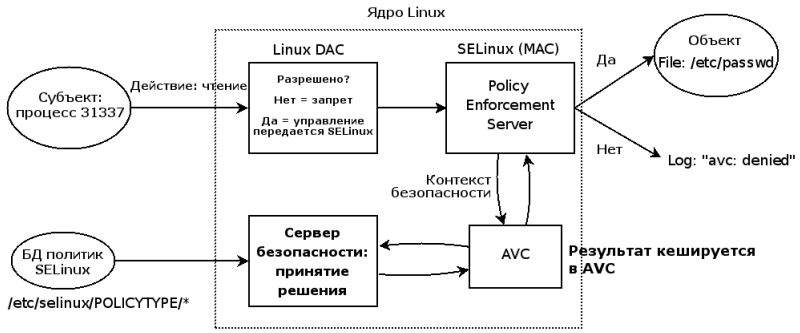
Одной из ключевых особенностей SELinux является возможность работы в различных режимах, что позволяет системным администраторам точно настраивать безопасность системы.

Существует три основных режима SELinux:

1. Enforcing. Режим SELinux по умолчанию и наиболее безопасный. В этом режиме SELinu
2. Permissive. Менее безопасен, чем принудительный режим, но все же обеспечивает заш
3. Disabled. Наименее безопасный режим, поскольку он не защищает системные ресурсы.

2.0.4 Политики SELinux

SELinux по-умолчанию работает в режиме Enforcing, когда любые действия, кроме разрешенных, автоматически блокируются, каждая программа, пользователь или сервис обладают только теми привилегиями, которые необходимы им для функционирования, но не более того. Это довольно жесткая политика, которая обладает как плюсами — наибольший уровень информационной безопасности, так и минусами — конфигурирование системы в таком режиме сопряжено с большими трудозатратами системных администраторов, к тому же, велик риск того, что пользователи столкнутся с ограничением доступа, если захотят использовать систему хоть сколько-нибудь нетривиальным образом. Такой подход допустим в Enterprise-секторе, но неприемлем на компьютерах конечных пользователей. Многие администраторы просто отключают SELinux на рабочих станциях, чтобы не сталкиваться с подобными проблемами.(рис. **??**).



Для того, чтобы избежать этого, для ключевых приложений и сервисов, таких как, например, httpd, named, dhcpd, mysqld, определены заранее сконфигурированные целевые политики, которые не позволят получить злоумышленнику доступ к важным данным. Те же приложения, для которых политика не определена, выполняются в домене unconfined_t и не защищаются SELinux. Таким образом, правильно выбранные целевые политики позволяют добиться приемлемого уровня безопасности, не создав при этом для пользователя лишних проблем.

2.0.5 Дискреционный контроль доступа

Контроль доступа — это стандарт, используемый для определения разрешений в системах. Контроль доступа можно разделить на два основных типа: дискреционный контроль доступа (DAC) и обязательный контроль доступа (MAC).[3]

В системе DAC UNIX разрешения представлены в следующем формате:

Здесь rwx означает чтение, запись и выполнение соответственно. В каждом поле показаны разрешения для владельца, члена группы и лица, не являющегося членом группы, соответственно. Владелец файла имеет возможность изменять права доступа к файлам, которыми он владеет. Мы можем использовать команду ls с флагом L в нижнем регистре, чтобы увидеть разрешения файла Unix.

2.0.6 Обязательный контроль доступа

Напротив, SELinux MAC назначает контекст безопасности каждому субъекту или объекту, где субъекты можно рассматривать как процессы, а объекты — это ресурсы, предоставляемые операционной системой или программой пользовательского пространства. Мы можем использовать команду ls с флагом Z, чтобы увидеть контекст SELinux файла Unix.[3]

Формат контекста безопасности выглядит следующим образом: пользователь:роль:тип:[:диапазон]

где

Пользователь

Пользователь представляет пользователя SELinux. Пользователь SELinux отделен от пользователя Linux, но может быть назначен нескольким пользователям Linux и помогает преодолеть разрыв между миром Linux и миром SELinux. Имена пользователей SELinux часто заканчиваются на _u.

Роль

Роль представляет роль, в которой должен находиться пользователь. Роль может иметь более одного пользователя SELinux, и они по соглашению заканчиваются на _r. Роли часто представляют собой работу пользователя Linux, например администратор, обычный пользователь, администратор базы данных и т. д.

Тип

Типы являются наиболее важной частью контекста, поскольку все правила основаны на типах, и поэтому тип субъекта, домена или объекта определяет его разрешения. Когда тип связан с процессом, он определяет, к каким процессам (или доменам) может получить доступ пользователь SELinux (субъект). Когда тип связан с объектом, он определяет, какие права доступа имеет пользователь SELinux к этому объекту. SELinux и соблюдение типов идут рука об руку. Пользователи, присутствующие в привилегированных доменах SELinux, обычно помеченных как неограниченные, часто могут указывать файлы политики SELinux, файлы *.te, чтобы создать определенные разрешения для типа/домена. Более

подробную информацию об этом можно найти в разделе «Политика безопасности».

Диапазон

Поле диапазона контекста — это расширенный параметр, который назначает диапазон чувствительности. Они представлены в формате s#, где # — номер. По соглашению низшие имеют меньшие привилегии, чем высшие. Чувствительности могут читать и записывать на свой собственный уровень чувствительности, но могут читать только с более низких и записывать только на более высокие. Вы можете определить одну чувствительность, чтобы она доминировала над другой, используя оператор доминирования в политике SELinux. Например, доминирование {s1, s2} означает, что s2 доминирует над s1. Когда диапазон определяется только чувствительностью, это называется многоуровневой безопасностью (MLS). В дополнение к MLS существует Multi-Category Security, где для расчета доступа используется комбинация чувствительности и категорий. MCS вводит идею категорий, которые представляют собой отсеки, внутри которых применяется чувствительность. В отличие от чувствительных, в категориях нет иерархии, но они используются для определения разных типов ресурсов. Например, финансовые документы могут относиться к работе, побочному бизнесу и личным инвестициям. Процесс может получить доступ только к объектам, которые относятся к тем же категориям, что и он сам. В целом диапазон может выглядеть так: s1-s15:c0.c700, что означает, что процесс имеет эффективную чувствительность s1 и чувствительность (максимальную) чувствительности. из 15, с возможностью доступа к контейнерам от 0 до 700. Следовательно, учитывая объект в s0:c0, вышеуказанный процесс может только читать из него, поскольку он имеет более низкий уровень чувствительности и находится в том же контейнере. [3]

2.0.7 Устранение проблем

Рано или поздно происходит ситуация, когда вы сталкиваетесь с ситуацией, когда SELinux запрещает вам доступ к чему-то. Есть несколько основных причин отказа доступа:

— Неправильно маркированный файл. — Процесс работает в неправильном контексте — Ошибка в политике. Процесс требует доступ к файлу, который не был учтен при создании политики. — Попытка вторжения.

Первые три причины отказа доступа разрешаются достаточно легко, в то время как во время попытки вторжения звучит сигнал тревоги и пользователю посылается соответствующее уведомление. Для того, чтобы разобраться с любой проблемой, достаточно просмотреть журнал SELinux. По умолчанию он записывается процессом auditd в файл /var/log/audit/audit.log. Если этот процесс не запущен, то SELinux ведет журнал в файле /var/log/messages, в этом случае все сообщения системы контроля доступа маркируются ключом AVC, что позволяет быстро отфильтровать нужные строки, например. при помощи команды grep. В последние версии дистрибутивов (начиная с CentOS 5), включена утилита с графическим интерфейсом пользователя, которая позволяет отображать журнал SELinux в удобном и понятном для пользователя виде. Вызвать её можно из консоли, набрав sealert -b. Утилита входит в состав пакета setroubleshoot. В том случае, если X-сервер не запущен, вы можете сгенерировать понятные и удобные для человека отчеты следующей командой:

```
sealert -a /var/log/audit/audit.log > /path/to/mylogfile.txt
```

2.0.8 Утилита audit2allow

Утилита audit2allow создает разрешающие правила политики SELinux из файлов журналов, содержащих сообщения о запрете операций.

Эта утилита сканирует журналы в поиске сообщений, появляющихся, когда система не дает разрешения на операцию. Далее утилита генерирует ряд

правил, которые, будучи загруженными в политику, могли бы разрешить эти операции. Однако, данная утилита генерирует только разрешающие правила Type Enforcement (TE). Некоторые отказы в использовании разрешений могут потребовать других изменений политики. Например, добавление атрибута в определение типа, для разрешения существующего ограничения (constraint), добавления разрешающего правила для роли или модификации ограничения (constraint). В случае сомнений для диагностики можно попробовать использовать утилиту audit2why. [5]

Следует с осторожностью работать с выводом данной утилиты, убедившись, что разрешаемые операции не представляют угрозы безопасности. Обычно бывает лучше определить новый домен и/или тип или произвести другие структурные изменения. Лучше избирательно разрешить оптимальный набор операций вместо того, чтобы вслепую применить иногда слишком широкие разрешения, рекомендованные этой утилитой. Некоторые запреты на использование разрешений бывают не принципиальны для приложения. В таких случаях вместо использования разрешительного правила («allow» rule) лучше просто подавить журналирование этих запретов при помощи правила «dontaudit».(рис. 2.2).

Опция	Значение опции
-a | --all	Прочесть входную информацию из журналов message и audit. Не используется вместе с опцией -i
-d | --dmesg	Прочесть входную информацию из вывода команды /bin/dmesg. Обратите внимание, что когда работает auditd, не все сообщения аудита доступны через dmesg. Вместо этого используйте "ausearch -m avc | audit2allow " или "-a".
-f | --fcfile <File Context File>	Добавить Файл Контекстов в генерируемый пакет модуля. Требует опцию -M.
-h | --help	Вывести краткую справку по использованию
-i <inputfile> | --input <inputfile>	Прочесть входную информацию из <inputfile>
-l | --lastreload	Прочесть только часть входной информации, начиная с момента последней перезагрузки политики
-m <modulename> | --module <modulename>	Генерировать модуль. Требуется вывод <modulename>
-M <modulename>	Генерировать загружаемый пакет модуля. Опция конфликтует с -o
-o <outputfile> | --output <outputfile>	Дописать вывод в <outputfile>

Рис. 2.2: Опции audit2allow

2.0.9 Утилита secon

Утилита secon - просмотреть контекст SELinux для файла, программы или ввода пользователя.

Просматривает часть контекста. Контекст берется из файла, идентификатора процесса, ввода пользователя или контекста, в котором была запущена утилита secon. (рис. 2.3).

Опция	>Значение опции
-V, --version	Посмотреть текущую версию secon
-h, --help	Вывести информацию по использованию secon
-P, --prompt	Вывести данные в формате, подходящем для подсказки
-u, --user	Показать пользователя контекста безопасности
-r, --role	Показать роль контекста безопасности
-t, --type	Показать тип контекста безопасности
-s, --sensitivity	Показать уровень чувствительности (sensitivity level) контекста безопасности
-c, --clearance	Показать уровень допуска (clearance level) контекста безопасности
-m, --mls-range	Показать для контекста безопасности в виде диапазона уровень чувствительности (sensitivity level) и уровень допуска (clearance)
-R, --raw	Вывести уровень чувствительности (sensitivity level) и уровень допуска (clearance) в формате без трансляции
-f, --file	Получить контекст заданного файла FILE

Рис. 2.3: Опции secon

3 Выводы

В заключении доклада о SELinux следует подчеркнуть необходимость осознания значимости этого модуля безопасности для Linux и призвать разработчиков и администраторов к уделению должного внимания его изучению и использованию. SELinux является важным инструментом для обеспечения безопасности системы, благодаря его способности к изоляции и защите компонентов, а также блокированию несанкционированного доступа к системным ресурсам. Повышение общего уровня знаний о SELinux среди профессионалов в области Linux может существенно улучшить безопасность систем и способствовать более эффективному управлению ими.

Список литературы

[1] - What is SELinux? | 12.01.2023 | URL: https://phoenixnap.com/kb/selinux [2] - SElINUX | URL: https://github.com/SELinuxProject/selinux-notebook [3] - Introduction to SELinux | 05.07.2023 | URL: https://github.blog/2023-07-05-introduction-to-selinux/ [4] - SELinux – описание и особенности работы с системой. Часть 1 | 20.01.2014 | URL: https://habr.com/ru/companies/kingservers/articles/209644/ [5] - SELinux - система принудительного контроля доступа | URL: https://redos.red-soft.ru/base/manual/safe-redos/selinux/