РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук
Кафедра прикладной информатики и теории вероятностей

Презентация №9

Модель Доступа AppArmor

Студент: Эйвази Мани

Группа: НПИбд-03-24

Студенческий билет №: 1032245107

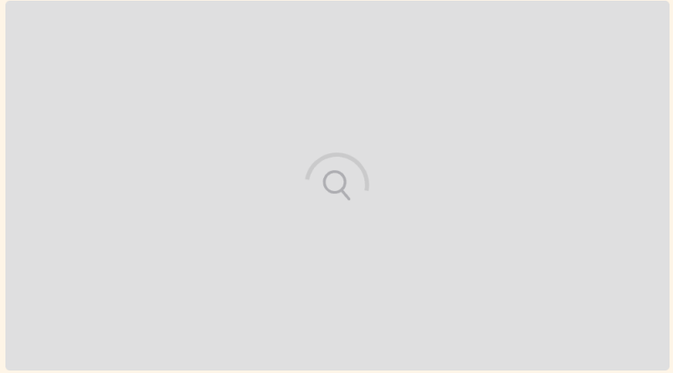
Модель Доступа AppArmor

Повышение безопасности Linux-систем через мандатный контроль доступа, привязанный к программам

ГЛАВА 1

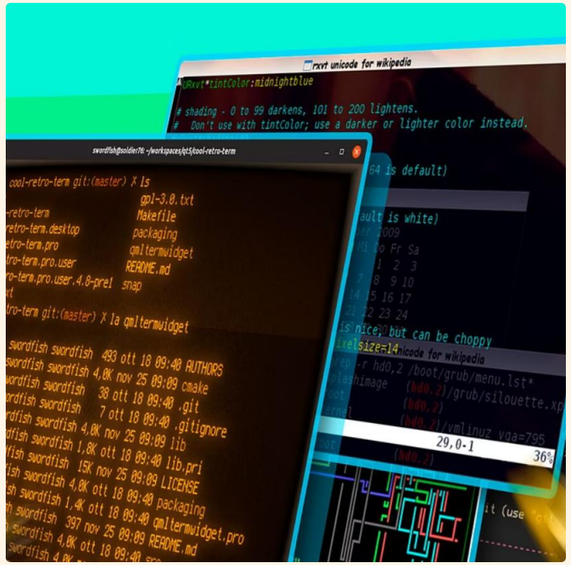
AppArmor как LSM-модуль

AppArmor (Application Armor) — это модуль безопасности ядра Linux, реализующий мандатный контроль доступа (MAC). Он работает как часть Linux Security Modules (LSM) и позволяет администраторам связывать профили безопасности с отдельными программами. В отличие от традиционной дискреционной системы контроля доступа (DAC), которая зависит от идентификаторов пользователя (UID) и группы (GID), AppArmor ограничивает действия программы независимо от привилегий пользователя, запустившего её.



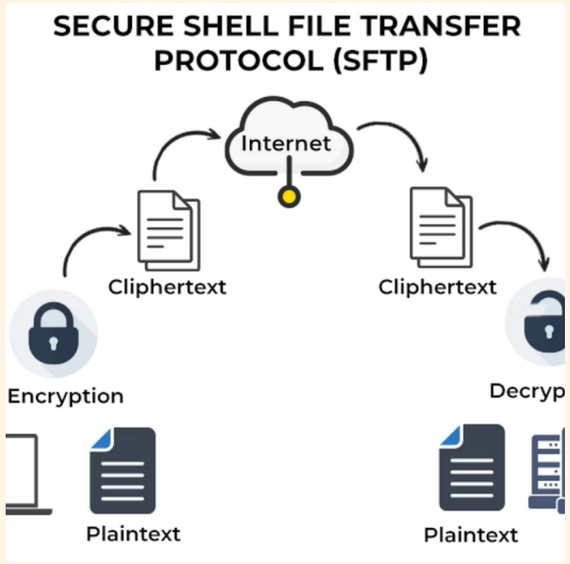
Одной из ключевых особенностей AppArmor является его простота и человекочитаемый синтаксис, что делает его более доступным для понимания и управления по сравнению с SELinux. В то время как SELinux оперирует сложными контекстами безопасности и требует глубоких знаний для настройки, AppArmor использует более интуитивный подход, фокусируясь на путях к файлам и сетевых операциях.

Ключевая Концепция: Профиль к Программе



Привязка к Программе

В основе AppArmor лежит идея связывания атрибутов доступа напрямую с исполняемым файлом программы. Это означает, что профиль безопасности определяет, что конкретная программа может делать или не делать, независимо от пользователя, который её запускает. Это радикально отличается от традиционных моделей, где права наследуются от пользователя.



Исполняемый Файл и Профиль

Каждый профиль AppArmor предназначен для защиты определённого исполняемого файла. Когда программа запускается, ядро Linux проверяет наличие активного профиля AppArmor для этой программы. Если профиль существует, все последующие действия программы (доступ к файлам, использование сетевых сокетов, запросы к ядру) фильтруются согласно правилам, определённым в профиле.

Режимы Работы Профилей

1

Режим Enforce (Принудительный)

В этом режиме AppArmor активно применяет все правила, определённые в профиле. Любое действие программы, которое нарушает эти правила, будет заблокировано. Нарушения также будут записаны в системный журнал. Это режим максимальной защиты, предназначенный для производственных систем после тщательного тестирования профиля.

2

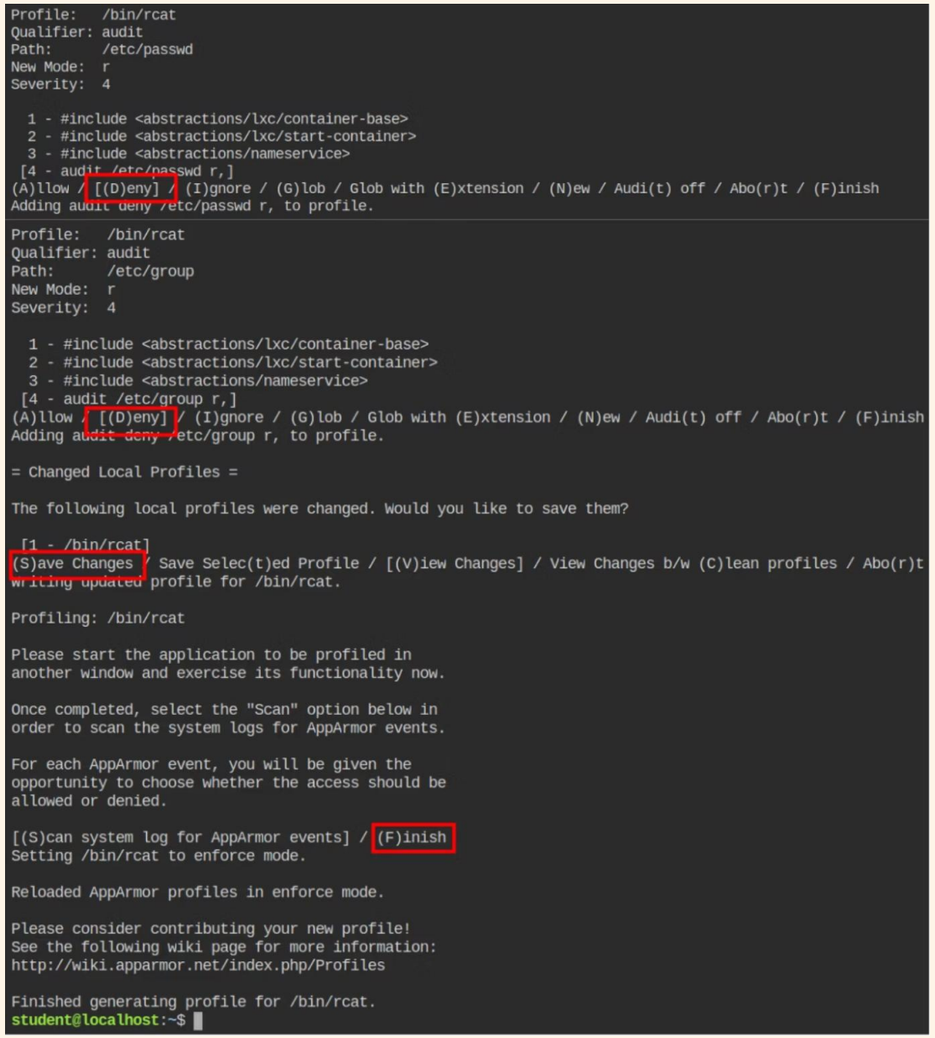
Режим Complain (Обучение)

В режиме complain (или learning) AppArmor не блокирует нарушения правил. Вместо этого он только записывает все запрещённые действия в системный журнал. Этот режим идеально подходит для создания новых профилей или тонкой настройки существующих, поскольку он позволяет выявить все необходимые привилегии программы без нарушения её функциональности.

Переключение между режимами выполняется с помощью утилит командной строки, что позволяет динамически адаптировать уровень защиты.

ГЛАВА 2

Структура Профиля AppArmor



- ***Путь к программе:*** *Имя файла профиля обычно соответствует полному пути к исполняемому файлу программы, для которой он предназначен. Например, для* `/usr/bin/firefox` *профиль будет называться* `usr.bin.firefox`*.*
- ***Правила доступа к ФС:*** *Определяют, к каким файлам и каталогам программа имеет доступ (чтение, запись, исполнение, блокировка).*
- ***Capability-права:*** *Контролируют специфические привилегии ядра, такие как создание сетевых сокетов, изменение системного времени или перезагрузка системы.*
- ***Сетевые операции:*** *Управляют доступом к сетевым протоколам (TCP, UDP, ICMP) и портам, позволяя ограничить входящие и исходящие соединения.*
- ***Дополнительные правила:*** *Включают правила для использования IPC, mount-операций и других системных вызовов.*

Каждый профиль — это текстовый файл, содержащий набор правил, написанных на простом и понятном языке, что способствует лёгкой аудируемости и модификации.

Синтаксис Профиля: Детали

Профили AppArmor используют простой и интуитивно понятный синтаксис. Рассмотрим основные элементы:

Include-файлы

Для обеспечения модульности и повторного использования правил, AppArmor поддерживает включение других файлов с правилами. Это позволяет создавать абстракции для общих ресурсов (например, `/etc/apparmor.d/tunables/global`*).*

```
#include <abstractions/base>
```

Deny-правила

Позволяют явно запретить доступ к определённым ресурсам, даже если они косвенно разрешены другими правилами или абстракциями.

```
deny /etc/shadow rw,
```

Маски Доступа

Основные маски доступа для файловой системы: `r` *(чтение),* `w` *(запись),* `m` *(загрузка в память),* `x` *(исполнение). Также существуют* `ux` *(безусловное исполнение) и* `px` *(исполнение в другом профиле).*

```
/etc/fstab r,
```

```
/usr/bin/ping x,
```

Пример для /bin/ping

Профиль может явно разрешать выполнение ping, а также ограничивать его доступ к другим файлам или сетевым операциям.

```
profile ping /bin/ping {
 #include <abstractions/base>
 network inet raw,
 /bin/ping ix,
 /etc/services r,
 deny /etc/hosts w,
}
```

Хранение и Загрузка Профилей

Управление профилями AppArmor централизовано и организовано для удобства администраторов.

Каталог Профилей

Все профили AppArmor хранятся в каталоге `/etc/apparmor.d/`*. Это стандартное место, где система ожидает найти файлы конфигурации профилей.*

Соглашение об Именах

Имя файла профиля обычно является "дефисированной" версией полного пути к исполняемому файлу. Например, профиль для `/usr/sbin/nginx` *будет называться* `usr.sbin.nginx`*. Символ слеша (/) заменяется на точку (.).*

Загрузка Профилей

Для загрузки, выгрузки и управления профилями используется утилита `apparmor_parser`*. Например,* `sudo apparmor_parser -r /etc/apparmor.d/usr.sbin.nginx` *перезагрузит профиль Nginx, а* `sudo apparmor_parser -R /etc/apparmor.d/usr.sbin.nginx` *выгрузит его.*

Эти соглашения помогают поддерживать порядок и облегчают автоматизацию управления профилями.

ГЛАВА 3

Инструменты Управления AppArmor

AppArmor предоставляет набор мощных утилит командной строки для мониторинга, создания и модификации профилей.

aa-status

Отображает текущий статус всех загруженных профилей AppArmor, указывая, какие из них находятся в режиме `enforce`*, а какие — в режиме* `complain`*.*

```
sudo aa-status
```

aa-enforce

Переводит указанный профиль из режима `complain` *в режим* `enforce`*, активируя принудительное применение правил.*

```
sudo aa-enforce /usr/bin/program
```

aa-complain

Переводит указанный профиль из режима `enforce` *в режим* `complain`*, что позволяет программе работать без блокировок, но с журналированием нарушений.*

```
sudo aa-complain /usr/bin/program
```

aa-genprof

Интерактивная утилита для создания нового профиля. Она помещает программу в режим `complain`*, предлагает запустить её и собрать все действия, затем на их основе создаёт черновик профиля.*

```
sudo aa-genprof /usr/bin/program
```

aa-logprof

Анализирует системные журналы на предмет нарушений AppArmor и предлагает обновить существующие профили, добавляя или изменяя правила, чтобы разрешить ранее заблокированные действия.

```
sudo aa-logprof
```

aa-autodep

Создаёт базовый профиль для программы, анализируя её зависимости и типичные пути доступа, что служит хорошей отправной точкой для дальнейшей настройки.

```
sudo aa-autodep /usr/bin/program
```

Диагностика и Журналирование

Эффективное использование AppArmor требует постоянного мониторинга и анализа журналов для выявления и устранения проблем с доступом.

Анализ Отказов

Все события, связанные с AppArmor, включая заблокированные или разрешённые действия, записываются в системные журналы. Основные места для проверки:

- ***dmesg:*** *Ядерный буфер сообщений, где можно увидеть низкоуровневые сообщения о блокировках AppArmor.*
- ***audit.log:*** *Системный журнал аудита (часто* `/var/log/audit/audit.log`*), который предоставляет детальную информацию о событиях безопасности, включая* `DENIED` *и* `ALLOWED` *записи от AppArmor.*

Понимание этих записей критически важно для отладки и создания профилей.

Примеры Записей

В журналах вы можете найти записи, подобные этим:

```
type=APPARMOR AUDIT_TYPE="AVC" comm="program" ...
profile="profile_name" operation="open" name="/path/to/file"
pid=1234 ... DENIED
```

Эта запись указывает на то, что программа с профилем `profile_name` *пыталась открыть файл* `/path/to/file`*, и это действие было заблокировано (*`DENIED`*).*

```
type=APPARMOR AUDIT_TYPE="AVC" comm="program" ...
profile="profile_name" operation="open" name="/path/to/file"
pid=1234 ... ALLOWED
```

Эта запись, наоборот, показывает, что доступ был разрешён (`ALLOWED`*). Такие записи особенно полезны при работе в режиме* `complain` *для сбора всех необходимых разрешений.*

Спасибо за внимание