Доклад

Дискреционные модели доступа. Списки управления доступом.

Легиньких Галина Андреевна

Содержание

Список иллюстраций

Список таблиц

1 Введение

В современном мире информационных технологий безопасность данных является одной из ключевых задач, требующих внимания со стороны разработчиков, администраторов и пользователей. Дискреционные модели доступа (DAC) представляют собой один из основных подходов к управлению доступом к ресурсам, позволяя пользователям определять права доступа к объектам в системе. Основной идеей этих моделей является то, что владельцы объектов имеют возможность управлять правами доступа к ним, что предоставляет гибкость и удобство в управлении.

Списки управления доступом (ACL) выступают в роли важного инструмента, реализующего дискреционные модели доступа. Они представляют собой структуры данных, в которых указаны пользователи и соответствующие им права доступа к объектам системы. Применение ACL позволяет эффективно управлять доступом к ресурсам, минимизируя риск несанкционированного доступа и обеспечивая защиту информации.

Цель работы

Изучить дискреционные модели доступа и списки управления доступом (ACL) как инструментов управления доступом к информационным ресурсам в информационных системах.

2 Основная часть

2.1 Политика безопасности

Технология защиты автоматизированных систем начала развиваться относительно недавно, но сегодня уже существует значительное число теоретических моделей, позволяющих описывать различные аспекты безопасности и обеспечивать средства защиты формально подтвержденной алгоритмической базой.

Под *политикой безопасности* понимается совокупность норм и правил, регламентирующих процесс обработки информации, выполнение которых обеспечивает защиту от определенного множества угроз и составляет необходимое условие безопасности системы. Формальное выражение политики безопасности называют *моделью безопасности*.

Основная цель создания политики безопасности системы и описания ее в виде формальной модели — это определение условий, которым должно подчиняться поведение системы, выработка критерия безопасности и проведение формального доказательства соответствия системы этому критерию при соблюдении установленных правил и ограничений.

Кроме того, формальные модели безопасности позволяют решить еще целый ряд задач, возникающих в ходе проектирования, разработки и сертификации защищенных систем, поэтому их используют не только теоретики информационной безопасности, но и другие категории специалистов, участвующих в процессе создания и эксплуатации защищенных информационных систем (производители, потребители, эксперты).

Модели безопасности обеспечивают системотехнический подход, включающий решение следующих важнейших задач:

- выбор и обоснование базовых принципов архитектуры защищенных автоматизированных систем, определяющих механизмы реализации средств и методов защиты информации;
- подтверждение свойства защищенности разрабатываемых систем путем формального доказательства соблюдения политики безопасности (требований, условий, критериев);
- составление формальной спецификации политики безопасности как важнейшей составной части организационного и документационного обеспечения разрабатываемых защищенных компьютерных систем.

Производители защищенных информационных систем используют модели безопасности в следующих случаях:

- при составлении формальной спецификации политики безопасности разрабатываемой системы;
- при выборе и обосновании базовых принципов архитектуры защищенной системы, определяющих механизмы реализации средств защиты;
- в процессе анализа безопасности системы, при этом модель используется в качестве эталонной модели;
- при подтверждении свойств разрабатываемой системы путем формального доказательства соблюдения политики безопасности.

Потребители путем составления формальных моделей безопасности получают возможность довести до сведения производителей свои требования в четко определенной и непротиворечивой форме, а также оценить соответствие защищенных систем своим потребностям.

Эксперты по квалификации в ходе анализа адекватности реализации политики безопасности в защищенных системах используют модели безопасности в качестве эталонов.

По сути, модели безопасности являются связующим элементом между производителями, потребителями, экспертами.

2.2 Дискреционная модель управления доступом (DAC)

Дискреционная модель управления доступом (Discretionary Access Control, DAC) предоставляет пользователям контроль над правами доступа к их собственным ресурсам. Владельцы файлов и других объектов могут самостоятельно определять, какие пользователи или группы имеют доступ к объекту, и какие действия они могут выполнять. (рис. 2.1)

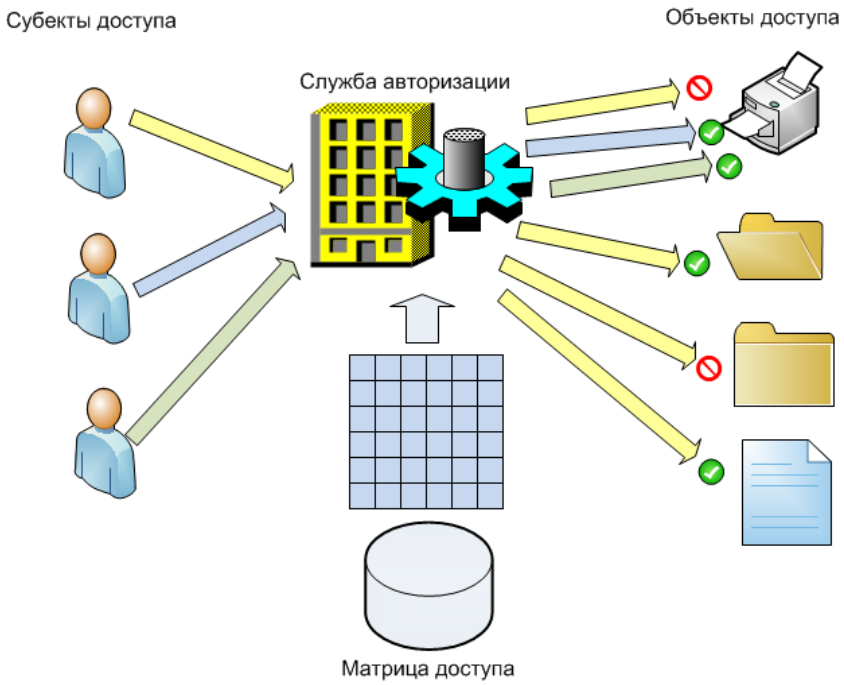

Рис. 2.1: DAC

2.2.1 Основные принципы DAC

Владение ресурсом: В DAC владелец ресурса имеет полный контроль над тем, кто и каким образом может получать доступ к этому ресурсу. Например, владелец файла может разрешить другим пользователям читать, изменять или исполнять файл.

Гибкость и децентрализация: В отличие от централизованных моделей управления доступом, таких как мандатная модель (MAC), в DAC каждый владелец ресурса управляет доступом к нему самостоятельно. Это позволяет системе быть более гибкой, особенно в условиях, где каждому пользователю необходимо предоставить возможность настраивать права доступа к своим данным.

2.2.2 Преимущества и недостатки DAC

Преимущества:

- **Гибкость**: DAC предоставляет пользователям свободу самостоятельно настраивать права доступа к их ресурсам.
- **Простота управления**: Модель интуитивно понятна и проста для пользователей, что делает её удобной для небольших систем.

Недостатки:

- **Уязвимость для ошибок**: Пользователь может случайно предоставить слишком большие права другим пользователям, что может привести к утечке данных.
- **Отсутствие централизованного контроля**: В крупных системах управление правами доступа становится сложным, так как каждый пользователь самостоятельно управляет доступом к своим ресурсам.

2.2.3 Матрица прав доступа

Матрица прав доступа — это структурированное представление прав доступа, показывающее, какие действия пользователи (субъекты) могут выполнять над ресурсами (объектами). Это важная концепция в модели DAC, которая помогает наглядно представить права доступа в системе.

Структура матрицы прав

Матрица прав представляет собой таблицу, где строки соответствуют субъектам (пользователям или группам), а столбцы — объектам (файлам, папкам, ресурсам системы). На пересечении строки и столбца записываются права доступа субъекта к объекту.

Пример матрицы прав:

Пользователь/Группа	Файл1	Файл2	Директория1
`user1`	r, w	r	r, w, x
`user2`	-	w	r
`group1`	r	-	-

В данном примере:

- Пользователь `user1` имеет права на чтение и запись файла `Файл1`, права на чтение файла `Файл2`, и полные права (чтение, запись, исполнение) для директории `Директория1`.
- Пользователь `user2` имеет только право на запись в файл `Файл2` и право на чтение директории.
- Группа `group1` может только читать файл `Файл1`.

Матрица прав доступа помогает системным администраторам или владельцам ресурсов легко определить, какие действия могут выполнять пользователи или группы над объектами.

2.2.4 Пример в Linux

Операционные системы, такие как Linux, также оперируют матрицей прав, хотя пользователи обычно взаимодействуют с правами доступа через команды и интерфейсы системы.

В Linux каждая файловая система содержит три базовых типа прав для объектов:

- **Чтение (r)** — возможность просматривать содержимое файла.
- **Запись (w)** — возможность изменять содержимое файла.
- **Исполнение (x)** — возможность запускать файл как программу.

Субъекты управления правами:

- **Владелец (user)** — пользователь, создавший файл или ресурс.
- **Группа (group)** — группа пользователей, которые могут получить доступ к ресурсу.
- **Прочие (other)** — все остальные пользователи.

2.3 Списки управления доступом (ACL)

Списки управления доступом (Access Control Lists, ACL) представляют собой расширение стандартных прав доступа в Linux, позволяя задавать права для конкретных пользователей и групп, не ограничиваясь базовой триадой (владелец, группа, прочие).

2.3.1 Пример использования ACL в Linux

Для использования ACL в Linux необходимо установить и активировать поддержку этой функциональности, если она не включена по умолчанию. Например, для системы Ubuntu можно использовать команду:

```
sudo apt-get install acl
```

После установки ACL можно работать с командами `setfacl` для назначения прав и `getfacl` для просмотра прав.

1. **Установка ACL для файла**:

   Допустим, у нас есть файл `example.txt`, и мы хотим предоставить пользователю `user1` право на чтение, а пользователю `user2` — право на запись:

   ```
   setfacl -m u:user1:r example.txt
   setfacl -m u:user2:w example.txt
   ```

2. **Просмотр ACL для файла**:

   Чтобы увидеть, какие ACL установлены для файла, используйте команду:

   ```
   getfacl example.txt
   ```

   Пример вывода команды:

   ```
   # file: example.txt
   # owner: root
   # group: root
   user::rw-
   user:user1:r--
   user:user2:-w-
   group::r--
   mask::rwx
   other::r--
   ```

3. **Удаление ACL**:

   Если необходимо удалить ACL для конкретного пользователя, используйте команду:

   ```
   setfacl -x u:user1 example.txt
   ```

   Это удалит права доступа, назначенные пользователю `user1`.

2.3.2 Преимущества и недостатки ACL

Преимущества:

- **Гибкость и детализация**: ACL позволяет настроить права доступа не только для владельца и группы, но и для каждого конкретного пользователя или группы.
- **Совместимость с существующими механизмами**: ACL дополняет стандартные механизмы управления доступом в Linux, позволяя использовать их совместно.

Недостатки:

- **Усложнение управления**: При большом количестве пользователей и объектов управление списками доступа может стать сложным и трудоёмким.
- **Зависимость от поддержки файловой системы**: Не все файловые системы поддерживают ACL. Например, в Linux поддержка ACL доступна в файловых системах, таких как ext4 и XFS.

2.4 Применение DAC и ACL в операционной системе Linux

В Linux права доступа традиционно управляются с использованием стандартной схемы DAC, которая предоставляет владельцу ресурса полный контроль над его доступом. Однако, благодаря поддержке ACL, система предоставляет пользователям возможность более гибкого управления правами доступа.

Пример использования базовых средств управления правами в Linux:

- **Команда `chmod`** используется для задания прав доступа на файлы и каталоги. Например:

```
chmod 755 example.txt
```

Здесь:

- 7 — полный доступ для владельца (чтение, запись, исполнение),
- 5 — доступ только на чтение и исполнение для группы и остальных пользователей.

- **Команда `chown`** позволяет изменить владельца файла:

```
chown user1 example.txt
```

- **Команда `chgrp`** изменяет группу, которой принадлежит файл:

```
chgrp group1 example.txt
```

Благодаря ACL, Linux позволяет создавать более сложные политики управления доступом, что особенно полезно в условиях многопользовательских систем и серверов.

2.5 Как посмотреть права доступа в Linux

Для просмотра прав доступа к файлу или директории в Linux используется команда `ls` с флагом `-l`, которая выводит детализированную информацию о правах доступа.

Пример:

```
ls -l example.txt
```

Результат может выглядеть так:

```
-rw-r--r-- 1 user1 group1 1234 Oct 3 12:34 example.txt
```

В данном примере: - Первые 10 символов показывают права доступа: - Первый символ (-) указывает на тип файла (в данном случае это обычный файл). -

Следующие три символа (`rw-`) указывают права владельца (чтение и запись). - Следующие три символа (`r--`) показывают права группы (только чтение). - Последние три символа (`r--`) указывают права остальных пользователей (только чтение).

Для файлов с установленными ACL вывод `ls -l` может содержать символ `+` после прав доступа, например:

```
-rw-r--r--+ 1 user1 group1 1234 Oct 3 12:34 example.txt
```

Это указывает на наличие дополнительных правил ACL для файла.

3 Заключение

Дискреционная модель управления доступом и списки управления доступом являются важнейшими механизмами контроля прав доступа в операционных системах. DAC предоставляет пользователям гибкость в управлении своими ресурсами, но при этом несёт риски, связанные с безопасностью, особенно в крупных системах. Матрица прав помогает визуализировать и систематизировать права доступа, упрощая администрирование. ACL в Linux позволяет более гибко управлять доступом, дополняя стандартные механизмы прав доступа. Однако правильное использование этих инструментов требует от пользователей и администраторов осторожности и понимания, чтобы избежать ошибок в настройке прав доступа и не допустить утечек информации.

4 Список литературы

1. Андреев А. Разбираем современные методы управления доступом в информационных системах [Электронный ресурс] // Habr [сайт]. – Режим доступа: https://habr.com/ru/articles/781096/, свободный (дата обращения: 03.10.2024).

2. Модель управления доступом и её реализация в Unix и Windows [Электронный ресурс] // УрФУ [сайт]. – Режим доступа: https://learn.urfu.ru/resource/index/data/resourc свободный (дата обращения: 03.10.2024).

3. Модели управления доступом [Электронный ресурс] // Studfile [сайт]. – Режим доступа: https://studfile.net/preview/10072102/page:13/, свободный (дата обращения: 03.10.2024).

4. Основы информационной безопасности [Электронный ресурс] // RV-Lab [сайт]. – Режим доступа: http://www.rv-lab.ru/it/is_2008/part3.htm, свободный (дата обращения: 03.10.2024).