Модели безопасности ОС

Основы информационной безопасности

Черная С. В.

18 апреля 2025

Российский университет дружбы народов, Москва, Россия

Объединённый институт ядерных исследований, Дубна, Россия

Докладчик

- Черная София Витальевна
- студентка группы НКАбд-01-23
- Российский университет дружбы народов
- 1132236043@rudn.ru

Актуальность

1. Рост киберугроз
2. Развитие технологий
3. Регуляторные требования
4. Увеличение числа пользователей

Цели и задачи

Цели:

1. Ознакомление с основами безопасности операционных систем и важностью защиты данных.
2. Рассмотрение различных моделей управления доступом и их применения в современных информационных системах.

Задачи:

1. Определение основных понятий и принципов работы операционных систем.
2. Выделение ключевых функциональных дефектов ОС, которые могут привести к утечке данных.
3. Изучение и описание различных моделей управления доступом (DAC, MAC, RBAC, ABAC) с указанием их преимуществ и недостатков.
4. Приведение примеров применения этих моделей в реальных системах.
5. Подчеркивание важности физической защиты и комплексного подхода к безопасности данных.

Введение

Операционная система (ОС) — это совокупность программ, управляющих ресурсами вычислительной системы для их эффективного использования и обеспечивающих интерфейс пользователя. ОС прошли несколько поколений в своем развитии. Первое поколение сосредоточилось на упрощении перехода между задачами пользователей, что вызвало проблемы безопасности данных. Второе поколение характеризовалось улучшением ввода-вывода и стандартизацией обработки прерываний, но вопросы безопасности данных оставались нерешенными. К концу 60-х годов начался переход к мультипроцессорным системам, что усугубило проблемы распределения ресурсов и их защиты. Это привело к разработке ОС с аппаратными средствами защиты, такими как защита памяти и контроль. Основная тенденция развития вычислительной техники — максимальная доступность для пользователей, что противоречит требованиям безопасности данных. Механизмы защиты в ОС включают все средства, обеспечивающие защиту данных, и такие ОС часто называют защищенными системами.

Безопасность ОС

Безопасность ОС — это состояние, при котором невозможно случайное или преднамеренное нарушение ее функционирования и безопасности управляемых ресурсов. Выделим особенности ОС, которые делают вопросы обеспечения безопасности отдельной категорией:

- Управление всеми ресурсами системы;
- Наличие встроенных механизмов, которые прямо или косвенно влияют на безопасность программ и данных, работающих в среде ОС;
- Обеспечение интерфейса пользователя с ресурсами системы;
- Размеры и сложность ОС.

Безопасность ОС

Большинство ОС обладают дефектами с точки зрения обеспечения безопасности данных в системе, что обусловлено выполнением задачи обеспечения максимальной доступности системы для пользователя.

Типовые функциональные дефекты ОС:

1. Слабые пароли – пользователи выбирают простые комбинации (например, 12345), которые легко взломать.
2. Незащищенное хранение паролей – если ОС хранит пароли в открытом виде, злоумышленник может их украсть.
3. Неограниченные попытки входа – без блокировки после нескольких ошибок можно подбирать пароли методом перебора.
4. Общая память – данные одной программы могут остаться в ОЗУ и попасть к другой (например, к вредоносной).
5. Избыточные привилегии – если программа получает больше прав, чем нужно, это

Модели управления доступом

Дискреционная модель (DAC)

Базируется на явно заданных для каждого субъекта (пользователя) правах доступа к объектам/сегментам информации. Они представляются в виде матрицы, в которой указываются полномочия субъекта относительно каждого объекта или сегмента информации.

Модель довольно простая в реализации. Не требует использования сложных технических средств. Но она не подходит для информационных систем (далее – ИС) с множеством субъектов (сотрудников). Чем больше сотрудников, тем сложнее организовать управление правами пользователей в компании на основе модели DAC по причине: - Сложности централизованного контроля, - Рассредоточенности управления, - Оторванности прав доступа от данных (их содержания), - Большого количества жестких связей между объектами и субъектами, а также зависимостей, которые сложно отслеживать.

Мандатная модель (MAC)

Основные принципы ее построения позаимствованы из правил секретного документооборота, использование которых практикуется во многих странах. В системе управления пользователями данных, применяющей эту модель, всем субъектам (сотрудникам) и объектам (единицам информации: файлам, папкам и так далее) назначаются метки (мандаты), соответствующие разным уровням конфиденциальности. Субъект имеет право на чтение только тех объектов, уровень конфиденциальности которых не выше его уровня. Право на запись/изменение у субъекта есть при условии, что уровень конфиденциальности объекта не ниже его.

При использовании этого подхода не требуется столь высокого уровня детализации отношений между объектами и субъектами, как в предыдущем случае (DAC-модель). Просто изменив метку (мандат), можно управлять доступом к объекту для большого количества субъектов. И наоборот – при изменении метки субъекта, ему можно открывать/запрещать доступ сразу к некоторому объему данных.

Ролевая модель (RBAC)

Подходит для информационных систем с множеством пользователей. Позволяет организовать управление доступом к данным с помощью объектно-ориентированного подхода. Его суть – внедрение между пользователями и их полномочиями (привилегиями) неких сущностей. Они называются ролями. Каждая роль дает сотруднику определенные права на доступ к информации. Одновременно могут быть активными несколько ролей. За счет этого обеспечивается гибкость системы.

Чаще всего RBAC используется в крупных IT-инфраструктурах. Иногда к ней «подмешиваются» элементы других подходов (в частности, мандатного).

Атрибутная модель (ABAC)

Нередко совмещается с ролевой моделью, что позволяет реализовать дополнительные меры по снижению рисков инцидентов за счет учета дополнительных атрибутов (времени, местоположения, кода устройства и т. п.). Например, при ее использовании сотрудникам с заранее определенными ролями доступ к определенным разделам внутренней корпоративной сети предоставляется в соответствии с конкретной комбинацией ID устройства / ID пользователя. Даже если злоумышленникам удастся завладеть данными для входа в корпоративную ИС, воспользоваться ими не получится: в доступе будет отказано, например, при попытке авторизации с устройства, которое не внесено в реестр отношений, создаваемый при внедрении ABAC.

Сравнение моделей

Дискреционная модель (DAC)

- **Принцип**: Права доступа назначает владелец объекта (файла, процесса).
- **Плюсы**:
  - Простота настройки (например, `chmod` в Linux).
  - Гибкость для небольших систем.
- **Минусы**:
  - Нет централизованного контроля.
  - Риск человеческих ошибок (например, случайный доступ к конфиденциальным файлам).

Мандатная модель (MAC)

- **Принцип**: Доступ на основе меток конфиденциальности (администратор задаёт правила).
- **Плюсы**:
  - Максимальная защита (например, запрет доступа к секретным данным даже для владельца).
- **Минусы**:
  - Сложность настройки (требует предварительного проектирования политик).
  - Жёсткие ограничения для пользователей.

Ролевая модель (RBAC)

- **Принцип**: Права выдаются по ролям (например, “администратор”, “гость”).
- **Плюсы**:
  - Удобство для больших организаций (роли = должности).
  - Лёгкость масштабирования.
- **Минусы**:
  - Требует чёткого распределения ролей.
  - Возможны конфликты (если роли пересекаются).

Атрибутная модель (ABAC)

- **Принцип**: Доступ на основе атрибутов (время, местоположение, устройство).
- **Плюсы**:
  - Гибкость (например, доступ только с рабочих ПК в рабочее время).
- **Минусы**:
  - Сложность реализации (нужны мощные системы управления).

Итог:

- DAC – для простых систем.
- MAC – для максимальной защиты.
- RBAC – для бизнеса.
- ABAC – для динамичных сред (облако).

Примеры

SELinux (Security-Enhanced Linux)** реализует мандатную модель в Linux, назначая каждому объекту (файлу, процессу) уровень безопасности. Доступ контролируется политиками, определяющими, какие процессы могут взаимодействовать с объектами. Например, процесс с низким уровнем безопасности не сможет получить доступ к файлу с высоким уровнем, даже если он принадлежит пользователю с правами доступа.

Примеры

Microsoft Windows** использует ролевую модель доступа через Active Directory, где пользователи назначаются в группы с определенными правами. Например, группа “Системные администраторы” имеет полный доступ к серверу, а группа “Пользователи” — только к определенным приложениям, что упрощает управление правами в больших организациях.

Примеры

Современные системы**, такие как AWS (Amazon Web Services), применяют атрибутную модель, предоставляя доступ на основе атрибутов пользователя (роли, местоположение, время доступа). Если сотрудник пытается получить доступ из неавторизованного местоположения, доступ будет отклонен, даже с правильными учетными данными.

Примеры

В Unix и Linux** права доступа к файлам управляются дискреционной моделью, где владелец файла устанавливает права для других пользователей и групп. Например, владелец может разрешить чтение и запись для себя, чтение для группы и запретить доступ для остальных.

Заключение

Безопасность операционных систем — это сложный баланс между удобством пользователей и защитой данных. Каждая модель управления доступом имеет свои сильные стороны и подходит для разных сценариев:

- **DAC (дискреционная)** – гибкая и простая, идеальна для персональных компьютеров (Windows, Linux).
- **MAC (мандтная)** – максимальная защита, используется в госструктурах и военных системах (SELinux).
- **RBAC (ролевая)** – удобна для бизнеса и корпоративных сетей (Active Directory).
- **ABAC (атрибутная)** – гибкость для облачных технологий и сложных ИТ-систем (AWS, современные SaaS-решения).

Выбор модели зависит от требований к безопасности и масштаба системы.