Механизмы защиты информации в современных операционных системах

Лекция 1: Формальные модели безопасности

Краткое содержание лекции

- Формальные модели доступа
  - Дискреционная
  - Мандатная
  - Ролевая
- Дискреционная модель доступа UNIX
- Мандатный и ролевой доступ в современных системах

Формальные модели доступа

- Назначение
  - Формальное выражение политики безопасности
- Основные понятия и определения
  - O – объекты системы,
  - S – субъекты системы,
  - R – права доступа
- Типы моделей
  - Дискреционная (свободная) – Discretionary Access Control,
  - Мандатная (нормативная) – Mandatory Access Control

Дискреционная модель доступа. Модель Харрисона-Руззо-Ульмана

- Множества:
  - объектов (O),
  - субъектов (S),
  - прав доступа (R)
- Матрица прав доступа M
  - ячейка M[s,o] содержит набор {r} прав доступа субъекта s к объекту o
- Состояние системы Q=(S,O,M)
- Эволюция системы во времени через изменение матрицы M

	O1	O2	…	On
S1	r,w	r		r
S2		r,w,x		r,x
…				
Sm	r,w	r,w,x		r

Дискреционная модель (2) Эволюция системы во времени

$command\ \alpha(x_1,..,x_k)$

$if\ \ r_1\ in\ M[x_{s_1},x_{o_1}]\ and$

$\quad r_2\ in\ M[x_{s_2},x_{o_2}]\ and$

$...$

$\quad r_k\ in\ M[x_{s_k},x_{o_k}]$

$then$

$op_1,op_1,...,op_n$

$enter\ r\ to\ M[s,o]$

$delete\ r\ \ from\ M[s,o]$

$create\ subject\ s$

$delete\ subject\ s$

$create\ object\ o$

$delete\ object\ o$

- Модель системы:
  - наборы прав доступа, начальных субъектов и объектов,
  - начальная матрица доступа
  - набор команд C
- Формальный критерий безопасности: **Начальное состояние Q является безопасным относительно права r если не существует последовательности команд, добавляющих в ячейку матрицы M право r, изначально отсутствующее в этой ячейке**
- Проблема безопасности системы:
  - Алгоритмически неразрешима,
  - Разрешима при ограничениях на команды из множества C
  - Ограничения на команды существенно снижают практическую применимость модели

Мандатная модель доступа. Модель Белла-ЛаПадулы

- Сопоставление субъектам и объектам уровней безопасности
- Наличие двух операций доступа – read и write
- Два простых правила:
  - Запрет чтения сверху
  - Запрет на запись вниз
- Модель системы: $\Sigma(v,R,T)$, где $T:(V\times R)\to V$ – функция переходов
- Понятие безопасности состояния: безопасность по чтению и записи
- Критерий безопасности системы – достижимость только безопасных состояний
- Проблема безопасности системы:
  - Основная теорема безопасности Белла-ЛаПадула:

    Система безопасна тогда и только тогда, когда начальное состояние безопасно и при любом переходе не возникает и не сохраняется небезопасных отношений доступа

- P.S. Уровень безопасности: число или нечто большее ?
  - Уровень допуска + доступные категории

Мандатная модель (2)
Модель Биба

- К свойствам конфиденциальности добавляются свойства целостности
- Характеристика целостности
  - Либо запрет на общение субъекта и объекта с разным уровнем целостности
  - Либо снижение уровня целостности при взаимодействии

- P.S. Мандатные модели – на каком уровне абстракции они действуют ?

Ролевой контроль доступа (Role Based Access Control)

- Замещение понятия “субъект” понятиями “пользователь” и “роль”
- Идея состоит в том, что полномочиями наделяются не конкретные пользователи, а абстрактные роли
- Ролевая модель:
  - U – множество пользователей
  - R – множество ролей
  - P – множество полномочий на доступ к объектам
  - S – множество сеансов пользователей
  - Отношение PA – отображает множество полномочий на множество ролей,
  - Отношение UA – отображает множество пользователей на множество ролей
  - Правила управления доступом определяются функциями:
    - user: S->U
    - roles: S->R
    - permissions : S->P

UA

PA

x			x
	x		
		x	x

user

roles

Ролевой контроль доступа (2)

- Критерий безопасности:

  Система считается безопасной, если любой пользователь системы, работающий в сеансе s, может осуществлять действия, требующие полномочия p, только в том случае, когда p є permissions(s)
- Управление доступом через назначение набора доступных ролей.
- Иерархия ролей.
- Различные типы ролевых моделей в зависимости от вида PA, UA, user и roles отражают различные виды распределения полномочий и ответственности:
  - Взаимоисключающие роли (статическое разделение обязанностей),
  - Ограничения на одновременное использование ролей в рамках одного сеанса (динамическое разделение обязанностей),
  - Количественные ограничения на назначение ролей,
  - Группирование ролей и полномочий

Типовая дискреционная модель доступа в UNIX

- Наличие у каждого объекта системы (файла) владельца (пользователя)
- Наличие идентификаторов пользователя и группы пользователя (символических для аутентификации в системе и числовых для определения прав доступа)
- Атрибуты объекта определяют права доступа к нему:
  - чтение, запись, выполнение
  - со стороны владельца, группы и остальных пользователей

  XYZ, где X – права владельца, Y – группы, Z – остальных пользователей.

  Каждое из чисел получается сложением битов, указывающих на наличие права: 4 – право на чтение, 2 – на запись, 1 – на выполнение.

  Альтернативное представление:

  –rw– r–– –r– user users data-type0.5.4.txt.html

  drwx ––– ––– user users test/

  drwx r–x r–x user users todo/

- Владелец объекта имеет возможность устанавливать права доступа к объекту по своему усмотрению

Типовая дискреционная модель доступа в UNIX (2)

- Суперпользователь
  - Root (UID = 0)
  - Имеет доступ ко всем объектам системы, а так же может менять права доступа к объектам и их владельцев
- Исполнение программ, права программы
  - По умолчанию, программа исполняется от имени пользователя, запустившего ее и получает эффективный идентификатор этого пользователя,
  - Тем самым программа получает права пользователя, запустившего ее,
  - login как начало сеанса работы пользователя в системе
- Иногда требуется, чтобы программа запускалась с правами владельца, а не реального пользователя
  - Выполнение специфичных операций,
  - Атрибут setUID

Мандатный и ролевой доступ в Linux. SELinux

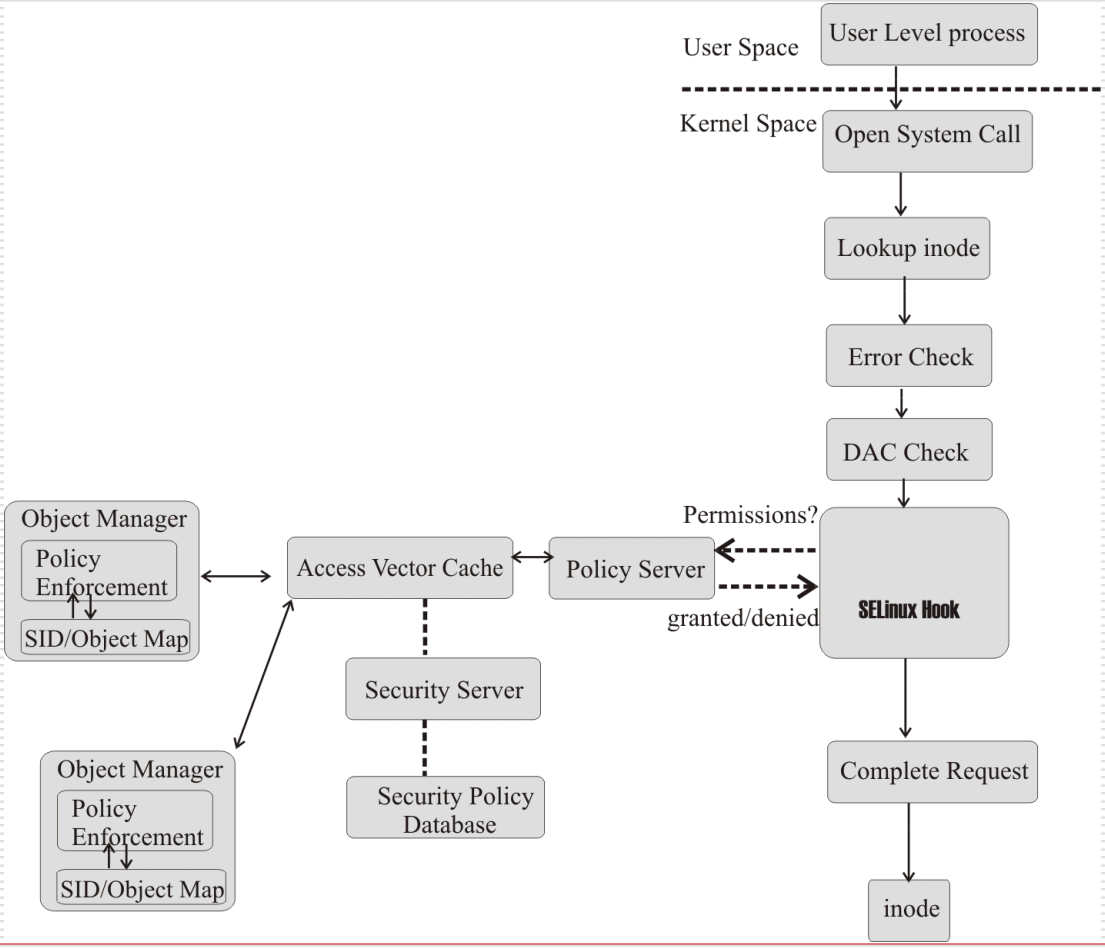
- Реализация системы принудительного контроля доступа, оформленная в виде модулей ядра
- Реализует мандатный и ролевой контроль доступа поверх стандартной дискреционной модели UNIX
- Контроль доступа осуществляется со стороны ядра операционной системы и прозрачно для приложений
- Управляет файловой системой, дескрипторами, сокетами, сообщениями и сетевыми интерфейсами
- Правила доступа задаются т.н. политиками SELinux
  - Задают типы ресурсов и домены приложений,
  - Описывают разрешенные операции доступа для приложений из доменов над ресурсами заданных типов,
  - Определяют порядок наследования доменов и ролей запускаемыми приложениями
- Различные варианты политик:
  - “целевая” политика (накладывает ограничения на стандартные приложения, не ограничивает остальные),
  - “строгая” политика и принцип наименьших привилегий (запрещено все, что не разрешено)

Мандатный и ролевой доступ в Linux (2). SELinux. Политики безопасности

```
# Определение типов для сетевых приложений и публичных документов
# и разрешение объектам этого типа получать доступ к публичных документов
type net_app_t;
/usr/sbin/net_app.*        --          net_app_t
type public_doc_t;
/home/user/public/*        --          public_doc_t
allow net_app_t public_doc_t :file {read write}


# Разрешение пользователям запускать утилиту tcpdump
domain_auto_trans(userdomain, netutils_exec_t, netutils_t)
in_user_role(netutils_t)
allow netutils_t user:chr_file rw_file_perms;
```

Схема принятия решения в SELinux



Мандатный и ролевой доступ в Linux (3). Другие системы

- GRSecurity
  - Набор патчей к ядру
  - RBAC
- AppArmor
  - Модуль ядра
  - Профили определяют к каким ресурсам и с какими правами может иметь доступ приложение

Вопросы ?