Слайд 1

Лекция 4
МОДЕЛИ БЕЗОПАСНОСТИ ОСНОВНЫХ ОПЕРАЦИОННЫХ СИСТЕМ

Слайд 2

Учебные вопросы:
1. Основные модели логического управления доступом в ОС.
2. Системы разграничения доступа.

Слайд 3

1. Основные модели разграничения доступа в ОС

Матричная модель доступа (модель Харрисона-Руззо-Ульмана)

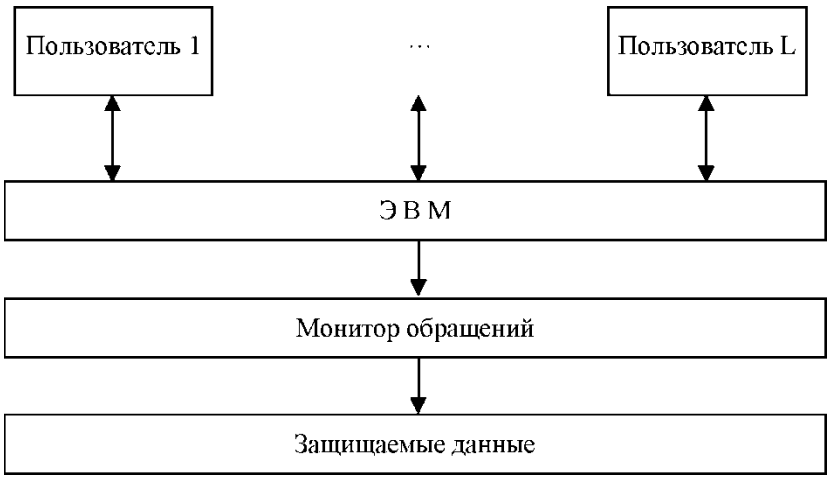

Рисунок 1 - Схема модели Харрисона, Руззо и Ульмана

Разработка и практическая реализация различных защищенных ОС привела Харрисона, Руззо и Ульмана к построению формальной модели защищенных систем. Схема модели Харрисона, Руззо и Ульмана (HRU-модели) приведена на рисунке 1.

Рассмотрим формальную постановку задачи в традиционной трактовке.

Имеется совокупность **субъектов** и набор **объектов**. ***Задача логического управления доступом состоит в том, чтобы для каждой пары "субъект-объект" определить множество допустимых операций и контролировать выполнение установленного порядка.***

Слайд 4

Отношение "субъекты-объекты" можно представить в виде **матрицы доступа**, в строках которой перечислены **субъекты**, в столбцах - **объекты**, а в клетках, расположенных на пересечении строк и столбцов, *записаны дополнительные условия* (например, время и место действия) и *разрешенные виды доступа*. Фрагмент матрицы может выглядеть, например, как показано на рисунке 2.

	Файл	Программа	Линия связи	Реляционная таблица
Пользователь 1	o r w с системной консоли	e	rw с 8:00 до 18:00	
Пользователь 2				a

Обозначение: "o" – разрешение на передачу прав доступа другим пользователям, "r" – чтение, "w" – запись, "e" – выполнение, "a" – добавление информации.

Рисунок 2 - Фрагмент матрицы доступа

Тема логического управления доступом - одна из сложнейших в области информационной безопасности.

Дело в том, что само понятие объекта (а тем более видов доступа) меняется от *сервиса к сервису*.

Для **операционной системы** к объектам относятся файлы, устройства и процессы. Применительно к файлам и устройствам обычно рассматриваются права на **чтение**, **запись**, **выполнение** (для программных файлов), иногда на **удаление** и **добавление**. Отдельным правом может быть возможность передачи полномочий доступа другим субъектам (так называемое **право владения**). Процессы можно создавать и уничтожать. Современные операционные системы могут поддерживать и другие объекты.

Для систем управления **реляционными базами данных** объект - это **база данных, таблица, представление, хранимая процедура**. К таблицам применимы операции поиска, добавления, модификации и удаления данных.

В результате при задании матрицы доступа нужно принимать во внимание не только принцип распределения привилегий для каждого сервиса, но и существующие связи между сервисами (приходится заботиться о согласованности разных частей матрицы).

Аналогичная трудность возникает при экспорте/импорте данных, когда информация о правах доступа, как правило, теряется (поскольку на новом сервисе она не имеет смысла).

Следовательно, обмен данными между различными сервисами представляет особую опасность с точки зрения управления доступом, а при проектировании и реализации разнородной конфигурации необходимо позаботиться о согласованном распределении прав доступа субъектов к объектам и о минимизации числа способов экспорта/импорта данных.

Матрицу доступа, ввиду ее разреженности (большинство клеток - пустые), неразумно хранить в виде двухмерного массива. Обычно ее хранят по столбцам, т.е. для каждого объекта поддерживается список "допущенных" субъектов вместе с их правами. Элементами списков могут быть имена групп и шаблоны субъектов, что служит большим подспорьем администратору.

Списки доступа - исключительно гибкое средство. С их помощью легко выполнить требование о гранулярности прав с точностью до пользователя. Посредством списков несложно добавить права или явным образом запретить доступ (например, чтобы наказать нескольких членов группы пользователей).

Слайд 6

Безусловно, списки являются лучшим средством произвольного управления доступом.

Подавляющее большинство операционных систем и систем управления базами данных реализуют именно произвольное управление доступом.

Основное достоинство произвольного управления - *гибкость*. К сожалению, у "произвольного" подхода есть ряд недостатков.

1) Рассредоточенность управления доступом ведет к тому, что доверенными должны быть многие пользователи, а не только системные операторы или администраторы. *Из-за рассеянности или некомпетентности сотрудника, владеющего секретной информацией, эту информацию могут узнать и все остальные пользователи. Следовательно, произвольность управления должна быть дополнена жестким контролем за реализацией избранной политики безопасности.*

2) Права доступа существуют отдельно от данных. *Ничто не мешает пользователю, имеющему доступ к секретной информации, записать ее в доступный всем файл или заменить полезную утилиту ее "троянским" аналогом. Подобная "разделенность" прав и данных существенно осложняет проведение несколькими системами согласованной политики безопасности и, главное, делает практически невозможным эффективный контроль согласованности.*

*Возвращаясь к вопросу **представления матрицы доступа**, укажем, что для этого можно использовать также функциональный способ, когда матрицу не хранят в явном виде, а каждый раз вычисляют содержимое соответствующих клеток.*

***Например,** при принудительном управлении доступом применяется сравнение меток безопасности субъекта и объекта.*

*Удобной надстройкой над средствами логического управления доступом является ограничивающий интерфейс, когда пользо*вателя *лишают самой возможности попытаться совершить несанкционированные действия, включив в число видимых ему объектов только те, к которым он имеет доступ. Подобный подход обычно реализуют в рамках системы меню (пользователю показывают лишь допустимые варианты выбора) или посредством ограничивающих оболочек, таких как restricted shell в ОС Unix.*

При принятии решения о предоставлении доступа обычно анализируется следующая **информация**:

1)**идентификатор субъекта** (идентификатор пользователя, сетевой адрес компьютера и т.п.). Подобные идентификаторы являются основой произвольного (или дискреционного) управления доступом;

2)**атрибуты субъекта** (метка безопасности, группа пользователя и т.п.).

Слайд 7

Системы в модели Харрисона-Руззо-Ульмана

Любая система в модели Харрисона-Руззо-Ульмана характеризуется матрицей доступа M, конечным количеством прав $R=\{r_1,r_2\ldots r_n\}$, объектов $O=\{o_1,o_2\ldots o_n\}$, субъектов $S=\{s_1,s_2,\ldots s_n\}$ и операций $A=\{a_1,a_2,\ldots a_n\}$. Система

является монооперационной, если каждая команда a_i данной системы выполняет лишь одну элементарную операцию ***op***.

Изменения в эту матрицу вводятся с помощью специальных команд.

Переходы между состояниями системы

Для того, чтобы был возможен переход из ***Q=(S,O,M)*** в ***Q'=(S',O',M'}***, нужно ввести некоторые элементарные операции. Для этой цели в данной модели существует шесть операторов ***op***:

1. Добавление права *r* в ячейку *M[s,o]*:

op = enter r into M[s,o]

2. Удаление права *r* из ячейки *M[s,o]*:

op = delete r from M[s,o]

3. Создание нового субъекта *s*:

op = create subject s'

4. Создание нового объекта *o*:

op = create object o'

5. Удаление субъекта s:

op = destroy subject s

6. Удаление объекта *o*:

op = destroy object o

Пример команды

Примером команды, которая является комбинацией элементарных операций ***op***, может служить создание файла ***F*** пользователем ***P*** и получение им прав на чтение ***read***:

Command CreateFile (F,P)
create object F,
enter read into M[P,F]
end.

Слайд 8

Критерий безопасности системы

Для заданной системы исходное состояние $Q_0=(S_0,O_0,M_0)$ называется безопасным относительно права ***r***, если не существует такой последовательности команд, которая изменила бы заданное начальное состояние системы так, что право ***r*** записалось бы в ячейку ***M[s;o]*** в которой оно отсутствовало в начальном состоянии Q_0. Если это условие не выполнено, то произошла **утечка информации**.

*Пусть в начальном состоянии в системе имеются три субъекта: **o**, **s** и **t**; **s** обладает правом записи по отношению к **t**, а **t** обладает некоторым правом **a** (которое может представлять собой либо **r**, либо **w**) по отношению к **o**. Покажем, как субъект **s** может получить право доступа **a** по отношению к субъекту **o**.*

1. Система находится в начальном состоянии.

*2. Субъект **s** создаёт новый субъект **x**, по отношению к которому автоматически получает права чтения и записи.*

*3. Субъект **s** передаёт субъекту **t** права чтения и записи по отношению к **x**.*

*4. Субъект **t** передаёт субъекту **x** право доступа **a** по отношению к **o**.*

*5. Субъект **s** получает от субъекта **x** право доступа **a** по отношению к **o**.*

Приведённая последовательность операций проиллюстрирована на рисунке 3.

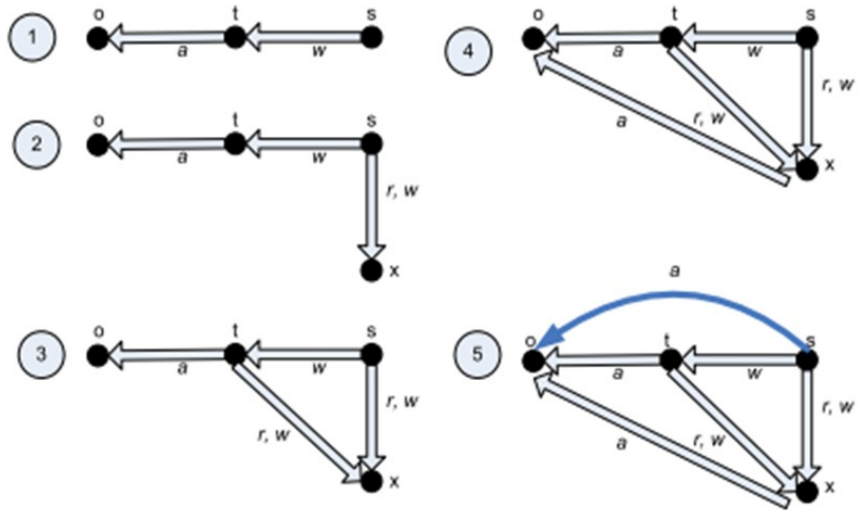

Рис. 3 Последовательность операций в модели HRU, приводящая к утечке информации

Слайд 9

Преимущества и недостатки

Преимущества

1. Простота и наглядность, так как для данной модели не требуется сложных алгоритмов.
2. Эффективность в управлении, так как возможно управление правами пользователей с точностью до операции.
3. Сильный критерий безопасности.

Недостатки

1. Не существует алгоритма проверки на безопасность для произвольной системы.
2. Уязвимость к атаке с помощью «троянского коня», так как в данной модели не существует контроля за потоками информации между субъектами.

Применение

Несмотря на то, что модель Харрисона-Руззо-Ульмана не предоставляет алгоритма проверки на безопасность для произвольной системы, данная модель нашла свою нишу в мире и используется при формальной верификации корректности построения систем разграничения доступа в высокозащищённых автоматизированных системах. На данный момент создание и развитие моделей управления доступом заключается в основном в проектировании различных модификаций модели Харрисона-Руззо-Ульмана и в поиске условий, при которых бы задача определения безопасности была алгоритмически разрешимой.

Сравнение с аналогом

Аналогом модели Харрисона-Руззо-Ульмана является модель Белла-Лападулы, которая предусматривает разделение объектов и субъектов на

уровни секретности, а также контролирует поток данных от субъекта к субъекту. Так субъект, имеющий менее высокий уровень секретности, чем у объекта, не получит права доступа к этому объекту. Это создает ряд преимуществ и недостатков по сравнению с моделью Харрисона-Руззо-Ульмана. Модель Белла — Лападулы не подвержена атакам с помощью троянских коней, более безопасна. Но модель Харрисона-Руззо-Ульмана более эффективна в управлении и проста в использовании. Поэтому каждая из этих моделей нашла свое применение.

Слайд 10

Модель безопасности Белла-ЛаПадулы

Модель Белла — Лападулы — модель контроля и управления доступом, основанная на мандатной модели управления доступом. В модели анализируются условия, при которых невозможно создание информационных потоков от субъектов с более высоким уровнем доступа к субъектам с более низким уровнем доступа (Рис.4).

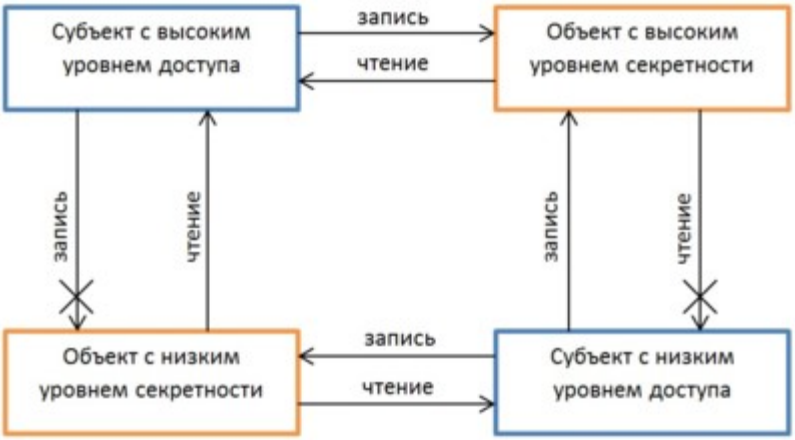

Рис. 4. Модель безопасности Белла-ЛаПадулы

Метки безопасности - основа мандатного управления доступом.

Основу *мандатной* политики (модель Белла-ЛаПадулы) безопасности составляет мандатное управление доступом:

- все субъекты и объекты должны быть идентифицированы;
- задан линейно упорядоченный набор *меток секретности*;
- каждому объекту системы присвоена метка секретности, определяющая ценность содержащейся в нем информации — его *уровень секретности;*
- каждому субъекту системы присвоена метка секретности, определяющая уровень доверия к нему — его *уровень доступа*;
- решение о разрешении доступа субъекта к объекту принимается исходя из типа доступа и сравнения метки субъекта и объекта.

История

Классическая модель Белла — Лападулы была описана в 1975 году сотрудниками компании MITRE Corporation Дэвидом Беллом и Леонардом Лападулой, к созданию модели их подтолкнула система безопасности для работы с секретными документами Правительства США.

Суть системы заключалась в следующем: *каждому субъекту (лицу, работающему с документами) и объекту (документам) присваивается*

метка конфиденциальности, начиная от самой высокой («особой важности»), заканчивая самой низкой («несекретный» или «общедоступный»). Причем субъект, которому разрешён доступ только к объектам с более низкой меткой конфиденциальности, не может получить доступ к объекту с более высокой меткой конфиденциальности. Также субъекту запрещается запись информации в объекты с более низким уровнем безопасности.

Особенности

Модель Белла — Лападулы является моделью разграничения доступа к защищаемой информации. Она описывается конечным автоматом с допустимым набором состояний, в которых может находиться информационная система. Все элементы, входящие в состав информационной системы, разделены на две категории — субъекты и объекты. Каждому субъекту присваивается свой уровень доступа, соответствующий степени конфиденциальности. Аналогично, объекту присваивается уровень секретности.

Слайд 11

Понятие защищённой системы определяется следующим образом: каждое состояние системы должно соответствовать политике безопасности, установленной для данной информационной системы. Переход между состояниями описывается **функциями перехода**. *Система находится в безопасном состоянии в том случае, если у каждого субъекта имеется доступ только к тем объектам, к которым разрешен доступ на основе текущей политики безопасности.* Для определения, имеет ли субъект права на получение определенного вида доступа к объекту, уровень секретности субъекта сравнивается с уровнем секретности объекта, и на основе этого сравнения решается вопрос, предоставить или нет запрашиваемый доступ.

Слайд 12

Наборы уровень доступа/уровень секретности описываются с помощью матрицы доступа. Основными правилами, обеспечивающими разграничение доступа, являются следующие:

Простое свойство безопасности (The Simple Security)

Субъект с уровнем доступа x_s может читать информацию из объекта с уровнем секретности x_0 тогда и только тогда, когда x_s преобладает над x_0. Это правило также известно под названием «нет чтения верхнего» (NRU). Например, если субъект, имеющий доступ только к несекретным данным, попытается прочесть объект с уровнем секретности совершенно секретно, то ему будет отказано в этом.

Свойство * (The *-property)

Субъект с уровнем секретности x_s может писать информацию в объект с уровнем безопасности x_0 только если x_0 преобладает над x_s. Это правило также известно под названием «нет записи вниз» (NWD). Например, если субъект, имеющий уровень доступа **совершенно секретно**, попытается записать в объект с уровнем секретности **секретно**, то ему будет отказано в этом.

Дискреционное свойство безопасности (The Discretionary Security Property)

Заключается в том, что права дискреционного доступа субъекта к объекту определяются на основе матрицы доступа.

Математическое описание модели

Обозначения

- *S — множество субъектов;*
- *O — множество объектов,* $S \subset O$*;*
- $R = \{r, w\}$ *— множество прав доступа,* ***r*** *— доступ на чтение,* ***w*** *— доступ на запись;*
- *L = {U, SU ,S ,TS} — множество уровней секретности, U — Unclassified (несекретный), SU — Sensitive but unclassified (конфиденциальный), S — Secret (секретный), TS — Top secret (сов.секретный);*
- $\Lambda = (L, \leq, \bullet, \otimes)$ *— решётка уровней секретности, где:*
  - $\leq$ *— оператор, определяющий частичное нестрогое отношение порядка для уровней секретности;*
  - $\bullet$ *— оператор наименьшей верхней границы;*
  - $\otimes$ *— оператор наибольшей нижней границы.*
- *V — множество состояний системы, представляемое в виде набора упорядоченных пар (F,M), где:*
  - $F : S \cup O \to L$ *— функция уровней секретности, ставящая в соответствие каждому объекту и субъекту в системе определённый уровень секретности;*
  - *M — матрица текущих прав доступа.*

Оператор отношения $\leq$ *обладает следующими свойствами:*

- *Рефлексивность: данное свойство означает, что между субъектами и объектами одного уровня безопасности передача информации разрешена.*
- *Антисимметричность: свойство означает, что если информация может передаваться как от субъектов и объектов уровня A к субъектам и объектам уровня B, так и от субъектов и объектов уровня B к субъектам и объектам уровня A, то эти уровни эквивалентны.*
- *Транзитивность: свойство означает, что если информации может передаваться от субъектов и объектов уровня A к субъектам и объектам уровня B, и от субъектов и объектов уровня B к субъектам и объектам уровня C, то она может передаваться от субъектов и объектов уровня A к субъектам и объектам уровня C.*

Смысл операторов наименьшей верхней границы и наибольшей нижней границы заключается в том, что для каждой пары элементов множества уровней безопасности всегда можно указать единственный элемент, ограничивающий ее сверху или снизу таким образом, что между ними и этим элементом не будет других элементов.

Система $\sum = (v_0, R, T)$ *в модели Белла — Лападулы состоит из следующий элементов:*

- v_0 *— начальное состояние системы;*
- R *— множество прав доступа;*
- T *— функция перехода, которая в ходе выполнения запросов переводит систему из одного состояния в другое.*

Слайд 13

Определения состояния безопасности

Состояние системы *(F,M)* называется **безопасным по чтению** (или simple-безопасным), если для каждого субъекта, осуществляющего в этом состоянии доступ по чтению к объекту, уровень безопасности субъекта доминирует над уровнем безопасности объекта.

Состояние системы *(F,M)* называется **безопасным по записи** в случае, если для каждого субъекта, осуществляющего в этом состоянии доступ по записи к объекту, уровень безопасности объекта доминирует над уровнем безопасности субъекта.

Состояние *(F, M)* называется **безопасным**, если оно безопасно по чтению и по записи.

Система называется **безопасной**, если её начальное состояние v_0 безопасно, и все состояния, достижимые из v_0 путём применения конечной последовательности запросов из *R*, безопасны.

Слайд 14

Недостатки

В силу своей простоты, классическая модель Белла — Лападулы имеет ряд ***серьёзных недостатков:***

1) Деклассификация

Данная уязвимость заключается в следующем: **классическая модель не предотвращает систему от деклассификации объекта (изменение уровня секретности объекта вплоть до «не секретно» по желанию «совершенно секретного» субъекта)**. *Например, пусть субъект с высоким уровнем доступа А читает информацию из объекта того же уровня секретности. Далее он понижает свой уровень доступа до низкого Б, и записывает считанную ранее информацию в объект, низкого уровня секретности Б. Таким образом, хотя формально модель нарушена не была, безопасность системы нарушена.*

Для решения этой проблемы вводят правила:

- **Правило сильного спокойствия** — уровни безопасности субъектов и объектов никогда не меняются в ходе системной операции.
- **Правило слабого спокойствия** — уровни безопасности субъектов и объектов никогда не меняются в ходе системной операции таким образом, чтобы нарушить заданную политику безопасности.

2) Удаленное чтение.

Данный недостаток проявляет себя в распределенных компьютерных системах. *Допустим, субъект А с высоким уровнем доступа пытается прочитать информацию из объекта Б с низким уровнем секретности.*

Может создаться впечатление, что если субъекту А будет разрешено чтение информации из объекта Б, никакая конфиденциальная информация не будет раскрыта.

Однако это не так.

Во время операции чтения между удаленными объектами происходит появления потока информации от читаемого объекта к запросившему доступ на чтение субъекту. Поток, который при этом появляется, является безопасным, так как информация недоступна неавторизированным субъектам. Однако в распределенной системе чтение инициируется запросом от одного объекта к другому. Такой запрос образует поток информации идущий в неверном направлении (запись в объект с более низким уровнем секретности). Таким образом, удаленное чтение в распределенных системах может произойти, только если ему предшествует операция записи вниз, что является нарушением правил классической модели Белла — Лападулы.

Слайд 15

Модель дискреционного доступа (DAC)

В дискреционной модели контролируется доступ субъектов (пользователей или приложений) к объектам, представляющим собой различные информационные ресурсы: файлы, приложения, устройства вывода и т. д.

Для каждого объекта существует субъект-владелец, который сам определяет тех, кто имеет доступ к объекту, а также разрешенные операции доступа. Основными операциями доступа являются READ (чтение), WRITE (запись) и EXECUTE (выполнение, имеет смысл только для программ). В модели дискреционного доступа для каждой пары субъект-объект устанавливается набор разрешенных операций доступа.

При запросе субъектом доступа к объекту система ищет субъекта в списке прав доступа объекта. Система разрешит доступ субъекта к объекту, если субъект присутствует в списке и имеет разрешенный требуемый тип доступа. Иначе доступ не предоставляется.

Классическая система дискреционного контроля доступа является «закрытой». Изначально объект не доступен никому, и в списке прав доступа описывается набор разрешений. Также существуют «открытые» системы, в которых по умолчанию все имеют полный доступ к объектам, а в списке доступа описывается набор ограничений.

Такая модель реализована в операционных системах Windows и Linux.

В Linux для каждого файла (все ресурсы в операционной системе Linux представимы в виде файлов, в том числе устройства ввода-вывода) устанавливаются разрешения доступа для трех категорий субъектов: владелец файла, члены той же группы, что и владелец, и все остальные пользователи. Для каждой из этих категорий устанавливаются права на чтение (r), запись (w) и выполнение (x). Набор прав доступа объекта может быть представлен в виде символьной строки. Например, запись «rwxr-xr-» означает, что владелец файла может делать с ним все, что угодно; члены

его группы могут читать и исполнять файл, но не могут записывать, а прочим пользователям доступно только чтение.

***Недостаток** модели DAC заключается в том, что субъект, имеющий право на чтение информации может передать ее другим субъектам, которые этого права не имеют, без уведомления владельца объекта. Таким образом, нет гарантии, что информация не станет доступна субъектам, не имеющим к ней доступа. Кроме того, не во всех автоматизированных ИС каждому объекту можно назначить владельца. Во многих случаях данные принадлежат не отдельным субъектам, а всей системе.*

Слайд 16

Основа дискреционной (избирательной) политики безопасности в ОС Windows:

1) все субъекты и объекты должны быть идентифицированы, идентификация проводится по идентификаторам защиты (Security Identifiers, SID). SID представляет собой числовое значение переменной длины:

S - R - I - SO - S1 - ... - Sn - RID

S - неизменный идентификатор строкового представления SID;

R - уровень ревизии (версия) – в настоящее время 1.

I - (identifier-authority) идентификатор полномочий. Представляет собой 48 битную строку, идентифицирующую компьютер или сеть, который(ая) выдал SID объекту.

Sn - 32-битные коды (количеством 0 и более) субагентов, которым было передано право выдать SID. Значение первых подчиненных полномочий общеизвестно.

RID - 32-битный относительный идентификатор. Он является идентификатором уникального объекта безопасности в области, для которой был определен SID. Например, 500 — обозначает встроенную учетную запись Administrator, 501 — обозначает встроенную учетную запись Guest, а 502 — RID для билета на получение билетов протокола Kerberos;

Слайд 17

Предопределенным пользователям и группам Windows выдает характерные SID, состоящие из SID компьютера или домена и предопределенного RID. В таблице 1 приведен перечень некоторых общеизвестных SID.

Таблица 1. Общеизвестные SID Windows

SID	Название	Описание
S-1-1-0	Все	Группа, в которую входят все пользователи
S-1-5-2	Сеть	Группа, в которую входят все пользователи, зарегистрировавшиеся в системе из сети
S-1-5-7	Анонимный вход	Группа, в которую входят все пользователи, вошедшие в систему анонимно
S-1-5-*домен*-500	Администратор	Учетная запись администратора системы. По умолчанию только эта запись обеспечивает полный контроль системы
S-1-5-домен-	Гость	Учетная запись пользователя-гостя

501		

Слайд 18

Соответствие имени пользователя и его SID можно отследить также в ключе реестра HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ ProfileList

Слайд 19

После аутентификации пользователя процессом Winlogon, все процессы, запущенные от имени этого пользователя будут идентифицироваться специальным объектом, называемым маркером доступа (access token). Если процесс пользователя запускает дочерний процесс, то его маркер наследуются, поэтому маркер доступа олицетворяет пользователя для системы в каждом запущенном от его имени процессе. Основные элементы маркера представлены на рис.1.

SID пользователя	**SID1 . SIDn Идентификаторы групп пользователя**	**DACL по умолчанию**	**Привилегии**	**Прочие параметры**

Рис. 1 Обобщенная структура маркера доступа.

Маркер доступа содержит идентификатор доступа самого пользователя и всех групп, в которые он включен. В маркер включен также DACL по умолчанию - первоначальный список прав доступа, который присоединяется к создаваемым пользователем объектам. Еще одна важная для определения прав пользователя в системе часть маркера - список его привилегий. Привилегии - это права доверенного объекта на совершение каких-либо действий по отношению ко всей системе.

Остальные параметры маркера носят информационный характер и определяют, например, какая подсистема создала маркер, уникальный идентификатор маркера, время его действия.

Необходимо также отметить возможность создания ограниченных маркеров (restricted token), которые отличаются от обычных тем, что из них удаляются некоторые привилегии и его SID-идентификаторы проверяются только на запрещающие правила. Создать ограниченный маркер можно программно, используя API-функцию CreateRestrictedToken, а можно запустить процесс с ограниченным маркером, используя пункт контекстного меню Windows “Запуск от имени другого пользователя”

Слайд 20

Маркер доступа идентифицирует субъектов-пользователей системы. С другой стороны, каждый объект системы, требующий защиты, содержит описание прав доступа к нему пользователей. Для этих целей используется **дескриптор безопасности (Security Descriptor, SD)**. Каждому объекту системы, включая файлы, принтеры, сетевые службы, контейнеры Active Directory и другие, присваивается дескриптор безопасности, который определяет права доступа к объекту и содержит следующие основные атрибуты (рис. 5.3):

- SID владельца, идентифицирующий учетную запись пользователя-

владельца объекта;

- пользовательский список управления доступом (Discretionary Access Control List, DACL), который позволяет отслеживать права и ограничения, установленные владельцем данного объекта. DACL может быть изменен пользователем, который указан как текущий владелец объекта.
- системный список управления доступом (System Access Control List, SACL), определяющий перечень действий над объектом, подлежащих аудиту;
- флаги, задающие атрибуты объекта.

Авторизация Windows основана на сопоставлении маркера доступа субъекта с дескриптором безопасности объекта. Управляя свойствами объекта, администраторы могут устанавливать разрешения, назначать право владения и отслеживать доступ пользователей.

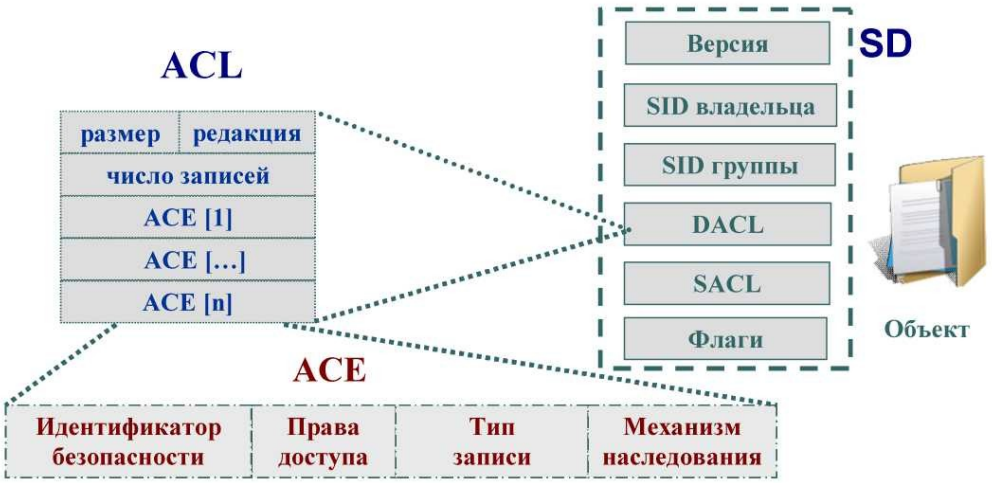

Рис.5.3 Структура дескриптора безопасности объекта Windows

Список управления доступом содержит набор элементов (Access Control Entries, ACE). В DACL каждый ACE состоит из четырех частей: в первой указываются пользователи или группы, к которым относится данная запись, во второй - права доступа, а третья информирует о том, предоставляются эти права или отбираются. Четвертая часть представляет собой набор флагов, определяющих, как данная запись будет наследоваться вложенными объектами (актуально, например, для папок файловой системы, разделов реестра).

Если список ACE в DACL пуст, к нему нет доступа ни у одного пользователя (только у владельца на изменение DACL). Если отсутствует сам DACL в SD объекта - полный доступ к нему имеют все пользователи.

Слайд 21

Если какой-либо поток запросил доступ к объекту, подсистема SRM осуществляет проверку прав пользователя, запустившего поток, на данный объект, просматривая его список DACL. Проверка осуществляется до появления разрешающих прав **на все** запрошенные операции. Если встретится запрещающее правило хотя бы **на одну** запрошенную операцию, доступ не будет предоставлен.

Рассмотрим пример на рис.5.4. Процесс пытается получить доступ к объекту с заданным DACL. В маркере процесса указаны SID запустившего его пользователя, а также SID групп, в которые он входит. В списке DACL объекта присутствуют разрешающие правила на чтение для пользователя с SID=100, и на запись для группы с SID=205. Однако, в доступе пользователю будет отказано, поскольку раньше встречается запрещающее запись правило для группы с SID=201.

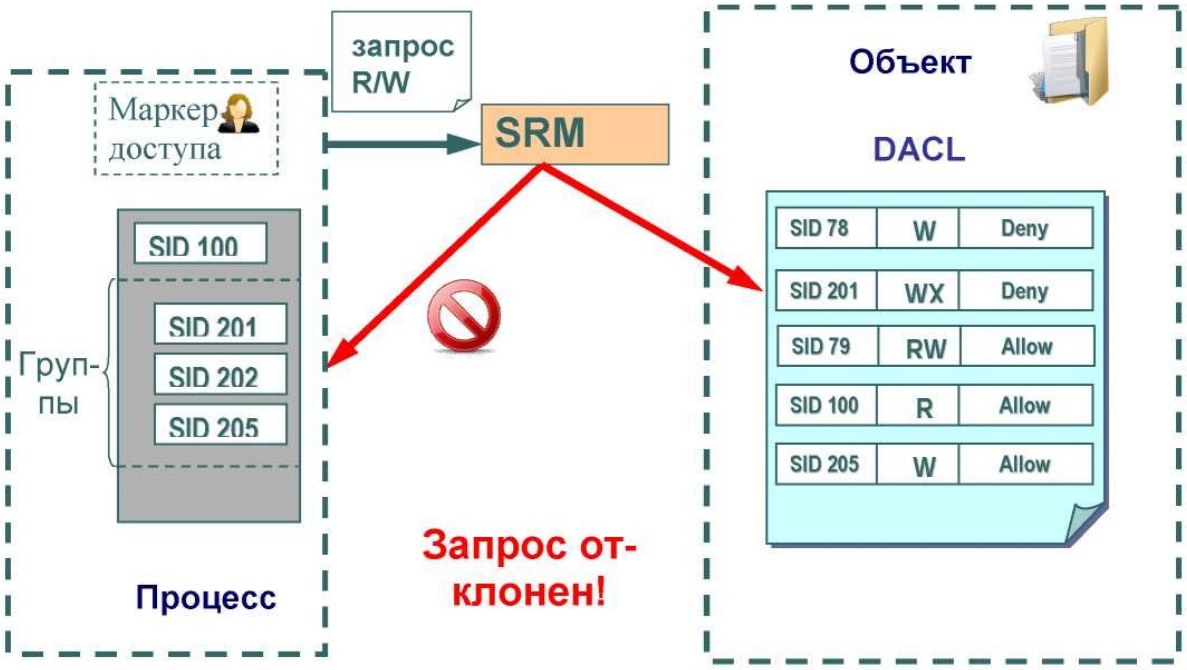

Рис. 5.4 Проверка прав доступа пользователя к объекту

Необходимо отметить, что запрещающее правило помещено в списке DACL на рисунке не случайно.

Запрещающие правила **всегда** размещаются перед разрешающими, то есть являются **доминирующими** при проверке прав доступа.

Слайд 22

Для определения и просмотра прав доступа пользователей к ресурсам можно использовать как графические средства контроля, так и консольные команды. Стандартное окно свойств объекта файловой системы (диска, папки, файла) на вкладке **Безопасность** (рис. 5.5) позволяет просмотреть текущие разрешения для пользователей и групп пользователей, редактировать их, создавать новые или удалять существующие.

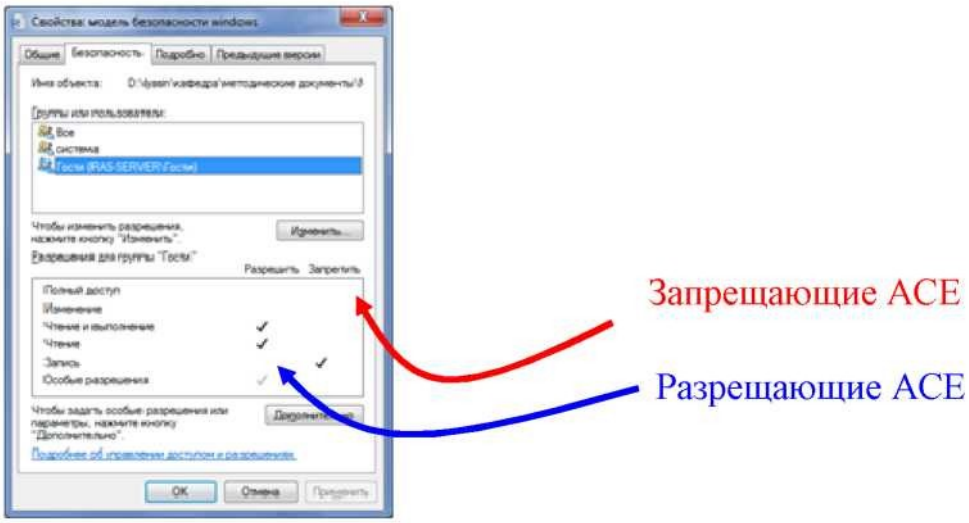

Рис. 5.5 GUI-интерфейс Windows для изменения прав доступа к объектам

При определении прав доступа к объектам можно задать правила их наследования в дочерних контейнерах.

Слайд 23

Ролевая модель контроля доступа (RBAC)

Управление доступом на основе ролей (англ. *Role Based Access Control, RBAC*) — развитие политики избирательного управления доступом, при этом права доступа субъектов системы на объекты группируются с учётом специфики их применения, образуя роли.

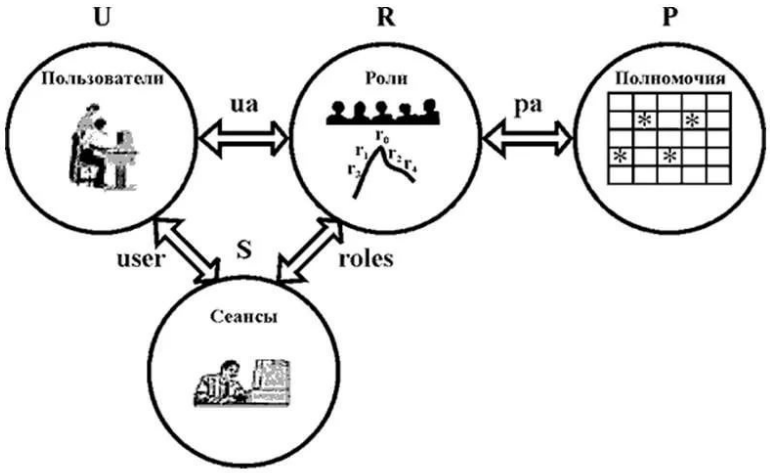

Рис. 5 Ролевая модель управления доступом RBAC

*Формирование ролей призвано определить чёткие и понятные для пользователей компьютерной системы **правила разграничения доступа**. Ролевое разграничение доступа позволяет реализовать гибкие, изменяющиеся динамически в процессе функционирования компьютерной системы **правила разграничения доступа**.*

*Такое разграничение доступа является составляющей многих современных компьютерных систем. Как правило, данный подход применяется в системах **защиты СУБД**, а отдельные элементы реализуются в **сетевых операционных системах**. Ролевой подход часто используется в системах, для пользователей которых чётко определён круг их должностных полномочий и обязанностей.*

Несмотря на то, что Роль является совокупностью прав доступа на объекты компьютерной системы, ролевое управление доступом отнюдь не является частным случаем избирательного управления доступом, так как его правила определяют порядок предоставления доступа субъектам компьютерной системы в зависимости от имеющихся (или отсутствующих) у него ролей в каждый момент времени, что является характерным для систем мандатного управления доступом. С другой стороны, правила ролевого разграничения доступа являются более гибкими, чем при мандатном подходе к разграничению.

Так как привилегии не назначаются пользователям непосредственно и приобретаются ими только через свою роль (или роли), управление индивидуальными правами пользователя по сути сводится к назначению ему ролей. Это упрощает такие операции, как добавление пользователя или смена подразделения пользователем.

История

Элементарные формы модели RBAC были осуществлены во множестве специальных форм на многих системах, начиная с 1970-х годов. Контроль доступа на основе ролей, используемый в настоящее время происходит из модели, предложенной Феррайоло (англ. Ferraiolo) и Куном (англ. Kuhn) (1992) и как образцовая модель позже усовершенствованная Санди (англ. Sandhu), Койн, Фейнштейн и Йоман (1996).

В 2004 году Американский национальный институт стандартов и Международный комитет по стандартам информационных технологий (ANSI/INCITS) принимают предложенную Санди, Феррайоло и Куном модель RBAC как единый стандарт.

Слайд 24

Базовая модель RBAC

Для определения модели RBAC используются следующие соглашения:

- S = Субъект (англ. *Subject*) = Человек или автоматизированный агент (множество пользователей);
- R = Роль (англ. *Role*) = Рабочая функция или название, которое определяется на уровне авторизации (множество ролей);
- P = Разрешения (англ. *Permissions*) = Утверждения режима доступа к ресурсу (множество прав доступа на объекты системы);
- SE = Сессия (англ. *Session*) = Соответствие между S, R и/или P
- SA = Назначение субъекта (англ. *Subject Assignment*)
- PA: $R \rightarrow 2^P$ — функция, определяющая для каждой роли множество прав доступа; при этом для каждого $p \in P$ существует $r \in R$ такая, *что* $p \in PA(r)$; (англ. *Permission Assignment*)
- RH = Частично упорядоченная иерархия ролей (англ. *Role Hierarchy*). RH может быть еще записана так:
- Один субъект может иметь несколько ролей.
- Одну роль могут иметь несколько субъектов.
- Одна роль может иметь несколько разрешений.
- Одно разрешение может принадлежать нескольким ролям.

Слайд 25

Роли назначаются субъектам, вследствие чего субъекты получают те или иные разрешения через роли.

RBAC требует именно такого назначения, а не прямого - назначение разрешений субъектам, иначе это приводит к сложно контролируемым отношениям между субъектами и разрешениями.[3]

На возможность наследования разрешений от противоположных ролей накладывается ограничительная норма, которая позволяет достичь надлежащего разделения режимов.

Возможности и применение

Технология управления доступом на основе ролей достаточно гибка и сильна, чтобы смоделировать как избирательное управление доступом (DAC), так и мандатное управление доступом (MAC).

До разработки RBAC, единственными известными моделями управления доступом были MAC и DAC: если модель была не MAC, то она была DAC, и наоборот. Исследования в 90-х показали, что RBAC не попадает ни в ту, ни в другую категорию.

Роли создаются внутри организации для различных рабочих функций. Определённым ролям присваиваются полномочия (permissions) для выполнения тех или иных операций. Штатным сотрудникам (или другим пользователям системы) назначаются фиксированные роли, через которые они получают соответствующие привилегии для выполнения фиксированных системных функций. В отличие от управления доступом на основе контекста (англ. context-based access control, CBAC), реализация RBAC в чистом виде не принимает во внимание текущую ситуацию (такую как, например, откуда было установлено соединение).

RBAC отличается от списков контроля доступа (англ. access control lists, ACL), используемых в традиционных избирательных системах управления доступом, тем, что может давать привилегии на сложные операции с составными данными, а не только на атомарные операции с низкоуровневыми объектами данных. Например, список контроля доступа может предоставить или лишить права записи в такой-то системный файл, но он не может ограничить то, каким образом этот файл может быть изменен.

Концепции иерархии ролей и ограничений позволяют создать или смоделировать контроль доступа на основе решетки (англ. lattice-based access control, LBAC) средствами RBAC. Таким образом, RBAC может быть основанием и расширением LBAC.

Для больших систем с сотнями ролей, тысячами пользователей и миллионами разрешений, управление ролями, пользователями, разрешениями и их взаимосвязями является сложной задачей, которую нереально выполнить малой группой администраторов безопасности. Привлекательной возможностью является использование самой RBAC для содействия децентрализованному управлению RBAC.

RBAC широко используется для управления пользовательскими привилегиями в пределах единой системы или приложения. Список таких

систем включает в себя Microsoft Active Directory, SELinux, FreeBSD, Solaris, СУБД Oracle, PostgreSQL 8.1, SAP R/3, Lotus Notes и множество других.

Слайд 26

2. Системы разграничения доступа

Конкретное воплощение модели разграничения доступа находят в системе разграничения доступа (СРД).

СРД – это совокупность реализуемых правил разграничения доступа в средствах вычислительной техники или автоматизированных системах.

Многие системы разграничения доступа базируются на концепции диспетчера доступа. В основе этой концепции лежит понятие *диспетчера доступа* — абстрактной машины, которая выступает посредником при всех обращениях субъектов к объектам. Диспетчер доступа использует базу данных защиты, в которой хранятся правила разграничения доступа и на основании этой информации разрешает, либо не разрешает субъекту доступ к объекту, а также фиксирует информацию о попытке доступа в системном журнале.

Слайд 27

Основными требованиями к реализации диспетчера доступа являются:

1. Требование полноты контролируемых операций. Проверке должны подвергаться все операции всех субъектов над всеми объектами системы. Обход диспетчера предполагается невозможным.

2. Требование изолированности. Защищенность диспетчера от возможных изменений субъектами доступа с целью влияния на процесс его функционирования.

3. Требование формальной проверки правильности функционирования.

4. Минимизация используемых диспетчером ресурсов.

База данных защиты строится на основе **матрицы доступа** или **одного из ее представлений**.

Слайд 28

Матрица доступа – это таблица, в которой строки соответствуют субъектам, столбцы – объектам доступа, а на пересечении строки и столбца содержатся правила (разрешения) доступа субъекта к объекту.

Основными недостатками такой матрицы являются ее чрезмерно большая размерность и сложность администрирования. Все взаимосвязи и ограничения предметной области приходится учитывать вручную. Примеры ограничений: права доступа субъекта к файлу не могут превышать его прав доступа к устройству, на котором этот файл размещен; группа пользователей наследует одинаковые полномочия и т. д.

Для преодоления этих сложностей матрица доступа в СРД часто заменяется некоторым ее **неявными представлениями**:

1. Списки управления доступом (Access Control Lists, ACL). Для каждого объекта задан список субъектов, имеющих ненулевые полномочия доступа к ним (с указанием этих полномочий). В результате серьезно экономится память, поскольку из матрицы доступа исключаются все нулевые

значения, составляющие большую ее часть. Списки управления доступом имеют недостатки:

• неудобство отслеживания ограничений и зависимостей по наследованию полномочий субъектов;

• неудобство получения сведений об объектах, к которым имеет какой-либо вид доступа субъект;

• так как списки управления доступом связаны с объектом, то при удалении субъекта возможно возникновение ситуации, при которой объект может быть доступен несуществующему субъекту.

Слайд 29

2. Списки полномочий субъектов. Аналогично ACL с той лишь разницей, что для каждого субъекта задан список объектов, доступ к которым разрешен с указанием полномочий доступа. Такое представление называется профилем субъекта. Оба представления имеют практически идентичные достоинства и недостатки.

3. Атрибутные схемы. Основаны на присвоении субъектам и/или объектам определенных меток, содержащих значения атрибутов. Элементы матрицы доступа не хранятся в явном виде, а динамически вычисляются при каждой попытке доступа для конкретной пары субъект-объект на основе их атрибутов. Помимо экономии памяти достигается непротиворечивость базы данных защиты, а также удобство ее администрирования. Основным недостатком является сложность задания прав доступа конкретного субъекта к конкретному объекту.

Слайд 30

Литература:

1. Учебное пособие / Под редакцией В.П. Иванникова. — Электронное издание. — М.: ИНТУИТ. РУ "Интернет университет информационных технологий", 2005. — 536 с. — ISBN: 5-9556-0044-2

2. Безбогов, А.А. Безопасность операционных систем : учебное пособие / А.А. Безбогов, А.В. Яковлев, Ю.Ф. Мартемьянов. – М. : "Издательство Машиностроение-1", 2007. – 220 с.

3. Шаньгин В. Ф. Информационная безопасность компьютерных систем и сетей: учеб. пособие. — М.: ИД «ФОРУМ»: ИНФРА-М, 2011. — 416 с.

4. Цирлов В.Л. Основы информационной безопасности: краткий курс/ В.Л. Цирлов. – Ростов-на-Дону: Феникс, 2008.

Слайд 31

Контрольные вопросы

1. Контроль доступа к данным, основные модели.
2. Матричная модель доступа (модель Харрисона-Руззо-Ульмана), краткая характеристика, достоинства и недостатки.
3. Системы в модели Харрисона-Руззо-Ульмана, переходы между состояниями системы, критерий безопасности.
4. Мандатная модель управления доступом (модель безопасности Белла-ЛаПадулы), краткая характеристика.

5. Основные правила, обеспечивающие разграничение доступа в мандатной модели управления доступом.

6. Модель дискреционного доступа, краткая характеристика, недостатки.

7. Ролевая модель контроля доступа, краткая характеристика.

8. Системы разграничения доступа, основными требованиями к реализации диспетчера доступа.

9. Неявные формы представления матрицы доступа, достоинства и недостатки.