

Модели безопасности на
основе дискреционной
ПОЛИТИКИ

Выделяют:

- теоретико-множественные (реляционные) модели разграничения доступа
 - пятимерное пространство Хартсона,
 - модели на основе матрицы доступа

 - модели распространения прав доступа
 - модель Харисона-Рузо-Ульмана (HRU),
 - модель типизованной матрицы доступа,
 - теоретико-графовая модель TAKE-GRANT
-

Модели безопасности на основе дискреционной политики

- Общие положения
 - Множество (область) безопасных доступов определяется дискретным набором троек **"Пользователь (субъект)-поток (операция)-объект"**.
 - Конкретные модели специфицируют
 - способ представления области безопасного доступа и
 - механизм проверки соответствия запроса субъекта на доступ области безопасного доступа.
 - Если запрос не выходит за пределы данной области, то он разрешается и выполняется.
 - При этом утверждается, что осуществление такого доступа переводит систему в безопасное состояние.
-

Модели на основе **матрицы доступа**.

Система защиты – совокупность множеств

- исходных объектов $O (o_1, o_2, \dots, o_M)$
- исходных субъектов $S (s_1, s_2, \dots, s_N)$, при этом $S \subseteq O$
- операций (действий) над объектами $Op (Op_1, Op_2, \dots, Op_L)$
- прав, которые м.б. даны субъектам по отношению к объектам $R (r_1, r_2, \dots, r_K)$ – т.н. "общие права"
- NxM матрица доступа A , в которой
 - каждому субъекту соответствует *строка*, а каждому объекту - *столбец*.
 - В ячейках матрицы располагаются права r соотв. субъекта над соотв. объектом в виде набора разрешенных операций Op_i

A =

Субъекты
доступа

		Объекты доступа					
		o_1	o_2	...	o_j	...	o_N
s_1			W				
s_2	r						
...							
s_i					r, W		
...							
s_M							e

Обозначения: W – "изменение объекта";
 r – "чтение объекта";
 e – "запуск объекта на выполнение".

$A[s_i, o_j] = a_{ij}$ - право r из R (т. е. не общее, а конкр. право)

Каждый элемент прав r_k специфицирует совокупность операций над объектом

$$r_k \sim (Op_{1k}, Op_{2k}, \dots, Op_{jk})$$

Реализация матриц доступа

- Матрица доступа –
 - ассоциированный с монитором безопасности объект,
 - содержащий информацию по политике разграничения доступа в конкретной системе.
- Вид МД определяется конкретными моделями и конкретными программно-техническими решениями КС, в которых она реализуется,
 - принцип (структура) организации МД,
 - размещение,
 - а также процессы создания, изменения МД

Способы организации информационной структуры матрицы доступа

- Централизованная единая информационная структура
 - СУБД, системная таблица с назначениями доступа

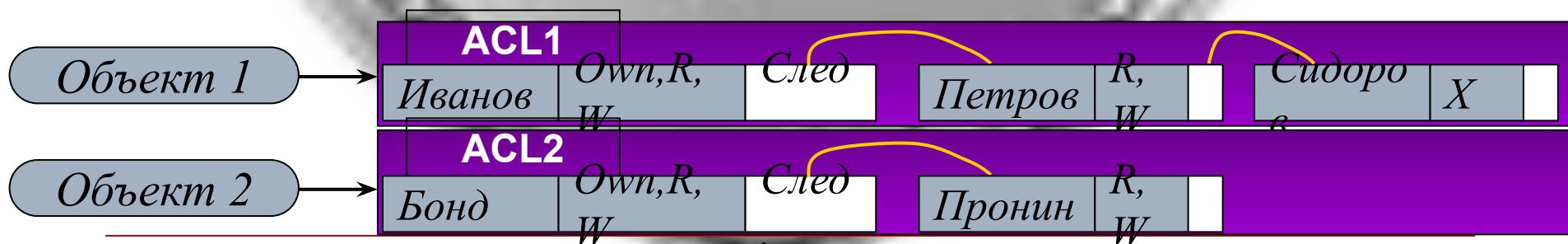
- Децентрализованная распределенная информационная структура
 - Биты защиты (UNIX)
 - Списки доступа (Windows)

Способы организации информационной структуры МД

□ Биты защиты (UNIX)

Биты Объект Объект 2	Владелец			Группа			Остальные польз-ли		
	Чтен	Запис	Выпол	Чтен	Запис	Выпол	Чтен	Запис	Выпол
	1	2	3	4	5	6	7	8	9
Объект 1	1	1	1	1	0	1	0	0	0
Объект 2	1	1	0	1	1	0	1	0	0

□ Списки доступа в файловой системе ОС Windows (Access Control List – ACL)



Способы организации информационной структуры МД. ACL - *Access Control List*

□ Два способа размещения ACL

<i>В спец. системной области</i>	<i>Вместе с объектом</i>
Объекты д.б. зарегистрированы в системе	Д.б. обеспечен контроль целостности ACL

Способы организации информационной структуры МД. ACL - *Access Control List*



□ Структура ACL на примере NTFS

- С каждым объектом *NTFS* связан т.н. дескриптор защиты, состоящий из:

<i>ID влад.</i>	<i>ID перв. гр. влад.</i>	<i>DACL</i>	<i>SACL</i>
-----------------	---------------------------	-------------	-------------

- *DACL* – последовательность произв. кол-ва дескрипторов контроля доступа – *ACE*, вида:

<i>Allowed / Denied</i>	<i>ID субъекта (польз., группа)</i>	<i>Права доступа (отображ-е)</i>	<i>Флаги, атрибуты</i>
-------------------------	-------------------------------------	----------------------------------	------------------------

- *SACL* – системные данные для генерации сообщений аудита

Разновидности моделей на основе МД (основаны на том, как формируются ячейки МД)

- Принудительное управление доступом (жесткость, но четкий контроль)
 - вводится т.н. **доверенный субъект** (администратор доступа), который и определяет доступ субъектов к объектам (централизованный принцип управления)
 - заполнять и изменять ячейки матрицы доступа может только администратор
- Добровольное управление доступом (гибкость, но более сложный контроль)
 - вводится т.н. **владение** (владельцы) объектами и доступ субъектов к объекту определяется по усмотрению владельца (децентрализованный принцип управления)
 - ~~в таких системах субъекты посредством запросов~~
могут изменять состояние матрицы доступа

-
- Пример моделей распространения прав доступа –

*модель
Харрисона-Руззо-Ульмана*

(HRU)

Системы с добровольным управлением доступом – модель Харрисона-Руззо-Ульмана

- Кроме элементарных операций доступа *Read*, *Write* и т.д.,
- вводятся также примитивные операции Op_k по изменению субъектами матрицы доступа:
 - *Enter r into (s,o)* - ввести право r в ячейку (s,o)
 - *Delete r from (s,o)* - удалить право r из ячейки (s,o)
 - *Create subject s* - создать субъект s (т.е. новую строку матрицы A)
 - *Create object o* - создать объект o (т.е. новый столбец матрицы A)
 - *Destroy subject s* - уничтожить субъект s
 - *Destroy object o* - уничтожить объект o

Системы с добровольным управлением доступом – модель Харрисона-Руззо-Ульмана

- Состояние системы Q изменяется при выполнении команд $C(a_1, a_2, \dots)$, изменяющих состояние матрицы доступа A .
- Команды инициируются пользователями-субъектами
- Структура команды

Название

[Условия]
(необяз.)

Операции

```
Command  $\alpha(x_1, \dots, x_k)$   
  if  $r_1$  in  $A[s_1, o_1]$  and  $r_2$  in  $A[s_2, o_2]$  ...  
  then;  $Op_2$  ; ...;  
end
```

x_i – идентификаторы
задействованных
субъектов или
объектов

Системы с добровольным управлением доступом – модель Харрисона-Руззо-Ульмана. Пример команд

Command "создать файл" (s, f) :

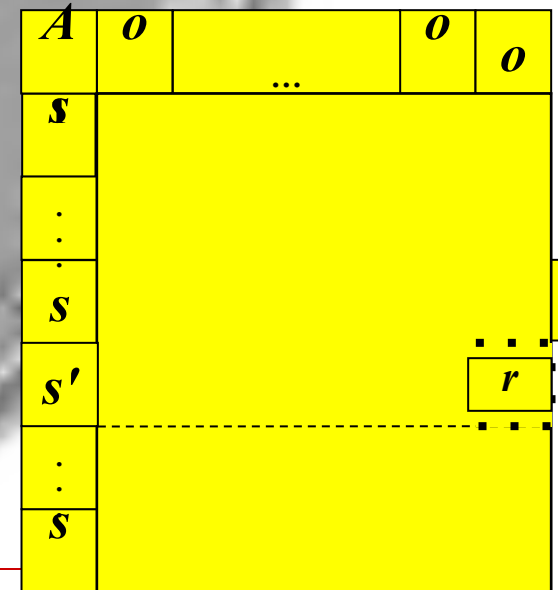
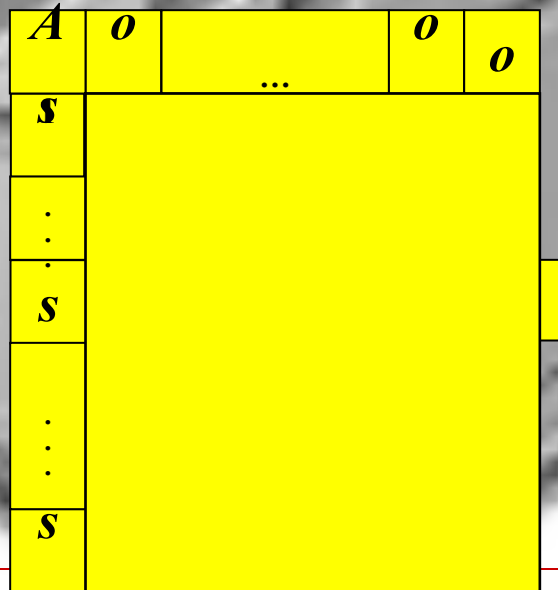
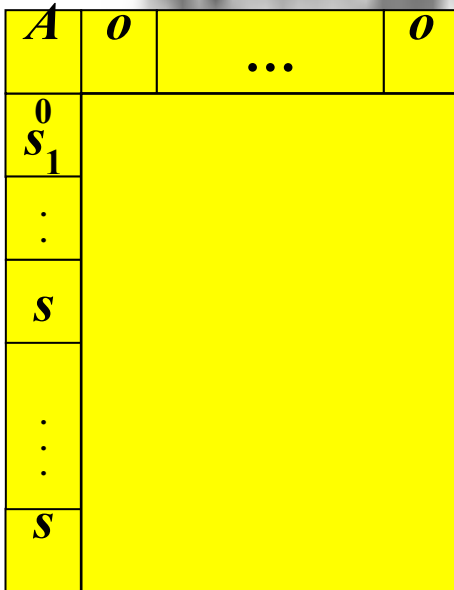
```

Create object  $f$  ;
Enter "own" into  $(s, f)$  ;
Enter "read" into  $(s, f)$  ;
Enter "write" into  $(s, f)$  ;
end
    
```

Command «ввести право чтения» (s, s', f) :

```

if  $own \subseteq (s, f)$  ;
then
    Enter  $r$  "read" into  $(s', f)$  ;
end
    
```



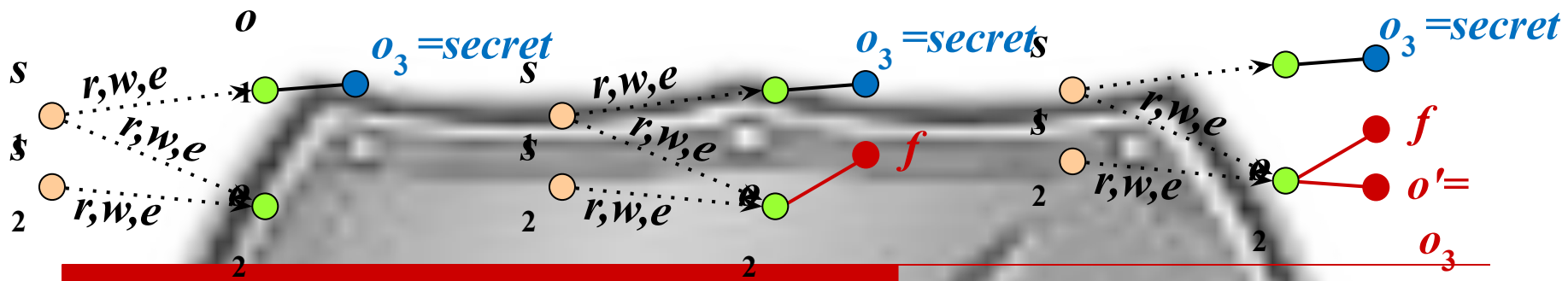
Основной критерий безопасности модели HRU

- $Q_0 = (S_0, O_0, A_0) \rightarrow Q_1 = (S_1, O_1, A_1)$
- Состояние системы с начальной конфигурацией Q_0 **безопасно по праву r** , если не существует (при определенном наборе команд и условий их выполнения) последовательности запросов к системе, которая приводит к записи права r в ранее его не содержащую ячейку матрицы $A_0[s, o]$
- Для модели HRU проблема безопасности
 - разрешима для *моно-операционных* систем
 - в общем случае – не разрешима



Проблема «тройных программ»

- проблема – отсутствие контроля за порождением потоков информации
- в частности, контроля за порождением субъектов, следствием чего могут быть так называемые *"тройные" программы*.
- В модели **HRU** "правильность", легитимность инициируемых из объектов-источников субъектов доступа никак не контролируется.
- В результате, злоумышленник в системе может осуществить неправомерный доступ к информации на основе подмены свойств субъектов доступа.



Command "создать файл"
 (s_2, f):
 if $write \in [s_2, o_2]$;
 then
 Create object f ;
 Enter "read" into $[s_2, f]$;
 Enter "write" into $[s_2, f]$;
 Enter "execute" into $[s_2, f]$;
 if $read \in [s_1, o_2]$;
 then
 Enter "read" into $[s_1, f]$;
 if $write \in [s_1, o_2]$;
 then
 Enter "write" into $[s_1, f]$;
 if $execute \in [s_1, o_2]$;
 then
 Enter "execute" into $[s_1, f]$;
 end

Command "запустить файл"
 (s_1, f):
 if $execute \in [s_1, f]$;
 then
 Create subject f' ;
 Enter "read" into $[f', o_1]$;
 Enter "read" into $[f', o_3]$;
 if $write \in [s_1, o_2]$;
 then
 Enter "write" into $[f, o_2]$;
 end

Command "скопировать файл o_3 программой f' в o_2 "
 (f', o_3, o_2):
 if $read \in [f', o_3]$ and
 $write \in [f', o_2]$
 then
 Create object o' ;
 Write (f', o_3, o') ;
 if $read \in [s_2, o_2]$;
 then
 Enter "read" into $[s_2, o']$;
 end