УДК 681.3

МОДЕЛИ БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ КРИТИЧНЫХ ИНФОРМАЦИОННО - ИЗМЕРИТЕЛЬНЫХ СИСТЕМ

(Часть 2.Модели дискреционного доступа)

Артамонов В.А.

Белорусский национальный технический университет, Минск, Республика Беларусь

Рассмотрен класс моделей информационных технологий критичных ИИС на основе предоставления прав в виде дискреционного доступа к объектам комплексных измерений. Представлены основные теоремы безопасности систем подобного класса.

Введение

Модели разграничения доступа, построенные по принципу предоставления прав, являются самой естественной основой для построения политик разграничения доступа к ресурсам критичных ИИС. Неформально право доступа может быть описано как "билет", в том смысле, что владение им разрешает доступ к некоторому объекту, описанному в данном разрешении.

Основными типами моделей, построенных на предоставлении прав, являются модели *дискреционного* **(*DAC)*** и *мандатного* ***(MAC)*** доступов [1]. Модели данного типа используются в большинстве реальных систем ИТ, синтезированных в настоящее время.

Основная часть

Рассмотрим наиболее изученную и часто реализуемую на практике дискреционную модель предоставления доступа [2]. В модели представлено четыре типа объектов, относящихся к безопасности: пользователи**(u),** задания**(j),** терминалы**(t)** и файлы**(f),** причем каждый объект описывается четырехмерным кортежем **(A, C, F, M),** включающим параметры безопасности:

*Компетенция **A*** - скаляр - элементы из набора иерархически упорядоченных уровней безопасности, таких как **НЕСЕКРЕТНО, КОНФИДЕНЦИАЛЬНО, СЕКРЕТНО, СОВЕРШЕННО СЕКРЕТНО.**

*Категория **C*** - дискретный набор рубрик. Категории не зависят от уровня безопасности. Пример набора рубрик: **ОГРАНИЧЕНО, ТАЙНО, ТОЛЬКО ДЛЯ ПРОСМОТРА, ВОЕННЫЙ , ПОЛИТИЧЕСКИЙ.**

*Полномочия **F*** - группа пользователей, имеющих право на доступ к определенному объекту.

*Режим **M*** - набор видов доступа, разрешенных к определенному объекту или осуществляемых объектом. Пример: **ЧИТАТЬ ДАННЫЕ, ПРИСОЕДИНЯТЬ ДАННЫЕ, ИСПОЛНИТЬ ПРОГРАММУ.**

Если **U={u}** обозначает набор всех пользователей, известных системе, а **F(i)** - набор всех пользователей, имеющих право использовать объект **i**, то для модели формулируются следующие правила:

1. Пользователь **u** получает доступ к системе **U ⇔ u ∈ U**.
2. Пользователь **u** получает доступ к терминалу **t ⇔ u ∈ F(t)** (то есть в том случае, когда пользователь **u** имеет право использовать терминал **t**).
3. Пользователь u получает доступ к файлу **j ⇔ A(j) ≥ A(f), C(j) ⊇C(f), M(j) ⊇ M(f)** и **u ∈ F(t)** ,то есть тогда и только тогда, когда выполняются условия:

-привилегии выполняемого задания шире привилегий файла или равны им;

-пользователь является членом **F(f).**

Задавая параметры безопасности **A, C, F, M,** можно сформировать матрицу определения параметров безопасности (табл. 1).

Таблица 1

Матрица определения параметров безопасности

Объект	A	C	F	M
Пользователь u	Const	Const	{u}	Const
Терминал t	Const	Const	{u(t,i)}	Const
Задание j	min(A(u),A(t))	C(u)∩C(t)	{u(j,i)}	M(u)∩M(t)
Существ. файл f(i)	Const	Const	{u(f,i)}	Const
Нов .файл f=g(f1,f2)	max(A(f1),A(f2))	C(f1)∪C(f2)	{u(f,j)}	M(f1)∪M(f2)

f1, f2 - старые файлы; новый файл **f** является некоторой их функцией.

Четырехмерный кортеж безопасности, полученный на основе прав задания, а не прав пользователя, используется в модели для управления доступом. Данный подход обеспечивает однородный контроль права на доступ над неоднородным множеством программ и данных, файлов, пользователей и терминалов. Например,

наивысшим полномочием доступа к файлу для пользователя **"СОВ. СЕКРЕТНО",** выполняющего задание с **"КОНФИДЕНЦИАЛЬНОГО"** терминала будет **"КОНФИДЕНЦИАЛЬНО".**

Далее рассмотрим модель, называемую пятимерным пространством безопасности [3]. В данной модели используется пятимерное пространство безопасности для моделирования процессов, установления полномочий и организации доступа на их основании. Модель имеет пять основных наборов:

A *- установленных полномочий;* ***U*** *- пользователей;* ***E*** *- операций;* ***R*** *-ресурсов;* ***S*** *- состояний.*

Область безопасности будет выглядеть как декартово произведение: **A×U×E×R×S**. Доступ рассматривается как ряд запросов, осуществляемых пользователями **u** для выполнения операций **e** над ресурсами **R** в то время, когда система находится в состоянии **s**. Например, запрос на доступ представляется четырехмерным кортежем **q = (u, e, r, s), u∈U, e∈E, s∈S, r≤ R.** Величины **u** и **s** задаются системой в фиксированном виде. Таким образом, запрос на доступ - подпространство четырехмерной проекции пространства безопасности. Запросы получают право на доступ в том случае, когда они полностью заключены в соответствующие подпространства.

Процесс организации доступа можно описать алгоритмически следующим образом. Для запроса **q**, где **q(u, e, r, s),** набора **U'** вполне определенных групп пользователей, набора **R'** вполне определенных единиц ресурсов и набора **P** правильных (установленных) полномочий процесс организации доступа будет состоять из следующих процедур:

1. Вызвать все вспомогательные программы, необходимые для предварительного принятия решений.

2. Определить из **U** те группы пользователей, к которым принадлежит **u**. Затем выбрать из **P** спецификации полномочий, которым соответствуют выделенные группы пользователей. Этот набор полномочий **F(u)** определяет привилегию пользователя **u**.

3. Определить из **P** набор **F(e)** полномочий, которые устанавливают **e** как основную операцию. Этот набор называется привилегией операции **e**.

4. Определить из **P** набор **F(R)** (привилегию единичного ресурса **R**) - полномочия, которые определяют поднабор ресурсов из **R'**, имеющего общие элементы с запрашиваемой единицей ресурса **R**.

Полномочия, которые являются общими для трех привилегий в процедурах 2, 3, 4, образуют **D(q)** (так называемый домен полномочий для запроса):

$$q: D(q) = F(u) \cap F(e) F(R) \qquad (1)$$

5. Удостовериться, что запрашиваемый ресурс **R** полностью включается в **D(q),** то есть каждый элемент из **R** должен содержаться в некоторой единице ресурса, которая определена в домене полномочий **D(q).**

6. Осуществить разбиение набора **D(q)** на эквивалентные классы так, чтобы два полномочия попадали в эквивалентный класс тогда и только тогда, когда они специфицируют одну единицу ресурса. Для каждого такого класса логическая операция **ИЛИ** (или **И**) выполняется с условиями доступа элементов каждого класса.

Новый набор полномочий - один на каждую единицу ресурса, указанную в **D(q),** есть **F(u, q)** - фактическая привилегия пользователя **u** по отношению к запросу **q**.

7. Вычислить **EAC** - условие фактического доступа, соответствующего запросу **q**, осуществляя логическое **И** (или **ИЛИ**) над условиями доступа членов **F(u, q).** Это **И**(или **ИЛИ**) выполняется над всеми единицами ресурсов, которые перекрывают единицу запрошенного ресурса.

8. Оценить **EAC** и принять решение о доступе:

-разрешить доступ к **R**, если **R** перекрывается;

-отказать в доступе в противном случае.

9. Произвести запись необходимых событий.

10. Вызвать все программы, необходимые для организации доступа после *"принятия решения"*.

11. Выполнить все вспомогательные программы, вытекающие для каждого случая из условия 8.

12. Если решение о доступе было положительным - завершить физическую обработку.

Следует отметить, что приведенная последовательность шагов не всегда необходима в полном объеме. Например, в большинстве реализаций шаги 2 и 6 осуществляются во время регистрации пользователя в системе.

Далее проанализируем проблемы безопасности моделей дискреционного доступа [2].

Рассмотрим типичную модель системы защиты, состоящую из следующих конечных наборов:

-общих прав **A={a_1..., an};**

-исходных субъектов S_0 и объектов O_0;

-команд **C** формы $\alpha(X_1, ..., X_n)$, где **α** - имя; $X_1, ..., X_n$ - формальные

параметры, указывающие на объекты.

Элементами матрицы доступа являются права доступа, взятые из набора общих прав. Состояния системы изменяются при изменении элементов матрицы доступа **M.** Существуют две теоремы о безопасности данного типа систем [3]. Первая относится к безопасности *монооперационных* систем. При этом, под монооперационной системой понимается система ИТ, в которой каждый запрос имеет только одну операцию.

Вторая теорема указывает на то, что проблема безопасности для системы с запросами общего вида является неразрешимой.

Теорема 1. *Существует алгоритм для определения, является или нет моно-операционная система безопасной для данного права* ***a****.*

Доказательство.

Доказательство строится на том факте, что для любой последовательности запросов $\mathbf{r_1}$, $\mathbf{r_2,...,\ r_n}$, которая приводит к утечке права **a** из начальной конфигурации ($\mathbf{S_0,\ O_0,\ M_0}$), существует последовательность запросов из данной конфигурации, которая также ведет к утечке права **a**, которая содержит только запрос **enter**, за исключением начального запроса **create subject**. Для того, чтобы сформировать данную последовательность, необходимо вставить запрос **create subject** в ее начало для создания нового субъекта. Обозначим данный запрос как $\mathbf{s_{init}}$. Далее удалим все запросы **delete и destroy** из последовательности $\mathbf{r_1}$, $\mathbf{r_2,...,\ r_n}$. Данная последовательность все равно будет приводить к утечке права **a**, так как условная часть запроса может протестировать только присутствие права, но не отсутствие прав. Далее рассмотрим самый правый в последовательности запрос **create subject**. Можно удалить данный запрос и просто заменить все ссылки на нового субъекта в дальнейших запросах ссылкой на $\mathbf{s_{init}}$. Продолжим данную процедуру до тех пор, пока не достигнем начального запроса **create subject**, который останется без изменений. Далее заменяются все запросы **create object** к системе с заменой ссылок на новые объекты в дальнейших запросах ссылками на $\mathbf{s_{init}}$. После этого из последовательности удаляются запросы **enter**, которые вводят право $\boldsymbol{a_i}$ в ячейку, которая уже содержит это право.

Результирующая последовательность приводит также к утечке права **a**, и имеет длину $\mathbf{I = (|\ A\ |\ *\ (|\ S_0\ |\ +\ 1)\ *\ (|\ O_0\ |\ +\ 1))\ +\ 1}$ запросов, так как каждый запрос, за исключением начального запроса **create object**, должен вводить новый символ из **A** в ячейку матрицы доступа, и количество ячеек в матрице не может быть больше, чем $\mathbf{(|\ S_0\ |\ +\ 1)\ *\ (|\ O_0\ |\ +1)}$. Следовательно, можно определить, является ли данная система безопасной просмотром всех возможных последовательностей запросов **enter** длиной меньше или равной чем l.

Теорема 2. *Проблема определения безопасности для данного права a в системе с запросами общего вида является неразрешимой.*

Доказательство.

Доказательство данной теоремы основывается на классическом свойстве формализма машины Тьюринга [4].

Рассмотрим машину Тьюринга. Каждая машина Тьюринга **T** имеет фиксированное число состояний **К** и конечный набор символов **Г**, размещаемых на ленте. Одним из этих символов является пробел **B**, который первоначально появляется в каждой ячейке ленты. Лента бесконечна только вправо. Машина Тьюринга снабжена сканирующей головкой, которая в каждый момент времени обозревает некоторую ячейку на ленте. Действия машины Тьюринга определяются функцией **δ** с аргументом из **К×Г** и значением **К×Г×(L, R)** (**L** означает переход головки влево, **R** - вправо). Если **δ (q, X) = (p, Y, L)** для состояний **p** и **q** и символов на ленте **X** и **Y**, то это означает, что машина **T**, находящаяся в состоянии **q** и наблюдающая ячейку, содержащую символ **X**, переходит в состояние **p**, стирает символ **X** и печатает символ **Y** в эту ячейку, сканирующая головка переходит при этом на одну клетку влево. Первоначально **T** находится в состоянии $\mathbf{q_0}$ - исходном состоянии и обозревает ячейку 1. Каждая ячейка первоначально содержит пробел. Существует особое состояние **q**, известное как конечное.

В соответствии с тезисом Черча, алгоритм, вычислимый на машине Тьюринга, вычислим и в интуитивном понятии. Таким образом, наша проблема сводится к проблеме останова машины Тьюринга.

Покажем, что вопрос обеспечения безопасности является неразрешимым. Это можно показать на примере системы защиты, которую можно смоделировать произвольной машиной Тьюринга.

Действия машины Тьюринга отражаются в командах следующим образом. Во-первых, если **δ(q, X) = (p, Y, L),** то существует команда $\mathbf{C_{qx}(s, s')}$ с условиями:

собственность **∈ (s, s'),**

c∈ (s', s'), x∈ (s', s').

s и **s'** должны соответствовать двум последовательным ячейкам на ленте; машина находится в состоянии **q**, наблюдает ячейку, соответствующую **s'**, и в этой ячейке записан символ **X**.

Команда $\mathbf{C_{qx}}$ интерпретируется следующим образом:

удалить q из (s', s'),

удалить X из (s', s'),

ввести p в (s', s'),

ввести Y в (s', s').

Если **δ(q, X) = (p, Y, R)** (то есть головка ленты перемещается вправо), то это возможно в зависимости от того, прошла ли головка текущий конец ленты.

Существует команда $\mathbf{C_{qx}(s, s')}$ с условиями:

собственность ∈ **(s, s'),**

q ∈ **(s, s), X** ∈ **(s, s)**

и интерпретация:

удалить q из **(s, s),**

удалить X из **(s, s),**

ввести p в **(s', s'),**

ввести Y в **(s, s).**

Существует также команда **Dqx(s, s')** с условиями:

конец ∈ **(s, s),**

q∈ **(s, s),X**∈ **(s, s)**

и интерпретация:

удалить q из **(s, s),**

удалить X из **(s, s),**

создать субъект s',

ввести B в **(s, s'),**

ввести p в **(s', s'),**

ввести Y в **(s, s)**.

Если мы начинаем с исходной матрицы, содержащей субъект $\mathbf{s_1}$ с правами $\mathbf{q_0}$, **B** ***пробел*** и ***собственность*** по отношению к самому себе, то матрица доступа будет всегда иметь ровно одно общее право, которое является состоянием системы. Это следует из того, что каждая команда удаляет состояние, известное из условия той команды, которая должна выполняться. Каждая команда также добавляет одно состояние в матрицу доступа. Нельзя вносить в матрицу более одного общего права, которое является символом ленты по такому же аргументу. Таким же образом конец появляется только в одной записи матрицы, а именно в диагональной записи для каждого созданного субъекта.

Таким образом, в каждой конфигурации системы защиты, полученной из исходной конфигурации, существует хотя бы одна команда, которую можно применить. Это объясняется тем, что машина Тьюринга имеет по крайней мере одно допустимое перемещение в любой ситуации. $\mathbf{C_{qx}}$ и $\mathbf{D_{qx}}$ никогда не используются одновременно.

Система защиты должна поэтому точно моделировать машину Тьюринга, используя описанное представление. Если машина Тьюринга вводит состояние $\mathbf{q_j}$, то система защиты может иметь утечку общего права $\mathbf{q_j}$, в противном случае она безопасна для $\mathbf{q_j}$. Поскольку нельзя предсказать, будет ли машина Тьюринга вводить $\mathbf{q_j}$, нельзя решить, является ли система защиты безопасной для $\mathbf{q_j}$.

Безопасными являются монотонные системы (системы, не содержащие операции **destroy** и **delete**), системы не содержащие операций **create** и моно-условные системы (системы, запрос к которым содержит только одно условие).

Заключение

К достоинствам моделей дискреционного доступа можно отнести хорошую гранулированность (относительно независимую характеристику защиты) и относительно простую реализацию. В качестве примера реализаций данного типа моделей можно привести так называемую матрицу доступа, строки которой соответствуют *субъектам* системы, а столбцы - *объектам*; элементы матрицы характеризуют *права доступа*. Проблемы, возникающие в системах, синтезированных на их основании, показаны в следующей части работы.

Список цитируемых источников

1. Хоффман Дж. Современные методы защиты информации. - М.: Сов.радио, 1990.
2. .Landwehr. Formal Models for Computer Security. ACM Computing Surveys. Vol. 13, N3.1994.
3. M.Harrison, W.Ruzzo, J.Uhlman "Protection in operating systems", Communications of the ACM, 1996.
4. John McLean "Security models", Enciclopedia of software engeneering, 1994.

Artamonov V.A.

The models of safety information technologies critical information-measuring systems (Part 2. The models of discretionary access)

The class of models of information technologies critical IMS on the basis of granting the rights in the form of discretionary access to objects of complex measurements is considered.The basic theorems of safety's systems of a similar class are presented.