УДК 004.451

Сравнительный анализ использования существующих моделей разграничения доступа в системах, обладающих равнозначными объектами

С.А. Лапин

Алтайский государственный университет (Барнаул, Россия)

Comparative Analysis of Existing Access Control Models in Systems with Interchangeable Objects

S.A. Lapin

Altai State University (Barnaul, Russia)

Рассматриваются особенности использования существующих моделей разграничения доступа применительно к системам, имеющим равнозначные объекты. Под равнозначными понимаются объекты, которые обладают одинаковыми функциональными возможностями, но с различными характеристиками. Такие системы могут быть ориентированы на выполнение некоторого множества задач, при решении каждой из которых применяются определенные требования. Примером такой системы может являться учреждение здравоохранения, где субъекты системы — врачи, предоставляющие услуги лечения пациентам, а объектами, к которым необходимо контролировать доступ — лекарственные препараты. При лечении одного и того же заболевания в распоряжение врача могут предоставляться различные препараты, имеющие одинаковые функциональные возможности, но различные характеристики. Рассматриваются как классические модели разграничения доступа (ХРУ, RBAC), так и модели, предназначенные для использования в динамических системах (TBAC, DEBAC). Анализ производится с точки зрения избыточности предоставляемых прав доступа, разделения их в соответствии с задачами, а также гибкости и сложности администрирования политики безопасности. На основе проведенного анализа обосновывается необходимость создания новой математической модели разграничения доступа, ориентированной на использование в рассматриваемых системах.

Ключевые слова: компьютерная безопасность, математические модели безопасности, разграничение доступа, динамические системы, задачи, ХРУ, RBAC, TBAC, DEBAC, требования.

The paper examines usage peculiarities of existing access control models in systems with interchangeable objects. Interchangeable objects are understood to be objects that have the same functionality but different characteristics. Such systems can be focused on performing of a set of tasks with certain specific requirements for each of tasks. A health care institution is an example of such system, where each subject of the system is a doctor providing treatment services to patients and objects with the access that requires monitoring are drugs. In order to treat the same disease, a doctor can be provided with a variety of drugs that have the same functionality, but different characteristics. Both classic access control models (HRU, RBAC) and models meant for dynamic systems (TBAC, DEBAC) are discussed. The analysis is conducted in the context of access rights redundancy, differentiation in accordance with the tasks, as well as flexibility and complexity of the security policy administration. In terms of the performed analysis the necessity to create a new mathematical access control model suited for the examined systems is proved.

Key words: computer security, mathematical security models, access control, dynamic systems, tasks, HRU, RBAC, TBAC, DEBAC, requirements.

DOI 10.14258/izvasu(2016)1-25

Введение. При построении системы обеспечения информационной безопасности важным ее компонентом является система разграничения доступа, цель которой — предотвращение несанкционированного доступа. На сегодняшний день разработано множество моделей разграниче-

ния доступа, основанных на различных парадигмах (матрице доступа, ролях, задачах, событиях и пр.), что объясняется обширной природой современных систем. Каждая из моделей имеет свои достоинства и недостатки при ее использовании в той или иной системе.

Целью работы является анализ использования существующих моделей разграничения доступа применительно к системам, в которых присутствуют равнозначные объекты. Под равнозначными будем понимать объекты, которые имеют одинаковые функциональные возможности, но различные характеристики. Равнозначные объекты по некоторому признаку, например по функциональности, объединяются в несколько подмножеств множества объектов, которые назовем группами равнозначных объектов. Такие системы могут быть ориентированы на выполнение некоторого множества задач, для решения которых могут применяться определенные требования. Примером такой системы может являться учреждение здравоохранения [1], где субъектами системы являются врачи, предоставляющие услуги лечения пациента, а объектами — лекарственные препараты.

Определим следующие требования, которым должна удовлетворять используемая модель разграничения доступа в подобного рода системах:

1. Права доступа должны предоставляться субъектам на основе решаемых ими задач.
2. С целью минимального выделения прав при решении каждой задачи доступ должен предоставляться только к одному равнозначному объекту из каждой группы.
3. Выполнение операций администрирования в системе должно приводить к минимальным изменениям политики безопасности, чтобы уменьшить вероятность совершения ошибок, при определении правил контроля доступа.

Рассмотрим систему, в которой имеются следующие элементы:

$S = \{s_1, s_2, s_3\}$ — множество субъектов. В качестве субъектов могут выступать, например, врачи медецинского учреждения;

$O = \{o_1, \cdots, o_7\}$ — множество объектов. Под объектами можно понимать лекарственные препараты;

$G = \{g_1 = \{o_1, o_3, o_6\}, g_2 = \{o_2, o_5\}, g_3 = \{o_4, o_7\}\}$ — множество групп объектов. Группы объектов формируются по некоторому общему признаку, которым обладают все объекты этой группы. Например, если под объектами рассматривать лекарственные препараты, то такие группы можно образовать по их предназначению;

$R = \{use\}$ — множество видов прав доступа субъектов к объектам. Для простоты и наглядности изложения рассматривается только одно право use — использовать объект;

$T = \{task_1, task_2, task_3\}$ — множество задач, которые могут выполняться в системе.

Пусть субъекты $s_1,\ s_2,\ s_3$ должны выполнять в системе задачи $task_1,\ task_2,\ task_3$ соответственно. Пусть также для выполнения задачи $task_1$ необходим вид доступа $r = \{use\}$, к объектам из групп $g_1,\ g_3$, для $task_2$ из $g_2,\ g_3$, а для $task_3$ из $g_1,\ g_2$. Общая структура такой системы представлена на рисунке 1.

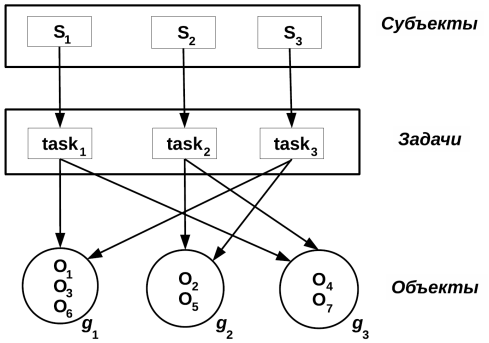

Рис. 1. Общая структура исследуемой системы

1. Использование классических моделей безопасности. Воспользуемся для решения поставленной задачи моделью ХРУ (Харрисона — Руззо — Ульмана) [2, 3]. Во введении уже определены такие элементы модели, как множество субъектов системы S, множество объектов O, множество видов прав доступа R. Разграничение прав доступа определяется матрицей доступа M, строками которой являются субъекты, а столбцы объектами. Таким образом, $M[s, o] \subseteq R$ — права доступа субъекта s к объекту o.

Так как в матрице доступов необходимо перечислить для каждого субъекта системы доступа, которыми он обладает к каждому объекту, то, очевидно, разграничение доступа к равнозначным объектам возможно только в случае, когда субъекту системы предоставляется доступ только к одному объекту из группы. Таким образом, можно ввести следующие условие для составления матрицы доступов M: если для $s \in S$ и $o \in g_i \in G\ \exists M[s, o] = R^{'}$, где $R^{'} \subseteq R$, то $\forall o^{'} \in g_i$, такого что $o \neq o^{'}$, $M[s, o^{'}] = \emptyset$.

При таком способе использования модели очевидны следующие недостатки:

1. Субъекту предоставляются недостаточные права доступа. Субъект имеет доступ только к одному объекту из каждой группы, что, при наличии в системе равнозначных объектов, может привести к некорректному решению поставленной задачи и повлечь за собой угрозы безопасности системы.

2. Множество прав доступа субъекта статично. Вне зависимости, какую задачу выполняет субъект в системе, множество его прав доступа остается постоянным.

Однако стоит отметить, что как формирование, так и дальнейшее администрирование политики безопасности, основанной на ХРУ, является достаточно простым. Процедуры добавления/удаления объектов или субъектов в системе приводят только к изменению матрицы доступа M, путем добавления/удаления соответствующих строк или столбцов.

Модель RBAC (Role-based Access Control) [3–5] контролирует доступ субъектов системы на объекты в соответствии с совокупностью действий и обязанностей, связанных с определенным видом деятельности субъектов. Такие полномочия представляют собой семантические конструкции, называемые ролями субъектов, которые лежат в основе модели разграничения доступа. Роли позволяют получить конкретным лицам доступ к объектам в той степени, в какой это необходимо им для выполнения своих обязанностей.

В данном случае RBAC возможно применить в рассматриваемой системе, если для каждой роли определять права доступа, исходя из всевозможных комбинаций использования равнозначных объектов. Таким образом, роли будут сформированы так, что каждая из них предоставляет доступ только к одному из равнозначных объектов группы. Например, субъект s_2 может обладать следующими правами доступа:

$PA(r_1) = \{o_2,\ o_4\} \times use$
$PA(r_2) = \{o_2,\ o_7\} \times use$
$PA(r_3) = \{o_5,\ o_4\} \times use$
$PA(r_4) = \{o_5,\ o_7\} \times use,$

где $\{r_1, r_2, r_3, r_4\}$ множество ролей, на которые может авторизоваться пользователь s_2.

В отличие от модели ХРУ, RBAC позволяет разграничить доступ субъектов в системе относительно выполняемых ими задач в отдельности и при этом предоставляет инструменты для разграничения доступа к равнозначным объектам. Более того, права доступа пользователя в системе не являются постоянными и могут изменяться в зависимости от того, с какой ролью авторизовался пользователь. Отметим следующие недостатки использования RBAC:

1. Решение, на какую из доступных пользователю ролей он будет авторизован, принимает сам пользователь. Как уже было сказано, это может повлиять на безопасность системы.

2. Значительно увеличивается количество определяемых ролей, по сравнению с реальным количеством функциональных обязанностей пользователя в системе. Например, пользователь s_2 выполняет в системе только одну функцию — решение задачи $task_2$. Однако с целью минимизации его прав для него определено четыре роли. Если обозначить за $G_{t_i} = \{g^1_{t_i}, g^2_{t_i}, \cdots, g^m_{t_i}\} \subseteq G$ множество групп равнозначных объектов, к которым должен иметь доступ пользователь системы для решения задачи t_i, то общее количество ролей, которые должны быть определены, можно записать как

$$\sum_{i=1}^{|T|} \prod_{j=1}^{|G_{t_i}|} |g^j_{t_i}|. \qquad (1)$$

3. Усложняются процедуры администрирования системы безопасности, как на этапе ее формирования, так и при внесении изменений. Данный недостаток является следствием предыдущего. Очевидно, что в реальных системах с большим количеством задач и равнозначных объектов количество определяемых ролей будет достаточно большим. В свою очередь, это может привести к возникновению ошибок, росту числа уязвимостей и пр., что негативно сказывается на системе безопасности.

2. Использование модели TBAC. В основе модели Task-based Authorization Controls (TBAC) [6] лежит понятие «задача». Права доступа пользователей изменяются с учетом специфики выполняемой задачи, на которую он авторизован в данный момент, что обеспечивает динамический контроль доступа. Таким образом, права доступа субъектов в системе предоставляются в зависимости от контекста задачи и не являются постоянными. Разрешения предоставляются и отменяются для каждой задачи в отдельности. TBAC определяет права доступа как $P \subseteq S \times O \times A \times U \times AS$.

Применим модель TBAC для решения поставленной задачи. Имеем следующие элементы:

$S = \{s_1, s_2, s_3\}$ — множество субъектов;

$O = \{o_1, o_2, \cdots, o_7\}$ — множество объектов;

$A = \{use\}$ — множество видов прав доступа субъектов к объектам;

$AS = \{as_1, as_2, \cdots, s_{16}\}$ — множество этапов авторизации.

В статье [6] модель описана не формально [7]. В частности, не конкретизируется, каким образом хранятся права доступа в системе. Поэтому введем следующие вспомогательные элементы:

M — матрица доступов, строки которой соответствуют этапам авторизации, а столбцы соответствуют объектам. $M[as, o] \subseteq A$ — права доступа субъекта, прошедшего этап авторизации as, на объект o.

$SAS : S \to 2^{AS}$ — функция, задающая для каждого субъекта подмножество этапов авторизации, которые он может пройти.

В TBAC каждое право доступа дается субъектам на некоторый промежуток времени U. Будем считать, для любого права доступа это значение является постоянным, поэтому оно не будет учитываться в дальнейшем изложении.

Тогда состояние системы в TBAC с учетом вспомогательных элементов можно записать как $P \subseteq S \times O \times M \times AS \times SAS$.

Определим значения функции SAS следующим образом:

$SAS(s_1) = \{as_1, as_2, \cdots, as_6\}$

$SAS(s_2) = \{as_7, as_8, as_9, as_{10}\}$

$SAS(s_3) = \{as_{11}, as_{12}, \cdots, as_{16}\}$

Зададим матрицу контроля доступа M:

	o_1	o_2	o_3	o_4	o_5	o_6	o_7
as_1	use			use			
as_2	use						use
as_3			use	use			
as_4			use				use
as_5				use		use	
as_6						use	use
as_7		use		use			
as_8		use					use
as_9				use	use		
as_{10}					use		use
as_{11}	use	use					
as_{12}	use				use		
as_{13}		use	use				
as_{14}			use		use		
as_{15}		use				use	
as_{16}					use	use	

В данном случае каждая задача из T делится в системе на несколько задач, каждая из которых соответствует своему этапу авторизации. В данном случае субъект s_1 может быть аутентифицирован на шесть этапов, заданных функцией SAS, каждый из которых представляет собой выполнение задачи $task_1$. Для каждого этапа авторизации в матрице доступа M определены права доступа в системе для субъекта, который его прошел. При этом права доступа в M заданы таким образом, чтобы субъект имел доступ только к одному объекту из группы.

Использование модели TBAC для решения поставленной задачи имеет следующие положительные стороны:

1. В соответствии с определениями модели разграничение прав доступа субъектов системы происходит в соответствии с задачами, которые они выполняют в отдельности.
2. Права субъектов в системе не являются постоянными и предоставляются только на время выполнения ими назначенной задачи.
3. Модель позволяет реализовать разграничение доступа к равнозначным объектам системы путем добавления нескольких этапов авторизации для задач.
4. Для добавления новых субъектов системы требуется только назначить для них этапы авторизации путем внесения изменений в функцию SAS. При этом не требуется вносить какие-либо изменения в матрицу доступа M.

Однако администрирование политики безопасности при таком подходе является затруднительным. Это связано с большим количеством этапов авторизации, значительно превышающее количество задач, которые решаются в системе. Как следствие увеличивается размер матрицы доступа M. Общее количество этапов авторизации, которое должно присутствовать в матрице M, будет определяться по формуле (1).

В реальных системах количество задач, субъектов и равнозначных объектов, как правило, является достаточно большим. Очевидно, что в таком случае процесс формирования матрицы доступа M будет являться трудоемким и занимающим значительное время. Более того, ее размер будет многократно увеличиваться, если в такую систему будут добавляться новые равнозначные объекты или задачи. Это может привести к ошибкам в системе разграничения прав доступа и, как следствие, приводить к неверному функционированию системы безопасности.

В работах [8,9] предлагаются модели, где права доступа для каждого этапа авторизации описываются в виде ролей. Таким образом, предлагается гибридная модель, построенная на основе TBAC и RBAC. Если использовать такой способ, то, очевидно, что указанные недостатки будут применимы и к ней. Это объясняется тем, что количество этапов авторизации будет сохраняться, и для каждого из них необходимо будет определить права соответствующей роли.

3. Использование модели DEBAC. Более общий подход по формированию правил контроля доступа представляет модель DEBAC (Dynamic Event-Based Access Control) [10]. Фундаментальным понятием модели DEBAC является понятие «событие». В DEBAC права пользователя изменяются только при наступлении какого-либо события, которое определено в системе. В соответствии с моделью имеем следующие элементы:

$O = \{o_1, o_2, \cdots, o_7\}$ — множество объектов;

$A = \{use\}$ — множество именованных действий;

$U = \{u_1, u_2, u_3\}$ — множество идентификаторов пользователей;

$C = \{c_0, c_1, c_2, \cdots, c_{16}\}$ — множество категорий;

$E = \{e_1, e_2, \cdots, e_{19}\}$ — множество идентификаторов событий;

$S = \{\mu\}$ — множество идентификаторов месторасположений;

$\mathbb{T}$ — конечное множество моментов времени;

W — конечное множество действий, связанных с событиями.

В DEBAC доступ к объекту определяется следующим образом: пользователю $u \in U$ разрешено выполнять действие $a \in A$ над объектом $o \in O$, который имеет месторасположение $s \in S$, если и только если u назначена категория $c \in C$, которой был назначен доступ к o.

Определим значения функции $privileges$, которая возвращает список пар $\{a_i, c_i\}$, т.е. действие и категорию, позволяющую выполнить такое действие для данного объекта в данном месторасположении, следующим образом:

$privileges(o_1, \mu) = \{c_1, c_2, c_{11}, c_{12}\} \times \{apply\}$
$privileges(o_2, \mu) = \{c_7, c_8, c_{11}, c_{13}, c_{15}\} \times \{apply\}$
$privileges(o_3, \mu) = \{c_3, c_4, c_{13}, c_{14}\} \times \{apply\}$
$privileges(o_4, \mu) = \{c_1, c_3, c_5, c_7, c_9\} \times \{apply\}$
$privileges(o_5, \mu) = \{c_1, c_3, c_5, c_7, c_9\} \times \{apply\}$
$privileges(o_6, \mu) = \{c_5, c_6, c_{15}, c_{16}\} \times \{apply\}$
$privileges(o_7, \mu) = \{c_2, c_4, c_6, c_8, c_{10}\} \times \{apply\}$

Как можно видеть, категория c_0 не позволяет выполнять никаких действий над объектами системы. Другими словами, пользователь, которому была назначена данная категория, не имеет никаких прав доступа в системе.

Зададим функцию $user(E)$, определяющую, в каких событиях могут участвовать пользователи:

$user(E_1) = u_1$, где $E_1 = \{e_1, e_2, \cdots, e_6, e_{17}\}$
$user(E_2) = u_2$, где $E_2 = \{e_7, e_8, \cdots, e_{10}, e_{18}\}$
$user(E_3) = u_3$, где $E_3 = \{e_{11}, e_{12}, \cdots, e_{16}, e_{19}\}$

Зададим следующие вспомогательные функции:

$Head(L)$ — функция, возвращающая последний добавленный элемент в список L;

$Check_0(e)$ — функция, возвращающая 1, если событие означает завершение выполнения задачи, иначе возвращающая 0.

Определим функцию $Check_0(e)$ в виде:

$$\begin{aligned} Check_0(e) \quad \rightarrow \quad & if\ e = e_{17}\ or\ e = e_{18}\ or\ e = e_{19} \\ & then\ 1 \\ & else\ 0 \end{aligned}$$

Функция $Estatus$ отвечает за назначение пользователю категории в соответствии с событием. Будем считать, что при наступлении события e_i пользователю может быть назначена категория c_i, где $1 \leqslant i \leqslant 16$. Тогда функция $Estatus(E)$ может быть определена следующим образом:

$$\begin{aligned} Estatus(event(e_i, u, w_i, T)) \quad \rightarrow \quad & if\ Check_0(e_i) = 0\ and\ Head(status(U, L)) = c_0 \\ & then\ cons(c_i, status(U, L)) \\ & else\ if\ Check_0(e_i) = 1 \\ & then\ cons(c_0, status(U, L)) \end{aligned}$$

где $status(U, L)$ — отображение событий с участием пользователя U на список L. При наступлении события e_i в момент времени t, в котором участвует пользователь u_i, происходит проверка события на то, не означает ли оно завершение выполнения задачи пользователем. Если нет и в данный момент он не выполняет никаких задач, то происходит изменение категории пользователя на соответствующую событию. Если же событие означает окончание выполнения задачи пользователем, то он лишается всех прав доступа в системе, путем назначения ему категории c_0.

Каждая задача из T делится в системе на несколько задач, каждой из которых соответствует событие в системе. Отметим положительные стороны использования модели DEBAC:

1. Разграничение прав доступа субъектов системы возможно реализовать в соответствии с отдельными задачами, которые они выполняют.
2. Права субъектов в системе не являются постоянными и предоставляются только на время выполнения ими назначенной задачи.
3. Модель позволяет реализовать разграничение доступа к равнозначным объектам системы, путем добавления нескольких событий, которые соответствуют одной задачи.

Недостатком использования DEBAC является процедура администрирования политики. В основе своей это связано с тем, что для каждого пользователя необходимо определить множество событий, в которых он может участвовать.

Пусть в системе имеется n задач, которые могут выполнять одни и те же m пользователей. Обозначим за $G_{t_i} = \{g^1_{t_i}, g^2_{t_i}, \cdots, g^m_{t_i}\} \subseteq G$ — множество групп равнозначных объектов, к которым должен иметь доступ субъект системы для решения задачи t_i. Тогда количество событий, которые будет необходимо определить, равно

$$\sum_{i=1}^{n} m \cdot (\prod_{j=1}^{|G_{t_i}|} |g_{t_i}^{j}| + 1).$$

Очевидно, что это влечет за собой усложнение формирования и дальнейшего сопровождения политики безопасности.

Выводы и заключение. В представленной работе было рассмотрено использование как классических моделей безопасности (XPY, RBAC), так и моделей, реализующих динамический контроль доступа (TBAC, DEBAC). Для каждой из рассмотренных моделей были отмечены достоинства и недостатки их использования. Показано, что более приемлемым подходом для решения поставленной задачи является использование динамических моделей, основанных на задачах или событиях. При этом их применение в рассматриваемом аспекте представляется громоздким и затруднительным с точки зрения администрирования.

Наличие недостатков использования рассмотренных моделей безопасности, в том числе, связано с тем, что в них предполагается решение задачи субъектом путем применения одного и того же множества объектов, к которому ему предоставляется доступ. Однако не учитываются такие особенности системы, как наличие равнозначных объектов и требований к процессу решения задачи. В зависимости от таких требований для решения одной и той же задачи субъекту могут предоставляться различные доступы к равнозначным объектам. Таким образом, возникает необходимость создания модели, учитывающей такие особенности системы.

В работе [11] предлагается модель разграничения доступа D-TBAC, учитывающая перечисленные особенности и позволяющая решить поставленную во введении задачу. D-TBAC предоставляет механизм, позволяющий выделять минимальные права доступа субъектам системы при решении ими задач с учетом требований к их выполнению.

Библиографический список

1. Лапин С.А. Применение модели разграничения доступа D-TBAC в медицинском учреждении для контроля доступа к лекарственным препаратам // Новые информационные технологии и системы : сб. науч. ст. XII Междунар. науч.-тех. конф. — Пенза, 2015.

2. Harrison M., Ruzzo W., Ullman J. Protection in Operating Systems // Commun. ACM. — New York, 1976. — V. 19, №8. DOI: 10.1145/360303.360333

3. Девянин П.Н. Модели безопасности компьютерных систем. Управление доступом и информационными потоками : учебное пособие для вузов. — М., 2013.

4. Sandhu R. Role-based Access Control // Advances in Computers. — 1998. — V. 46. DOI: 10.1016/S0065-2458(08)60206-5

5. Ferraiolo D., Sandhu R., Gavrila S., Kuhn R., Chandramouli R. Proposed NIST Standard for Role-based Access Control // ACM Trans. Inf. Syst. Secur. — 2001. — V. 4, №3. DOI: 10.1145/501978.501980

6. Thomas R., Sandhu R. Task-Based Authorization Controls (TBAC): A Family of Models for Active and Enterprise-Oriented Autorization Management // Proceedings of the IFIP TC11 WG11.3 Eleventh International Conference on Database Securty XI: Status and Prospects. — London, 1998.

7. Cvrček D. Access Control in Workflow Systems // MOSIS'99 Proceedings. — Rožnov pod Radhoštěm, 1999.

8. Zhang C., Hu Y., Zhang G. Task-Role Based Dual System Access Control Model // International Journal of Computer Science and Network Security — 2006. — V. 7, №6.

9. Лепешкин О.М., Харечкин П.В. Функционально-ролевая модель управления доступом в социотехнических системах // Известия Южного федерального университета. Технические науки. — 2009. — Т. 100, №11.

10. Bertolissi C., Fernández M., Barker S. Dynamic Event-Based Access Control as Term Rewriting // Proceedings of the 21st Annual IFIP WG 11.3 Working Conference on Data and Applications Security. — California, 2007. DOI: 10.1007/978-3-540-73538-0_15

11. Lapin S. Access control model D-TBAC subject to the requirements to tasks' performing // Proceedings of the 8th International Conference on Security of Information and Networks. — New York, 2015. DOI: 10.1145/2799979.2800034