УДК 681.3.07

Модель контроля доступа D-TBAC с учётом требований к выполнению задач

С. А. Лапин*

В настоящей статье представлена модель контроля доступа D-TBAC, расширяющая TBAC. Представленная модель учитывает требования к выполнению задач. Формально определены элементы модели, её свойства, а также правила преобразования состояний.

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

1. Введение

В современных компьютерных системах предъявляются повышенные требования к уровню их безопасности. Одним из компонентов системы безопасности является разграничение доступа пользователей на объекты системы.

В существующих моделях разграничения доступа, в том числе ориентированных на задачи, предполагается, что одна и та же задача будет решена субъектом системы через использование одного множества объектов, к которому ему предоставляется доступ. Однако такие модели не учитывают, что в системе могут присутствовать равнозначные объекты, которые имеют одинаковые функциональные возможности, но обладают различными характеристиками [1, 2]. Равнозначные объекты по некоторому признаку, например, по функциональности, объединяются в несколько подмножеств множества объектов, которые назовём группами равнозначных объектов. Такие системы могут быть ориентированы на выполнение множества задач, для решения которых могут предъявляться определённые требования. От того, каким из равнозначных объектов воспользуется субъект при решении поставленной перед ним задачи, может зависеть как безопасность её решения, так и безопасность всей системы в целом. Поэтому в зависимости от различных факторов следует предоставлять субъекту различные доступы к равнозначным объектам. Набор таких факторов будет формировать определённые требования к решению задач субъектами системы.

Таким образом, целью представленной работы является формальное описание модели контроля доступа, которая позволяет:

1) выделять минимально необходимые права доступа субъектам системы при наличии в ней объектов с одинаковыми функциональными возможностями;
2) предъявлять набор требований к процессу выполнения задачи субъектом системы;
3) использовать модель в динамических системах.

Для достижения поставленной цели в статье решаются следующие задачи:

1) определение множества элементов, из которых состоит предлагаемая модель;
2) определение свойств предлагаемой модели, описывающих безопасность системы;
3) определение множества запросов, с помощью которых изменяется состояние системы;
4) определение правил преобразования состояний системы, функционирующей в соответствии с предлагаемой моделью.

*Публикация подготовлена в рамках поддержанного РГНФ научного проекта № 16-33-01160

2. Связанные работы

В этом разделе обсуждаются способы решения поставленной во введении задачи на основе анализа существующей литературы. Рассматриваются как классические модели разграничения доступа (ХРУ, RBAC), так и модели, предназначенные для использования в динамических системах (TBAC, DEBAC), с точки зрения избыточности предоставляемых прав доступа, разделения их в соответствии с задачами, а также гибкости и сложности администрирования политики безопасности.

В модели Харрисона–Руззо–Ульмана (ХРУ) [3, 4] разграничение прав доступа определяется матрицей доступа, строками которой являются субъекты, а столбцы – объектами. Т.к. в матрице доступов необходимо перечислить для каждого субъекта системы доступы, которыми он обладает к каждому объекту, то, очевидно, разграничение доступа к равнозначным объектам возможно только в случае, когда субъекту предоставляется доступ только к одному объекту из группы. 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 от того, какую задачу выполняет субъект в системе, множество его прав доступа остаётся постоянным.

Основной моделью, реализующей мандатное разграничение прав доступа, является классическая модель Белла–ЛаПадулы (Bell–LaPadula) [5]. Модель ориентирована прежде всего на контроль за информационными потоками, возникающими в системе, на которые накладывается ряд ограничений, препятствующих возникновению таких потоков от объектов с большим уровнем конфиденциальности к объектам с меньшим уровнем конфиденциальности [3]. Каждому объекту в системе назначается определенный уровень конфиденциальности. Такие уровни образуют решетку уровней конфиденциальности. Применительно к рассматриваемой задаче наиболее выгодно назначать различные метки конфиденциальности равнозначным объектам в зависимости от предъявляемых требований. Однако использование такого подхода имеет следующие сложности, обуславливаемые определениями модели Белла–ЛаПадулы:

1. Назначенные уровни конфиденциальности объектов распространяются на все субъекты системы.
2. При наличии нескольких требований к выполнению одной задачи необходима функция, которая по некоторому правилу в соответствии с набором требований назначит уровни конфиденциальности для объектов системы, используя единственную решетку уровней конфиденциальности.
3. Не определены условия, по которым следует изменять метки конфиденциальности равнозначных объектов в соответствии с требованиями, предъявляемыми к решению задачи.
4. Возникают сложности в администрировании политики безопасности. Изменение меток конфиденциальности объектов для каждой задачи, которая назначается для выполнения субъекту, приводит к непрозрачности политики безопасности и, как следствие, может повлечь за собой ошибки, приводящие к компрометации защищаемой информации.

Широко применяется модель контроля доступа на основе ролей (Role-based Access Control, RBAC) [3, 6, 7], с помощью которой возможно выполнить тонкую настройку правил контроля доступа [8, 9]. Модель RBAC контролирует доступ субъектов системы на объекты в соответствии с совокупностью действий и обязанностей, связанных с определённым видом деятельности субъектов. Такие полномочия представляют собой семантические конструкции, называемые ролями субъектов, которые лежат в основе политики разграничения доступа. Роли поз-

воляют конкретным лицам получить доступ к объектам в той степени, в какой это необходимо им для выполнения своих обязанностей. Однако данная модель имеет целый ряд недостатков для применения в динамических системах, которые перечислены в [10, 11, 12, 13, 14, 15, 16]. В рассматриваемом аспекте для каждой роли необходимо 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. 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 [9]. Отметим следующие недостатки использования RBAC:

1. Решение, на какую из доступных пользователю ролей авторизоваться, принимает сам пользователь. При этом им могут быть допущены ошибки, связанные с выбором роли, а это может повлиять на безопасность системы.
2. Значительно увеличивается количество определяемых ролей по сравнению с реальным количеством функциональных обязанностей пользователя в системе.
3. Усложняются процедуры администрирования системы безопасности как на этапе её 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.

В основе модели Task-based Authorization Controls (TBAC) [17] лежит понятие «задача». Права доступа пользователей изменяются с учётом специфики выполняемой задачи, на которую он авторизован в данный момент, что обеспечивает динамический контроль доступа [15, 18]. Использование модели TBAC в данном контексте 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 путём добавления нескольких этапов авторизации для задач.
4. Для добавления новых субъектов системы необходимо только назначить для них этапы авторизации.

Однако администрирование политики безопасности при таком подходе является затруднительным. Это связано с большим количеством этапов авторизации, значительно превышающим количество задач, которые решаются в системе. Более того, её 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, к неверному функционированию системы безопасности.

В работах [10, 11, 16] предлагаются гибридные варианты моделей контроля доступа, построенных на основе RBAC и TBAC. Такие модели предоставляют механизмы по изменению доступов субъектов в процессе функционирования системы. Однако они сохраняют недостатки как RBAC, так и TBAC.

Более общий поход по формированию правил контроля доступа для динамических систем представлен моделью Dynamic Event-Based Access Control (DEBAC) [14]. Фундаментальным понятием модели DEBAC является «событие». В DEBAC права пользователя изменяются только при наступлении какого-либо события, которое определено в системе. В соответствии

с набором правил, которые должны быть применены при обработке данного события, могут измениться и права доступа пользователя в системе. Стоит отметить, что наступление события влечёт за собой и изменение требований по доступу пользователя к объектам системы. Отметим положительные стороны использования модели DEBAC:

1. В соответствии с определениями модели разграничение прав доступа субъектов системы происходит в соответствии с отдельными задачами, которые они выполняют.
2. Права субъектов в системе не являются постоянными и предоставляются только на некоторый период времени.
3. Модель позволяет реализовать разграничение доступа к равнозначным объектам системы путём добавления нескольких событий, которые соответствуют одной задаче.

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

Таким образом, использование как классических моделей безопасности (XPУ, RBAC), так и моделей, реализующих динамический контроль доступа (TBAC, DEBAC), имеет недостатки в рассматриваемом аспекте. При этом более приемлемым подходом для решения поставленной задачи является использование динамических моделей, основанных на задачах или событиях. Однако 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 [19] неформально описывается модель D-Task-based Authorization Controls (D-TBAC), где D означает множество требований в соответствии с элементами системы, учитывающая перечисленные особенности и позволяющая решить поставленную во введении задачу. Она предоставляет механизм, позволяющий выделять минимальные права доступа субъектам системы при решении ими задач с учетом требований к их выполнению. Однако требуется её формальное определение.

3. Модель D-TBAC

Предлагаемая модель состоит из следующих элементов:

S — множество субъектов;

T — множество задач;

O — множество объектов;

$G = \{g_i\} \mid \forall g_i \subseteq O$ и $\forall g_i, g_j \in G \;\; g_i \cap g_j = \emptyset$, где $g_i \neq g_j$ — множество групп объектов;

R — множество видов доступа (например, read, write, execute);

$P = 2^{G \times R}$ — множество операций над группами объектов;

D — множество требований;

L — множество решёток уровней требований и свойств объектов;

$ST : S \to 2^T$ — функция, определяющая для каждого субъекта множество задач, которые он может выполнять;

$TP : T \to 2^P$ — функция, определяющая для каждой задачи множество операций;

$TD : T \to 2^D$ — функция, определяющая для каждой задачи множество требований, которые должны быть выполнены при её выполнении;

$DL : D \to L$, $\forall d \in D\ \exists!(l_d, \leq) \in L : DL(d) = (l_d, \leq)$ — функция, задающая для каждого требования решётку, на которой определяется его уровень;

$f_d : d \to (l_d, \leq) \cup \{none\}$, где $d \in D$ и $DL(d) = (l_d, \leq)$ — функция, определяющая уровень требования, где $none$ означает, что уровень требования не определён;

$OL : O \to L$, $\forall o \in O\ \exists!(l_d, \leq) \in L : OL(o) = (l_d, \leq)$ — функция, задающая для каждого объекта решетку его свойств;

$f_o : o \to (l_d, \leq)$, где $o \in O$ и $OL(o) = (l_d, \leq)$ — функция, определяющая свойство объекта.

Определение 1. Требование $d \in D$ называется значимым для подмножества объектов $O_d \subseteq O$, если $\forall o \in O_d$ выполняется равенство $DL(d) = OL(o)$.

$DO \subseteq D \times O \mid \forall (d, o) \in DO$ верно $DL(d) = OL(o)$ — множество пар значимых требований и объектов.

Для любой пары $(d, o) \in DO$ определим двойку функций $(f_d, f_o) \in DL(d) \times OL(o)$, определяющих уровень требования $d \in D$ и уровень свойств объекта $o \in O$, для которого такое требование является значимым.

Тогда $(s, t, (f_d, f_o)) \in Y \subset S \times T \times (DL(d) \times DO(o))$, где $\forall (d, o) \in DO$ — тройка, определяющая уровень требования $d \in D$ к объекту $o \in O$ при решении задачи $t \in T$ субъектом $s \in S$.

$B \subseteq 2^{S \times O \times R}$ — множество возможных множеств текущих доступов в системе;

$CT \subseteq 2^{S \times T}$ — множество текущих задач в системе.

Структура элементов модели D-TBAC представлена на рис. 1.

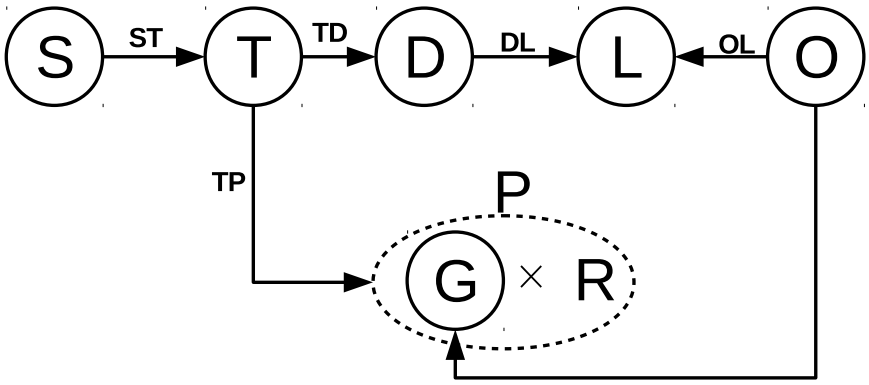

Рис. 1. Структура элементов модели D-TBAC

Определение 2. Пусть определены: множества субъектов S, объектов O, задач T, групп объектов G, видов доступа R, операций P, требований D, решёток требований и свойств объектов L, множество доступов B, а также функции, задающие задачи для субъектов ST, права доступа задач TP, требования для задач TD, решётки для требований DL, решётки свойств для объектов OL, а также функция Y.

Тогда $V = (S, T, O, G, R, P, D, L, ST, TP, TD, DL, OL, CT, B, Y)$ — состояние системы.

Введем обозначения:

Q — множество команд в системе;

$W : Q \times V \to V$ — функция переходов, где $W(q, v) = v^*$ означает, что система по команде $q \in Q$ из состояния v перешла в состояние v^*;

Определение 3. Под системой будем понимать конечный автомат $\sum(Q, V^*, W, v_0)$, где V^* — множество всех возможных состояний системы, v_0 — начальное состояние системы.

Утверждение 1. *В модели предполагается, что множества S, T, O, G, R, P, D, L и функции ST, TP, TD, DL, OL не меняются в процессе функционирования системы.*

Принимая во внимание утверждение 1, будем использовать сокращённое обозначение для состояния системы $V = (CT, B, Y)$.

Утверждение 2. *В модели D-TBAC предполагается, что в состоянии v_0:*

- $CT = \emptyset$ *— выполняемые задачи в данный момент времени отсутствуют;*
- $B = \emptyset$ *— субъекты не имеют прав доступа на объекты системы;*
- $\forall(d, o) \in DO$ $(f_d = none, f_o) => \forall(s, t, (f_d, f_o)) \in Y$ *верно* $(s, t, (f_d = none, f_o))$ *— уровни требований не определены.*

В модели рассматриваются следующие запросы, входящие во множество Q:

- *set_demad(s,t, d,val)* — запрос, задающий для требования $d \in D$ его уровнень $f_d = val$ для выполнения задачи $t \in T$ субъектом $s \in S$;
- *start_task(s,t)* — запрос запуска процесса решения задачи $t \in ST(s)$ субъектом $s \in S$;
- *stop_task(s)* — запрос прекращения решения задач субъектом $s \in S$.

Безопасность системы определяется с помощью четырёх свойств:

- *d-свойство* — свойство требования;
- *t-свойство* — свойство задачи;
- *f-свойство* — свойство полноты;
- *!-свойство* — свойство единственности.

Определение 4. Объект $o \in O$ называется соответствующим значимому требованию $d \in D$ относительно $(f_d, f_o) \in DL(d) \times OL(o)$, если $f_d = f_o$.

Определение 5. Объект $o \in O$ называется минимально удалённым от уровня значимого требования относительно $(f_d \neq none, f_o) \in DL(d) \times OL(o)$, когда $f_o < f_d$ и $\nexists o' \in O \mid o, o' \in g$, для которого выполняется $f_o < f_{o'} \leq f_d$.

Определение 6. Объект $o \in O$ обладает *d-свойством* относительно $(f_d, f_o) \in DL(d) \times OL(o)$, если $o \in O$ соответствует значимому требованию $d \in D$ или минимально удалён от него.

Определение 7. Доступ $(s, o, r) \in S \times O \times R$ обладает *d-свойством* относительно $(s, t, (f_d, f_o))$, когда объект $o \in O$ обладает *d-свойством* относительно (f_d, f_o).

Определение 8. Состояние системы $(CT, B, Y) \in V$ обладает *d-свойством*, когда выполняется одно из условий:

- если $CT \neq \emptyset$, то каждый доступ $(s, o, r) \in B$ обладает *d-свойством* относительно $(s, t, (f_d, f_o)) \in Y$;
- если $CT = \emptyset$, то $B = \emptyset$.

Обладание системы в каждом её состоянии *d-свойством* означает, что любой субъект $s \in S$ в каждый момент времени при выполнении задачи $t \in T$ имеет доступ к объектам, которые максимально могут удовлетворить предъявляемым требованиям.

Определение 9. Доступ $(s, o, r) \in S \times O \times R$ обладает *t-свойством* относительно $(s, t) \in S \times T$, когда выполняются два условия:

1. $t \in ST(s)$;
2. $\exists g \in G \mid o \in g$ и $(g, r) = p \in TP(t)$.

Определение 10. Состояние системы $(CT, B, Y) \in V$ обладает *t-свойством*, когда выполняется одно из условий:

- если $CT \neq \emptyset$, то каждый элемент $(s, o, r) \in B$ обладет *t-свойством* относительно элемента $(s, t) \in CT$;
- если $CT = \emptyset$, то $B = \emptyset$.

Обладание системы в каждом её состоянии *t-свойством* означает, что каждый доступ, который присутствует в данном состоянии системы, относится к выполнению задачи $t \in T$ субъектом $s \in S$.

Определение 11. Состояние системы $(CT, B, Y) \in V$ обладает *f-свойством*, когда выполняется одно из условий:

- если $CT \neq \emptyset$, то $\forall (s,t) \in CT$ и $\forall g \in G \mid (g,r) = p \in TP(t)$, верно что $\exists! o \in g \mid (s,o,r) \in B$;
- если $CT = \emptyset$, то $B = \emptyset$.

Обладание системы в каждом её состоянии *f-свойством* означает, что каждый субъект в любой момент времени при выполнении задачи имеет все необходимые доступы к объектам.

Определение 12. Множество текущих задач CT обладает *!-свойством*, когда выполняется условие: $\forall s \in S$ верно $|CT \cap (s \times T)| \leqslant 1$.

Определение 13. Состояние системы $(CT, B, Y) \in V$ обладает *!-свойством*, когда CT обладает *!-свойством*.

Обладание системы в каждом её состоянии *!-свойством* означает, что любой субъект $s \in S$ в каждый момент времени может выполнять не более одной задачи $t \in T$.

Определение 14. Состояние системы $(CT, B, Y) \in V$ называется безопасным, когда оно обладает одновременно *d-свойством*, *t-свойством*, *f-свойством* и *!-свойством*.

Определение 15. Система $\sum(Q, V^*, W, v_0)$ называется безопасной, когда каждое её состояние безопасно.

Правила преобразования состояний системы в модели D-TBAC показаны в табл. 1.

В модели определено множество объектов системы. При этом объекты по некоторому признаку, например, по функциональности, объединяются, образуя непересекающееся множество групп G. Операции, которые возможно выполнять над группой в процессе решения задачи $t \in T$, определяет функция TP. Если над группой $g \in G$ допускается выполнение операции $p \in P$, то её выполнение допускается над любым объектом $o \in G$.

В модели представлено множество требований D. Отображение TD ставит в соответствие каждой задаче подмножество требований, которые должны быть выполнены при её выполнении. Каждое требование может иметь различный уровень. Поэтому для каждого требования существует своя решётка уровней $(l_d, \leq) \in L$. Уровень требования определяется значением функции f_d.

Каждый объект из каждой группы имеет уровень характеристик своих свойств относительно значимого для него требования. Уровень таких свойств отображается на решётку уровней требования и свойств объекта функцией f_o. В одной группе не может существовать несколько объектов, имеющих одинаковый уровень характеристик. В группе может не существовать объекта с уровнем своих характеристик, совпадающим с требуемым уровнем значимого для него требования. В таком случае из группы предоставляется доступ на объект, уровень свойств которого минимально отличается от требуемого и не превышает его.

Состояние системы может изменяться только путём применения в ней трёх команд. Команда *set_demad(s,t,d,val)* устанавливает уровень требования $d \in D$ для решения задачи $t \in T$ субъектом $s \in S$. Значение этой функции определяет, к какому объекту будет предоставлен доступ субъекту $s \in S$ при выполнении задачи $t \in T$. Изменение уровня требования в процессе выполнения задачи субъектом не влияет на его текущие доступы в системе. *start_task(s,t)* запускает процесс решения задачи $t \in T$ субъектом $s \in S$. При выполнении такой команды в системе субъекту предоставляются все необходимые доступы, которые удовлетворяют обозначенным четырём свойствам. *stop_task(s)* прекращает выполнение всех задач в системе субъектом $s \in S$. Т.к. из *!-свойства* следует, что в системе может выполняться только одна задача одним субъектом, то при выполнении данной команды система переходит в состояние, из которого исключены все доступы субъекта $s \in S$. Стоит обратить внимание, что важен

Т а б л и ц а 1. Правила преобразования состояний системы в модели D-TBAC

Команда	Условия выполнения	Исходное состояние $V = (CT, B, Y)$	Результирующее состояние $V' = (CT', B', Y')$
set_demad(s,t, d,val)	отсутствуют	$s \in S$, $t \in T$, $d \in D$, $DL(d) = (l_d, \leq)$, $val \in (l_d, \leq)$	$CT' = CT$ $B' = B$, $Y' = Y \setminus \{(s,t,(f_d,f_o))\} \cup$ $\cup\{(s,t,(val,f_o))\}, \forall(s,t,(f_d,f_o)) \in Y$
start_task(s,t)	$\nexists(s,t^*) \in CT$, где $t^* \in T$	$s \in S$ $t \in ST(s)$	$Y' = Y$, $CT' = CT \cup (s,t)$ и для CT' верно *!-свойство*, $B' = B \cup B_{new}$, где $B_{new} \subset s \times O \times R \subset S \times O \times R$ и $\forall(s,o,r) \in B_{new}$ верно *d-свойство*, *t-свойство* и для $((s,t), B_{new}, Y')$ верно *f-свойство*
stop_task(s)	отсутствуют	$s \in S$	$Y' = Y$, $CT' = CT \setminus CT_{old}$, $CT_{old} = s \times T \subset S \times T$, $B' = B \setminus B_{old}$, $B_{old} = s \times O \times R \subset S \times O \times R$

порядок использования данных команд: не допускается применение команды *start_task(s,t)*, если в системе субъект $s \in S$ уже выполняет какую-либо задачу.

4. Применение модели D-TBAC

Для иллюстрации использования предлагаемой модели контроля доступа рассмотрим учреждение здравоохранения. Каждый субъект системы — врач, предоставляющий услуги лечения пациента. Пусть объектами, к которым необходимо контролировать доступ, являются лекарственные препараты. Для лечения одного и того же заболевания в распоряжении врача могут находиться различные препараты, имеющие одинаковые функциональные возможности (обезболивающие, успокоительные и пр.), но различные характеристики (стоимость, влияние на организм, побочные эффекты, привыкание и др.). Поэтому к процессу лечения пациента применяются определённые требования. При выполнении задачи субъект должен иметь доступ только к одному препарату из каждой группы объектов. В соответствии с определениями модели D-TBAC имеем следующие элементы:

$S = \{doctor_1, doctor_2, doctor_3\}$ — множество субъектов состоит из множества врачей;

$T = \{treatment_1, treatment_2\}$ — множество задач состоит из двух способов лечения;

$O = \{drug_1, drug_2, drug_3, drug_4, drug_5, drug_6, drug_7\}$ — множество объектов состоит из имеющихся в больнице препаратов;

$G = \{g_1 = \{drug_1, drug_3, drug_6\}, g_2 = \{drug_2, drug_5\}, g_3 = \{drug_4, drug_7\}\}$ — множество групп объектов образуется исходя из функционального назначения препаратов;

$R = \{apply\}$ — множество видов доступа состоит из одного элемента — «применить»;

$P = \{p_1 = g_1 \times apply, p_2 = g_2 \times apply, p_3 = g_3 \times apply\}$ — множество операций над группами объектов;

$D = \{price, effect, sideEffect\}$ — к задачам лечения пациента могут предъявляться требования по стоимости, эффекту и побочному влиянию на организм лекарственных препаратов;

$L = \{(l_{price}, \leq) = \{low, medium, high\}, (l_{effect}, \leq) = \{low, high\}\}, (l_{sideEffect}, \leq) = \{low, high\}\}$ — множество решёток, задающих уровни требований и свойства объектов относительно требований;

$ST(doctor_1) = ST(doctor_2) = \{treatment_1\}$, $ST(doctor_3) = \{treatment_2\}\}$ — для каждого врача определены задачи, которые он может выполнять;

$TP(treatment_1) = \{p_1, p_2\}$, $TP(treatment_2) = \{p_2, p_3\}$ — функция, определяющая права доступа при решении задач;

$TD(treatment_1) = \{price, effect\}, TD(treatment_2) = \{price, sideEffect\}$ — функция TD задаёт для каждой задачи множество требований, которые должны быть выполнены при её решении;

$DL(price) = (l_{price}, \leq)$, $DL(effect) = (l_{effect}, \leq)$, $DL(sideEffect) = (l_{sideEffect}, \leq)$ — для каждого требования задаётся своя решётка уровней;

Для каждого препарата определим решётку, отображающую уровень его свойств относительно значимого требования, следующим образом:

$OL(drug_1) = OL(drug_3) = OL(drug_6) = (l_{effect}, \leq)$,

$OL(drug_2) = OL(drug_5) = (l_{price}, \leq)$,

$OL(drug_4) = OL(drug_7) = (l_{sideEffect}, \leq)$.

Для каждого объекта $o \in O$ определён уровень его характеристик f_o относительно значимого к нему требования. Значения функции f_o для каждого объекта представлены на рис. 2.

$DO = \{(effect, drug_1), (effect, drug_3), (effect, drug_6), (price, drug_5),$

$(price, drug_6), (sideEffect, drug_4), (sideEffect, drug_7)\}$ — множество пар значимых требований и объектов.

Структура связей между элементами модели D-TBAC применительно к медицинскому учреждению показана на рис. 3.

Состояние системы может изменяться путём использования трёх команд: *set_demad*, *start_task*, *stop_task*. Пусть начальное состояние системы $V = (CT, B, Y)$ будет:

- $CT = \emptyset$;
- $B = \emptyset$;
- $\forall(s, t, (f_d, f_o)) \in Y$ верно $(s, t, (f_d = none, f_o))$.

Пусть субъекту $doctor_1$ необходимо выполнить задачу $treatment_1$ в определённых требованиях. Тогда последовательность преобразований системы в соответствии с моделью D-TBAC будет выглядеть, как представлено в табл. 2, в которой через $V = (CT, B, Y)$ обозначено исходное состояние системы (до выполнения команды), а через $V' = (CT', B', Y')$ обозначено результирующее состояние системы (после выполнения команды).

Аналогичным образом модель позволяет выполнить задачу $treatment_1$ субъектом $doctor_2$, но уже в других требованиях к применению лекарственных препаратов. При этом различные уровни требований, которые могут быть установлены субъектам $doctor_1$ и $doctor_2$ для выполнения задачи $treatment_1$, не имеют корреляции относительно множества доступов для

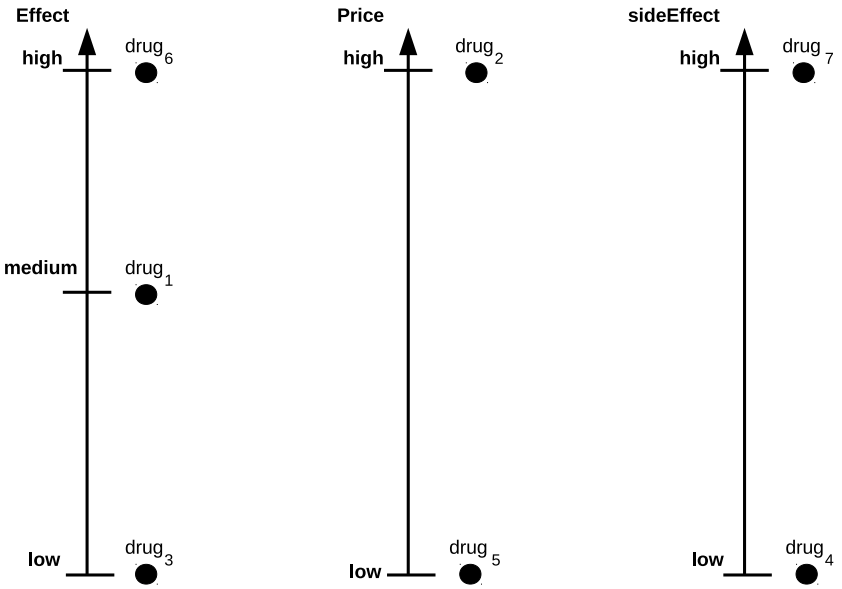

Рис. 2. Характеристики лекарственных препаратов относительно значимых требований

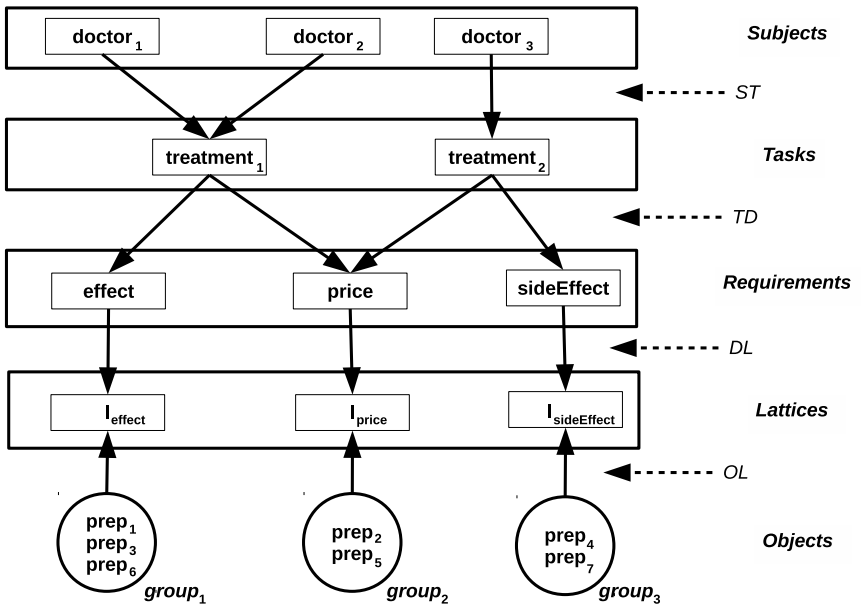

Рис. 3. Структура элементов модели D-TBAC применительно к медицинскому учреждению

данных субъектов между собой. Аналогично требования, установленные для решения задачи $treatment_2$ субъектом $doctor_3$, не влияют на доступы субъектов $doctor_1$ и $doctor_2$. Таким образом, каждый врач учреждения здравоохранения имеет доступ к минимальному набору лекарств, необходимых для выполнения своей задачи, при этом установленные требования к процедуре лечения никак не влияют на множество доступов других врачей, даже при условии выполнения одной и той же задачи.

Т а б л и ц а 2. Последовательность преобразований состояния системы

Запрос	Результирующее состояние
$set\_demad(doctor_1,$ $treatment_1,$ $effect,$ $medium)$	$CT^{'} = CT, B^{'} = B,$ $Y^{'} = Y \setminus Y_{old} \cup Y_{new},$ $Y_{old} = \{(doctor_1, treatment_1, (f_{effect} = none, f_{drug_3} = low)),$ $(doctor_1, treatment_1, (f_{effect} = none, f_{drug_1} = medium)),$ $(doctor_1, treatment_1, (f_{effect} = none, f_{drug_6} = high))\},$ $Y_{new} = \{(doctor_1, treatment_1, (f_{effect} = medium, f_{drug_3} = low)),$ $(doctor_1, treatment_1, (f_{effect} = medium, f_{drug_1} = medium)),$ $(doctor_1, treatment_1, (f_{effect} = medium, f_{drug_6} = high))\},$
$set\_demad(doctor_1,$ $treatment_1,$ $price,$ $high)$	$CT^{'} = CT, B^{'} = B,$ $Y^{'} = Y \setminus Y_{old} \cup Y_{new},$ $Y_{old} = \{(doctor_1, treatment_1, (f_{price} = none, f_{drug_6} = low)),$ $(doctor_1, treatment_1, (f_{price} = none, f_{drug_2} = high))\},$ $Y_{new} = \{(doctor_1, treatment_1, (f_{price} = high, f_{drug_6} = low)),$ $(doctor_1, treatment_1, (f_{price} = high, f_{drug_2} = high)),$
$start\_task(doctor_1,$ $treatment_1)$	$Y^{'} = Y,$ $CT^{'} = CT \cup \{doctor_1, treatment_1\},$ $B^{'} = B \cup B_{new},$ $B_{new} = \{(doctor_1, drug_1, apply), (doctor_1, drug_2, apply)\}$
$stop\_task(doctor_1)$	$Y^{'} = Y,$ $CT^{'} = CT \setminus \{doctor_1, treatment_1\},$ $B^{'} = B \setminus B_{old},$ $B_{old} = \{(doctor_1, drug_1, apply), (doctor_1, drug_2, apply)\}$

5. Заключение

В настоящей статье предложена и формально определена модель D-TBAC, расширяющая TBAC. Как показано в разделе 2, применение существующих моделей разграничения доступа в рассматриваемых системах является затруднительным. Это связано с тем, что в них предполагается, что одна и та же задача будет решена субъектом системы через использование одного множества объектов, к которому ему предоставляется доступ. Однако не учитывается, что к процессу решения задачи могут предъявляться определённые требования. В зависимости от таких требований для решения одной и той же задачи субъекту могут предоставляться различные доступы к равнозначным объектам.

В разделе 3 формально определены множества элементов, из которых состоит модель D-TBAC, свойства представленной модели, описывающие безопасность системы, а также правила, в соответствии с которыми система переходит из одного состояние в другое. Модель D-TBAC предоставляет механизм, позволяющий выделить минимальные права доступа субъектов системы к равнозначным объектам. В разделе 4 показано, каким образом возможно применить модель D-TBAC в учреждении здравоохранения.

Среди достоинств предлагаемой модели можно обозначить следующие:

1. Модель позволяет предъявлять субъекту требования к процессу решения поставленной задачи.
2. Модель предоставляет прозрачные правила для разграничения прав доступа на равнозначные объекты из каждой группы.
3. Модель выделяет минимально необходимые права доступа субъектам системы при наличии в ней равнозначных объектов.

4. Разграничение прав доступа субъектов системы на объекты системы происходит в соответствии с задачами, которые они выполняют в данный момент времени.
5. Модель позволяет реализовать динамический контроль доступа.
6. Простота администрирования политики безопасности, построенной в соответствии с моделью.

Представленная в данной статье модель D-TBAC может найти применение в динамических системах, в которых стоит вопрос выбора объектов для решения определенного круга задач, в кибер-физических системах, в информационных и телекоммуникационных системах [1]. Однако модель нуждается в дальнейшем её исследовании. Особый интерес представляют исследования, ориентированные на применение модели D-TBAC в рамках существующих моделей безопасности в качестве их дополнения и построения гибридных моделей.

Из всего вышесказанного следует, что поставленная во введении цель достигнута, а обозначенные задачи решены.

Литература

1. *Лапин С. А.* Применение модели контроля доступа D-TBAC для разграничения доступа пользователей к каналу связи в системах предоставления мультимедиа контента // Сборник научных статей международной конференции «Ломоносовские чтения на Алтае: фундаментальные проблемы науки и образования». Барнаул, 20–24 октября, 2015. С. 1129–1132.
2. *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, 15–19 ноября, 2015. С. 184—187.
3. *Девянин П. Н.* Модели безопасности компьютерных систем. Управление доступом и информационными потоками. Учебное пособие для вузов. 2-е изд., испр. и доп. М.: Горячая линия–Телеком, 2013. 338 с.
4. *Harrison, Michael A. and Ruzzo, Walter L. and Ullman, Jeffrey D.* Protection in Operating Systems // Commun. ACM. 1976. V. 19, № 8. P. 461–471.
5. *Bell D., LaPadula L.* Secure Computer Systems: Unified Exposition and Multics Interpretation // Bedford, Mass. MITRE Corp. 1976. MTR 2997. Rev. 1.
6. *Sandhu R.* Role-based Access Control // Advances in Computers. 1998. V. 46. P. 237–286.
7. *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. P. 224–274.
8. *Головин А. В., Поляков В. В., Лапин С. А.* Ролевое разграничение доступа для автоматизированного рабочего места пользователя при оперативном удаленном управлении конфиденциальной информацией // Доклады Томского государственного университета систем управления и радиоэлектроники. 2010. Т. 21, № 1. С. 143—145.
9. *Лепешкин О. М., Харечкин П. В.* Анализ моделей разграничения доступа, реализованных в современных социотехнических системах // Инфокоммуникационные технологии. 2008. Т. 6, № 2. С. 91–93.
10. *Лепешкин О. М., Харечкин П. В.* Функционально-ролевая модель управления доступом в социотехнических системах // Известия Южного федерального университета. Технические науки. 2009. Т. 100, № 11. С. 52—57.
11. *Zhang C., Hu Y., Zhang G.* Task-role based dual system access control model // IJCSNS International Journal of Computer Science and Network Security. 2006. V. 6, № 7 P. 211–215.

12. *Ferraiolo D., Cugini J., Kuhn R.* Role-based access control (RBAC): Features and motivations // Proceedings of 11th annual computer security application conference. New Orleans, Louisiana, December 11–15, 1995. P. 241–248.
13. *Freudenthal E., Pesin T., Port L., Keenan E., Karamcheti V.* dRBAC: distributed role-based access control for dynamic coalition environments // Proceedings of the 22-nd International Conference on Distributed Computing Systems (ICDCS'02). Vienna, Austria, July 2–5, 2002. P. 411–420.
14. *Bertolissi, Clara and Fernández, Maribel and Barker, Steve* Dynamic event-based access control as term rewriting // 21st Annual IFIP WG 11.3 Working Conference on Data and Applications Security. Redondo Beach, CA, USA, July 8–11, 2007. P. 195–210.
15. *Lu, Yahui and Zhang, Li and Sun, Jiaguang* Task-activity based access control for process collaboration environments // Computers in Industry. 2009. V. 60, № 6. P. 403–415.
16. *Oh S., Park S.* Task-role-based access control model // Information Systems. 2003. V. 28, № 6. P. 533–562.
17. *Thomas R. K., Sandhu R. S.* Task-based authorization controls (TBAC): A family of models for active and enterprise-oriented authorization management // Proceedings of the IFIP TC11 WG11.3 Eleventh International Conference on Database Securty XI: Status and Prospects. Lake Tahoe, California, USA, August 10–13, 1997. P. 166–181.
18. *Cvrček D.* Access Control in Workflow Systems // MOSIS'99 Proceedings. Rožnov pod Radhoštěm, CZ, April 27–29, 1999. P 93–100.
19. *Лапин С. А.* Неформальное описание модели разграничения доступа на основе задач с учетом требований к их выполнению // Проблемы правовой и технической защиты информации. 2015. № 3. С. 52—57.

Статья поступила в редакцию 01.12.2015;
переработанный вариант – 28.03.2016.

Лапин Сергей Александрович
аспирант кафедры прикладной физики, электроники и информационной безопасности АлтГУ (656049, Барнаул, пр. Ленина, 86), тел. 8 (3852) 364809, e-mail: lapinsa567@gmail.com.

Access control model D-TBAC with the requirements to carry out tasks

S. Lapin

This article describes access control model D-TBAC which extends TBAC. The presented model takes into account the requirements for performing the tasks. Model elements, properties, and rules of the states transformation are formally defined.

Keywords: IT-security, mathematical models of security, access control, dynamical systems, tasks, TBAC, requirements.