УДК 004.056 **DOI:10.30837/rt.2021.2.205.05**

В.В. ВИЛИГУРА

АНАЛИЗ ФОРМАЛЬНЫХ МОДЕЛЕЙ УПРАВЛЕНИЯ ДОСТУПОМ И ОСОБЕННОСТИ ИХ ПРИМЕНИМОСТИ ДЛЯ БАЗ ДАННЫХ

Введение

Обеспечить безопасность информационной системы (ИС) легче, если есть четкая модель того, что нужно защищать и кому и что разрешено делать [1]. Поэтому неотъемлемой частью любого проекта по созданию или оценке безопасности ИС, в том числе и баз данных (БД), как отмечается в работе [2], является наличие *модели безопасности*, под которой понимается формальное представление политики безопасности [3, 4]. Основная цель создания политики безопасности ИС и описания ее в виде формальной модели – это определение условий, которым должно подчиняться поведение системы, определение показателей и критерия безопасности, а также проведение формального доказательства соответствия защищаемой системы этому критерию при соблюдении установленных правил и ограничений [4].

Формальная модель политики безопасности, представленная в виде математических выражений, схем, диаграмм, алгоритмов, играет важную роль в процессах разработки и исследования информационных систем в целом и БД в частности, так как обеспечивает системный подход. Модели безопасности позволяют решить задачи, возникающие в ходе разработки и исследования защищенных систем. Так, используя эти модели, заказчики могут в четко определенной форме формулировать требования к создаваемым защищенным ИС, которые соответствуют политике безопасности, принятой в организации, а также оценить соответствие защищенных систем своим потребностям. Разработчики на основе моделей безопасности составляют спецификацию политики безопасности разрабатываемой системы. Эксперты используют модели безопасности в качестве эталонов в ходе анализа адекватности реализации политики безопасности в защищенных системах. На основе этих моделей они создают методики оценки защищенности конкретных ИС, осуществляют сертификацию разработанных систем по требованиям защиты информации. Благодаря формальным моделям можно, опираясь на объективные и неопровержимые положения математической теории, доказать безопасность системы. Разработка соответствующей модели – это первый шаг на пути к созданию теоретических основ обеспечения безопасности в ИС [5].

В данной работе рассмотрим основные положения наиболее распространенных моделей безопасности, основанных на контроле доступа субъектов к объектам, и общем критерии безопасности ИС, который формулируется следующим образом: информационная система безопасна тогда и только тогда, когда субъекты не имеют никаких возможностей нарушать (обходить) установленную в ней политику безопасности.

Модели управления доступом

Наиболее распространенной парадигмой моделей обеспечения безопасности данных в целом и моделей управления доступом, в частности, является субъектно-объектная абстракция. Формально субъектно-объектная модель ИС это тройка: $\langle S, O, Op \rangle$, в которой $S = \{s_1, s_2, ..., s_m\}$ – субъекты (пользователи, процессы); $O = \{o_1, o_2, ..., o_n\}$ – объекты; $Op = \{op_1, op_2, ..., op_L\}$ – множество операций над объектами. Под *субъектом* понимается активная сущность ИС, которая может изменять состояние системы посредством порождения процессов над объектами, в том числе, порождать новые объекты и инициировать порождение новых субъектов. Под *объектом* понимается пассивная сущность ИС, процессы над которой могут в определенных случаях быть источником порождения новых субъектов [6]. Для баз данных такими объектами могут быть следующие: таблицы, представления (англ. view), домены (типы), атрибуты, кортежи, процедуры, функции, триггеры, синонимы, про-

фили и некоторые другие. Активность понимается как возможность выполнять операции над объектами (пассивными сущностями).

Остановимся более подробно на моделях управления доступом:

- модели безопасности на основе дискреционной политики;
- модели безопасности на основе мандатной политики;
- модели безопасности на основе ролевой политики.

Модели безопасности на основе дискреционной политики

Работы по моделям дискреционного доступа к информации в ИС появились еще в 60 – 70-х годах прошлого столетия. Они достаточно широко освещены в научной литературе. Наиболее известные из них – модель ADEPT-50 [7], пятимерное пространство Хартсона [8], модель Хариссона – Руззо – Ульмана [9], модель Take-Grant [10]. Авторами этих моделей был внесен значительный вклад в теорию безопасности компьютерных систем. Их работы заложили основу для последующего создания и развития защищенных ИС.

В теоретическом и практическом плане наибольшее развитие и применение получили дискреционные модели, основанные на *матрице доступа* – таблице (M), описывающей права доступа субъектов (S) к объектам (O), строки которой соответствуют субъектам доступа $s_1, s_2, ..., s_m$, столбцы – объектам доступа $o_1, o_2, ..., o_n$, а в ячейках (элементах матрицы $M[s_i, o_j]$) записываются разрешенные операции (виды доступа) $op_1, op_2, ..., op_L$ соответствующего субъекта над соответствующим объектом.

В приведенной на рис. 1 матрице доступа M виды доступа op_l ($l = 1...L$) допускают следующие операции (виды доступа): чтение (r_d), запись с модификацией (w), запись без модификации (только с новой записью или дописыванием в файл) (a), запуск объекта на выполнение (e). Однако, как отмечается в [5], при необходимости элементы матрицы могут содержать указатели на процедуры. Эти процедуры исполняются при каждой попытке доступа к заданному объекту. Тем самым решение о доступе может приниматься на основании более сложных зависимостей не столь очевидных, как в простой матрице доступа.

	o_1	o_2	…	o_i		o_n
s_1	r_d	r_d, w		e		r_d
s_2	r_d, a	-		r_d		e
…						
s_i	r_d	-		-		r_d
…						
s_m	r_d, w	-		e		e

Рис. 1. Матрица доступа M

Данная модель предполагает, что все попытки доступа к объектам перехватываются и проверяются специальным управляющим процессом. Таким образом, субъект s_i получит инициируемый им доступ op_l к объекту o_j только в случае, если элемент матрицы $M[s_i, o_j]$ имеет значение op_l.

По принципу управления доступом выделяются два подхода [2, 6]:

- принудительное управление доступом;
- добровольное управление доступом.

Принцип принудительного управления доступом основывается на парадигме доверенных субъектов. Согласно этому принципу право создания и изменения матрицы доступа имеют только субъекты администратора системы, который при регистрации для работы в

системе нового пользователя создает с соответствующим заполнением новую строку матрицы доступа, а при возникновении нового объекта, подлежащего избирательному доступу, образует новый столбец матрицы доступа.

Как можно заметить, принудительный способ обеспечивает жесткое централизованное управление доступом. Как итог, ему не хватает гибкости и точности настройки системы разграничения доступа в зависимости от потребностей и полномочий пользователей, которые имеют наиболее полное представление о содержимом и конфиденциальности объектов (активов), как их владельцы.

Принцип добровольного управления доступом основывается на понятии владения объектами. *Владельцем* объекта o_j доступа называется субъект s_i, инициализировавший процесс, в результате которого объект возник в системе. Тем самым, в дополнение к основным положениям субъектно-объектной модели вводится специальное отображение множества объектов на множество субъектов доступа, называемое владением, ставящее в каждый фиксированный момент времени каждому объекту системы подмножество субъектов доступа, инициализированных пользователем-владельцем объекта.

Добровольное управление доступом выражается следующим правилом: *права доступа к объекту определяют (устанавливают) их владельцы*. Из данного правила следует, что формирование элементов матрицы доступа осуществляют субъекты, являющиеся владельцами соответствующих объектов.

На практике в большинстве случаев, в том числе и для БД, применяется комбинированный способ управления доступом, когда определенная часть полномочий на доступ к объектам устанавливается администратором (привилегированным пользователем), а другая часть – владельцами объектов.

Во многих ИС права владения объектами могут передаваться. В результате при добровольном управлении доступом реализуется полностью децентрализованный принцип управления процессом разграничения доступа. Такой подход обеспечивает гибкость формирования правил разграничения доступа для конкретной совокупности пользователей к активам, но приводит к усложнению общего контроля состояния безопасности активов в системе [6]. Это, в свою очередь, требует дополнительного исследования условий и процессов распространения прав доступа.

В теоретическом плане впервые данная проблема была исследована Харрисоном, Руззо и Ульманом, они разработали специальную формальную модель дискреционного доступа (сокращенно модель HRU) [9].

В данной модели дополнительно к субъектам S, объектам O (для того чтобы включить в область действия модели и отношения между субъектами, принято считать, что все субъекты одновременно являются и объектами: $S \subseteq O$) и конечному набору прав доступа $R = (r_1, r_2, ..., r_K)$ ($M[s,o] \subseteq R$) вводится пространство состояний системы $Q = (S, O, M)$. Пространство состояний системы образуется декартовым произведением множеств составляющих ее объектов, субъектов и прав: $S \times O \times M$. Любой элемент матрицы M содержит набор прав субъекта s_i к объекту o_j, принадлежащих множеству прав доступа R. Поведение системы во времени рассматривается как последовательность состояний $\{Q_v\}$, каждое последующее состояние является результатом применения некоторой команды ($\alpha_\tau \in C$, где C – конечный набор команд) к предыдущему: $Q_{v+1} = \alpha_\tau(Q_v)$.

Критерий безопасности в модели HRU формулируется следующим образом.

Система является безопасной относительно права r_k ($k = 1...K$; для простоты часто в дальнейшем вместо r_k будем использовать обобщенное обозначение $r \in R$), если для заданного начального состояния $Q_0 = (S_0, O_0, M_0)$ не существует применимой к Q_0 последова-

тельности команд, в результате которой право r будет занесено в элемент матрицы M, в которой оно отсутствовало в начальном состоянии Q_0.

Другими словами, это означает, что субъект никогда не получит право доступа r к объекту, если он не имел его изначально. Если же право r оказалось в ячейке матрицы M, в которой оно изначально отсутствовало, то говорят, что произошла утечка права r [11].

Из критерия безопасности следует, что для данной модели ключевую роль играет выбор значений прав доступа и их использование в условиях команд. По сути, данная модель описывает не только доступ субъектов к объектам, а распространение прав доступа от субъекта к субъекту, поскольку именно изменение содержания элементов матрицы доступа определяет возможность выполнения команд, в том числе команд, модифицирующих саму матрицу доступа, которые потенциально могут привести к нарушению критерия безопасности.

Однако Харрисон, Руззо и Ульман доказали, что в общем случае не существует алгоритма, который может для произвольной системы, ее начального состояния $Q_0 = (S_0, O_0, M_0)$ и общего права r решить, является ли данная конфигурация безопасной. Помимо проблем с неопределенностью распространения прав доступа в системах на основе модели HRU была подмечена еще одна серьезная проблема – уязвимость по отношению к атаке с помощью «троянского коня». В таких моделях контролируются только операции доступа субъектов к объектам, а не потоки информации между ними. Поэтому, например, когда некоторая троянская программа переносит информацию из доступного некоторому пользователю объекта в объект, доступный злоумышленнику (например, из одной таблицы в другую), то формально никакое правило дискреционной политики безопасности не нарушается, но утечка информации происходит.

Таким образом, дискреционная модель HRU в своей общей постановке не дает гарантий безопасности системы. Однако именно она послужила основой для целого класса моделей политик безопасности таких, например, как модель типизованной матрицы доступа (модель Type Access Matrix – TAM [12]), модель TAKE-GRANT [10] и некоторых других, которые используются для управления доступом и контроля за распространением прав в различных системах [4]. Развитие моделей дискреционного управления доступом заключается преимущественно в построении всевозможных модификаций модели HRU, а также в поиске минимально возможных ограничений, которые можно наложить на описание системы, чтобы вопрос ее безопасности был вычислительно разрешимым [11].

Так, для смягчения условий, в которых можно проводить формальное доказательство безопасности, а также для введения контроля за порождением объектов, как отмечается в работе [6], была предложена модель TAM. Модель типизованной матрицы доступа является обобщением модели HRU, которую можно рассматривать как частный случай TAM с одним единственным типом для всех объектов и субъектов [4]. С другой стороны, любую систему TAM можно выразить через систему HRU, введя для обозначения типов специальные права доступа, а проверку типов в командах заменив проверкой наличия соответствующих прав доступа. Тем не менее введение строгого контроля типов ($T = (t_1, t_2, ..., t_\Theta)$, с одним из которых создается любой объект (включая субъекты)) в модель HRU позволило доказать критерий безопасности систем для более приемлемых ограничений, что существенно расширило область ее применения [13]. Так, используя специальные типы, например «файл», «программа» для объектов; «user», «administrator», «auditor» для субъектов, в ИС на основе TAM можно организовать эффективный контроль порождения субъектов с нейтрализацией проблемы троянских программ [6].

Еще одной моделью, имеющей важное теоретическое значение в исследовании процессов распространения прав доступа в системах, основанных на политике дискреционного доступа, является модель TAKE-GRANT. Данная модель ориентирована на анализ путей распространения прав доступа в системах дискреционного управления доступом. Исходя из основных положений субъектно-объектной парадигмы построения ИС, модель TAKE-

GRANT использует аппарат теории графов для моделирования системы разграничения доступа и процессов ее изменения.

Состояние системы описывается соответствующим ему графом доступов ($G = (S, O, E)$), в котором множество вершин – это множество объектов O субъектов S доступа, причем $S \subseteq O$, а множество ребер – это множество E установленных прав доступа (x, y, α) субъекта x к объекту y с правом α из конечного набора прав $\alpha \subseteq R(r_1, r_2, ..., r_K) \bigcup \{t, g\}$, в том числе с двумя особыми правами – правом *take* (t – право брать права доступа у какого-либо объекта по отношению к другому объекту) и правом *grant* (g – право предоставлять права доступа к определенному объекту другому субъекту). При этом, в отличие от модели HRU, в модели TAKE-GRANT возможно наличие прав доступа не только у субъектов к объектам, но и у объектов к объектам. Основная цель модели TAKE-GRANT – определение и обоснование алгоритмически проверяемых условий проверки возможности утечки права доступа по исходному графу доступов, соответствующего некоторому состоянию системы [13]. Используя правила преобразования «*take*», «*grant*», «*create*» и «*delete*», можно воспроизвести состояния, в которых будет находиться система в зависимости от распределения и изменения прав доступа. Следовательно, можно проанализировать возможные угрозы для данной системы [14].

Модель TAKE-GRANT, как и ее расширенная модель [15], в которой $\alpha \subseteq R(r_1, r_2, ..., r_K) \bigcup \{t, g\} \{r_d, w\}$ (где r_d (*read*) – право на чтение или информационный поток на чтение, w (*write*) – право на запись или информационный поток на запись), играет важную методологическую роль, предоставляя теоретико-графовый инструмент анализа систем разграничения доступа с точки зрения санкционированного и несанкционированного со стороны определенных субъектов распространения прав доступа в рамках дискреционной политики [6].

Основные решения этих моделей нашли применение при предоставлении пользователям прав на выполнение тех или иных операций с теми или иными объектами СУБД (в большей мере реляционных). А именно, дискреционная модель разграничения доступа в СУБД основывается на следующих основных положениях.

1. Все субъекты S и объекты O БД должны быть идентифицированы, то есть каждой сущности (активной, пассивной) должен быть присвоен уникальный идентификатор.

2. Для каждого объекта o_j БД должен быть определен пользователь-владелец $s^{owner_j} = s_i \in S$.

3. Для субъектов S должны быть определены права доступа (иначе называемые привилегиями, или полномочиями) к разным объектам O. Например, в стандарте SQL определены такие типы привилегий доступа, как [16, 17]: SELECT, INSERT, DELETE, UPDATE, REFERENCES, USAGE, TRIGGER, EXECUTE, UNDER. Информация о привилегиях (аналог матрицы доступов M) сохраняется в системном каталоге СУБД в виде полномочий, выраженных с помощью некоторого языка описания. Эта информация используется системой для принятия решения о выполнении запрошенных субъектом операций над данными. При этом для принятия решения о том, выполнять запрошенные субъектом операции над данными или нет, система должна уметь аутентифицировать запрашивающий субъект.

В общем случае набор привилегий зависит от реализации СУБД (определяется производителем). Достаточно часто выделяют такие категории привилегий, как: *системные* (это право субъекта $s_i \in S$ выполнять некоторые административные действия с объектами O базы данных и с самой БД, например, такие как: создание базы данных, создание / удаление / изменение таблицы, создание/удаление представления, процедуры, триггера и т. д.) и *объектные* (права (разрешения) субъекта $s_i \in S$ на выполнение определенного действия с кон-

кретным объектом o_j схемы БД, таким как конкретная таблица, представление, последовательность, процедура, функция и т. д.).

Следует отметить еще одну особенность, присущую традиционным СУБД, связанную с особой ролью в обеспечении безопасности данных, наряду с привилегиями, представлений. Создавая представление и давая субъекту разрешение на доступ к нему, а не к исходной таблице, можно тем самым ограничить доступ субъекта, позволив ему обращаться только к определенным столбцам и строкам, на что еще в свое время обратил внимание автор монографии [5]. Таким образом, представления позволяют осуществлять контроль над тем, какие данные доступны тому или иному субъекту [18].

4. Владелец объекта s^{owner_j} должен обладать правом определения прав доступа к объекту другим субъектам $s_i \in S$.

Субъект получает/теряет определенные права (привилегии) двумя способами: первый связан с созданием объекта o_j и его владением, второй – путем передачи / отзыва определенных прав одним субъектом – другому [16]. Для этих целей, например, в стандарте SQL предусмотрены команды (операторы) GRANT/REVOKE, которые позволяет одному пользователю назначать (предоставлять) определенные права другому или отзывать их у другого. При этом «даритель» (лицо, предоставляющее право – англ. grantor) привилегий не лишается. С каждой привилегией ассоциировано право передачи (использование конструкции WITH GRANT OPTION), при котором имеется возможность передавать не только права на указанные действия, но и право передавать эти права другим субъектам. Назначенные привилегии могут быть в любой момент отозваны. Отзыв одной привилегии нередко связан с выполнением ряда каскадных действий по отмене иных привилегий: если отзывается привилегия, назначенная другим субъектам с правом передачи, она должна быть отозвана и у этих субъектов. Конструкции CASCADE, RESTRICT оператора REVOKE определяют, каким образом должна производиться отмена привилегий. Так, например, конструкция CASCADE отменяет привилегии не только для субъекта, который непосредственно упоминался в операторе GRANT при предоставлении ему соответствующих привилегий, но и для всех субъектов, которым этот субъект, воспользовавшись правом передачи WITH GRANT OPTION, предоставил привилегии.

Поскольку в процессе назначения привилегий и передачи прав на них собственно привилегии зачастую перекрываются и зависимости между теми, кто их предоставлял и теми, кто их получал, становятся все более сложными, структуры привилегий полезно представлять в виде графов (подобно теоретико-графовому инструменту модели TAKE-GRANT), называемых диаграммами назначения (англ. *grant diagrams*) [16]. СУБД поддерживают внутреннее представление подобных диаграмм для отслеживания событий назначения/отзыва привилегий.

5. В системе существует привилегированный пользователь $s^{pv} \in S$, обладающий правом полного доступа к любому объекту. При этом следует отметить, что реализация этого положения не должна позволять такому пользователю использовать свои полномочия незаметно для реального владельца объекта (что зачастую сегодня не всегда выполняется).

Подводя итог, можно сделать выводы о достоинствах и недостатках дискреционной политики управления доступом и моделях безопасности, построенных на ее основе.

К достоинствам дискреционной политики доступа можно отнести относительную простоту и гибкость реализации системы управления доступом. Это подтверждается действующими ИС, безопасность которых обеспечивается выполнением требований именно данной политики. Например, классическая модель HRU до сих пор широко используется при проведении формальной верификации корректности построения систем разграничения доступа в высоко защищенных автоматизированных системах [11, 19].

К недостаткам дискреционных моделей относятся:

– статичность установленных правил управления доступом. Данные модели, как правило, не учитывают динамику изменений состояний ИС;

– невозможность контролировать в полной мере потоки информации между объектами. Это обостряет проблему, связанную с уязвимостью по отношению к атаке с помощью вредоносных программ вида троянский конь;

– сложность отслеживания предоставляемых субъектам привилегий при их большом количестве;

– отсутствие механизма управления доступом к конфиденциальной информации. Как следствие, возможна утечка конфиденциальных данных.

Кроме этого, при использовании дискреционных моделей возникает вопрос о задании правил управления (распространения прав) доступом и анализа их влияния на безопасность ИС. В общем случае при использовании таких моделей достаточно сложно (алгоритмически трудноразрешимая задача) проверить, приведут ли действия субъекта, руководствующегося некоторым набором правил, к нарушению безопасности или нет.

Все это предопределило дальнейший поиск и разработку других моделей управления доступом.

Модели безопасности на основе мандатной политики

Исследователи и критики дискреционной политики, понимая, что основная проблема подобных моделей заключается в отсутствии контроля за информационными потоками, стали анализировать каким образом ее можно разрешить. Их внимание привлекло решение подобных задач в секретном делопроизводстве, в котором критерием безопасности является невозможность получения информации из документов определенного уровня безопасности (уровня / грифа секретности) субъектом доступа, чей уровень безопасности (допуска), ниже, чем уровень безопасности соответствующих документов.

Разграничение доступа и порядок работы с конфиденциальными документами организуются на основе парадигмы градации доверия определенным группам сотрудников в отношении секретов определенной степени важности. С этой целью вводится система уровней безопасности (уровней конфиденциальности/секретности). Сотрудники с самым высоким уровнем безопасности (уровнем доверия, допуска), могут работать с документами самой высокой степени (грифа) секретности. На более низком уровне безопасности (доверия, допуска), вводятся ограничения в отношении работы с документами более высокого уровня безопасности (секретности) и т. д. Соответственно, все сотрудники получают допуск к работе с секретными документами определенного уровня, а документы снабжаются специальной меткой (грифом секретности), отражающей требования к уровню безопасности при работе с ними.

Таким образом, если в дискреционных моделях управление доступом происходит путем предоставления субъектам полномочий для осуществления определенных операций над конкретными объектами, то мандатные модели управляют доступом неявным образом – с помощью назначения всем сущностям системы (субъектам, объектам) уровней безопасности, которые определяют все допустимые взаимодействия между ними. Следовательно, мандатное управление доступом не различает сущностей, которым присвоен одинаковый уровень безопасности, и на их взаимодействия ограничения отсутствуют [4]. То есть мандатный подход к разграничению доступа, основываясь только лишь на парадигме ранжированного доверия, без учета специфики других характеристик субъектов и объектов, приводит в большинстве случаев к избыточности прав доступа для конкретных субъектов в пределах соответствующих классов безопасности, что противоречит самому понятию разграничения доступа. Поэтому в тех ситуациях, когда управление доступом требует более гибкого подхода, мандатный принцип разграничения доступа дополняется дискреционным внутри соответствующих классов безопасности. В теоретических моделях для этого вводят матрицу доступа, раз-

граничивающую разрешенный по мандатному принципу доступ к объектам одного уровня безопасности [6].

Наиболее широкое распространение среди моделей мандатного управления доступа (многоуровневой защиты) получила модель Белла – ЛаПадулы [20]. Основными элементами данной модели являются:

– S – множество субъектов;

– O – множество объектов;

– $R = \{read, write, append, execute\} = \{r_d, w, a, e\}$ – множество видов доступа (a=*append* – доступ на запись в конец объекта);

– $B = \{b \subseteq S \times O \times R\}$ – множество возможных множеств текущих доступов в системе;

– Λ_L – решетка уровней безопасностей L (например, $L = \{U, SU, C, S, TS\}$, где U – *unclassified* (*несекретно*), SU – *sensitive but unclassified* (*чувствительно, но несекретно*), C – *confidential* (*конфиденциально*), S – *secret* (*секретно*), TS – top secret (*совершенно секретно*), $U < SU < C < S < TS$). Решеткой Λ_L называется алгебраическая система вида $(L, \leq, \sqcup, \otimes)$, где $\leq$ – оператор, определяющий частичное нестрогое отношение порядка для уровней безопасностей L; $\sqcup$ – оператор наименьшей верхней границы; $\otimes$ – оператор наибольшей нижней границы.

– $M = \{M_1, M_2, ..., M_c\}$ – множество возможных матриц доступов, где $c = n \cdot m \cdot 2^p$, $p = |R|$, $M_\eta[s, o] \subseteq R$ – права доступа субъекта s к объекту o, $\eta = 1..c$;

– $(f_s, f_o, f_c) \in F = L^S \times L^O \times L^S$ – тройка функций (f_s, f_o, f_c), задающих соответственно: $f_s : S \to L$ – уровень безопасности (доступа) субъектов; $f_o : O \to L$ – уровень безопасности объектов; $f_c : S \to L$ – текущий уровень безопасности (доступа) субъектов, при этом для любого $s \in S$ выполняется неравенство $f_c(s) \leq f_s(s)$;

– $V = B \times M \times F$ – множество состояний системы;

– Q – множество запросов системе;

– D – множество ответов по запросам, например $D = \{yes, no, error\}$;

– $W \subseteq Q \times D \times V \times V$ – множество действий системы, где четверка $(q, d, v^*, v) \in W$ означает, что система по запросу q с ответом d перешла из состояния v в состояние v^*;

– $T = \{1, 2, ..., t, ...\}$ – множество дискретных моментов времени;

– X – множество функций $x : T \to Q$, задающих все возможные последовательности запросов к системе;

– Y – множество функций $y : T \to D$, задающих все возможные последовательности ответов (решений – decision) системы по запросам;

– Z – множество функций $z : T \to V$, задающих все возможные последовательности состояний системы.

Основные свойства (правила, обеспечивающие разграничение доступа) безопасности модели Белла – ЛаПадулы:

1. *Свойство простой безопасности* (*simple security property – ss*). Субъект на уровне безопасности $l \in L$ может проводить операцию чтения только в отношении объектов своего или более низкого уровня. Это свойство также известно, как правило *no read up* (NRU) – нет чтения вверх.

2. *Свойство* * (**-property*). Субъект с заданным уровнем безопасности $l \in L$ может осуществлять запись только в объекты своего или более высокого уровня. Это свойство также известно, как правило *no write down* (NWD) – нет записи вниз.

3. *Свойство дискреционной безопасности* (*discretionary security – ds*). Права дискреционного доступа субъекта к объекту определяются на основе матрицы доступа M. Термин «дискреционная» безопасность уместен в контексте конкретных решений этой модели, поскольку в модель включена возможность изменять M (структуру разрешений) [20].

Характеризуя понятия безопасного состояния, Белл и ЛаПадула предложили следующее определения.

Состояние $(b, m, f) \in V$ называется *безопасным по чтению* (или *просто безопасным*) тогда и только тогда, когда для каждого субъекта, осуществляющего в этом состоянии доступ чтения к объекту, уровень безопасности этого субъекта доминирует над уровнем безопасности этого объекта. *Состояние* ($(b, m, f) \in V$) называется *безопасным по записи* (или **-безопасным*) тогда и только тогда, когда для каждого субъекта, осуществляющего в этом состоянии доступ записи к объекту, уровень безопасности этого объекта доминирует над уровнем безопасности этого субъекта. *Состояние системы* $(b, m, f) \in V$ *безопасно* тогда и только тогда, когда оно безопасно и по чтению, и по записи. А *система* $\Sigma(Q, D, W, z_0) \subset X \times Y \times Z$ (где z_0 – начальное состояние системы) называется *безопасной* тогда и только тогда, когда она обладает *s*-свойством, *-свойством, *ds*-свойством одновременно. Другими словами система $\Sigma(Q, D, W, z_0)$ безопасна тогда и только тогда, когда ее начальное состояние z_0 безопасно и все состояния, достижимые из z_0 путем применения конечной последовательности запросов из Q, безопасны.

Графическое представление модели Белла – ЛаПадулы показано на рис. 2.

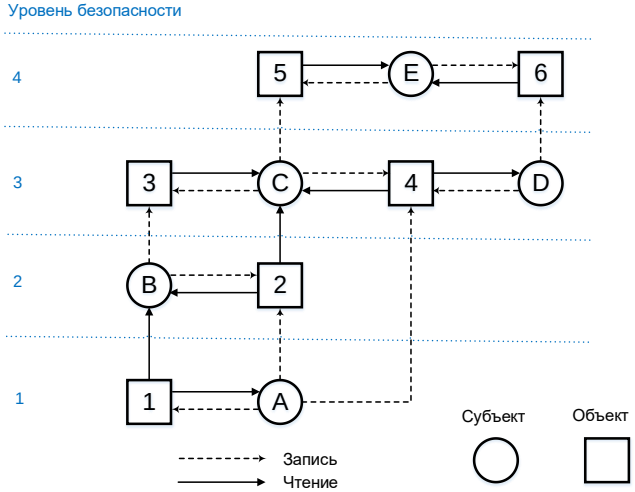

Рис. 2. Многоуровневая модель безопасности Белла – ЛаПадулы

На рис. 2 сплошная стрелка от объекта к субъекту показывает, что субъект осуществляет чтение объекта (информационный поток идет от объекта к субъекту). Пунктирная стрелка от субъекта к объекту показывает, что субъект осуществляет запись в объект (информационный поток идет от субъекта к объекту). Таким образом, направления информационных потоков указываются стрелками. При этом, как видно из рис. 2, например, субъект B может читать данные из объекта 1, но не может считывать данные из объекта 3.

Для СУБД особенно большое внимание методам мандатного управления доступом стало уделяться в начале 1990-х годов. Это было связано, как отмечается в работе [21], с тем фактом, что согласно требованиям Министерства обороны США любая используемая в этом ведомстве СУБД должна была поддерживать принципы мандатного управления доступом. Поэтому разработчикам СУБД пришлось вступить в соперничество за скорейшую разработку методов такого управления.

Мандатная модель управления доступа в СУБД основывается на следующих основных положениях:

1. Все субъекты S и объекты O БД должны быть идентифицированы.

2. Должна быть определена решетка уровней безопасностей L.

3. Каждому объекту БД $o \in O$ должен быть присвоен уровень безопасности (метка конфиденциальности), задающий установленные ограничения на доступ к данному объекту.

4. Каждому субъекту БД $s \in S$ должен быть присвоен уровень безопасности – уровень доступа, задающий уровень полномочий данного субъекта.

5. Субъект $s \in S$ может получить доступ к объекту БД $o \in O$ только в случае, когда уровень доступа субъекта позволяет предоставить ему данный доступ к объекту с заданным уровнем конфиденциальности, и реализация доступа не приведет к возникновению информационных потоков от объектов с высоким уровнем конфиденциальности к объектам с низким уровнем конфиденциальности.

Отличие мандатного управления доступа от дискреционного заключается в том, что в мандатной модели контролируются не операции, выполняемые субъектом над объектом, а потоки информации, которые могут быть только двух видов: либо от субъекта к объекту (запись), либо от объекта к субъекту (чтение).

В реляционной модели в качестве структуры, обладающей меткой, естественно выбрать кортеж (строку). Такое решение позволяет обеспечить достаточную избирательность доступа. Местом хранения самих меток может быть выбран соответствующий атрибут кортежа, определенный на домене уровней безопасностей L. При этом должен быть решен вопрос с механизмом формирования и изменения этого атрибута.

Один из подходов к построению системы управления доступом в реляционных СУБД на основе мандатной политики описан в группе патентов [22 – 24]. Предлагаемый подход обеспечивает контроль доступа на уровне строк (англ. *Row Level Security* – RLS) в таблице реляционной БД. В публикациях такой механизм упоминается под различными названиями: детальный контроль доступа (англ. *Fine Grained Access Control* – FGAC), виртуальная приватная база данных (англ. *Virtual Private Database* – VPD) и некоторых других [25, 26]. Суть детального контроля доступа (FGAC) состоит в следующем. Таблица базы данных содержит столбец метки безопасности, в строки которого записываются конкретные значения меток, определенные в иерархии уровней безопасностей L. Когда субъект $s \in S$ запрашивает доступ к строке (являющейся в рассматриваемом случае объектом доступа $o \in O$) таблицы, механизм безопасности сравнивает уровень безопасности субъекта с уровнем безопасности (меткой конфиденциальности), указанном в соответствующем атрибуте строки. Если уровень безопасности субъекта s преобладает над уровнем безопасности, указанном в соответствующем атрибуте строки, субъекту предоставляется доступ к строке.

Метка определяет уровень безопасности для субъекта в многоуровневой схеме безопасности и определенные привилегии для доступа к данным БД. Кроме того, метка безопасности может определять категории безопасности в рамках того уровня безопасности, к которому субъекту разрешен доступ. Примером категории безопасности может быть некоторый проект, к которому субъект допущен. Например, некоторому субъекту может быть разрешено просматривать данные, обозначенные определенными уровнями безопасности, такими как уровни безопасности: U, SU, C, S, TS. Этому же субъекту также может быть разрешен доступ к данным, относящимся к определенным категориям, таким как, например проекты $A1$, $A2$ и $A3$. Значение, хранящееся в метке, формируется некоторым способом, который позволяет понятно выразить информацию об уровне и категории безопасности для системы безопасности. Примером метки после такого преобразования (кодирования) может быть следующая: $SA1$, где $'S'$ указывает уровень безопасности – секретно, а $'A1'$ задает категорию безопасности, которая является идентификатором проекта, к которому субъект допущен.

Доступ к соответствующей строке разрешен только в том случае, если безопасность субъекта преобладает над безопасностью строки, в которой выполняются оба следующих условия:

– уровень безопасности, указанный меткой безопасности субъекта, больше или равен уровню безопасности, указанному меткой безопасности строки;

– категории безопасности, связанные с меткой безопасности строки, являются надлежащим подмножеством категорий безопасности, связанных с меткой безопасности субъекта.

Реализации технологии мандатного доступа в различных СУБД могут отличаться. Например, в СУБД Oracle она накладывается на реализацию дискреционной модели. А именно, возможность доступа к данным проходит проверку, содержащую два этапа: сначала проверяются стандартные ограничения дискреционного доступа. Затем для субъектов, прошедших проверку первого этапа, проверяется возможность доступа к объектам, базирующаяся на ограничениях мандатной модели. Механизм VPD в Oracle позволяет регламентировать доступ к частям таблицы.

Начиная с версии 8.1.7 в Oracle появилось другое средство – Oracle Label Security (OLS). В результате, реализация технологии мандатного доступа, основывающаяся на механизме OLS, стала опираться не только на дискреционную модель доступа (вначале проверяются права субъекта на выполнение соответствующей операции над таблицей), но и на механизм VPD (если у субъекта есть соответствующие привилегии, проверяется, не прикреплены ли к таблице какие-либо политики VPD). И только после всего этого проверяется наличие политик Oracle Label Security, назначенных защищаемой таблице: сравниваются метки, присвоенные отдельным строкам, с авторизацией меток пользователей, разрешая или запрещая доступ. Важным этапом развития по отношению к механизму виртуальных частных баз данных в OLS стала возможность формировать составную метку доступа.

В интерпретации Oracle (механизм OLS) метка безопасности – это тройка:

$$\lambda = \langle \lambda_1, \lambda_2, \lambda_3 \rangle, \tag{1.2}$$

где λ_1 – элемент линейно упорядоченного множества Θ_1 (множество уровней чувствительности / секретности). Задание в метке доступа уровня секретности является обязательным. Теоретически допускается до 10 000 уровней. Пример различных способов задания уровней секретности приведен в табл. 1;

Таблица 1

Пример различных способов задания уровня секретности

Числовая форма	***Длинная форма***	***Краткая форма***
40	HIGHLY_SENSITIVE	HS
30	SENSITIVE	S
20	CONFIDENTIAL	C
10	PUBLIC	P

λ_2 – подмножество элементов из множества Θ_2 (множество отделений (англ. compartments)). Отделения определяют области, которые описывают чувствительность помеченных данных, обеспечивая более тонкий уровень детализации в пределах уровня. Пример различных способов задания отделений приведен в табл. 2.

Таблица 2

Пример различных способов задания отделения

Числовая форма	***Длинная форма***	***Краткая форма***
85	FINANCIAL	FINCL
65	CHEMICAL	CHEM
45	OPERATIONAL	OP

Отделения не являются обязательными. Метка может содержать ноль или более отделений, то есть не все метки могут иметь отделения. OLS позволяет определять до 10 000 отделений [27];

λ_3 – элемент иерархически упорядоченных элементов (дерева) множества Θ_3 (множество групп (англ. groups)). Все данные, относящиеся к определенному отделению, могут

иметь группу этого отделения в метке. Группы полезны для контролируемого распространения данных и своевременного реагирования на организационные изменения. Группы иерархичны, то есть группа может быть связана с родительской группой. Пример различных способов задания групп приведен в табл. 3.

Таблица 3

Пример различных способов задания группы

Числовая форма	*Длинная форма*	*Краткая форма*	*Родительская группа*
1000	WESTERN_REGION	WR	
1100	WR_SALES	WR_SAL	WR
1200	WR_HUMAN_RESOURCES	WR_HR	WR
1300	WR_FINANCE	WR_FIN	WR
1310	WR_ACCOUNTS_PAYABLE	WR_AP	WR_FIN
1320	WR_ACCOUNTS_RECEIVABLE	WR_AR	WR_FIN

Группы необязательны. Метка может содержать ноль или более групп. Oracle Label Security позволяет определять до 10 000 групп.

В действительности метка безопасности реализуется добавлением специального столбца, содержащего значение, интерпретируемое как тройка $\lambda = \langle \lambda_1, \lambda_2, \lambda_3 \rangle$. Символьные строковые представления меток используют следующий синтаксис:

```
LEVEL:COMPARTMENT1,...,COMPARTMENTn:GROUP1,...,GROUPn.
```

Пример допустимых меток:

```
SENSITIVE:FINANCIAL,CHEMICAL:EASTERN_REGION,WESTERN_REGION
CONFIDENTIAL:FINANCIAL:VP_GRP
SENSITIVE
HIGHLY_SENSITIVE:FINANCIAL
SENSITIVE::WESTERN_REGION
```

С текстовой строкой, представляющей метку, связывается так называемый числовой тег метки (англ. *label tag*). Именно этот тег метки, а не текстовая строка, сохраняется в столбце метки защищаемой таблицы. В отличие от простых меток, соответствующих уровням конфиденциальности информации, составные метки имеют более сложные правила упорядочения и обеспечивают большую функциональность систем защиты информации.

Подводя итог, можно сделать следующие выводы.

Модель Белла – ЛаПадулы сыграла огромную роль в развитии теории компьютерной безопасности, и ее положения были введены в качестве обязательных требований к системам, обрабатывающим информацию, содержащую государственную тайну, в стандартах защищенных ИС. Однако при практической реализации модели Белла – ЛаПадула возникает ряд проблем, например, таких как [28]:

1) завышение уровня безопасности (для некоторой информации может быть определен уровень безопасности выше необходимого);

2) запись вслепую (например, в ситуации, когда субъект производит запись объекта с более высоким уровнем безопасности, операция не нарушает правила NWD, однако после ее завершения субъект не может проверить правильность выполнения записи объекта путем выполнения контрольного чтения, так как это нарушает правило NRU);

3) привилегированные субъекты. Эта проблема связана с работой администратора (системного, базы данных), которая подразумевает выполнение в системе таких критических операций, как добавление и удаление пользователей, восстановление системы после сбоев, аварий, установка программного обеспечения, устранение ошибок. Однако такие операции не вписываются в рамки модели, что означает невозможность осуществления правильного администрирования без нарушения правил данной модели.

Расширения модели Белла – ЛаПадулы [29, 30], связанные с поиском условий и ограничений, повышающих ее безопасность, также не снимают всех недостатков мандатного доступа. В частности, мандатный доступ отчасти снимает проблему троянских программ – толь-

ко с точки зрения опасных потоков «сверху вниз». В пределах же одного класса безопасности

вопросы доступа решаются, как и в дискреционных моделях, – на основе матрицы доступа. Следовательно, для полного устранения проблемы троянских программ в системах мандатного доступа также требуется более тщательный и детализированный контроль информационных потоков.

В целом же основным недостатком многоуровневых моделей является невозможность управления доступом к конкретным объектам на основе учета индивидуальных особенностей каждого из субъектов.

Таким образом, оба рассмотренные выше подхода не в полной мере могут эффективно и гибко управлять безопасным доступом к данным. Следовательно, оба подхода как бы предполагают поиск различных компромиссов между эффективностью, гибкостью и безопасностью. Очевидно, что оптимальное решение вопросов безопасности должно вырабатываться с применением обоих видов моделей.

Модели безопасности на основе ролевой политики

В основе рассмотренных моделей безопасности лежат отношения между отдельным субъектом и объектом доступа, определяемые либо внешним фактором (дискреционный доступ), либо уровнем безопасности (мандатный доступ).

Вместе с тем, анализ различных ИС, и в первую очередь информационных систем организационного управления, показывает, что в реальной жизни все данные системы принадлежат некоторой организации, а не конкретному пользователю (субъекту). Сотрудники этой организации выполняют определенные функциональные обязанности не от своего личного имени, а в рамках некоторой должности, которую можно трактовать как определенную роль, представляющую собой некоторую обобщенную сущность, выражающую определенный тип функций и статус сотрудника (подчиненность, права и полномочия).

Таким образом, политика разграничения доступа в таких ИС должна строиться на основе функционально-ролевых отношений, складывающихся в предметной области. Концепция управления доступом на основе ролей связана с многопользовательскими системами, впервые появившимися в 1970-х годах [31]. Несколько позже появилось формальное выражение – управление доступом на основе ролей (англ. *Role-Based Access Control* – RBAC), используемое сегодня. Основой ролевых моделей является введение в субъектно-объектную модель ИС дополнительной категории активных сущностей – ролей. Ролевая модель определяет особый тип политики, основанный на компромиссе между гибкостью управления доступом, характерной для дискреционных моделей, и жесткостью правил контроля доступа, присущей мандатным моделям. В ролевой модели классическое понятие субъект разделяется на две составляющие: пользователь и роль. *Пользователь* – это человек, работающий с системой и выполняющий определенные служебные обязанности. *Роль* – это активно действующая в системе абстрактная сущность, с которой связывается определенный набор полномочий (привилегий), необходимых для осуществления определенной деятельности.

Управление доступом при использовании ролевой политики осуществляется в два этапа:

1. Создание ролей и определение их полномочий (прав доступа к объектам).
2. Назначение ролей пользователям системы.

Следует отметить, что пользователь может быть ассоциирован с несколькими ролями. Данная возможность значительно упрощает администрирование сложных систем.

Управление доступом в ролевых системах требует разбиения процесса функционирования системы и работы пользователя на сеансы, в каждом из которых выделяются фазы [6]:

– авторизации пользователя в текущем сеансе с одной или несколькими разрешенными для него ролями;

– разрешения/запрещения доступа пользователю к объектам системы в рамках полномочий соответствующих ролей, с которыми этот пользователь авторизован в текущем сеансе.

Нетрудно видеть, что ролевые модели сочетают в себе как мандатный подход к организации доступа – через определенную агрегацию субъектов и объектов доступа (распределение ролей между сеансами и пользователями, а также полномочий между ролями) и тем самым обеспечивают жесткость правил разграничения доступа, так и дискреционный подход (связывание с ролью определенного набора полномочий на объекты системы), обеспечивающий гибкость в настройке системы разграничения доступа в ИС для конкретной предметной области.

Ролевая модель описывает систему в виде следующих множеств [31]:

– U – множество пользователей;

– R – множество ролей;

– P – совокупность полномочий на доступ к объектам (реализованная, например, в виде матрицы доступа);

– C – множество сеансов работы пользователей с системой.

Ролевые отношения устанавливаются следующими отображениями множеств сущностей системы:

– $PA \subseteq P \times R$ – отображение множества полномочий на множество ролей, задающее для каждой роли установленный набор полномочий (отношение типа «многие ко многим»);

– $UA \subseteq U \times R$ – отображение множества пользователей на множество ролей, определяющее набор ролей, доступных данному пользователю (отношение типа «многие ко многим»).

Управление доступом в системе осуществляется на основе введения следующих функций:

– $user : C \to U$ – функция, которая ставит в соответствие каждому сеансу $c_i \in C$ одного пользователя $u \in U$: $u = user(c_i)$ (не меняется в течение сеанса);

– $roles : S \to 2^R$ – функция, которая ставит в соответствие каждому сеансу c_i набор ролей из множества R, доступных в данном сеансе: $roles(c_i) = \{r \mid (user(c_i), r) \in UA\}$ (набор ролей может меняться со временем);

– $permissions : C \to P$ – функция, которая ставит в соответствие каждому сеансу c_i набор доступных в нем полномочий: $permissions(c_i) = \bigcup_{r \in roles(c_i)} \{p \mid (p, r) \in PA\}$, иначе говоря, совокупность полномочий всех ролей, доступных в данном сеансе.

Взаимосвязь пользователей, ролей, полномочий (привилегий) и сеансов показана на рис. 3.

Основное правило (критерий безопасности) ролевого доступа определяется следующим образом: система считается безопасной, если и только если любой пользователь $u \in U$ в системе, работающий в сеансе $c \in C$, может осуществлять действия, требующие полномочий $p \in P$, только в том случае, если $p \in permissions(c)$.

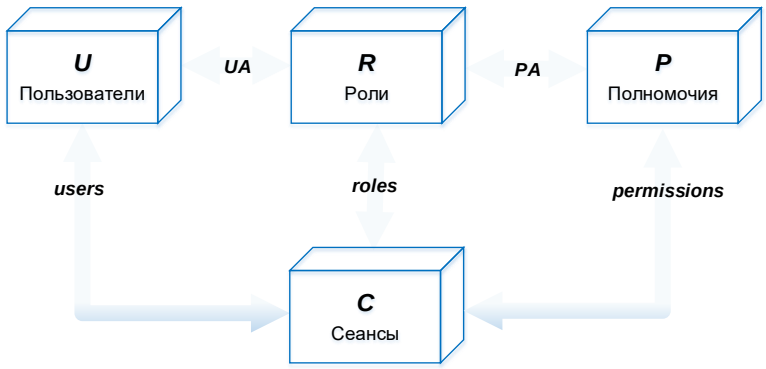

Рис. 3. Взаимосвязь ролей, полномочий, пользователей и сеансов

На практике управление доступом в ИС при использовании ролевой модели осуществляется главным образом не с помощью назначения новых полномочий ролям, а путем задания отношения *UA* (определения ролей, доступных данному пользователю) и функции *roles*, определяющей доступный в сеансе набор ролей. Поэтому многочисленные интерпретации ролевой модели [2, 4, 6, 31]:

– с иерархической организацией системы ролей ($PH \subseteq R \times R$);

– с взаимоисключающими на любые (все) сеансы ролями (модель статического распределения обязанностей);

– с взаимоисключающими на один сеанс ролями (модель динамического распределения обязанностей);

– с количественными ограничениями по ролям; с группированием ролей и полномочий, различаются видом функций *user*, *roles* и *permissions*, а также ограничениями, накладываемыми на отношения *PA* и *UA*.

Использование ролевой модели позволяет повысить эффективность администрирования сложных ИС. Поэтому данный подход является востребованным, в том числе и для СУБД.

Большое число пользователей, статус которых требует различных полномочий (привилегий) для доступа к ресурсам базы данных, создает значительный объем монотонной и утомительной работы администратору БД (АБД). В связи с этим, например, в стандарт SQL была добавлена концепция ролей как именованного набора привилегий [18]. В большинстве современных реализаций SQL роли могут быть предоставлены отдельным идентификаторам пользователей точно так же, как и отдельные привилегии. Кроме того, большинство реализаций СУБД поставляется с набором предопределенных ролей. Например, привилегии, которые обычно необходимы для работы АБД, часто предоставляются поставщиком СУБД в виде роли.

Приобретаемые преимущества от использования ролей в СУБД.

– Роли могут существовать до того, как будут определены учетные записи пользователей. Например, можно создать роль для отдела обработки заказов, вместо того, чтобы все его сотрудники совместно использовали некоторый общий идентификатор пользователя (*user-id*). Когда в отдел приходит новый сотрудник, одна инструкция GRANT с указанием роли предоставляет ему все необходимые для работы в отделе привилегии.

– Роли сохраняются, когда учетные записи пользователей удаляются (например, при увольнении). АБД, удаляя учетную запись некоторого пользователя (*user-id*), больше не беспокоится о потере прав. Роль может быть легко передана другому человеку, принятому на эту должность.

– Роли поддерживают стандартные привилегии. При применении ролей в организации легко обеспечить одинаковые привилегии для всех людей, выполняющих одинаковую работу.

– Роли устраняют рутинную работу по предоставлению привилегий отдельным пользователям. Роли позволяют предоставлять множество привилегий одной простой командой. При добавлении/удалении привилегий к/из роли все изменения немедленно отражаются на всех пользователях, которым предоставлена данная роль.

Создание и назначение привилегий при помощи ролей выполняется достаточно просто с помощью операторов CREATE, GRANT и REVOKE. Например, в стандарте SQL для создания роли используется следующий оператор:

```
CREATE ROLE name_role;.
```

Назначить роли привилегии можно путем выполнения, например, такого оператора:

```
GRANT name_ privilege1, name_ privilege2,… ON name_object TO
name_role;.
```

Предоставить роль пользователям можно так:

```
GRANT name_role TO user_1, user_2, …;.
```

Однако следует заметить, что поддержка ролей в разных реализациях SQL несколько отличается. К тому же не все СУБД поддерживают механизм ролей. Среди современных СУБД ролевая модель разграничения доступа поддерживается, например, в Microsoft SQL Server, Oracle, IBM DB2, PostgreSQL, MySQL 8.0 и некоторых других.

В заключение следует отметить, что ролевые модели позволяют реализовать гибкие, динамически изменяющиеся в процессе функционирования ИС правила разграничения доступа, эффективность которых особенно заметно проявляется при организации доступа к ресурсам сложных информационных систем с большим количеством пользователей и объектов, в том числе благодаря возможности построения иерархий ролей. Оперировать ролями гораздо удобнее, чем субъектами, поскольку это более соответствует распространенным технологиям обработки информации, предусматривающим разделение обязанностей и сфер ответственности между пользователями.

Вместе с тем, в ролевых моделях нет строгих доказательств безопасности системы в соответствии с определенными формализованными критериями. Такой подход позволяет получать простые и понятные правила контроля доступа, которые легко могут быть применены на практике, но лишает систему теоретической доказательной базы. В отличие от других рассмотренных ранее моделей ролевая модель практически не гарантирует безопасность с помощью формального доказательства, а только определяет характер ограничений, соблюдение которых и служит критерием безопасности системы. Поэтому безопасность ролевых моделей основывается на контрольных механизмах дискреционных или мандатных моделей, средствами которых регулируется доступ ролевых субъектов к объектам системы. К недостаткам ролевого разграничения доступа также следует отнести: возможность внесения избыточности (дублирования) при предоставлении пользователям прав доступа, сложность конструирования ролей.

Анализируя разнообразие формальных моделей и множество подходов к их реализации достаточно сложно определиться, какая из них более предпочтительна. И это естественно, так как каждая из них имеет свои преимущества, которые необходимо использовать именно в конкретной ситуации, точно так же, как свести при этом к минимуму недостатки каждой из них. При этом следует понимать, что сама модель безопасности не обеспечивает защиту, а только предоставляет принцип построения защищенной ИС, реализация которого и должна обеспечить заложенные в модели свойства безопасности. Безопасность системы в равной степени определяется тремя факторами [4]: свойствами самой модели (одной или нескольких), ее (их) адекватностью угрозам, воздействующим на систему, и тем насколько она (они) корректно реализована(ы).

В сложившейся ситуации, принимая во внимание научно-практические достижения в области информационной безопасности, современное состояние развития информационных систем, квалификацию злоумышленников, положения и рекомендации различных нормативно-правовых актов, целесообразным представляется проведение дальнейших исследований, результатом которых являлась бы некоторая методология, учитывающая, в том числе, возможности комплексного использования рассмотренных выше моделей, для обоснования, построения и оценки безопасности проектируемых и эксплуатируемых защищенных ИС.

Выводы

1. Проведенный анализ формальных моделей управления доступом позволил выявить их основные достоинства и слабые стороны, которые в зависимости от конкретной ситуации, целесообразно разумно использовать, в том числе и комплексно. При этом следует исходить из того, что сама модель безопасности не обеспечивает защиту, а только предоставляет принцип построения защищенной ИС, БД реализация которого должна обеспечить заложенные в модели свойства безопасности. Безопасность системы в равной степени определяется: свойствами самой модели (одной или нескольких), ее (их) адекватностью угрозам, воздействующим на систему, и тем насколько она (они) корректно реализована(ы).

2. К достоинствам моделей безопасности, построенных на основе дискреционной политики управления доступом, следует отнести ее относительную простоту и гибкость. Модели безопасности на основе дискреционной политики целесообразно применять при проведении формальной верификации корректности построения систем разграничения доступа в хорошо защищенных информационных системах и базах данных. Однако следует учитывать, что этим моделям свойственны недостатки, ограничивающие их применение, а именно: статичность установленных правил управления; невозможность контролировать в полной мере потоки данных между объектами; сложность отслеживания предоставляемых субъектам привилегий при их большом количестве и неконтролируемой передачи (распространении прав) от одного субъекта к другому; отсутствие механизма управления доступом к конфиденциальной информации.

3. Важным выгодным отличием мандатного управления доступа от дискреционного является то, что в мандатной модели контролируются не операции, выполняемые субъектом над объектом, а потоки информации, которые могут быть только двух видов: либо от субъекта к объекту (запись), либо от объекта к субъекту (чтение). Основной целью мандатного разграничения доступа к объектам является предотвращение утечки информации из объектов с высоким уровнем безопасности к субъекту с низким уровнем безопасности (противодействие созданию каналов передачи информации «сверху вниз»). Однако, несмотря на то, что модели безопасности на основе мандатной политики доступа играют значимую роль в теории информационной безопасности и их положения введены в качестве обязательных требований к системам, обрабатывающим информацию, содержащую государственную тайну, а также в стандартах защищенных ИС, при практической реализации этих моделей может возникнуть ряд проблем, таких как: завышение уровня безопасности; запись вслепую; проблема привилегированных субъектов, выполняющих операции, не вписывающиеся в рамки модели (это означает невозможность осуществления правильного администрирования без нарушения правил данной модели). К тому же модель мандатного управления доступом, решая проблему троянских программ, снимает ее только с точки зрения опасных потоков «сверху вниз», но не в пределах одного класса безопасности, когда ее приходится решать аналогично дискреционным моделям, на основе матрицы доступа. Следовательно, для полного устранения проблемы троянских программ в системах мандатного доступа требуется более тщательный и детализированный контроль информационных потоков. В целом же основным недостатком многоуровневых моделей является невозможность управления доступом к конкретным объектам на основе учета индивидуальных особенностей каждого из субъектов.

В отношении различных современных СУБД следует заметить, что сегодня они имеют отличающиеся реализации технологии мандатного доступа.

4. Модели безопасности на основе ролевой политики позволяют реализовать гибкие, динамически изменяющиеся в процессе функционирования информационных систем, баз данных, правила разграничения доступа, эффективность которых особенно заметно проявляется при организации доступа к ресурсам систем с большим количеством пользователей и объектов, в том числе благодаря возможности построения иерархий ролей. Вместе с тем, в ролевых моделях нет строгих доказательств безопасности системы в соответствии с определенными формализованными критериями. Такой подход позволяет получать простые и понятные правила контроля доступа, которые достаточно легко можно применить на практике, но лишает систему теоретической доказательной базы. К недостаткам ролевого разграничения доступа также следует отнести возможность внесения избыточности (дублирования) при предоставлении пользователям прав доступа, сложность конструирования ролей.

Список литературы:

1. Tanenbaum A. S., Herbert Bos H. Modern Operating Systems. Fourth edition. Pearson, 2015. 1136 p.
2. Смирнов С. Н. Безопасность систем баз данных. Москва : Гелиос АРВ, 2007. 352 с.
3. Cunha M. M., Oliveira E. F., Tavares A. J., Ferreira L. G. Handbook of Research on Social Dimensions of Semantic Technologies and Web Services. Hershey, PA: IGI Global, 2009. 1180 p.

4. Зегжда Д. П., Ивашко А. М. Основы безопасности информационных систем. Москва : Горячая линия – Телеком, 2000. 452с.
5. Хоффман, Л. Дж. Современные методы защиты информации. Москва : Сов. радио, 1980. 264 с.
6. Гайдамакин Н. А. Теоретические основы компьютерной безопасности. Екатеринбург : Изд-во Уральск. ун-та, 2008. 212 с.
7. Weissman C. Security controls in the ADEPT-50 time-sharing system // Proceedings of the November 18-20, 1969, fall joint computer conference. 1969. P. 119-133.
8. Hartson H. R., Hsiao D. K. A Semantic Model for Database Protection Languages. Systems for Large Data Bases. North-Holland, Amsterdam : Publishing Co., 1976. P. 27-42.
9. Harrison M. A., Ruzzo W. L., Ullman J. D. Protection in Operating Systems // Communications of the ACM, 1976. № 19(8). P. 461–471.
10. Lipton R. J., Snyder L. A linear time algorithm for deciding subject security // Journal of the ACM (JACM), 1977. № 24(3). P. 455-464.
11. Цирлов В. Л. Основы информационной безопасности автоматизированных систем. Ростов-на-Дону : Феникс, 2008. 173 с.
12. Sandhu R. S. The Typed Access Matrix Model // Proceedings of IEEE Symposium on Security and Privacy, Oakland, California, May 4-6, 1992, P. 122-136.
13. Девянин П. Н. Модели безопасности компьютерных систем. Управление доступом и информационными потоками. 2-е изд. Москва : Горячая линия – Телеком, 2013. 338 с.
14. Скакун В. В. Защита информации в базах данных и экспертных системах. Минск : БГУ, 2015. 140 с.
15. Frank J., Bishop M. Extending the take-grant protection system // Technical Report, Department of Computer Science, University of California at Davis, 1996. 14 p. URL: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.51.907&rep=rep1&type=pdf. (accessed on: 04.02.2021).
16. Garcia-Molina H., Ullman J. D., Widom J. Database Systems. The Complete Book, 2th ed. Pearson Prentice Hall, 2009. 1203 p.
17. ISO/IEC 9075-2:2016 Information technology. Database languages. SQL. Part 2: Foundation (SQL/Foundation). URL: https://www.iso.org/standard/63556.html. (accessed on: 04.02.2021).
18. Грофф Д. Р., Вайнберг П. Н., Оппель Э. Д. SQL : полное руководство, 3-е изд. ; пер. с англ. Москва : Вильямс, 2015. 960 с.
19. Марков А. С., Цирлов В. Л., Барабанов А. В. Методы оценки несоответствия средств защиты информации. Москва : Радио и связь, 2012. 192 с.
20. Bell D. E., LaPadula L. J. Secure Computer Systems: Unified Exposition and Multics Interpretation (MTR-2997 Rev. 1). Bedford, Mass.: MITRE Corp., 1976. 129 p.
21. Date C. J. An Introduction to Database Systems, 8th ed. New York, USA : Pearson Education, Inc., 2004. 983 p.
22. Patent US 2004/0044655A1, United States, Row-level security in a relational database management system / Curt Cotner, Gilroy, CA (US); Roger Lee Miller, San Jose, CA (US); International Business Machines Corporation, Armonk, NY (US). N 10/233,397; Mar. 4, 2004.
23. Patent 8,131,664 B2, United States, Row-level security in a relational database management system / Curt Cotner, Gilroy, CA (US); Roger Lee Miller, San Jose, CA (US); International Business Machines Corporation, Armonk, NY (US). N 12/242,241; Mar. 6, 2012.
24. Patent 8.478,713 B2, United States, Row-level security in a relational database management system / Curt Cotner, Gilroy, CA (US); Roger Lee Miller, San Jose, CA (US); International Business Machines Corporation, Armonk, NY (US). N 15/343,568; Jan. 16, 2018.
25. Кайт Т. Oracle для профессионалов ; пер. с англ. СПб. : ООО «ДиаСофтЮП», 2003. 672 с.
26. Нанда А., Фейерштейн С. Oracle PL/SQL для администраторов баз данных ; пер. с англ. СПб : Символ-Плюс, 2008. 496 с.
27. Oracle Database 19c. Administrator's Guide. Understanding Data Labels and User Labels. URL: https://docs.oracle.com/en/database/oracle/oracle-database/19/olsag/understanding-data-labels-and-user-labels.html#GUID-2C0383D3-4AA5-4263-B938-827E2CCC40C0 (accessed on: 04.02.2021).
28. Щербаков А. Ю. Современная компьютерная безопасность. Теоретические основы. Практические аспекты. Москва : Книжный мир, 2009. 352 с.
29. McLean J. The specification and modeling of computer security // Computer. 1990. № 23(1). P. 9-16.
30. McLean J. Security models. Encyclopedia of software engineering, Vol. 2. Wiley, 1994. P. 1136-1145.
31. Sandhu R.S., Coyne E. J., Feinstein H. L., Youman C. E. Role-based access control models // IEEE Computer. 1996. № 2. P. 38-47.

Поступила в редколлегию 15.03.2021

Сведения об авторе:
Вилигура Владислав Викторович – Харьковский национальный университет имени В.Н. Каразина, аспирант кафедры безопасности информационных систем и технологий, факультета компьютерных наук; Украина, e-mail: viligura93@gmail.com; ORCID: https://orcid.org/0000-0002-1137-2382