УДК 004.056 **DOI:10.30837/rt.2021.1.204.04**

В.И. ЕСИН, д-р техн. наук, С.Г. РАССОМАХИН, д-р техн. наук, В.В. ВИЛИГУРА

АНАЛИЗ ФОРМАЛЬНЫХ МОДЕЛЕЙ ОБЕСПЕЧЕНИЯ ЦЕЛОСТНОСТИ ДАННЫХ И ИХ ПРИМЕНИМОСТЬ ДЛЯ БАЗ ДАННЫХ

Введение

Концепции и принципы управления безопасностью, определяющие основные параметры, необходимые для безопасной среды, цели и задачи, которых должны достичь как разработчики политик, так и разработчики систем, чтобы создать безопасное решение, являются неотъемлемыми элементами политики безопасности, формальное представление (в виде математических выражений, схем, диаграмм, алгоритмов и т. д.) которой называют моделью безопасности. Модели безопасности играют важную роль в процессах разработки и исследования защищенных компьютерных систем, так как обеспечивают системотехнический подход. Рассмотрение моделей безопасности целесообразно по нескольким причинам. Во-первых, они могут быть непосредственно использованы для анализа безопасности как существующих, так и перспективных информационных систем (ИС) и их основного функционального компонента – базы данных (БД), особенно в случаях, когда требуется получение гарантий защищенности ИС. Классические модели безопасности ИС позволяют формально анализировать свойства различных механизмов защиты ИС. Во-вторых, существующие модели безопасности могут быть использованы в качестве основы для разработки более совершенных моделей, позволяющих более точно описывать и исследовать особенности функционирования механизмов защиты современных ИС. В-третьих, владение знаниями о моделях безопасности ИС предоставляет специалисту в области компьютерной безопасности возможности для строгого научного и теоретически обоснованного изложения результатов прикладных исследований [1].

Среди формальных моделей безопасности в данной работе рассмотрим модели обеспечения целостности данных и особенности их применения для баз данных.

Модели обеспечения целостности данных

Обеспечение информационной безопасности невозможно без рассмотрения концепции защиты надежности/достоверности (англ. reliability) и правильности/корректности (англ. correctness) данных, представляющей суть обеспечения их целостности. Для многих, особенно невоенных организаций, целостность важнее конфиденциальности. Трудно представить систему, для которой были бы не важны свойства целостности. Например, если вы публикуете информацию в Интернете на Web-сервере и вашей целью является сделать ее доступной для самого широкого круга людей, то конфиденциальность в данном случае не требуется. Однако требования целостности остаются актуальными. Многочисленные атаки направлены на нарушение целостности. К ним можно отнести вредоносные модификации, выполняемые вирусами или другими зловредными программами, ошибки в приложениях. При этом нарушения целостности не ограничиваются преднамеренными атаками. Ошибка пользователя, недосмотр или неумелость являются причиной многих случаев несанкционированного изменения информации. События, которые приводят к нарушениям целостности, включают изменение или удаление файлов, данных в БД, ввод неверных данных, изменение конфигурации, ошибки в командах, внедрение вируса и выполнение вредоносного кода. Нарушение целостности может произойти из-за действий любого пользователя, включая администраторов. Они также могут возникать из-за недосмотра в политике безопасности или из-за неправильно настроенного контроля безопасности.

Целостность целесообразно рассматривать с трех сторон [2]:

– предотвращения (препятствия) внесения изменений неавторизованными субъектами;

– предотвращения внесения авторизованными субъектами несанкционированных изменений, например, ошибок;

– поддержания внутренней и внешней согласованности объектов, чтобы их данные были правильным и истинным отражением реального мира, а любые отношения (связи) с любым дочерним, равным или родительским объектом были действительными (англ. valid), согласованными (англ. consistent) и проверяемыми (англ. verifiable).

Правильно реализованная защита целостности предоставляет средства для авторизованных изменений, одновременно защищая от злонамеренных несанкционированных действий (таких как вирусы и вторжения), а также от ошибок, допущенных авторизованными пользователями (таких как ошибки или недосмотры/оплошности). Это гарантирует, что данные остаются *правильными* (отсутствуют логические ошибки в структуре и в значениях данных), *неизменными* (тождественность данных определенному эталону), *неискаженными* (отсутствие подделки данных) и *сохраненными*. Если механизм безопасности обеспечивает целостность, он обеспечивает высокий уровень гарантии того, что данные, объекты и ресурсы не будут изменены по сравнению с их первоначальным защищенным состоянием.

В зависимости от того, насколько тот или иной аспект области использования данных является наиболее важным, выделяют методы и средства, обеспечивающие их целостность, в смысле [3]:

– правильности, неискаженности и неизменности данных, основывающиеся на так называемых моделях целостности данных;

– неискаженности данных при передаче в линиях связи и хранении в информационных системах, основывающиеся на криптографии (например, использовании таких криптографических примитивов как: цифровая подпись, криптографические хеш-функции, коды проверки подлинности);

– параллельного выполнения транзакций в клиент-серверных системах (транзакции играют важную роль в механизме обеспечения целостности базы данных).

Существуют многочисленные контрмеры, которые могут гарантировать целостность данных при различных возможных угрозах [2]. В том числе обеспечить безопасность легче, если есть четкая модель того, что нужно защищать и кому и что разрешено делать [4]. Поэтому неотъемлемой частью любого проекта по созданию или оценке безопасности ИС и баз данных в том числе, как отмечается в [5], является наличие модели безопасности. Ниже, в первую очередь, остановимся на анализе некоторых наиболее известных моделях безопасности, связанных с аспектами, рассматриваемыми в работе, – формальных моделях целостности данных.

Модель Кларка – Вилсона

Исходя из важности обеспечения целостности данных было разработано несколько моделей безопасности, к числу которых можно отнести модели, предложенные Кларком с Вилсоном и Бибом.

Модель Кларка – Вилсона [6] является описательной. В ней не содержится каких бы то ни было строгих математических выражений. Модель Кларка – Вилсона – это основа и руководство для формализации политик безопасности, а не модель конкретной политики безопасности. В ней подчеркивается важность утверждения руководством процессов и политик безопасности, которым должна следовать организация [7]. Ее, скорее всего, целесообразно рассматривать как совокупность практических рекомендаций по построению системы обеспечения целостности в ИС.

Для лучшего понимания данной модели проведем некоторую формализацию, введя определенные обозначения:

– S – множество субъектов;

– D – множество данных в ИС (множество объектов), причем $D = CDI \bigcup UDI$, $CDI \bigcap UDI = \varnothing$ где CDI (*constrained data items* – «*ограниченный элемент данных*») – данные

(любой элемент данных), целостность которых контролируется (защищена моделью безопасности); *UDI* (*unconstrained data items* – «*неограниченный элемент данных*») – данные, целостность которых не контролируется моделью безопасности;

– *IVP* (*integrity verification procedure*) – процедура проверки целостности *CDI* (процедура, которая сканирует элементы данных и подтверждает их целостность, например путем расчета контрольной суммы или используя возможности современной блокчейновой модели, как это показано в работе [8]);

– *TP* (*transformation procedure*) – процедура преобразования – компонент, который может инициировать транзакцию (последовательность операций), переводящую систему из одного состояния в другое. Процедуры преобразования единственные процедуры, которым разрешено изменять *CDI*. Ограниченный доступ к *CDI* через *TP* составляет основу модели целостности Кларка – Вилсона.

Модель Кларка – Вилсона основывается, как и дискреционные модели разграничения доступа, на тройках: «*субъект – операция (транзакция), не нарушающая целостность – объект*». Субъекты не имеют прямого доступа к объектам. Доступ к объектам можно получить только через *TP*.

В модели выделяются два основных механизма, обеспечивающих базовый контроль доступа и целостность. А именно – правильно сформированная транзакция сохраняет целостность данных и предотвращает произвольное манипулирование данными субъектами. Следует заметить, что концепция правильно сформированной транзакции отлично вписывается в стандартную концепцию транзакций в традиционных СУБД [9]. Разделение обязанностей требует, чтобы каждая критическая операция состояла из двух или более частей, каждая из которых должна выполняться другим субъектом или субъектом с другой ролью.

Модель состоит из двух наборов правил: сертификации (С), которая проводится сотрудником по вопросам безопасности (администратором безопасности), владельцем системы, хранителем системы (англ. system custodian), и правил исполнения (Е), которое осуществляется системой. Правила исполнения соответствуют функциям безопасности, независимым от приложений, а правила сертификации позволяют включать в модель определения целостности для конкретных приложений. Желательно минимизировать правила сертификации, поскольку процесс сертификации сложен, подвержен ошибкам и должен повторяться после каждого изменения процедура преобразования (программы).

Несколько перефразированные относительно оригинала правила модели Кларка – Вилсона приведены ниже:

1. Правило С1. В системе должны иметься *IVP*, способные подтвердить целостность любого *CDI* (в оригинальной работе [6] оно формулируется таким образом: «Все *IVP* должны надлежащим образом гарантировать, что все *CDI* находятся в валидном состоянии на момент работы *IVP*»; под понятием «валидное (valid) состояние» авторы понимают такое состояние системы, при котором в любой момент времени *CDI* удовлетворяют требованиям целостности).

2. (С2) Все процедуры преобразования *TP* должны быть реализованы корректно, в том смысле, что не должны нарушать целостности данных (то есть они должны перевести *CDI* в допустимое конечное состояние, учитывая, что он находится в допустимом состоянии с самого начала), и применяться только по отношению к списку элементов *CDI*, устанавливаемых администратором безопасности (отношение $(TP_i,(CDI_a,CDI_b,CDI_c,...))$).

3. (Е1) Система должна контролировать допустимость применения *TP* к элементам *CDI* в соответствии со списками, указанными в правиле С2.

4. (Е2) Система должна поддерживать список разрешенных конкретным пользователям процедур преобразования *TP* с указанием допустимого для каждой $TP_i \in TP$ и данного субъекта ($s_j \in S$) набора обрабатываемых элементов *CDI* (то есть тройки: $(s_j,TP_i,(CDI_a,CDI_b,CDI_c,...))$.

5. (C3) Список, определенный правилом E2, должен отвечать требованию разграничения функциональных обязанностей (в том числе совместного выполнения).

6. (E3) Система должна аутентифицировать всех пользователей (каждый субъект), пытающихся выполнить какую-либо процедуру преобразования TP.

7. (C4) Каждое применение TP должно регистрироваться в специальном элементе CDI – журнале регистрации, содержащем информацию, достаточную для восстановления полной картины каждого применения этой процедуры преобразования, и доступном только для добавления в него информации.

8. (C5) Любая TP, которая принимает UDI в качестве входных данных, может выполнять только допустимые преобразования для любого возможного значения UDI. TP либо принимает (конвертирует в CDI), либо отклоняет UDI. То есть специальные TP могут корректно обрабатывать UDI, превращая их в CDI.

9. (E4) Только специально уполномоченный субъект (пользователь, агент, которому разрешено сертифицировать объекты) может изменять списки, определенные в правилах C3 и E2. Этот субъект не имеет права выполнять какие-либо действия, если он уполномочен изменять регламентирующие эти действия списки.

Роль каждого из девяти правил модели Кларка – Вилсона в обеспечении целостности данных в работе [10] соотносится с так называемыми теоретическими принципами политики контроля целостности:

1) корректность транзакций;
2) аутентификация пользователей;
3) минимизация привилегий;
4) разграничение функциональных обязанностей;
5) аудит произошедших событий;
6) объективный контроль;
7) управление передачей привилегий;
8) обеспечение непрерывной работоспособности;
9) простота использования защитных механизмов.

Соответствие правил модели Кларка-Вилсона первым шести перечисленным выше принципам показано в табл. 1.

Таблица 1

Правило модели Кларка – Вилсона	Принципы политики контроля целостности
C1	1,6
C2	1
C3	4
C4	5
C5	1
E1	3,4
E2	1,2,3,4
E3	2
E4	4

Как видно из табл. 1, принципы политики контроля целостности 1 (корректность транзакций) и 4 (разграничение функциональных обязанностей) реализуются большинством правил модели Кларка – Вилсона, что соответствует ее основной идее.

На рис. 1 представлена схема применения данных правил для управления работой системы и данными. UDI представляют данные, существующие вне защищенной системы. Правила сертификации обеспечивают правильную проверку таких данных при входе в систему. Например, правило C5 требует, чтобы правильно сформированные TP, которые преобразуют UDI в CDI, выполняли только проверенные преобразования. Правила C1 и C2 требуют, чтобы CDI удовлетворяли требованиям целостности в начальном состоянии и после последующих преобразований. Правило C4 требует регистрации всех транзакций, как это обычно это бывает с базами данных. Ведение журнала базы данных в большей мере предназначено

для восстановления данных после сбоя, отказа (для отката – возврата к предыдущему состоянию), а ведение журнала в модели Кларка – Вилсона – для аудита. Хотя в базах данных может вестись и журнал аудита. Правило С3 требует соответствующего разделения обязанностей. Поскольку данные могут быть введены только в соответствии с правилами сертификации, для систем, которые нас интересуют, следует, что все данные в базе данных должны быть *CDI*.

Правила исполнения предотвращают изменение *CDI* способами, противоречащими *IVP*. Правила Е2 – Е4 относятся к авторизации доступа *TP*. В то время как Е1 гарантируют, что только правильно сформированные сертифицированные (проверенные) *TP* могут использоваться для изменения *CDI*.

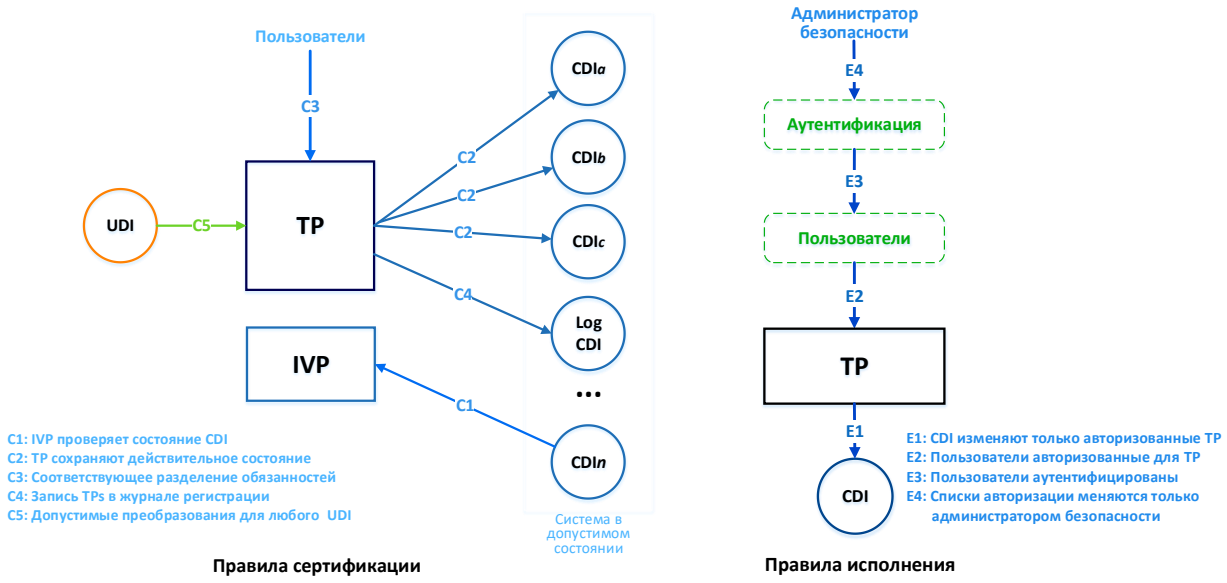

Рис. 1. Схема применения правил модели Кларка – Вилсона

Основной недостаток, обычно упоминаемый для модели Кларка – Вилсона, заключается в том, что *IVP* и связанные с ними методы непросто реализовать в реальных компьютерных системах [11]. Например, основной проблемой реализации механизмов контроля целостности файловых объектов является их достаточно сильное влияние на загрузку вычислительного ресурса системы, что обусловливается следующими причинами [12]: во-первых, может потребоваться контроль больших объемов информации, что связано со значительной продолжительностью выполнения процедуры *IVP*; во-вторых, может потребоваться непрерывное поддержание файлового объекта в эталонном состоянии. В связи с этим возникает вопрос: какой должна быть периодичность запуска процедуры *IVP*? Если выполнять ее часто, то это приведет к существенному снижению производительности системы, если редко, то эффективность такого контроля может оказаться низкой. Поэтому одной из основных задач при реализации механизмов контроля целостности файловых объектов является выбор принципов и механизмов запуска процедуры проверки целостности *CDI*. Другая проблема реализации механизма контроля целостности – это контроль целостности самой контролирующей программы, если контроль целостности реализуется программно. Все это требует определенной дополнительной проработки и принятия соответствующих решений, зависящих, как правило, от особенностей конкретных ИС.

Однако в контексте СУБД отмеченный выше общий недостаток модели Кларка – Вилсона, обусловленный сложностью реализации *IVP* и связанных с ними методов в значительной степени можно преодолеть. Так, например, для реляционных СУБД некоторые ограничения целостности заложены в теории: целостность сущностей, ссылочная целостность. Другие могут быть указаны как статические ограничения с помощью SQL (так называемая декларативная поддержка ограничений целостности). Третьи – как динамические ограничения целостности (так называемая процедурная поддержка ограничений целостности), которые

могут быть реализованы с помощью триггеров и хранимых программ. Все они обеспечивают целостность CDI, к которым осуществляется доступ и их модификация с помощью процедур преобразования TP.

Таким образом, традиционные СУБД поддерживают многие механизмы модели Кларка – Вилсона. Однако реализации, основанные на стандартном SQL, требуют некоторых компромиссов. Например, популярный принцип распространения (предоставления) прав доступа WITH GRANT OPTION (получателю передаваемых привилегий дается привилегия на дальнейшую передачу полученных привилегий, включая привилегию на передачу привилегий) противоречит модели Кларка – Вилсона (правилу E4). Актуальными для СУБД также остаются вопросы, связанные с механизмами контроля целостности хранимых процедур, функций (как файловых объектов). Это обусловливает необходимость дополнительных исследований в соответствующих направлениях.

В целом же, безусловными достоинствами этой модели являются ее относительная простота и легкость совместного использования с другими моделями безопасности.

Модель Биба

Модель Биба [13] была разработана после модели Белла – ЛаПадулы [14]. С точки зрения содержания и формального (математического) представления, эта модель является инверсией мандатной модели Белла – ЛаПадулы, проблема которой заключается в том, что она разработана для сохранения конфиденциальности, не гарантируя при этом целостность данных.

Основные элементы модели Биба:

- S – множество субъектов;
- O – множество объектов, причем $S \bigcap O = \varnothing$;
- $\Lambda_{LI} = (LI, \leq, \square, \otimes)$ – решетка уровней целостности, например: $LI = \{important, very\ important, crucial\}$, где *important*<*very important*<*crucial*;
- $RI = \{modify, invoke, observe, execute\}$ – множество видов доступа, где *modify* – доступ субъекта на модификацию объекта (аналог доступа *write* в модели Белла – ЛаПадулы), *invoke* – доступ на обращение субъекта к субъекту (например, программное средство для доступа к объекту); *observe* – доступ субъекта к объекту на чтение (аналог доступа *read* в модели
Белла – ЛаПадулы), *execute* – доступ на выполнение;
- $B = \{b \subseteq S \times O \times RI\}$ – множество возможных множеств текущих доступов в системе;
- $(i_S, i_O, i_C) \in I = LI^S \times LI^O \times LI^S$ – тройка функций (i_S, i_O, i_C), задающих: $i_S : S \to LI$ – уровень целостности субъектов; $i_O : O \to LI$ – уровень целостности объектов; $i_C : S \to LI$ – текущий уровень целостности субъектов, при этом для каждого $s \in S$ выполняется условие $i_C(s) \leq i_S(s)$;
- $V = B \times I$ – множество состояний системы.

Основные свойства или аксиомы модели Биба (в соответствии с политикой строгой целостности) можно сформулировать следующим образом:

1. *Простое свойство целостности* (*The simple integrity property*). Субъект с уровнем целостности $i_S(s)$ может читать (наблюдать – *observe*) информацию, содержащуюся в объекте с уровнем целостности $i_O(o)$ тогда и только тогда, когда уровень целостности объекта $i_O(o)$ преобладает над уровнем целостности субъекта $i_S(s)$ $(i_S(s) \leq i_O(o))$; другими словами, субъект не может прочитать объект на более низком уровне целостности (так называемое правило *no read-down* (NRD)).

2. *Свойство целостности* * (*The * integrity property*). Субъект с уровнем целостности $i_S(s)$ может изменять (*modify*) информацию, содержащуюся в объекте с уровнем целостности

$i_o(o)$, тогда и только тогда, когда уровень целостности субъекта $i_s(s)$ преобладает над уровнем целостности объекта $i_o(o)$ ($i_o(o) \le i_s(s)$); другими словами, субъект не может изменять объект на более высоком уровне целостности (так называемое правило *no write-up* (NWU)).

3. *Свойство вызова* (*invoke*) указывает на то, что субъектам разрешено вызывать субъекты только равного или более низкого уровня, то есть для $\forall\, s[1], s[2] \in S$, $s[1]$ может вызвать $s[2]$ только тогда, когда $i_s(s[2]) \le i_s(s[1])$.

Два первых свойства этой модели есть инверсия двух соответствующих свойств модели Белла – ЛаПадулы. А именно – правило NRD является полной противоположностью правила NRU модели Белла – ЛаПадулы, за исключением того, что в модели Биба используются уровни целостности, а не уровни безопасности (конфиденциальности), как в модели Белла – ЛаПадулы. Правило NWU мандатной модели целостности Биба является полной противоположностью правилу NWD модели Белла – ЛаПадулы для случая уровней целостности, а не безопасности.

Диаграмму информационных потоков, соответствующую модели Биба в системе с двумя уровнями целостности, можно представить следующим образом (рис. 2).

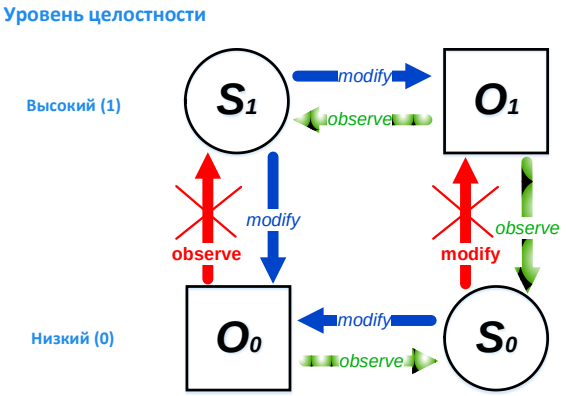

Рис. 2. Диаграмма информационных потоков в системе с двумя уровнями целостности

Многие критикуют модель Биба за то, что она использует целостность как некую меру, ставя под сомнение правомочность отображения свойства данных «целостность» дискретно-упорядоченным множеством. Действительно, в большинстве приложений целостность данных рассматривается как некое свойство (двоичный атрибут), которое либо сохраняется, либо не сохраняется. Тогда введение иерархических уровней целостности может представляться излишним. Однако если уровни целостности в модели Биба рассматривать как уровни достоверности/правильности (различные синтаксические, семантические ошибки могут по-разному влиять на правильность программного кода, вызывая, например, ошибки (errors) или предупреждения (warnings) при трансляции), а соответствующие информационные потоки – как передачу информации из более достоверной совокупности данных в менее достоверную и наоборот, то модель Биба является совершенно адекватной алгебраической структурой.

Поскольку формальное описание модели Биба очень близко с описанием модели Белла – ЛаПадулы, то она, естественно, обладает большинством достоинств и недостатков присущих этой модели.

В реальных ИС редко встречаются системы защиты, ориентированные исключительно на обеспечение конфиденциальности или исключительно на обеспечение целостности информации. Строя защищенные системы, многие хотели бы сочетать оба механизма, используя для этого различные формальные модели безопасности, в том числе такие, как модели Белла – ЛаПадулы и Биба. Это непростая задача. Возможные варианты совместного использования моделей Белла – ЛаПадулы и Биба и возникающие при этом осложнения приведены ниже [3, 15, 16]:

1. Две модели могут быть реализованы в системе независимо друг от друга. В этом случае субъектам S и объектам O независимо присваиваются уровни конфиденциальности и

уровни целостности на основе двух различных решеток. Решение о безопасности доступа принимается одновременно по правилам обеих моделей.

Нетрудно видеть, что при таком подходе к организации доступа возможны неразрешимые ситуации, например, когда по правилам модели Белла – ЛаПадулы доступ может быть разрешен, а по правилам модели Биба – нет, или наоборот.

2. Логическое объединение моделей на основе одной общей решетки уровней безопасности (конфиденциальности/целостности).

В таких системах разрешенными являются только доступы субъектов к объектам в пределах одного уровня безопасности (рис. 3).

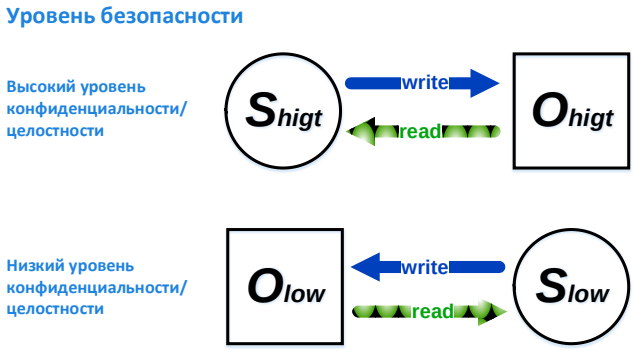

Рис. 3. Совместное использование моделей Белла – ЛаПадулы и Биба
(доступ в пределах одного уровня безопасности)

3. Логическое объединение моделей на основе одной общей решетки, но с двумя метками безопасности: по конфиденциальности и по целостности с противоположным характером их определения. Субъекты и объекты с высокими требованиями конфиденциальности (например, секретные данные и доверенные по секретам пользователи) располагаются на высоких уровнях иерархии решетки. Субъекты и объекты с высокими требованиями целостности (например, системное программное обеспечение и программисты) располагаются на нижних уровнях иерархии решетки (рис. 4).

Рис. 4. Совместное использование моделей Белла – ЛаПадулы и Биба
(на основе единой решетки с двумя метками безопасности)

Несмотря на сложность классификации субъектов и объектов доступа, именно третий вариант находит применение в современных ИС, в частности в СУБД, где реализуется мандатная политика безопасности [3].

Поскольку субъекты и объекты с высокой целостностью находятся внизу иерархии, а компоненты с низкой целостностью – наверху иерархии, то правила *no read up* и *no write down* имитируют мандатную модель целостности Биба в структуре модели Белла – ЛаПадулы. То есть чтение сверху в иерархии модели Белла – ЛаПадулы является чтением снизу в иерархии модели Биба. Аналогично, запись наверх в модели Белла – ЛаПадулы является

записью вниз в модели Биба. На практике это позволяет за счет размещения системных файлов (объектов O), в том числе относящихся к СУБД, и субъектов-администраторов (их процессов) в нижней части иерархии модели Белла – ЛаПадулы, обеспечить защиту целостности таких объектов от обычных субъектов-пользователей (и их процессов), поскольку правило *no write down* не позволяет им осуществлять запись в системные файлы. Кроме этого, если рассматривать исполнение как чтение, то субъекты-администраторы (и их процессы) не смогут исполнять программы вне высшего уровня целостности (или нижнего уровня иерархии модели Белла – ЛаПадулы).

Данная схема обеспечивает защиту системных файлов от вредоносных программ типа «троянский конь» ввиду того, что, если подобная зловредная программа находится на одном из верхних уровней, она никогда не сможет исказить системные файлы из-за необходимости выполнения правила *no write down*. Таким образом, такое объединение моделей осуществляет защиту секретности для верхних уровней определенной иерархии и защиту целостности для нижних уровней [16].

В заключение целесообразно отметить, что существующие теоретические разработки и практические реализации обеспечения безопасности ИС основываются не только на парадигме формального моделирования политики безопасности, но и на другой не менее важной парадигме – криптографии, нацеленной на решение определенных задач. Причем эти различные по происхождению и решаемым задачам подходы дополняют друг друга: криптография предлагает актуальные методы и примитивы для защиты информации, обеспечивая идентификацию, аутентификацию, шифрование, контроль целостности данных, а формальные модели безопасности предоставляют разработчикам защищенных ИС основополагающие принципы, которые лежат в основе архитектуры защищенной системы и определяют концепцию ее построения [17].

Целесообразным представляется проведение дальнейших исследований, результатом которых являлась бы некоторая методология комплексного использования различных моделей безопасности при проектировании и эксплуатации соответствующих ИС и их основного функционального компонента – БД, ведущая к повышению эффективности их защиты.

Выводы

1. Анализ формальных моделей обеспечения целостности данных выявил, что каждая из них, имея определенные преимущества и недостатки, имеет право на использование. Главным решающим фактором является оценка конкретной ситуации, которая позволит сделать правильный выбор, в том числе и комплексного их применения.

2. Модель Кларка – Вилсона является описательной, не содержащей строгих математических выражений. Данная модель – это основа и руководство для формализации политик безопасности, а не модель конкретной политики безопасности. Ее целесообразно рассматривать как совокупность практических рекомендаций по построению системы обеспечения целостности в ИС.

Безусловными достоинствами этой модели являются ее относительная простота и легкость совместного использования с другими моделями безопасности. Основным недостатком модели Кларка – Вилсона считается сложность реализации в реальных ИС методов и процедур проверки целостности данных. При этом в контексте традиционных баз данных от этого недостатка можно в значительной степени избавиться благодаря тому, что в реляционных БД некоторые ограничения целостности заложены в теории (целостность сущностей, ссылочная целостность), другие могут быть указаны как статические ограничения с помощью SQL (декларативная поддержка ограничений целостности), третьи – как динамические ограничения целостности (процедурная поддержка ограничений целостности). Однако для БД актуальными остаются вопросы, связанные с механизмами контроля целостности хранимых процедур, функций (как файловых объектов). Это обусловливает необходимость проведения дополнительных исследований в соответствующих направлениях.

3. Модель Биба, с точки зрения содержания и формального представления, является инверсией мандатной модели Белла – ЛаПадулы, проблема которой заключается в том, что она разработана для хранения секретов, не гарантируя при этом целостность данных. Поскольку формальное описание модели Биба очень близко с описанием модели Белла – ЛаПадулы, то она, естественно, обладает большинством достоинств и недостатков присущих модели безопасности на основе мандатной политики доступа.

На практике для создания защищенных ИС как систем, обеспечивающих конфиденциальность и целостность данных, важным является объединение моделей Белла – ЛаПадулы и Биба, причем объединение на основе одной общей решетки, но с двумя метками безопасности: по конфиденциальности и по целостности, с противоположным характером их определения.

Список литературы:

1. Девянин П. Н. Модели безопасности компьютерных систем. Управление доступом и информационными потоками. 2-е изд. Москва : Горячая линия–Телеком, 2013. 338 с.
2. Chapple M., Stewart J. M., Gibson D. CISSP Certified Information Systems Security Professional Official Study Guide, 8th ed. Sybex, John Wiley & Sons, Inc.: Indianapolis, Indiana, 2018. 1050 p.
3. Гайдамакин Н. А. Теоретические основы компьютерной безопасности. Екатеринбург : Изд-во Уральского ун-та, 2008. 212 с.
4. Tanenbaum A. S., Herbert Bos H. Modern Operating Systems. Fourth edition. Pearson, 2015. 1136 p.
5. Смирнов С. Н. Безопасность систем баз данных. Москва : Гелиос АРВ, 2007. 352 с.
6. Clark D. D., Wilson D. R. A Comparison of Commercial and Military Computer Security Policies // Proceedings of the 1987 IEEE Symposium on Research in Security and Privacy (SP'87), May 1987, Oakland, CA : IEEE Press, 1987. P. 184–193.
7. Gollmann D. Computer Security. 3rd ed. Wiley, 2011. 436 p.
8. Yesin V.I., Yesina M.V., Vilihura V.V. Monitoring the integrity and authenticity of stored database objects // Telecommunications and Radio Engineering. 2020. Vol. 79, Issue 12. P. 1029-1054.
9. Sandhu R. S., Jajodia S. Data and database security and controls // Handbook of information security management, Auerbach Publishers. 1993. P. 481-499.
10. Девянин П. Н., Михальский О. О., Правиков Д. И. и др. Теоретические основы компьютерной безопасности. Москва : Радио и связь, 2000. 192 с.
11. Ge X., Polack F., Laleau R. Secure databases: an analysis of Clark-Wilson model in a database environment // International Conference on Advanced Information Systems Engineering. Springer, Berlin, Heidelberg, 2004. P. 234-247.
12. Щеглов А. Ю. Защита компьютерной информации от несанкционированного доступа. СПб. : Наука и Техника, 2004. 384 с.
13. Biba K. J. Integrity considerations for secure computer systems. MTR-3153-REV-1. Mitre Corp Bedford MA, 1977. 64 p.
14. 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.
15. Цирлов В. Л. Основы информационной безопасности автоматизированных систем. Ростов-на-Дону : Феникс, 2008. 173 с.
16. Зегжда Д. П. Информационная безопасность. Москва : МГТУ им. Н.Э. Баумана, 2010. 236 с.
17. Зегжда Д. П., Ивашко А. М. Основы безопасности информационных систем. Москва : Горячая линия–Телеком, 2000. 452с.

Поступила в редколлегию 08.01.2021

Сведения об авторах:

Есин Виталий Иванович – д-р техн. наук, профессор, Харьковский национальный университет имени В.Н. Каразина, профессор кафедры безопасности информационных систем и технологий, факультета компьютерных наук; Украина; e-mail: v.i.yesin@karazin.ua; ORCID: https://orcid.org/0000-0003-1977-7269

Рассомахин Сергей Геннадьевич – д-р техн. наук, профессор, Харьковский национальный университет имени В.Н. Каразина, заведующий кафедрой безопасности информационных систем и технологий, факультета компьютерных наук; Украина; e-mail: rassomakhin@karazin.ua; ORCID: https://orcid.org/0000-0003-1394-3588

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