Математические структуры и моделирование
2016. № 2(38). С. 92–102

УДК 004.056.53

ПРИМЕНЕНИЕ ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ МОДЕЛЕЙ РАЗГРАНИЧЕНИЯ ДОСТУПА К АНАЛИЗУ БЕЗОПАСНОСТИ РЯДА КОМПЬЮТЕРНЫХ СИСТЕМ

С.В. Белим
профессор, д.ф.-м.н., e-mail: sbelim@mail.ru
С.В. Усов
к.т.н., e-mail: raintower@mail.ru

Омский государственный университет им. Ф.М. Достоевского

Аннотация. В работе проводится анализ подсистем безопасности ОС семейства Windows и СУБД Oracle в рамках объектно-ориентированной модели HRU.

Ключевые слова: модели безопасности, разграничение доступа, Windows, Oracle, HRU.

Введение

Современное состояние индустрии производства информационных систем характеризуется использованием широкого спектра парадигм и технологий их проектирования и реализации. Существенную долю рынка занимают информационные системы, построенные с использованием объектно-ориентированной парадигмы. Вместе с тем, существующий научно-методический аппарат верификации подсистем обеспечения информационной безопасности в значительной степени опирается на классические субъектно-объектные модели и не может быть непосредственно применён для анализа объектно-ориентированных систем. Однако с появлением объектно-ориентированных моделей систем безопасности [1] появляется возможность построения методики проверки объектно-ориентированных компьютерных систем на возможность утечки права доступа согласно объектно-ориентированной парадигме.

В данной работе проводится анализ подсистем безопасности двух широко распространённых компьютерных систем (таких как ОС Windows и СУБД Oracle) с точки зрения возможности описания их в рамках объектно-ориентированной модели Харрисона-Руззо-Ульмана, также даются рекомендации по настройке их подсистем безопасности с целью сведения к случаям, допускающим автоматическую проверку наличия несанкционированного доступа. Кроме того, предложена методика, позволяющая анализировать системы на наличие возможностей несанкционированного доступа, а также вырабатывать рекомендации по проектированию подсистемы безопасности новых объектно-ориентированных систем.

1. Объектно-ориентированная модель системы безопасности

Объектно-ориентированная дискреционная модель разграничения доступа OOHRU является развитием классической модели Харрисона-Руззо-Ульмана [2], предложенной авторами в 70-х годах для анализа безопасности субъектно-объектных систем.

Компьютерная система рассматривается в виде множества объектов $\mathbf{O}$, представляющихся наборами открытых полей и скрытых полей, а также методов обработки полей. Каждый из объектов принадлежит какому-то классу k из множества всех классов системы $\mathbf{K}$, причём объекты одного класса обладают одинаковым набором полей и методов. Для построения системы дискреционного разграничения доступа каждый объект системы дополняется скрытым полем M, или матрицей доступа – таблицей, содержащей информацию о всех разрешённых видах доступа к полям и методам данного объекта. В частности, ячейка o'$.M[o, f]$ матрицы доступа объекта o' содержит все права доступа, которыми обладает объект o на поле f объекта o' [1].

Определены элементарные операторы, преобразующие матрицу доступа:

1. $Create(o, k)$ — создаёт объект o класса $k \in \mathbf{K}$, если $o \in \mathbf{O}$.
2. $Destroy(o)$ — уничтожает объект o, если $o \in \mathbf{O}$.
3. $Enter(r, o, o'.f)$ — вносит право доступа r в $o'.M[o, f]$, если $o, o' \in \mathbf{O}$.
4. $Delete(r, o, o'.f)$ — удаляет право r доступа из $o'.M[o, f]$, если $o, o' \in \mathbf{O}$.
5. $Grant(o, o'.s)$ — разрешает вызов объекту o метода $o'.s$, если $o, o' \in \mathbf{O}$.
6. $Deprive(o, o'.s)$ — запрещает вызов объекту o метода $o'.s$, если $o, o' \in \mathbf{O}$.

Состояния компьютерной системы в модели HRU изменяются под воздействием запросов на модификацию матрицы доступа в виде команд следующего формата:

if <конъюнкция логических выражений вида $r \in o'.M[o, f]$ или $o'.M[o, s] = 1$>

then <последовательность элементарных операторов>.

На место аргументов команды подставляются объекты либо классы, участвующие в качестве переменных в условной части либо в элементарных операторах.

Определение 1. HRU-модель системы безопасности называется монооперационной, если каждая команда в этой системе содержит только один элементарный оператор.

Определение 2. HRU-модель системы безопасности называется моноусловной, если каждая команда в этой системе содержит только одно условие.

Определение 3. HRU-модель системы безопасности объектно-ориентированной компьютерной системы называется монотонной, если команды этой системы не содержат операторов $Delete$, $Deprive$ и $Destroy$.

Определение 4. HRU-модель системы безопасности объектно-ориентированной компьютерной системы называется однородной, если все объекты одного класса этой модели обладают одним и тем же набором прав доступа.

Было показано, что модель OOHRU допускает проверку возможности утечки права доступа в следующих случаях [1, 3]:

1) монооперационная OOHRU;
2) монотонно-моноусловная OOHRU;
3) однородная OOHRU.

Результаты для однородного случая также переносятся на более широкий класс иерархических OOHRU.

2. Иерархическая модель OOHRU

Модель безопасности OOHRU называется иерархической (или моделью с иерархией), если на множестве объектов **O** задан частичный порядок-иерархия «$\leqslant$», и в любой момент работы системы для любых двух объектов $o, o' \in \mathbf{O}$ таких, что $o' \leqslant o$, для любого поля или метода x, общего для объектов o и o', и для любого поля или метода x' объекта $o'' \in \mathbf{O}$ верно следующее: $o''.M[o, x'] \subset o''.M[o', x']$ и $o'.M[o'', x] \subset o.M[o'', x]$.

Однако в объектно-ориентированных системах классы (и, соответственно, объекты этих классов) уже связаны частичным отношением наследственности, поэтому в данном случае иерархия строится естественным образом.

Стоит выделить два способа задания иерархии на объектах одного и того же класса. Случай, когда объекты одного класса находятся на одном ярусе иерархии, соответствует однородной иерархической модели OOHRU. В этой ситуации достаточно ограничиться рассмотрением иерархии на классах. В более общем случае объекты одного класса могут находиться на разных ярусах иерархии, изменять со временем своё положение в иерархии, а потому не могут считаться сравнимыми. Тем не менее, если $o' \leqslant o$, где o' — объект класса k', а o — объект класса k, то любой объект класса k' будет меньше либо равен любому объекту класса k с точки зрения множества наборов прав. Как следствие, семейство множеств наборов прав, допустимых для объектов рассматриваемого класса, должно быть ограничено снизу: нижняя грань по включению этого семейства принимается за множество наборов прав самого класса.

Элементарные операторы в иерархическом случае имеют более сложный вид, а в неоднородном случае ещё и являются перегруженными [3]. В качестве примера приведём вид оператора $Enter$ в однородном случае:

$Enter(r, o^k, o'^{k'}.f)$ — вносит право доступа r в $k'.M[k, f]$. Данный элементарный оператор может быть выполнен, только если выполнены следующие условия (так называемые условия целостности):

А. Для любого класса k_1, являющегося потомком класса k, $r \in o'^{k'}.M[k_1, o'^{k'}.f]$;

Б. Для любого класса k_2, являющегося родителем класса k', $r \in k_2.M[o^k, k_2.f]$.

Условиями целостности также дополняются операторы $Delete$, $Grant$, $Deprive$.

3. Классификация моделей

Проведём классификацию объектно-ориентированных моделей безопасности с дискреционным разграничением доступа на основе построений, описанных ранее. Все модели разделим на две группы: без наследования классами прав доступа (группа F) и с наследованием классами прав доступа (группа H). Каждая модель, в силу своей специфики, требует своего набора элементарных операторов для рассмотрения вопросов безопасности. В каждой из моделей присутствует свой аналог монооперационной и монотонно-моноусловной системы, допускающих алгоритмическую проверку безопасности. Таким образом, получаем следующую классификацию:

- F1. Базовая (неоднородная) модель объектно-ориентированной системы без наследования. В данной модели каждый объект принадлежит определённому классу, при этом все классы независимы.
- FZ. Безусловная неоднородная модель, включающая в себя всевозможные команды.
- H0. Однородная модель с иерархией. В данной модели все объекты класса обладают одинаковым набором прав доступа к полям и методам всех объектов другого класса, при этом классы связаны между собой иерархией наследования.
- H1. Неоднородная модель с иерархией. Отличается от модели F1 заданием строгой связи — иерархии — на множестве классов.
- H2. Финально-неоднородная модель с иерархией. В этой модели неоднородными могут быть лишь классы, не имеющие наследников.
- HZ. Безусловная неоднородная модель с иерархией, включающая в себя всевозможные команды.

Для каждой из моделей ранее были выявлены условия, при которых соответствующая компьютерная система допускает алгоритмическую проверку безопасности. Следует иметь в виду, что все полученные условия являются достаточными, но не являются необходимыми. То есть, если эти условия выполнены, то возможно построение программного комплекса, который в каждый момент времени может автоматически провести проверку о наличии утечки прав доступа к объектам компьютерной системы.

Рассмотрим применимость каждой из моделей к конкретным информационным системам или информационным подсистемам и условия, которым должны удовлетворять эти системы для автоматической проверки своей безопасности.

4. Модель F1. Операционные системы семейства Windows

Операционные системы семейства Windows защищённой линии (NT/2000/XP/Vista/Seven) [4] реализованы на основе объектно-ориентированного подхода и имеют собственную подсистему безопасности. Объектом операционной системы может быть любой ресурс (файл, каталог, устройство и т. д.). Принципиально не отличающиеся объекты (в частности, создающиеся с

использованием одних и тех же конструкторов и т. д.) можно рассматривать как объекты одного класса. Субъектами операционной системы выступают процессы, которые при объектно-ориентированном подходе к моделированию политик безопасности ассоциируются с методами соответствующих объектов.

Каждый объект имеет набор полей и прав доступа к ним. Все права доступа разделены на три группы: стандартные, специальные и общие права доступа. Общие права доступа выдаются на доступ ко всему объекту в целом и представляют собой композицию специальных и стандартных прав доступа. То есть общие права доступа могут быть записаны как набор специальных и стандартных прав, следовательно, являются просто составными правами. Стандартные права доступа дают возможность работать с полями, одинаковыми для всех объектов системы (заголовок, атрибуты и т. д.). Стандартные права доступа реализованы с помощью сервисов операционной системы. Таким образом, доступ к объекту с одним из стандартных прав реализует работу с полями public в рамках рассматриваемой модели. Специальные права доступа обрабатываются самим объектом. То есть объект содержит методы доступа к своим специфическим полям. Если происходит запрос на доступ к объекту по специальному праву, то активизируется метод объекта, обрабатывающий соответствующее поле. Например, для файлов, хранящихся на жёстком диске, запись в содержимое файла требует вызова особых сервисов, содержащихся в драйвере NTFS и присущих классу объектов «файл». То есть специальные права доступа к объектам представляют собой права вызова методов объекта, соответствующие поля объектов относятся к типу private.

Матрица доступов для каждого объекта реализована с помощью разделительного списка контроля доступов (DACL). DACL приписывается каждому объекту компьютерной системы. В заголовке объекта хранится ссылка на дескриптор безопасности объекта, в котором содержится ссылка на список контроля доступов. Список контроля доступов представляет собой линейный список из записей с набором полей. В рамках рассматриваемой модели интерес представляют три поля. Первое поле показывает тип записи — разрешающая или запрещающая. Соответственно эти два типа реализуют политику безопасности по «белому списку» и по «чёрному списку». Политика безопасности по «белому списку» является более жёсткой, так как реализует принцип «запрещено все, что не разрешено». Второе поле содержит имя пользователя, которому соответствует эта запись. Пользователи кодируются с помощью идентификатора безопасности (SID), который является индивидуальным. Третье поле содержит маску доступов, показывающую множество разрешённых или запрещённых доступов стандартного, специального или общего вида.

Построим соответствие между моделью F1 и подсистемой безопасности операционной системы Windows. Реализация элементарных операторов представлена в таблице 1.

Обратим внимание, что команда $CreateProcess()$, создающая субъект (процесс) в рамках субъектно-объектной парадигмы, при объектно-ориентированном подходе к системам безопасности создаёт именно объект, содержащий метод, соответствующий процессу.

Таблица 1. Соответствие между элементарными операторами субъектно-объектной и объектно-ориентированной КС

Элементарный оператор модели F1	Компонент подсистемы безопасности ОС Windows
$Create(o, k)$ — создаёт объект o класса k	Команды создания различных объектов. Например, $CreateFile()$, $CreateProcess()$ и т.д.
$Destroy(o)$ — уничтожает объект o	Команды удаления объектов. Например, $DeleteFile()$.
$Enter(r, o, o'.f)$ — вносит право доступа r в $o'.M[o, f]$	Команды внесения записей в DACL, содержащих стандартные права доступа. Например, $AddAce()$.
$Delete(r, o, o'.f)$ — удаляет право r доступа из $o'.M[o, f]$	Команды удаления записей в DACL, содержащих стандартные права доступа. Например, $DeleteAce()$.
$Grant(o, o'.s)$ — разрешает вызов объекту o метода $o'.s$	Команды внесения записей в DACL, содержащих специальные права доступа. Например, $AddAce()$.
$Deprive(o, o'.s)$ — запрещает вызов объекту o метода $o'.s$	Команды удаления записей в DACL, содержащих специальные права доступа. Например, $DeleteAce()$

Выпишем условия, при которых система будет монооперационной:

1. Запрещены команды, вносящие запрещающие ACE в DACL объектов. Этот запрет вытекает из того, запрещающие ACE являются разрешающими сразу для большого количества прав, которые не запрещены.
2. Запрещены команды, вносящие общие права доступа в DACL объектов. Общие права доступа представляют собой композицию стандартных и специальных прав, то есть происходит внесение сразу нескольких прав в одной команде.
3. Запрещены команды, вносящие сразу несколько прав в DACL.
4. За одно обращение к объекту может быть внесено или удалено не более одного права доступа.

Выпишем условия, при которых система будет монотонно-моноусловной:

1. В каждом ACE должен быть прописан конкретный пользователь, для которого внесено соответствующее право доступа. Нельзя назначать права доступа для групп пользователей и других составляющих домена. Это требование вытекает из моноусловности, так как использование групп пользователей требует разбиения условия проверки на несколько условий, связанных с отдельными пользователями.
2. Запрещено автоматически удалять объекты информационной системы.

Удалять разрешается только пользователю, который является внешним по отношению к информационной системе. Предполагается, что пользователи действуют корректно.

3. Запрещено автоматически удалять записи в DACL. Удаление также могут проводить только пользователи.

Оба рассмотренных случая могут быть реализованы с помощью стандартных средств администрирования операционной системы. Выполнение этих требований при дальнейшей эксплуатации операционной системы также может быть гарантировано средствами администрирования. Таким образом, возможна такая настройка операционной системы Windows, при которой проверка наличия или отсутствия утечек прав доступа будет производиться автоматически. Следует отметить, что такая настройка системы снижает её функциональность. Однако это обычный эффект от применения средств защиты информации.

5. Модель H0. Объектно-ориентированные базы данных стандарта ISO/IEC SQL:1999

В качестве объектов в базах данных можно рассматривать любую сущность, содержащую конечную информацию: таблицы, представления, строки, столбцы. В рамках объектно-ориентированного подхода мы будем отождествлять с объектом в первую очередь ячейку таблицы базы данных. Под классом же можно понимать набор объектов, одинаковых с точки зрения устройства, например, столбец таблицы. Отдельные классы создаются для представления групп пользователей. Субъектами с точки зрения субъектно-объектного подхода могут быть пользователи, роли и хранимые процедуры. При объектно-ориентированном подходе каждая из перечисленных сущностей может быть интерпретирована как метод некоторого класса. Права доступов на объект обычно задаются либо администратором СУБД, либо владельцем объекта. В зависимости от конкретной СУБД механизмы обеспечения безопасности данных могут отличаться, но описанные выше элементы дискреционных политик присутствуют всегда.

Объектно-ориентированный подход к построению баз данных в последнее время получает все более широкое распространение, соответствующие базы данных получили название объектно-реляционных. Новые базы данных описаны в международном стандарте ISO/IEC SQL:1999. Данный стандарт реализован в СУБД Oracle, начиная с версии 8i [5].

В СУБД Oracle кроме стандартных типов полей допускается создание классов полей, определяемых пользователем. Созданные классы могут использоваться для определения столбцов таблиц. Классы могут включать как поля, так и собственные методы. Реализованы все стандартные свойства объектно-ориентированного подхода: инкапсуляция, наследование и полиморфизм. Определена процедура наследования классов, что приводит к иерархии классов.

Имеет смысл различать два способа администрирования БД Oracle. Первый способ не позволяет пользователям создавать собственные новые типы, а также новые роли. В процессе функционирования системы безопасности БД

Таблица 2. Соответствие между элементарными операторами модели H0 и командами СУБД Oracle

Элементарный оператор модели H0/H2	Компонент подсистемы безопасности СУБД Oracle
$Create(o, k)$ — создает объект o класса k	DECLARE o k; Либо вызов в OCI: $OCIObjectNew()$; Если объект является строкой таблицы: INSERT INTO $T(k)$ O; Если объект — пользователь: CREATE USER o.
$Destroy(o)$ — уничтожает объект o	Удаление нетабличного объекта происходит автоматически. Либо вызов в OCI: $OCIObjectMarkDelete()$; При удалении строки из таблицы: DELETE FROM $T(k)$ o; (или оператором DELETE_OBJECT модуля UTL_REF) Если объект — пользователь: DROP USER o.
$Enter(r, o, k'.f)$ — назначает привилегию r объекту (пользователю) o на поле f любого объекта класса (любой ячейки таблицы) k'	GRANT r ON $k'.f$ TO o.
$Delete(r, o, k'.f)$ — отзывает привилегию r у объекта (пользователя) o на поле f любого объекта класса (любой ячейки таблицы) k'	REVOKE r ON $k'.f$ FROM o.
$Grant(o, k'.s)$ — разрешает вызов объекту (пользователю) o метода $k'.s$	GRANT EXECUTE ON $k'.s$ TO o.
$Deprive(o, k'.s)$ — запрещает вызов объекту (пользователю) o метода $k'.s$	REVOKE EXECUTE ON $k'.s$ FROM o

Oracle пользователь может лишь создать объект (или таблицу объектов-строк) ранее определённого типа, либо назначить себе или другому пользователю уже существующую роль. Во втором способе указанные ограничения не накладываются, однако, подобный контроль над разграничением доступа оправдан лишь в системах, все пользователи которых доверяют друг другу (например, в среде разработчиков).

Напомним, классические модели дискреционного разграничения доступа таковы, что набор прав, набор команд и набор классов системы не изменяется в процессе функционирования системы. Таким образом, иерархическая модель OOHRU описывает именно первый способ администрирования базы данных. А именно, привилегии — это права доступа в модели безопасности, роли — фиксированные наборы прав, таким образом, назначение или отзыв роли в БД Oracle соответствует исполнению команды в OOHRU. Наконец, объектному типу (или таблице объектов-строк) соответствует класс объектно-ориентированной модели безопасности.

Иерархия классов в модели OOHRU, описывающей СУБД, не обязательно является естественной. Однако исполнение команды в любом случае осуществляется в соответствии с условиями целостности. Например, пользователь $Prepod$ получил от пользователя $Dekan$ привилегию UPDATE столбца $Marks$ таблицы $MarkTable$ с опцией распространения этой привилегии (GRANT OPTION). После чего $Prepod$ назначил привилегию UPDATE столбца $Marks$ пользователю $Student$. Если теперь $Dekan$ отзовёт привилегию UPDATE у пользователя $Prepod$, этот эффект должен каскадным образом распространиться и на пользователя $Student$. Таким образом, не произойдёт нарушения условий целостности в иерархии, в которой класс $Dekan$ наследует права от класса $Prepod$, в свою очередь наследующего права у класса $Student$ (обладающего наименьшим количеством прав).

По построению СУБД Oracle разрешена выдача прав на класс в целом, но не на отдельный объект, которым является ячейка таблицы. Таким образом, мы имеем дело с однородной объектно-ориентированной моделью с иерархией (H0). Однако если рассматривать базу данных, пользователи которой не являются фиксированными (в частности, все время могут появляться новые пользователи, обладающие своими собственными схемами и обладающие привилегиями независимо друг от друга), сопоставление каждому отдельному пользователю своего «субъектного» типа в модели безопасности OOHRU не представляется возможным. В этом случае имеет смысл представлять СУБД Oracle финально-неоднородной моделью OOHRU с иерархией, где пользователям будут соответствовать объекты финальных классов, в то время как объекты непосредственно самой базы данных будут относиться к однородным классам. Соответствие между операторами OOHRU для этой модели и командами СУБД Oracle приведены в таблице 2.

Выше под $T(k)$ понимается таблица объектов-строк типа (класса) k, которая создаётся оператором CREATE TABLE $T(k)$ OF k.

В отдельных случаях может быть удобно интерпретировать именно объектный тип СУБД Oracle в качестве объекта (не класса!) модели безопасности

OOHRU, в этом случае аналогом команды создания объекта *o* в модели безопасности будет следующая команда SQL: CREATE TYPE *o* AS OBJECT. В свою очередь, удаляется объектный тип командой DROP TYPE *o*.

Таким образом, иерархическая финально-неоднородная объектно-ориентированная модель применима для анализа безопасности объектно-реляционных баз данных. Если же СУБД можно описать однородной моделью H0, то в данной системе можно произвести проверку возможности несанкционированного доступа.

6. Перспективы применения: методика проверки объектно-ориентированной системы безопасности на наличие возможности несанкционированного доступа

На основе результатов, полученных для описанных выше моделей, были предложены методика и алгоритмы установления возможности несанкционированного доступа в объектно-ориентированных системах безопасности.

В результате применения методики нам либо удастся доказать безопасность объектно-ориентированной системы, либо доказать, что система допускает утечку права доступа, либо выяснить, что система не относится ни к одному из известных классов, допускающих проверку безопасности.

Однако наибольшими перспективами предложенная методика обладает в сфере проектирования новых подсистем безопасности, которые заведомо будут принадлежать к одному из классов, допускающих проверку возможности утечки права доступа.

Литература

1. Белим С.В., Белим С.Ю., Усов С.В. Объектно-ориентированная модификация модели безопасности HRU //Проблемы информационной безопасности. Компьютерные системы. 2010. № 1. С. 6–14.
2. Harrison M.A., Ruzzo W.L., Ulman J.D. Protection in Operating Systems // Communications of the ACM, 1975. P. 14–25.
3. Усов С.В. Неоднородные объектно-ориентированные модели с иерархией // Проблемы обработки и защиты информации. Книга 3. Модели разграничения доступа. Коллективная монография. Омск : «Полиграфический центр КАН», 2013.
4. Руссинович М. Соломон Д. Внутреннее устройство Microsoft Windows: Windows Server 2003, Windows XP и Windows 2000. Мастер-класс. Пер. с англ. 4-е изд. М. : Издательско-торговый дом «Русская редакция», 2005. 992 с.
5. Кайт Т. Oracle для профессионалов: архитектура, методики программирования и особенности версий 9i, 10g и 11g, 2-е издание. М. : «Вильямс», 2011. 848 с. С. 93–114.

APPLICATION OF OBJECT-ORIENTED MODELS OF ACCESS PERMISSIONS TO THE SAFETY ANALYSIS OF A NUMBER OF COMPUTER SYSTEMS

S.V. Belim
Professor, Dr.Sc. (Phys.-Math.), e-mail: sbelim@mail.ru
S.V. Usov
Ph.D. (Eng.), e-mail: raintower@mail.ru

Dostoevsky Omsk State University

Abstract. This paper analyzes the security sub-systems of Windows NT operating system and DBMS Oracle 8i + in the terms of object-oriented model HRU.

Keywords: discretionary safety models, Windows, Oracle, HRU.