Математические структуры и моделирование
2016. № 4(40). С. 151–163

УДК 004.056.53

О СВЯЗИ МЕЖДУ ОБЪЕКТНО-ОРИЕНТИРОВАННОЙ ДИСКРЕЦИОННОЙ И СУБЪЕКТНО-ОБЪЕКТНОЙ МАНДАТНОЙ МОДЕЛЯМИ БЕЗОПАСНОСТИ

С.В. Усов
к.т.н., e-mail: raintower@mail.ru

Омский государственный университет им. Ф.М. Достоевского

Аннотация. В статье рассмотрены объектно-ориентированная модель Харрисона-Руззо-Ульмана и субъектно-объектные модели с мандатным разграничением доступа. Показано, что модель Белла-Лападулы и классические мандатные модели могут быть реализованы с помощью объектно-ориентированной модели HRU.

Ключевые слова: дискреционные модели безопасности, мандатные модели безопасности, разграничение доступа, HRU, модель Белла-Лападулы.

Введение

Как дискреционные, так и мандатные политики безопасности известны еще с 70-х годов прошлого столетия, и традиционно базируются на субъектно-объектной парадигме компьютерной системы. Однако в связи с возрастающей актуальностью объектно-ориентированного подхода к построению компьютерных систем, возникает необходимость в пересмотре классических политик безопасности. Так, например, в [1] была предложена объектно-ориентированная модель разграничения доступа, базирующаяся на модели HRU (Харрисона-Руззо-Ульмана) [2], однако, обладающая более широкими возможностями, в частности, в рамках охвата компьютерных систем, которые можно описать с помощью этой модели.

Получают более широкое применение и мандатные политики безопасности [3], [4]. В частности, мандатная модель используется как семейством операционных систем Windows (начиная с Vista применяется для контроля целостности), так и семейством операционных систем Linux (доступна в качестве расширений). Цель данной работы — установить взаимосвязь между объектно-ориентированными дискреционными системами безопасности и мандатными системами безопасности, эксплуатирующими субъектно-объектный подход.

Прежде всего необходимо ответить на вопрос, позволяет ли инструментарий объектно-ориентированной модели HRU реализовать мандатную политику безопасности, и если позволяет, то с какими ограничениями.

В работе [1] была предложена иерархическая модель OOHRU, устройство которой подразумевает, что объект o, находящийся на более низком уровне

иерархии, чем объект o', обладает меньшим набором прав (как в отношении доступа к другим объектам, так и в отношении ограничения доступа других объектов по отношению к себе) по сравнению с объектом o'. Такая структура в точности повторяет решётку ценностей мандатной политики безопасности, что позволяет сделать предположение о структурной близости данных моделей.

1. Объектно-ориентированная модель безопасности с дискреционным разграничением доступа (OOHRU)

Компьютерная система в OOHRU рассматривается в виде множества объектов $\mathbf{O}$, разбитых по множеству классов $\mathbf{K}$ (все объекты одного класса имеют одинаковый набор полей и методов), обладающих открытыми полями $f \in \mathbf{F}$ и скрытыми полями $p \in \mathbf{P}$, а также методами обработки полей $s \in \mathbf{S}$. Здесь $F = \bigcup_{k \in \mathbf{K}} k.\mathbf{F}$ — множество всевозможных открытых полей всех объектов и классов, $k.\mathbf{F}$ — множество открытых полей класса k (каждый объект класса k обладает тем же набором $k.\mathbf{F}$ открытых полей), аналогично определяются $\mathbf{P}$ и $\mathbf{S}$. Причём если поле $k.f$ наследуется классом k у класса k', то соответствующее поле класса k' мы будем для удобства обозначать именно $k'.f$, подчёркивая тем самым их взаимосвязь (таким образом, $f \in k.\mathbf{F}$ и $f \in k'.\mathbf{F}$). Пусть $\mathbf{O}^k \in \mathbf{O}$ — множество объектов класса $k \in \mathbf{K}$. В случае, если требуется уточнить класс объекта, поле f объекта $o^k \in \mathbf{O}^k$ будем обозначать $o^k.f$, поле f класса k — $k.f$. Для скрытых полей класса будем использовать аналогичные обозначения.

Для построения модели дискреционного разделения доступов для каждого объекта и для каждого класса вводится дополнительное скрытое поле M, содержащее локальную матрицу доступов, и методы работы с матрицей доступов. Модификация матриц доступа производится посредством выполнения команд системы безопасности, о которых будет сказано ниже.

Модель безопасности OOHRU называется иерархической (или моделью с иерархией), если на множестве объектов $\mathbf{O}$ задан частичный порядок-иерархия, 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 $x \in \mathbf{X}$, общего для объектов o и o', и для любого поля или метода $x' \in \mathbf{X}$ объекта $o'' \in \mathbf{O}$ верно следующее: $o''.M[o, x'] \subset o''.M[o', x']$ и $o'.M[o'', x] \subset o.M[o'', x]$. Здесь и далее $\mathbf{X}$ — множество всевозможных полей и методов всех существующих в системе на данный момент времени объектов, «$\leqslant$» — отношение частичного порядка.

Состояние системы в модели HRU изменяется под действием команд, которые состоят из условной части и последовательности элементарных операторов [2], которая выполняется, только если истинна условная часть. Список элементарных операторов в OOHRU включает [1]:

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

6. $Deprive(r, o^k, o'^{k'}.s)$ — запрещает вызов объектом o^k метода $o'^{k'}.s$.

Изменения, производимые операторами, отражаются в матрицах доступа объектов системы. Подробное описание модели OOHRU можно найти в [1].

2. Мандатные политики безопасности

Мандатные политики безопасности оперируют понятиями уровня секретности информации и уровня доверия к пользователю. На множестве уровней секретности (уровней доверия) задано отношение нестрогого порядка. Таким образом, получаем частично упорядоченное множество L, в отдельных случаях являющееся решёткой (например, если L линейно упорядочено). Такую решётку будем называть решёткой ценностей.

Типичным примером мандатной политики безопасности является общепринятая для секретного документооборота в большинстве стран модель MLS, основанная на решётке ценностей. На множестве объектов $\mathbf{O}$ системы определяется функция ценности C, сопоставляющая каждому объекту один из уровней решётки ценностей L. Поток информации от объекта o к объекту o' допускается, только если $C(o)$ не превосходит $C(o')$.

Другим примером может служить модель Белла-ЛаПадулы [3], [4], в которой, однако, мандатная политика безопасности совмещается с дискреционной. Опишем эту модель подробнее.

Система представляется совокупностью множества объектов $\mathbf{O}$ доступа, множества субъектов $\mathbf{S}$ доступа, множества видов доступа $A = \{read, write, append, execute\}$ и матрицы доступов M, аналогичной используемой в модели HRU. Кроме того, заданы решётка ценностей L (обычно это линейно-упорядоченное множество, содержащее четыре классических уровня секретности: Unclassified, Confidencial, Secret, TopSecret, перечислены в порядке возрастания секретности) и тройка отображений $f = (f_S, f_C, f_O)$. Множество всех таких троек f обозначим через $\mathbf{F}$. $f_S : \mathbf{S} \to L$ определяет максимальный уровень допуска субъекта, $f_C : \mathbf{S} \to L$ — текущий уровень допуска субъекта, а $f_O : \mathbf{O} \to L$ — уровень секретности объекта. Дополнительно может определяться иерархия H объектов системы (например, на основе отношения вложенности папок [4]), для любой пары «родитель-потомок» этой иерархии уровень секретности родителя не может превосходить уровень секретности потомка (но может совпадать с ним). Отображение $f_H : \mathbf{O} \to L$ сопоставляет каждому объекту системы его место в иерархии, причём все объекты o, находящиеся в иерархии на одной позиции $h = f_H(o)$, имеют один и тот же уровень секретности $f_O(o)$. Поэтому в дальнейшем под уровнем секретности объекта мы будем понимать именно его позицию h в иерархии.

Множество текущих доступов в системе можно записать как $B \subset \mathbf{S} \times \mathbf{O} \times A$. В каждый момент времени система находится в определённом состоянии, являющемся декартовым произведением $d = \mathbf{B} \times \mathbf{M} \times f \times H$, переход в другое состояние осуществляется посредством выполнения одной из системных команд из множества Γ, а также исполнения запросов на доступ из множества Q, также являющегося подмножеством $\mathbf{S} \times \mathbf{O} \times A$. В отличие от дискреционной

модели Харрисона-Руззо-Ульмана, вид команд из множества Γ не специфицирован, однако, приводится список их возможностей [3]:

1. Изменить положение объекта в иерархии H, изменить текущий уровень доверенности субъекта или уровень секретности объекта, то есть изменить функции f_H, f_C и f_O.

2. Добавить или удалить право доступа субъекту на объект, то есть изменить содержание матрицы доступов M.

3. Создать новый объект или удалить группу объектов, то есть изменить иерархию H.

В то же время запросы из множества Q служат для оперирования (создания или прекращения) текущими потоками между субъектами и объектами. Подобные запросы не специфицированы в HRU, поэтому мы не будем останавливаться на них подробно.

Безопасность системы определяется с помощью трех свойств: ss-свойства, *-свойства и ds-свойства.

Доступ $b = (s, o, r) \in B$ обладает ss-свойством относительно $f = (f_S, f_C, f_O)$, если

1. $r = read$ или $write$, и $f_O(o) \leqslant f_S(s)$.

2. $r = execute$ или $append$.

Доступ $b = (s, o, r) \in B$ обладает *-свойством относительно $f = (f_S, f_C, f_O)$, если

1. $r = read$ и $f_O(o) \leqslant f_S(s)$.

2. $r = append$ и $f_S(s) \leqslant f_O(o)$.

3. $r = write$ и $f_O(o) = f_S(s)$.

4. $r = execute$.

Доступ $b = (s, o, r) \in B$ обладает ds-свойством относительно $f = (f_S, f_C, f_O)$, если $r \in M[s, o]$.

Состояние системы обладает ss-свойством (*-свойством, ds-свойством) относительно $f = (f_S, f_C, f_O)$, если каждый доступ b в этом состоянии обладает тем же свойством.

Состояние системы называется безопасным, если оно обладает всеми тремя свойствами. Реализация системы называется безопасной, если каждое состояние её безопасно. Ограничения на команды в безопасной реализации системы описаны в так называемой Basic Security Theorem [3]. Перечислим эти ограничения для перехода системы из состояния (B, M, f, f_H) в состояние (B', M', f', f'_H):

1) любой доступ $(s, o, r) \in B' \setminus B$ обладает ss-свойством относительно f';

2) если $(s, o, r) \in B$ и не обладает ss-свойством относительно f', то $(s, o, r) \notin B'$;

3) любой доступ $(s, o, r) \in B' \setminus B$ обладает *-свойством относительно f';

4) если $(s, o, r) \in B$ и не обладает *-свойством относительно f', то $(s, o, r) \notin B'$;

5) для любого доступа $(s, o, r) \in B' \setminus B$ верно, что $r \in M[s, o]$;

6) если $(s, o, r) \in B$ и $r \notin M[s, o]$, то $(s, o, r) \notin B'$.

Basic Security Theorem утверждает, что система безопасна тогда и только тогда, когда начальное состояние системы безопасно, и для всех переходов системы в последующие состояния выполнены шесть вышеперечисленных условий.

Заметим, что ss-свойство следует из *-свойства, что вызвано историческими причинами [4]. В первоначальной работе Белла и ЛаПадулы *-свойство отсутствовало, и было введено позднее, чтобы избавить систему, защищённую по БЛП, от уязвимости к атакам вида «троянский конь».

Исходя из тех соображений, что право доступа $write$ является комбинированным, по сути совмещая в себе права как на чтение объекта, так и на запись в объект, в то время как $read$ подразумевает доступ только на чтение, $append$ — только на запись, а $execute$ вообще не подразумевает прямого доступа субъекта к данным, мы можем заменить четыре права доступа четырьмя комбинациями всего двух пар доступа, $read$ (только чтение) и $write$ (только запись). В этом случае *-свойство можно сформулировать заметно проще:

Доступ $b = (s, o, r) \in B$ обладает *-свойством относительно $f = (f_S, f_C, f_O)$, если

1. $r = read$ и $f_O(o) \leqslant f_S(s)$,
2. $r = write$ и $f_S(s) \leqslant f_O(o)$.

Что касается иерархии H, то будем следовать замечанию Белла, изложенному в [4], что естественно полагать, что субъект, имеющий доступ к подпапке, имеет доступ и к документам самой папки, а субъект, не имеющий доступа к папке, не сможет просматривать и вложенные в неё подпапки. Это означает, что если $read \in M[s, o]$ и $f_H(o) \geqslant f_H(o')$, то $read \in M[s, o']$, и наоборот, если $write \in M[s, o]$ и $f_H(o) \leqslant f_H(o')$, то $write \in M[s, o']$.

3. Связь между моделями Белла-ЛаПадулы (МБЛ) и OOHRU

Основной результат данной работы заключается в том, что субъектно-объектная модель Белла-ЛаПадулы (МБЛ) может быть реализована объектно-ориентированной моделью OOHRU.

Рассмотрим два принципиально отличающихся случая, для каждого сформулируем и докажем отдельную теорему.

Будем называть МБЛ ds-свободной, если матрица доступов не накладывает дополнительных (относительно *-свойства) ограничений на доступ субъектов к объектам.

МБЛ однозначно определяется множеством своих состояний (включая начальное) и способов перехода из одного состояния в другое, то есть набором команд из множеств Q и Γ, описанных выше. Поэтому в каждый момент времени t допустимо рассматривать такую модель Σ как набор $(D(t), \Gamma)$, где D — множество состояний системы, Γ — набор команд системы, для которых выполнены условия Basic Security Theorem. В худшем случае множество $\mathrm{B}(t)$ текущих доступов может содержать всевозможные доступы, не противоречащие безопасности состояния $d(t)$ системы в момент времени t. Будем считать,

что $\mathrm{B}(t)$ именно таково, что избавляет нас от рассмотрения запросов на доступ из множества Q.

С другой стороны, связанную с МБЛ объектно-ориентированную модель Σ' будем рассматривать как набор $(D'(t), \Gamma')$, где $D'(t) = (\mathbf{O'}(\mathbf{t}), M'(t), \mathbf{K}, F', R)$ — множество состояний системы, $\mathbf{O'}$ — множество объектов системы, M' — множество прав доступа, оформленное в виде матрицы доступов, $\mathbf{K}$ — множество классов системы (возможно, в виде иерархии; зависит от L и H), F' — отображение классов системы на решётку ценностей, R — множество видов доступа, наконец, Γ' — набор команд системы (зависит от L, f_S, H).

Будем говорить, что объектно-ориентированная модель Σ' реализует субъектно-объектную МБЛ Σ, если существует взаимно-однозначное отображение ϕ, определённое на каждом из элементов модели Σ, устанавливающее соответствие между состояниями и командами систем Σ и Σ', такое что любому переходу системы Σ из состояния d в состояние d', совершаемому в результате выполнения команды γ, соответствует переход системы Σ' из $\phi(d)$ в состояние $\phi(d')$ в результате выполнения команды $\phi(\gamma)$.

Теорема 1. *Для любой безопасной ds-свободной МБЛ существует реализующая её иерархическая модель OOHRU.*

Доказательство. Построим искомую систему OOHRU и одновременно — требуемое отображение ϕ.

Во-первых, в МБЛ будем рассматривать только два вида доступа, $read$ и $write$. Причина тому была приведена в конце предыдущего параграфа. Соответственно, в OOHRU сохранятся те же виды доступа.

Во-вторых, иерархию классов в OOHRU будем строить на основе решётки L ценностей и иерархии H уровней секретности объектов.

Введём множество служебных классов $K' = \{k^0, k^{read}, k^{write}, k^{system}\}$. Класс k^0 — корневой и не обладает никакими правами доступа, в то время как объекты других классов обладают полным множеством прав доступа к нему. Класс k^{read} обладает полным набором прав чтения, но полностью лишён прав записи. Класс k^{write}, напротив, обладает полным набором прав записи, но полностью лишён прав чтения. Эти три класса могут не содержать объектов, либо методы данных классов лишены функциональности, а поля — подлежащей защите информации. Последний класс k^{system} обладает полным набором прав как по записи, так и по чтению, и содержит доверенный объект — объект администратора системы.

Объекты в конструируемой объектно-ориентированной системе будут принадлежать к одному из четырёх типов: содержащие единственное поле $field$, в которое можно писать, содержащие единственное поле $field$, которое можно читать, содержащие единственный метод $read$ с правом доступа по чтению, содержащие единственный метод $write$ с правом доступа по записи. Данные ограничения необходимы только для доказательства, в реальной системе можно обойтись и без них.

Кроме того, каждому уровню секретности объектов модели Белла-ЛаПадулы сопоставим два класса в объектно-ориентированной модели, а каж-

дому уровню допуска субъектов модели Белла-ЛаПадулы — два семейства классов. В рамках первого семейства представлены классы, позволяющие реализовать возможности субъектов мандатной модели по чтению, в рамках второго — по записи. Данное сопоставление реализуем в виде отображения $F\text{'} : \mathbf{K} \setminus \mathbf{K'} \to L \cup H$, где соблюдаются следующие разбиения:

$$\mathbf{K} = \mathbf{K'} \cup \mathbf{KOR} \cup \mathbf{KOW} \cup \mathbf{KSR} \cup \mathbf{KSW},$$

так что $F\text{'} : \mathbf{KOR} \cup \mathbf{KOW} \to H$ и $F\text{'} : \mathbf{KSR} \cup \mathbf{KSW} \to L$.

$$\mathbf{KOR} = \bigcup_{h \in H} k_h^{OR},$$

$$\mathbf{KOW} = \bigcup_{h \in H} k_h^{OW},$$

$$\mathbf{KSR} = \bigcup_{l \in L} \mathbf{K}_l^{SR},$$

$$\mathbf{KSW} = \bigcup_{l \in L} \mathbf{K}_l^{SW},$$

так что $F\text{'}(k_h^{OR}) = F\text{'}(k_h^{OW}) = h$ и $F\text{'}(\mathbf{K}_l^{RS}) = F\text{'}(\mathbf{K}_l^{SW}) = l$,

$$\mathbf{K}_l^{SR} = \bigcup_{s \in \mathbf{S}} k_{(l,s)}^{read},$$

$$\mathbf{K}_l^{SW} = \bigcup_{s \in \mathbf{S}} k_{(l,s)}^{write},$$

где $\mathbf{S}$ — множество субъектов в модели Белла-ЛаПадулы.

Здесь $k_{(l,s)}^{read}$, например, обозначает класс, находящийся в иерархии на уровне l, в котором может быть создан объект $o_{(l,s)}^{read}$ с единственным методом, реализующий функционал субъекта МБЛ $s \in \mathbf{S}$ по чтению. И такой объект существует в модели OOHRU тогда и только тогда, когда соответствующий субъект в МБЛ находится на уровне безопасности l, т.е. $f_C(s) = l \leqslant f_S(s)$. Одновременно с ним существует и объект $o_{(l,s)}^{write}$ класса $k_{(l,s)}^{write}$, реализующий функционал субъекта $s \in \mathbf{S}$ по записи. Значение $f_S(s)$ для каждого субъекта при этом закладывается на уровне создания системы.

В свою очередь, k_h^{OR} обозначает класс, находящийся в иерархии на уровне h, в котором может быть создан объект $o_{(h,o)}^{OR}$ с единственным полем, содержащим информацию объекта МБЛ $o \in \mathbf{O}$, предназначенную для чтения. И такой объект существует в модели OOHRU тогда и только тогда, когда соответствующий объект в МБЛ находится на уровне иерархии h, т.е. $f_H(o) = h$. Одновременно с ним существует и объект $o_{(h,o)}^{OW}$ класса k_h^{OW}, находящийся на том же уровне h иерархии и содержащий идентичную информацию, однако, к этому объекту могут обращаться только объекты из семейства классов KSW для выполнения операции записи.

Таким образом, $o^{read}_{(l,s)}$ может получить право читать информацию из $o^{OR}_{(h,o)}$, но не из $o^{OW}_{(h,o)}$, в то время как соответствующий тому же субъекту МБЛ s объект $o^{write}_{(l,s)}$ может обладать правом писать в $o^{OW}_{(h,o)}$, но не в $o^{OR}_{(h,o)}$. По завершении операции записи в объект $o^{OW}_{h,o}$, его содержание копируется системным объектом в $o^{OR}_{h,o}$.

Отображение F' таково, что для любых двух уровней безопасности $l, m \in L$, $l \leqslant m$, верно:

$k^0 \leqslant k^{read}_{(l,s)} \leqslant k^{read}_{(m,s)} \leqslant k^{read}$, но

$k^0 \leqslant k^{write}_{(m,s)} \leqslant k^{write}_{(l,s)} \leqslant k^{write}$.

Кроме того, для любых элементов $h \leqslant g$ решётки H выполнено:

$k^0 \leqslant k^{OR}_h \leqslant k^{OR}_g$, но

$k^0 \leqslant k^{OW}_g \leqslant k^{OW}_h$.

Наконец, $k^0 \leqslant k^{system}$.

Матрицы доступа методов этих объектов также индуцируются матрицей доступа субъекта s. Для ds-свободных МБЛ это означает, что если $f_C(s) = l$, то

$read \in o^{OR}_{(h,o)}.M[o^{read}_{(l,s)}, o^{OR}_{(h,o)}.field] \Leftrightarrow f_O(o) \leqslant l$,

$write \in o^{OW}_{(h,o)}.M[o^{write}_{(l,s)}, o^{OW}_{(h,o)}.field] \Leftrightarrow f_O(o) \geqslant l$.

Для завершения доказательства нам достаточно представить реализацию команд МБЛ средствами модели OOHRU. С учётом того, что в ds-свободной МБЛ матрица доступов не претерпевает изменений, если уровни допуска субъектов и уровни секретности объектов не изменяются, достаточно представить только команды, соответствующие изменению функций f_H, f_C и f_O, а также созданию и удалению объектов. Условные части команд отсутствуют, поскольку для соблюдения *-свойства достаточно условий целостности из элементарных операторов модели OOHRU.

1. Присвоение субъекту s с текущим уровнем допуска l нового уровня допуска m.

Команда $ChangeSubjectSecurityLevel[l, m](o^{read}_{(l,s)} : k^{read}_{(l,s)}; o^{write}_{(l,s)} : k^{write}_{(l,s)}; o^{read}_{(m,s)} : k^{read}_{(m,s)}; o^{write}_{(m,s)} : k^{write}_{(m,s)})$

$Create(o^{read}_{(m,s)}, k^{read}_{(m,s)})$,

$Create(o^{write}_{(m,s)}, k^{write}_{(m,s)})$,

$Destroy(o^{read}_{(l,s)})$,

$Destroy(o^{write}_{(l,s)})$.

Данная команда присутствует в системе только для значений $l, m \leqslant f_S(s)$.

2. Присвоение объекту o с текущим уровнем секретности h нового уровня секретности g.

Команда $ChangeObjectSecurityLevel[h, g](o^{OR}_{(h,o)} : k^{OR}_{(h,o)}; o^{OR}_{(g,o)} : k^{OR}_{(g,o)}; o^{OW}_{(h,o)} : k^{OW}_{(h,o)}; o^{OW}_{(g,o)} : k^{OW}_{(g,o)}; o^{system} : k^{system})$

$Create(o^{OR}_{(g,o)}, k^{OR}_{(g,o)})$,

$Create(o^{OW}_{(g,o)}, k^{OW}_{(g,o)})$,

$Enter(read, o^{system}, o^{OR}_{(h,o)})$,

$Enter(write, o^{system}, o^{OR}_{(g,o)})$,

$Enter(write, o^{system}, o^{OW}_{(g,o)})$,
$Destroy(o^{OR}_{(h,o)})$,
$Destroy(o^{OW}_{(h,o)})$.

Здесь доступ системного объекта к перемещаемому объекту необходим для того, чтобы можно было скопировать информацию. В МБЛ подобные операции могут быть осуществлены только доверенным субъектом, роль которого в OOHRU отведена объекту системного класса.

3. Создание объекта o уровня секретности h.
Команда $CreateObject[h](o^{OR}_{(h,o)} : k^{OR}_{(h,o)}; o^{OW}_{(h,o)} : k^{OW}_{(h,o)}; o^{system} : k^{system})$
$Create(o^{OR}_{(h,o)}, k^{OR}_{(h,o)})$,
$Create(o^{OW}_{(h,o)}, k^{OW}_{(h,o)})$,
$Enter(write, o^{system}, o^{OR}_{(h,o)})$,
$Enter(write, o^{system}, o^{OW}_{(h,o)})$.

4. Удаление объекта o.
Команда $DestroyObject[h](o^{OR}_{(h,o)} : k^{OR}_{(h,o)}; o^{OW}_{(h,o)} : k^{OW}_{(h,o)})$
$Destroy(o^{OR}_{(h,o)})$,
$Destroy(o^{OW}_{(h,o)})$.

При этом условие Белла [4] «вместе с папкой удаляются и содержащиеся в ней подпапки» можно реализовать цепочкой команд, удаляющих объекты, являющиеся потомками удаляемого, либо представив папку с вложенными подпапками средствами одного объекта. При создании объекта возможности доступов этого объекта к другим объектам, а также других объектов к этому объекту, совпадают с соответствующими возможностями класса этого объекта, то есть согласуются исключительно с *-свойством и никогда не изменяются.

■

Замечание 1. Доказанное утверждение справедливо не только для ds-свободных МБЛ, а вообще для всех, наследующих права доступа, то есть обладающих следующим свойством:
$read \in M[s, o]$и$f_C(s) \leqslant f_C(s') \Rightarrow read \in M[s', o]$,
$read \in M[s, o]$и$f_H(o) \geqslant f_H(o') \Rightarrow read \in M[s, o']$,
$write \in M[s, o]$и$f_C(s) \geqslant f_C(s') \Rightarrow write \in M[s', o]$,
$write \in M[s, o]$и$f_H(o) \leqslant f_H(o') \Rightarrow write \in M[s, o']$.

Замечание 2. Если множество субъектов в системе не является известным заранее, утверждение теоремы остаётся верным с учётом небольшого изменения в доказательстве. А именно, вместо того, чтобы сопоставлять каждому субъекту решётку классов, будем пользоваться единой решёткой классов для всех субъектов. То есть вместо класса $k^{read}_{(l,s)}$, соответствовавшего субъекту s, используем класс k^{read}_{l}, единый для всех субъектов. Сопоставление каждому субъекту классов собственной решётки необходимо для ситуации, в которой один субъект может получить право на активизацию другого субъекта, однако, в МБЛ право execute применяется по отношению к объектам.

Замечание 3. При необходимости выделение индивидуальной иерархии классов для каждого объекта МБЛ также возможно. Например, если объект имеет достаточно сложную структуру, а не ограничивается единственным полем. В этом случае разбиение $\mathbf{KOR} \cup \mathbf{KOW}$ на классы полностью аналогично тому, что было применено в доказательстве теоремы при реализации средствами OOHRU субъектов МБЛ.

Замечание 4. Модель MLS с точки зрения МБЛ является безопасной ds-свободной, поскольку каждое её состояние обладает *-свойством (в его упрощённой формулировке), и лишена дополнительных ограничений на доступ в виде матрицы доступа. А значит, утверждение теоремы 1 справедливо и для MLS.

Замечание 5. Доказательство теоремы 1 возможно провести и другим способом, без выделения отдельных классов, отвечающих за чтение и запись. Например, можно использовать конструкцию, в которой отсутствие права чтения интерпретируется как наличие права записи (с определёнными оговорками).

Теорема 2. *Для любой безопасной МБЛ существует реализующая её модель OOHRU.*

Доказательство. Доказательство этой теоремы в целом повторяет доказательство теоремы 1 за рядом отличий. Ограничимся только перечислением этих отличий.

Во-первых, иначе происходят разбиения классов:

$$\mathbf{K} = \mathbf{K}' \cup \mathbf{KO} \cup \mathbf{KS},$$

так что $F\text{'} : \mathbf{KO} \to H$ и $F\text{'} : \mathbf{KS} \to L$.

$$\mathbf{KO} = \bigcup_{h \in H} \mathbf{K}_h^O,$$

$$\mathbf{KS} = \bigcup_{l \in L} \mathbf{K}_l^S,$$

так что $F\text{'}(\mathbf{K}_h^O) = h$ и $F\text{'}(\mathbf{K}_l^S) = l$,

$$\mathbf{K}_h^O = \bigcup_{o \in \mathbf{O}} k_{(h,o)},$$

$$\mathbf{K}_l^S = \bigcup_{s \in \mathbf{S}} k_{(l,s)},$$

где $\mathbf{O}$ — множество объектов, а $\mathbf{S}$ — множество субъектов в модели Белла-ЛаПадулы.

Здесь $k_{(l,s)}$, например, обозначает класс, находящийся в иерархии на уровне l, в котором может быть создан объект $o_{(l,s)}$ с двумя методами, реализующими функционал субъекта МБЛ $s \in \mathbf{S}$ по чтению и по записи соответственно. И такой объект существует в модели OOHRU тогда и только тогда, когда

соответствующий субъект в МБЛ находится на уровне безопасности l, т.е. $f_C(s) = l \leqslant f_S(s)$.

В свою очередь, $k_{(h,o)}$ обозначает класс, находящийся в иерархии на уровне h, в котором может быть создан объект $o_{(h,o)}$ с единственным полем, содержащим информацию объекта МБЛ $o \in \mathbf{O}$. И такой объект существует в модели OOHRU тогда и только тогда, когда соответствующий объект в МБЛ находится на уровне иерархии h, т.е. $f_H(o) = h$.

Таким образом, для каждого объекта МБЛ строится одно семейство классов в OOHRU, параметризованное уровнями секретности, а не два (отдельно для чтения и записи), как в доказательстве теоремы 1.

Во-вторых, модель OOHRU не будет обладать иерархией: даже если $l = F'(k_{(l,s)}) \leqslant F'(k_{(m,s)}) = m$, из этого не следует, что $k_{(l,s)} \leqslant k_{(m,s)}$.

В-третьих, матрицы доступа методов этих объектов также индуцируются матрицей доступа субъекта s. Это означает, что если $f_C(s) = l$, то

$$read \in o_{(h,o)}.M[o_{(l,s)}, o_{(h,o)}.field] \Leftrightarrow read \in M[s,o] \text{и} f_O(o) \leqslant l,$$

$$write \in o_{(h,o)}.M[o_{(l,s)}, o_{(h,o)}.field] \Leftrightarrow write \in M[s,o] \text{и} f_O(o) \geqslant l.$$

Наконец, список команд теперь имеет следующий вид:

1. Добавление права на чтение субъектом s уровня допуска l объекта o уровня секретности h.

Команда $EnterRead[l,h](o_{(l,s)} : k_{(l,s)}; o_{(h,o)} : k_{(h,o)})$

$Enter(read, o_{(l,s)}, o_{(h,o)}.field)$.

Такие команды существуют для всех l и h таких, что h не превосходит l, то есть для любого объекта o такого, что $f_H(o) = h$, выполняется $f_O(o) \leqslant l$.

2. Удаление права на чтение субъектом s уровня допуска l объекта o уровня секретности h.

Команда $DeleteRead[l,h](o_{(l,s)} : k_{(l,s)}; o_{(h,o)} : k_{(h,o)})$

$Delete(read, o_{(l,s)}, o_{(h,o)}.field)$.

3. Добавление права на запись субъектом s уровня допуска l в объект o уровня секретности h.

Команда $EnterRead[l,h](o_{(l,s)} : k_{(l,s)}; o_{(h,o)} : k_{(h,o)})$

$Enter(write, o_{(l,s)}, o_{(h,o)}.field)$.

Такие команды существуют для всех l и h таких, что h не меньше l, то есть для любого объекта o такого, что $f_H(o) = h$, выполняется $f_O(o) \geqslant l$.

4. Удаление права на запись субъектом s уровня допуска l в объект o уровня секретности h.

Команда $DeleteRead[l,h](o_{(l,s)} : k_{(l,s)}; o_{(h,o)} : k_{(h,o)})$

$Delete(write, o_{(l,s)}, o_{(h,o)}.field)$.

5. Присвоение субъекту s с текущим уровнем допуска l нового уровня допуска m.

Команда $ChangeSubjectSecurityLevel[l,m,\alpha](o_{(l,s)} : k_{(l,s)}; o_{(m,s)} : k_{(m,s)})$

$if \alpha \in o.M[o_{(l,s)}, o.field]$

$Create(o_{(m,s)}, k_{(m,s)})$,

$Enter(\alpha, o_{(m,s)}, o)$,

$Destroy(o_{(l,s)})$.

Данная команда присутствует в системе только для значений $l, m \leqslant f_S(s)$. α — набор прав, которыми обладает объект $o_{(l,s)}$ на поля других объектов o. Мы должны сохранить этот набор при изменении уровня допуска субъекта s в МБЛ. Для каждого набора прав α существует своя команда, и при переносе объекта с набором прав, в точности совпадающим с α, в другой класс в OOHRU из множества команд выбирается соответствующая.

6. Присвоение объекту o с текущим уровнем секретности h нового уровня секретности g.

Команда $ChangeObjectSecurityLevel[h, g, \alpha](o_{(h,o)} : k_{(h,o)}; o_{(g,o)} : k_{(g,o)}; o^{system} : k^{system})$

$if \alpha \in o_{(h,o)}.M[s, o_{(h,o)}.field]$

$Create(o_{(g,o)}, k_{(g,o)})$,

$Enter(read, osystem, o_{(h,o)})$,

$Enter(write, osystem, o_{(g,o)})$,

$Enter(\alpha, s, o_{(g,o)})$

$Destroy(o_{(h,o)})$.

Здесь α — это множество прав, которыми обладают объекты s на поле объекта $o_{(h,o)}$. Мы должны сохранить этот набор при изменении уровня секретности объекта o в МБЛ.

7. Создание объекта o уровня секретности h.

Команда $CreateObject[h](o_{(h,o)} : k_{(h,o)}; o^{system} : k^{system})$

$Create(o_{(h,o)}, k_{(h,o)})$,

$Enter(write, o^{system}, o_{(h,o)})$.

8. Удаление объекта o.

Команда $DestroyObject[h](o_{(h,o)} : k_{(h,o)})$

$Destroy(o_{(h,o)})$.

■

О безопасности МБЛ.

Важно отметить, что в обоих случаях мы построили дискреционную модель, безопасную с точки зрения МБЛ, однако, возможность проверки её безопасности с точки зрения дискреционных политик безопасности не установлена.

С другой стороны, Basic Security Theorem неоднократно подвергалась критике, в том числе и со стороны МакЛина [5], по ряду причин. Так, фактически Basic Security Theorem лишь утверждает выполнение определённого ряда свойств, но каким образом эти свойства влияют на фактическую безопасность системы, не ясно. Хуже того, Basic Security Theorem допускает деклассификацию всех субъектов и объектов до самого низкого уровня секретности без нарушения определения безопасности, поэтому МакЛин предложил определять безопасность системы не с точки зрения состояний, а с точки зрения переходов между состояниями.

Таким образом, построенные нами модели OOHRU, реализующие безопасные модели Белла-ЛаПадулы, также отнести к безопасным относительно утечки права доступа было бы преждевременно.

Литература

1. Усов С.В. Неоднородные объектно-ориентированные модели с иерархией // Проблемы обработки и защиты информации. Книга 3. Модели разграничения доступа. Коллективная монография / Под общей редакцией С.В. Белима. Омск : ООО «Полиграфический центр КАН», 2013. С. 93–114.
2. Harrison M.A., Ruzzo W.L., Ulman J.D. Protection in Operating Systems // Communications of the ACM. 1975. P. 14–25.
3. Bell, David Elliott and LaPadula, Leonard J. Secure Computer System: Unified Exposition and Multics Interpretation. Technical report 2997, rev. 1. MITRE, 1996.
4. Bell, David Elliott. Looking Back at the Bell-LaPadula Model // 21st Annual Computer Security Applications Conference. Tucson, Arizona, USA, 2005. P. 337–351.
5. McLean J. The Specification and Modeling of Computer Security // Computer. 1990. N. 23(1). P. 9–16.

ON THE RELATION BETWEEN THE OBJECT-ORIENTED DISCRETIONARY SECURITY MODEL AND THE SUBJECT-OBJECT MANDATORY MODEL

S.V. Usov
Ph.D. (Phys.-Math.), e-mail: raintower@mail.ru

Omsk State University n.a. F.M. Dostoevskiy

Abstract. The article deals with object-oriented Harrison-Ruzzo-Ullman access control model and subject-object model with mandatory access control. It is shown that the Bell-LaPadula model and classic mandatory model can be implemented with object-oriented HRU model.

Keywords: access control, discretionary safety models, mandatory security models, HRU, Bell-LaPadula model.

Дата поступления в редакцию: 31.10.2016