Отношение взаимоисключения на множестве ролей в моделях управления доступом

Н.Ф. Богаченко
nfbogachenko@mail.ru

Омский государственный университет им. Ф.М. Достоевского, Омск, Россия

Аннотация

В данной работе подробно рассмотрено наиболее распространенное ограничение модели RBAC_2 – «взаимоисключающие роли». Это ограничение интерпретировано в терминах бинарных отношений. Исследованы свойства построенного отношения. В частности, рассмотрены варианты транзитивного и нетранзитивного отношений. В качестве языка описания ограничения «взаимоисключающие роли» предложено использовать XML-подобный язык GraphML.

Введение

Основные элементы математической модели ролевого управления доступом (Role Base Access Control, RBAC) изложены в работах [1, 2, 4, 5]. Принято говорить о семействе ролевых моделей. Модель RBAC_0 – базовая модель, предъявляющая минимум требований к системе, которая может поддерживать ролевое разграничение доступа. RBAC_1 включает требования модели RBAC_0 и концепцию иерархии ролей. RBAC_2 содержит требования модели RBAC_0 и ограничения, налагаемые на различные компоненты модели. Например, взаимоисключающие роли, количественные ограничения по ролям, концепция условных ролей и т.д. RBAC_3 – объединяющая модель, включает требования моделей RBAC_1 и RBAC_2 и, по транзитивности, модели RBAC_0. Кроме обычных ролей в систему могут быть введены административные роли, что приводит к возможности описания систем с изменяемой защитой [3].

В данной работе подробно рассматривается наиболее распространенное ограничение модели RBAC_2 – «взаимоисключающие роли». Например в современных системах управления ресурсами предприятия (Enterprise Resource Planning Systems, ERP-системах) понятие взаимоисключающих ролей введено в рамках RBAC-модели управления доступом для реализации концепции разделения полномочий.

1 Бинарные отношения модели RBAC_2

Организация ролевой системы управления доступом состоит из двух этапов:

1. Выделение множества ролей и определение их полномочий.
2. Назначение ролей пользователям системы.

Для формального описания модели RBAC_0 необходимо определить следующие множества и отображения:

- Множество ролей R, множество полномочий P, множество пользователей U.

In: Sergey V. Belim, Nadezda F. Bogachenko (eds.): Proceedings of the Workshop on Data Analysis and Modelling (DAM 2016), Omsk, Russia, October 2016, published at http://ceur-ws.org

- Отображение $RP : R \longrightarrow 2^P$, которое каждой роли r сопоставляет множество полномочий $r.p \subseteq P$.
- Отображение $UR : U \longrightarrow 2^R$, которое каждому пользователю u сопоставляет множество ролей $u.r \subseteq R$, на эти роли пользователь может быть авторизован.

Нетрудно заметить, что отображения RP и UR соответствуют первому и второму этапам построения модели RBAC_0.

Управление доступом осуществляется при помощи отображения $F_{\text{session roles}} : U \longrightarrow 2^R$, которое пользователю u сопоставляет подмножество ролей $u.sr \subseteq u.r \subseteq R$, на эти роли пользователь может быть авторизован в текущем сеансе работы с системой.

В рамках модели RBAC_2 ограничение «взаимоисключающие роли» как правило представляет собой задание специального отображения $F_{\text{mutually exclusive}} : R \longrightarrow 2^R$, которое каждой роли r сопоставляет подмножество «несовместных» с ней ролей $r.mer \subseteq R$ [1]. Различают статический и динамический методы распределения обязанностей. В первом случае отображение $F_{\text{mutually exclusive}}$ накладывает ограничения на отображение UR:

$$(r \in u.r) \wedge (r' \in r.mer) \Longrightarrow (r' \notin u.r), \tag{1}$$

во втором – на отображения $F_{\text{session roles}}$:

$$(r \in u.sr) \wedge (r' \in r.mer) \Longrightarrow (r' \notin u.sr). \tag{2}$$

Другими словами, пользователь u, авторизованный на роль r, не может быть одновременно авторизован на «несовместную» с ней роль r' ни на этапе организации ролевой модели (ограничение (1)), ни в процессе управления доступом (ограничение (2)).

В дальнейшем будем рассматривать динамический метод. Тем не менее последующие рассуждения очевидным образом переносятся и на статический случай. Ограничение (2) можно интерпретировать в терминах бинарных отношений: пусть на множестве ролей R задано бинарное отношение *взаимоисключения* «$\nleftrightarrow$»:

$$r \nleftrightarrow r' \Longleftrightarrow (r \in u.sr) \wedge (r' \notin u.sr).$$

Очевидно, что это отношение обладает следующими свойствами:

- Антирефлексивность: $\forall r \in R : \neg(r \nleftrightarrow r)$, так как $\neg((r \in u.sr) \wedge (r \notin u.sr))$.
- Симметричность: $\forall r, r' \in R : (r \nleftrightarrow r') \Longrightarrow (r' \nleftrightarrow r)$, в силу коммутативности конъюнкции.

Если считать, что авторизация пользователя осуществляется по принципу «все, что не запрещено – разрешено», то дополнением к отношению взаимоисключения будет являться отношение *совместимости* «$\leftrightarrow$»:

$$r \leftrightarrow r' \Longleftrightarrow (r \in u.sr) \wedge (r' \in u.sr).$$

Действительно, несложно понять, что $\neg(r \nleftrightarrow r') \Longleftrightarrow (r \leftrightarrow r')$. Свойства, которыми очевидным образом обладает отношение совместимости:

- Рефлексивность: $\forall r \in R : (r \leftrightarrow r)$.
- Симметричность: $\forall r, r' \in R : (r \leftrightarrow r') \Longrightarrow (r' \leftrightarrow r)$.

Таким образом достаточно задать одно из отношений: совместимость или взаимоисключение, тогда второе отношение строится как дополнение заданного до универсума – декартова произведения $R \times R$.

Так как множество ролей конечно, то для задания бинарных отношений можно использовать графовое представление. Отношению $\mathcal{R}$ сопоставляется ориентированный граф $G = G(V, E)$, в котором множество вершин V совпадает с множеством ролей R, а множество дуг E определяется следующим правилом: $(u, v) \in E \Longleftrightarrow u\mathcal{R}v$. Такой граф допускает наличие петель. Если отношение $\mathcal{R}$ является симметричным, то ориентированный граф можно заменить неориентированным.

Граф отношения взаимоисключения обозначим $G^{\nleftrightarrow}$. Несложно понять, что отображение $F_{\text{session roles}}$ связано с графом $G^{\nleftrightarrow}$ правилом: пользователь u не может быть авторизован на смежные вершины-роли в одном сеансе работы в системе.

На этом этапе возникает задача поиска подмножества ролей, на которые одновременно может быть авторизован пользователь u. С точки зрения теоретико-графовой модели необходимо найти независимое

множество вершин (множество, в котором никакие две вершины не смежны) правильного подграфа графа $G^{\nleftrightarrow}$, порожденного множеством вершин $u.sr$. Точный алгоритм поиска всех независимых множеств вершин представляет собой полный перебор элементов булеана множества $u.sr$ и имеет трудоемкость $\mathrm{O}(n^2 \cdot 2^n)$. Чаще задача формулируется в оптимизационной постановке: требуется найти максимальное независимое множество вершин. Трудоемкость точного алгоритма остается прежней, так как и в этом случае требуется организовать полный перебор. Если же возможно применение приближенного алгоритма, то используется «жадная» стратегия: на каждом шаге выбирается вершина с минимальной степенью и удаляются смежные с ней вершины. Известно, что эта эвристика имеет трудоемкость $O(n)$ при условии, что граф на вход подается в виде списка смежности и вершины упорядочены по неубыванию степеней.

2 Свойство транзитивности отношения взаимоисключения

Одним из принципиальных вопросов, которые следует обсудить при построении отношения взаимоисключения – это свойство транзитивности. Рассмотрим сначала традиционный подход, при котором отношение взаимоисключения считается транзитивным: $\forall r_1, r_2, r_3 \in R : (r_1 \nleftrightarrow r_2) \wedge (r_2 \nleftrightarrow r_3) \Longrightarrow (r_1 \nleftrightarrow r_3)$. Если искусственно изменить свойство антирефлексивности отношения взаимоисключения на рефлексивность (что принципиально не скажется на решаемой задаче авторизации пользователя), тогда отношение взаимоисключения будет отношением эквивалентности на множестве ролей. В каждый класс эквивалентности попадут роли попарно несовместные, а граф $G^{\nleftrightarrow}$ будет обладать следующим свойством: каждая его компонента связности – полный подграф. Возвращение к требованию антирефлексивности приведет к тому, что в графе $G^{\nleftrightarrow}$ исчезнут петли.

Удобным способом представления бинарного отношения на конечном множестве является матричная форма. Пусть $|R| = n$. Тогда матрица $\mathbf{M}^{\nleftrightarrow}$ отношения взаимоисключения имеет размерность $n \times n$ и определяется по правилу:

$$[\mathbf{M}^{\nleftrightarrow}]_{ij} = \begin{cases} 1, & \text{если } r_i \nleftrightarrow r_j, \\ 0, & \text{иначе.} \end{cases}$$

Несложно заметить, что матрица бинарного отношения есть матрица смежности его графового представления.

Утверждение 1. *Трудоемкость алгоритма проверки того факта, что неориентированный граф порядка n является графом некоторого транзитивного отношения взаимоисключения, не превосходит* $\mathrm{O}(n^3)$.

Доказательство. Пусть неориентированный граф G задан матрицей смежности $\mathbf{M}$ размерности $n \times n$. Необходимо проверить, является ли отношение $\mathcal{R}$, порождаемое ребрами графа G, антирефлексивным, симметричным и транзитивным. Отношение $\mathcal{R}$ будет антирефлексивным, если на главной диагонали матрицы смежности стоят нули. Трудоемкость проверки этого факта равна $\mathrm{O}(n)$. Симметричность отношения $\mathcal{R}$ следует из симметричности матрицы смежности неориентированного графа. Чтобы проверить транзитивность отношения $\mathcal{R}$ необходимо построить матрицу $\mathbf{M}^* = \mathbf{M} \circ \mathbf{M}$, где под операцией «$\circ$» подразумевается такое произведение матриц, при котором умножение элементов соответствует логической операции конъюнкция, а сложение элементов – логической операции дизъюнкция. Отношение $\mathcal{R}$ является транзитивным, если $[\mathbf{M}^*]_{ij} \leqslant [\mathbf{M}]_{ij}$ $(i, j \in \{1, ..., n\})$. Таким образом трудоемкость проверки транзитивности определяется трудоемкостью алгоритма произведения двух матриц и не превосходит $\mathrm{O}(n^3)$. В итоге трудоемкость алгоритма анализа графа G также не превосходит $\mathrm{O}(n^3)$. ■

Одной из объективных причин построения отношения взаимоисключения является возможность получения пользователем недопустимого, с точки зрения безопасности системы, набора ролей или полномочий. В частности, нельзя в одном сеансе работы с системой совмещать роли администратора и аудитора безопасности, нельзя занимать сразу две должности по основному месту работы, нельзя одновременно разрешать доступ к ценной информации и запуск недоверенных процессов, нельзя в финансовых процедурах совмещать роли контролера-аудитора и кассира-оператора и т. д. Для дальнейшего анализа отношения взаимоисключения рассмотрим несколько примеров.

Пример 1. Пусть между ролями r_1, r_2 и r_3 полномочия распределены следующим образом: $r_1.p = \{p_1, p_2\}$, $r_2.p = \{p_3, p_4\}$ и $r_3.p = \{p_1, p_3, p_4\}$. И пусть набор полномочий $\{p_1, p_2, p_3, p_4\}$ считается недопустимым с позиций безопасности системы. Рассмотрим пользователя u, который может быть авторизован на роли из набора $u.r = \{r_1, r_2, r_3\}$. Тогда в одном сеансе работы с системой пользователя u нельзя одновременно

авторизовать на роли r_1 и r_2, а также на роли r_1 и r_3, но имеется возможность одновременной авторизации на роли r_2 и r_3. Выпишем все пары ролей, связанных отношением взаимоисключения «$\nleftrightarrow$», учитывая его симметричность и антирефлексивность: $\{(r_1, r_2), (r_2, r_1), (r_1, r_3), (r_3, r_1)\}$. Очевидно, что это отношение не является транзитивным, так как в представленном множестве присутствуют, например, пары (r_3, r_1) и (r_1, r_2), но нет пары (r_3, r_2).

Пример 2. На рисунке 1 представлены стандартные роли службы Reporting Services технологии SQL Server для предоставления доступа к операциям сервера отчетов [6]. С каждой стандартной ролью связан набор задач (полномочий). Для обеспечения всех составляющих информационной безопасности пользователь не может быть одновременно авторизован на роли из множества $\{r_1, r_2, r_3, r_4, r_5\}$ и роли из множества $\{r_6, r_7\}$. Тогда отношение взаимоисключения «$\nleftrightarrow$» задается матрицей $\mathbf{M}^{\nleftrightarrow}$:

$$\mathbf{M}^{\nleftrightarrow} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \tag{3}$$

В силу того, что $[\mathbf{M}^{\nleftrightarrow} \circ \mathbf{M}^{\nleftrightarrow}]_{11} = 1 > 0 = [\mathbf{M}^{\nleftrightarrow}]_{11}$, построенное отношение взаимоисключения не является транзитивным.

Роли / **Полномочия**	r_1	r_2	r_3	r_4	r_5	r_6	r_7
Использование отчетов	1			1			
Создание связанных отчетов	1	1			1		
Управление всеми подписками	1						
Управление источниками данных	1	1			1		
Управление папками	1	1			1		
Управление моделями	1	1					
Управление отдельными подписками	1		1	1	1		
Управление журналом отчета	1						
Управление отчетами	1	1			1		
Управление ресурсами	1	1			1		
Установка политики безопасности для элементов	1						
Просмотр источников данных	1				1		
Просмотр отчетов	1		1	1	1		
Просмотр моделей	1		1	1			
Просмотр ресурсов	1		1	1	1		
Просмотр папок	1		1	1	1		
Выполнение определений отчетов						1	1
Управление заданиями						1	
Управление свойствами сервера отчетов						1	
Управление ролями						1	
Управление общими расписаниями						1	
Управление безопасностью сервера отчетов						1	
Просмотр свойств сервера отчетов							1
Просмотр общих расписаний							1

Рис. 1: Стандартные роли и полномочия службы Reporting Services; r_1 – диспетчер содержимого, r_2 – издатель, r_3 – браузер, r_4 – построитель отчетов, r_5 – мои отчеты, r_6 – системный администратор, r_7 – пользователь системы.

Исходя из вышесказанного следует отказаться от обязательной транзитивности отношения взаимоисключения. В этом случае изменятся и свойства графа $G^{\nleftrightarrow}$: исчезнет требование полноты компонент связности. Трудоемкость, обозначенная в утверждении 1, определяется алгоритмом произведения двух матриц. Если проверка транзитивности не требуется, то трудоемкость анализа графа становится линейной. Антирефлексивность отношения взаимоисключения обеспечивается отсутствием петель в графе. Поэтому справедливо следующее утверждение.

Утверждение 2. *Любой простой неориентированный граф порождает в общем случае нетранзитивное отношение взаимоисключения.*

3 Представление графа отношения взаимоисключения

При программно-технической реализации ограничений модели RBAC_2 возникает подзадача формального описания отношения взаимоисключения. Как отмечалось ранее, матрица отношения взаимоисключения представляет собой матрицу смежности некоторого простого неориентированного графа. Поэтому возможно матричное представление ограничения «взаимоисключающие роли». С другой стороны, исходя из объективных факторов, граф отношения $G^{\nsim}$ будет достаточно разреженным (см. пример 2). В этом случае предпочтительней представление графа списками смежности или списками ребер.

Предлагается для задания графа $G^{\nsim}$ не разрабатывать собственный язык представления графов, а выбрать один из общепризнанных стандартов для представления теоретико-графовых моделей. Это приведет как к сокращению времени разработки, так и к совместимости со многими библиотеками и прикладными программами для работы с графами.

Наиболее распространены следующие механизмы описания графов: DOT (используется в программном средстве визуализации графов Graphviz), GraphML (язык описания графов на основе XML), другие диалекты XML для описания графов (GML, GXL, XGMML), DGML (язык разметки ориентированных графов, используемый в графах архитектуры Visual Studio). Все они, по сути, представляют собой списки вершин и ребер графа.

Для описания графа отношения взаимоисключения предлагается использовать стандарт GraphML [7]. Язык GraphML имеет XML-синтаксис, что обеспечивает его совместимость с другими форматами, основанными на XML. Кроме того, GraphML обладает собственным механизмом расширения, что позволяет хранить дополнительную информацию о вершинах или ребрах графа. Для импорта/экспорта GraphML имеются специальные программные инструменты и библиотеки. Чтобы реализовать собственный GraphML-ридер, можно использовать какой-либо доступный XML-парсер, адаптировав его под свои задачи.

Пример 3. Граф G отношения взаимоисключения определяемого матрицей (3) описывается следующим фрагментом GraphML-документа:

```
<graph id="G" edgedefault="undirected">
    <node id="r1"/>
    <node id="r2"/>
    <node id="r3"/>
    <node id="r4"/>
    <node id="r5"/>
    <node id="r6"/>
    <node id="r7"/>
    <edge source="r1" target="r6"/> <edge source="r1" target="r7"/>
    <edge source="r2" target="r6"/> <edge source="r2" target="r7"/>
    <edge source="r3" target="r6"/> <edge source="r3" target="r7"/>
    <edge source="r4" target="r6"/> <edge source="r4" target="r7"/>
    <edge source="r5" target="r6"/> <edge source="r5" target="r7"/>
</graph>
```

Заключение

Представление ограничения «взаимоисключающие роли» модели RBAC_2 не в виде отображения, а в форме бинарного отношения взаимоисключения на множестве ролей позволяет задать это ограничение простым неориентированным графом. В качестве языка описания такого графа предлагается использовать XML-подобный язык GraphML.

Исследование свойств отношения взаимоисключения позволило отказаться от транзитивности, оставив в качестве обязательных требований симметричность и антирефлексивность.

Благодарности

Автор выражают благодарность С.В. Белиму за полезные замечания и предложения в процессе обсуждения тематики статьи.

Список литературы

[1] D. Ferraiolo, J. Cugini, R. Kuhn. Role-based access control: Features and motivations. In Proceedings of Annual Computer Security Applications Conference, IEEE Computer Society Press, 1995, pp. 249–255.

[2] D.F. Ferraiolo, D.R. Kuhn. Role-Based Access Controls. In Proceedings of 15th National Computer Security Conference, Baltimore MD, 1992, pp. 554–563.

[3] M. Nyanchama, S.L. Osborn. Access Rights Administration inRole-Based Security Systems. In Proceedings of the IFIP WG11.3 Working Conference on Database Security VII, North-Holland, 1994, pp. 37–56.

[4] R. Sandhu, E. Coyne, H. Feinstein, C. Youman. Role Based Access Control: A multidimensional view. In Proceedings of 10th Annual Computer Security Applications Conference, Orlando, 1994, pp. 54–62.

[5] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, C.E. Youman. Role-Based Access Control Models. *IEEE Computer*, 1996, N. 29(2), pp. 38–47.

[6] URL: `https://msdn.microsoft.com/ru-ru/library/ms157363.aspx`.

[7] URL: `http://graphml.graphdrawing.org/primer/graphml-primer.html`.

The Mutual Exclusion Relation on a Set of Roles in Access Control Models

Nadezda F. Bogachenko

The most widespread restriction of the $RBAC_2$ model is the "mutually exclusive roles". This restriction is interpreted in terms of the binary relations. Properties of the constructed relation are studied. In particular, transitive and intransitive relations are considered. The XML-like GraphML language is offered to use as language of the description of restriction "mutually exclusive roles".