Математические структуры и моделирование
2011, вып. 22, с. 114–120

УДК 004.056

ЭЛЕМЕНТАРНЫЕ ОПЕРАТОРЫ ПОСТРОЕНИЯ РОЛЕВОЙ ПОЛИТИКИ БЕЗОПАСНОСТИ

С. В. Белим, Н. Ф. Богаченко

Строится набор элементарных операторов, позволяющий модифицировать дерево ролей и отслеживать передачу прав доступа.

Введение

Ролевое разграничение доступа к информации получило широкое распространение не только в системах управления базами данных, но и в операционных системах. Основное отличие ролевой политики безопасности от мандатной и дискреционной состоит в возможности управления привилегиями. Под привилегией понимается возможность осуществления некоторых действий в системе в целом. Безусловно, для получения привилегии необходимо разрешение на доступ к некоторым служебным объектам системы, которое может регламентироваться любой политикой безопасности. Однако ролевое разграничение доступа позволяет наиболее естественно организовать сопоставление привилегий отдельным пользователям или группам пользователей, а также осуществить наследование привилегий между ролями.

Ролевую политику безопасности принято анализировать исходя из иерархии ролей, наиболее удобным представлением которой являются ориентированные графы, называемые далее ролевыми графами (если исходя из контекста понятно, что речь идёт об ориентированном графе, то орграф будем называть просто графом). Однако на сегодняшний день отсутствует формальный подход, позволяющий на основе некоторого набора формальных операций отслеживать любые преобразования ролевого графа. Целью данной статьи является создание модели ролевого разграничения доступа, исходя из набора примитивных операторов по аналогии с моделью HRU [4, с. 48].

1. Безопасность системы

Будем считать, что в системе существует некоторое множество ролей R, наделённых полномочиями из множества P. Далее всюду будем считать, что множества R и P конечны. Между ролями задана иерархия, определяемая ориентированным графом G. Вершины данного графа соответствуют ролям, а дуги —

Омский государственный университет им. Ф. М. Достоевского.
E-mail: sbelim@omsu.ru

наследованиям. То есть, если в графе присутствует дуга от вершины r к вершине r', значит, роль r авторизована на роль r'. Авторизация одной роли на другую подразумевает полное наследование её привилегий. Очевидно, что роль r может унаследовать только привилегии ролей, связанных с ней ориентированными путями. Более строго, если существует ориентированный путь $p(r, r')$ в графе G от роли-вершины r до роли-вершины r', то роль r наследует привилегии r'. Будем обозначать набор привилегий роли r через $r.p$. Тогда предыдущее утверждение может быть записано следующим образом:

$$\text{if } \exists\, p(r, r') \Rightarrow r.p \supseteq r'.p.$$

Введём обозначение для множества ролей, до которых существует путь от вершины r в ролевом графе G:

$$PR(r) = \left\{ r' \middle| \exists\, p(r, r') \right\}.$$

Как легко понять, это множество тех ролей, от которых привилегии передаются роли r.

Перейдём теперь к вопросам безопасности системы с ролевым разграничением доступа.

Определение 1. Будем считать, что происходит ***утечка привилегии*** p, если роль r получает её несанкционированно.

Определение 2. Система с ролевым разграничением доступа ***безопасна***, если в ней не происходит утечка привилегий.

Следует отметить, что данное выше определение безопасности является алгоритмическим. По сути, система будет безопасной, если возможна реализация алгоритма, следящего за передачей привилегий в системе. Задача проверки безопасности системы сводится к исследованию возможности передачи привилегий по дугам графа. Как было сказано выше, привилегии роли r могут передаваться только от ролей из множества $PR(r)$. То есть роли из множества $PR(r)$ влияют на r.

Определение 3. Будем говорить, что роль r' ***не влияет*** на роль r (обозначение $r' : |r$), если добавление любой привилегии p в множество привилегий $r'.p$ не приводит к изменению множества привилегий $r.p$.

Легко понять, что $(R \setminus PR(r)) : |r$.

Теорема 1. *Если в системе ролевое дерево неизменно, то система является безопасной.*

Доказательство. Для доказательства данной теоремы достаточно показать, что для любой роли r алгоритм поиска множества $PR(r)$ будет конечным, а также будет конечным само множество $PR(r)$. Конечность множества $PR(r)$

вытекает из того, что $PR(r) \subseteq R$, а множество R конечно. Поиск же множества $PR(r)$ может быть осуществлён с помощью построения матрицы достижимости [6, с. 176] для ориентированного графа G и имеет полиномиальную сложность [5, с. 48]. Таким образом строится множество $PR(r)$, которое является конечным и неизменным, а затем осуществляется контроль за тем, чтобы нелегитимная привилегия не была внесена в список привилегий как самой роли r, так и всех ролей из множества $PR(r)$, что может быть осуществлено прямым перебором за конечное число шагов. ■

Определение 4. Подграф $GR(r)$ ролевого графа G, вершинами которого является множество $PR(r)$, а дугами — соответствующие дуги между этими вершинами из графа G, будем называть ***графом влияния*** на роль r.

Предложение 1. *Если ролевой граф G является деревом, граф влияния $GR(r)$ на произвольную роль r будет также деревом.*

Доказательство. По свойствам ориентированного дерева [5, с. 239] подграф, определяемый множеством узлов, достижимых из вершины r, является ордеревом с корнем r. ■

Предложение 2. *Если ролевой граф G является решёточным, граф влияния $GR(r)$ на произвольную роль r будет также решёточным.*

Доказательство. Согласно [3], решёточный граф — это ориентированный граф, вершины которого образуют решётку, при этом отношение порядка определяется ориентированными путями графа.

В графе влияния $GR(r)$ для любой вершины r_i существует ориентированный путь $p(r, r_i)$ по определению. Следовательно, $\sup\{r, r_i\} = r$, $\inf\{r, r_i\} = r_i$.

Пусть теперь r_i и r_j — две различные вершины графа влияния $GR(r)$, отличные от r. Очевидно, что $\inf\{r_i, r_j\} \in GR(r)$, так как в этом графе содержатся все вершины, достижимые из r, а значит, и достижимые из r_i и r_j. Пусть $\sup\{r_i, r_j\} = r_s$. Докажем, что $r_s \in GR(r)$. Допустим это не так. По определению графа влияния: $r \geqslant r_i$ и $r \geqslant r_j$, следовательно, r является верхней гранью этих вершин. Так как r_s — наименьшая из всех верхних граней, то $r \geqslant r_s$, следовательно, существует ориентированный путь $p(r, r_s)$ — противоречие с тем, что $r_s \notin GR(r)$.

Таким образом, для любых двух вершин графа влияния $GR(r)$ наибольшая нижняя и наименьшая верхняя грани также принадлежат этому графу, следовательно $GR(r)$, — решёточный. ■

Для анализа путей передачи привилегий между ролями не обязательно рассматривать граф влияния полностью, можно удалить некоторые дуги, дублирующие друг друга при передаче привилегий роли r. При таком преобразовании могут «потеряться» некоторые привилегии ролей из множества $PR(r)$, однако перед нами стоит задача обеспечения безопасности роли r. Такую процедуру преобразования будем называть ***оптимизацией графа влияния***. Очевидно, что в результате оптимизации можно получить разные графы. Оптимизированный граф влияния с минимальным количеством дуг будем называть ***минимальным***.

Теорема 2. *Минимальный граф влияния является деревом.*

Доказательство. По определению, любая вершина r' графа влияния $GR(r)$ достижима из вершины r (существует ориентированный путь $p(r, r')$).

Как было показано в работе [2], в ролевом графе G отсутствуют ориентированные циклы. Отсюда следует, что полустепень захода (число входящих дуг) вершины r в графе влияния $GR(r)$ равна нулю ($d^+(r) = 0$), так как иначе в исходном графе G существовал бы ориентированный цикл $(r', r) \cup p(r, r')$ для некоторой вершины r'.

Очевидно, что полустепень захода всех остальных вершин графа влияния $GR(r)$ больше или равна единице: $\forall r' \in GR(r) : (r' \neq r) \Rightarrow (d^+(r') \geqslant 1)$. Покажем, что в минимальном графе влияния $\forall r' \in GR(r) : (r' \neq r) \Rightarrow (d^+(r') = 1)$. Пусть это не так, тогда $\exists r'' \in GR(r) : (r'' \neq r) \wedge (d^+(r'') > 1)$. Следовательно, в графе $GR(r)$ существует как минимум две различные дуги (r_1, r'') и (r_2, r''). А так как в графе $GR(r)$ любая вершина достижима из вершины r, то в этом графе существуют ориентированные пути $p(r, r_1)$ и $p(r, r_2)$, отличные друг от друга по крайней мере конечными вершинами. В соответствии с принципом наследования привилегий: $PR(r'') \subseteq PR(r_1) \subseteq PR(r)$ и $PR(r'') \subseteq PR(r_2) \subseteq PR(r)$. Но тогда к графу $GR(r)$ можно применить процедуру оптимизации графа влияния, удалив одну из дуг (r_1, r'') или (r_2, r''), что противоречит с его минимальностью.

В итоге, опираясь на определение ориентированного дерева [5, с. 238], теорема доказана. ■

Замечание 1. Если ролевой граф — дерево, то на основании предложения 1 и теоремы 2 для произвольной роли r граф влияния единственен (так как он является деревом и не подлежит оптимизации).

Теорема 3. *Трудоёмкость поиска графа влияния на произвольную вершину ролевого дерева не превосходит* $O(n^2)$.

Доказательство. Пусть ролевой граф G на n вершинах задан списками смежности L [5, с. 202]. Воспользуемся алгоритмом поиска в ширину (в глубину), представленным, например, в работе [5, с. 203]. Для построения множества $PR(r_i)$ достаточно в этом алгоритме начать обход с вершины r_i. Очевидно, результатом работы алгоритма на ориентированном графе будет множество вершин, достижимых из начальной, т. е. множество $PR(r_i)$. Трудоёмкость подобного алгоритма равна $O(n^2)$ [1].

Для задания графа влияния $GR(r_i)$ достаточно из исходных списков смежности L выбрать те, которые соответствуют вершинам множества $PR(r_i)$. Трудоёмкость этого шага зависит от реализации списков, но в любом случае не превышает $O(n^2)$. ■

Замечание 2. Вообще говоря, трудоёмкость алгоритма поиска в ширину равна $O(m)$, где m — число дуг графа. Для достаточно «плотных» графов $m = O(n^2)$. Если же граф, в котором ведётся поиск, является деревом, то $m = n - 1$. Таким образом, если ролевой граф — дерево, то трудоёмкость поиска графа влияния

на произвольную вершину может быть понижена до $O(n)$ при соответствующей реализации списков смежности.

Замечание 3. Пусть теперь ролевой граф G задан матрицей смежности M размерности $n \times n$. Очевидно, переход от такого представления к спискам смежности требует не более $O(n^2)$ операций, что оставляет справедливым утверждение теоремы 3.

Если же в нашем распоряжении имеется матрица достижимости M^* исходного ориентированного графа, то вновь трудоёмкость поиска графа влияния остаётся $O(n^2)$. Действительно, напомним, что элемент m^*_{ij} матрицы достижимости равен 1, если в орграфе существует ориентированный путь из вершины r_i в вершину r_j, и 0 — в противном случае. Тогда вершины, принадлежащие графу влияния $GR(r_i)$, — это те вершины r_j, для которых элемент $m^*_{ij} = 1$, а также сама вершина r_i. Таким образом трудоёмкость поиска множества $PR(r_i)$ равна $O(n)$. Осталось построить матрицу смежности $MR(r_i)$ графа влияния $GR(r_i)$. Для этого достаточно выбрать из матрицы смежности M ролевого графа G строки и столбцы, соответствующие множеству $PR(r_i)$. Трудоёмкость этого шага равна $O(n^2)$. Но здесь стоит заметить, что сама процедура построения матрицы достижимости имеет трудоёмкость $O(n^3)$ [5, с. 48].

2. Элементарные операторы

Для анализа преобразований введём набор элементарных операторов, преобразующих граф G.

1. $Auth(r_1, r_2)$ — авторизация роли r_1 на роль r_2, добавляет дугу от r_1 к r_2. Данная операция приводит к тому, что множество привилегий роли $r_1.p$ изменяется и становится равным $r_1.p \cup r_2.p$. Привилегии же роли r_2 остаются неизменными.

2. $DeleteA(r_1, r_2)$ — отменяет авторизацию роли r_1 на роль r_2, удаляет дугу от r_1 к r_2. Новое множество привилегий роли r_1 имеет вид

$$(r_1.p \setminus r_2.p) \cup \left(\bigcup_{r' \in (PR(r_1) \setminus r_2)} r'.p \right).$$

Простая разность множеств привилегий $r_1.p \setminus r_2.p$ может приводить к неверному результату, так как возможна ситуация, когда одна и та же привилегия наследуется от нескольких ролей.

3. $CreateR(r)$ — создаёт роль r, добавляет в граф вершину, не связанную с другими вершинами. Данный оператор сам по себе никак не влияет на распределение полномочий и не может приводить к нарушению безопасности.

4. $DeleteR(r)$ — удаляет роль r, удаляет в графе вершину. Условием выполнения оператора является изолированность вершины. Для удаления вершины со связями необходимо предварительно удалить все дуги с помощью оператора $DeleteA()$. Как легко понять, данный оператор сам по себе также не приводит к перераспределению полномочий.

5. $EnterP(p, r)$ — добавляет привилегию p в множество привилегий роли r. Соответственно данная привилегия также добавляется всем ролям, доминирующим над данной ролью.

6. $DeleteP(p, r)$ — удаляет привилегию p из множества привилегий роли r. После этого привилегия p удаляется также и у ролей, доминирующих над r, если они не наследуют эту привилегию от других ролей.

Для выполнения сложных преобразований системы из операторов могут быть составлены команды:

$$\begin{aligned} &Command\ C\{\\ &\alpha_1,\ \alpha_2,\ \ldots,\ \alpha_n;\\ &\}. \end{aligned}$$

Здесь $\alpha_1, \alpha_2, \ldots, \alpha_n$ — элементарные операторы.

Теорема 4. *Для любых двух ролевых графов G и G' существует команда C, преобразующая G в G'.*

Доказательство. Для доказательства теоремы построим одну из возможных команд преобразования ролевого графа G в G'. Применим для каждой из дуг графа G команду $DeleteA()$, в результате чего получим множество изолированных вершин. Применяя нужное количество раз оператор $CreateR()$ или $DeleteR()$, добьёмся того, чтобы количество вершин стало равным количеству вершин в графе G'. Далее с помощью оператора $Auth()$ создадим дуги, соответствующие дугам графа G'. Распределив полномочия с помощью оператора $EnterP()$, получим граф, изоморфный G'. ■

Таким образом, описанный выше набор элементарных операторов является полным и на его основе можно построить любой ролевой граф.

3. Классы безопасных систем

Следует отметить, что команды в системе выполняются атомарно, т. е. подсистема безопасности может производить проверку только после завершения команды. Выявим несколько классов безопасных систем. Для этого рассмотрим ограничения, которые при накладывании на систему могут гарантировать безопасность.

1. *Статический ролевой граф.* К этому классу относятся системы, у которых ролевой граф остаётся неизменным, т. е. отсутствует администрирование системы в процессе её функционирования. Как было показано в теореме 1, такие системы являются безопасными. Для проверки безопасности системы со статическим ролевым деревом могут применяться теоремы поиска ориентированных путей из теории графов.

2. *Фиксированное количество административных ролей.* Новые роли может создавать конечное фиксированное множество ролей (администраторов). Расширение предыдущего класса.

3. *Монооперационные системы.* Каждая команда содержит только один элементарный оператор. Можем после каждой команды проводить проверку системы, причём выполняется она за конечное число шагов.

4. *Системы с выделенным оператором создания административных ролей.* Если команда включает создание административной роли, то не включает других ролей.

Заключение

Таким образом, для ролевой политики безопасности имеется возможность формализации путём определения элементарных операторов преобразования ролевого графа. Их использование, в свою очередь, позволит получать более строгие доказательства безопасности систем с ролевым разграничением доступа к информации в соответствии с определёнными формализованными критериями.

Литература

1. Алексеев В. Е., Таланов В. А. Графы и алгоритмы. URL: http://www.intuit.ru/department/algorithms/gaa/4/ (дата обращения: 16.10.2010).
2. Белим С. В., Белим С. Ю., Богаченко Н. Ф. Теоретико-графовый анализ ролевой политики безопасности // Математические структуры и моделирование. Омск: УниПак. 2009. Вып. 19. С. 85–96.
3. Белим С. В., Богаченко Н. Ф., Ракицкий Ю. С. Совместная реализация мандатного и ролевого разграничения доступа к информации в компьютерных системах // Математические структуры и моделирование. Омск: УниПак. 2009. Вып. 20. С. 141–152.
4. Гайдамакин Н. А. Разграничение доступа к информации в компьютерных системах. Екатеринбург: Изд-во Урал. ун-та, 2003. 328 с.
5. Новиков Ф. А. Дискретная математика для программистов: учебник для вузов. СПб.: Питер, 2001. 304 с.
6. Хаггарти Р. Дискретная математика для программистов: пер. с англ. М.: Техносфера, 2005. 400 с.