Секция 4

МАТЕМАТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

УДК 65.012.810(075.8)

ПОИСК tg-ПУТЕЙ И ОСТРОВОВ ДЛЯ МОДЕЛИ БЕЗОПАСНОСТИ TAKE-GRANT

Д. М. Бречка

Минимальным требованием безопасности компьютерных систем является наличие в них разграничения доступа к информации [1], то есть организация системы таким образом, что пользователи системы получают доступ лишь к той информации, которая необходима им для выполнения своих функциональных обязанностей. Для достижения этих целей разработан ряд математических моделей.

Одной из наиболее проработанных моделей является модель дискреционного доступа Take-Grant [2 – 4]. Компьютерная система в модели Take-Grant представляется в виде ориентированного графа (*графа доступов*), вершинами которого являются субъекты и объекты системы, а дугами — установленные права субъектов на объекты. Помимо стандартного набора прав, таких, как, например, право чтения или право записи, в модели Take-Grant вводятся два дополнительных права: $Take$ (t) — право брать права у какого-либо субъекта и $Grant$ (g) — право передавать свои права какому-либо другому субъекту. Анализируя исходный граф системы, можно выяснить, возможно ли получение доступа каким-либо субъектом к какому-либо объекту за некоторое число шагов, то есть существует ли возможность модифицировать исходный граф так, что между двумя вершинами появится дуга с правом, которого нет в исходном графе. В рамках модели сформулированы две теоремы, которые оговаривают условия для возникновения возможности доступа.

Из теоретического описания модели ясны условия, при которых возникает возможность доступа, однако способы проверки наличия этих условий не оговариваются.

При оценке безопасности состояний системы, для которой граф доступов состоит только из вершин-субъектов, необходимо проверить наличие *tg-пути* между двумя вершинами (путь в графе, все дуги которого содержат права t или g) [2, 3].

Для графа из вершин-субъектов необходимо проверить наличие tg-пути между двумя вершинами; воспользуемся для этих целей алгоритмом Дейкстры [5].

Построим по исходному графу доступов G_0 граф G_0' следующим образом:

1) множество вершин графа G_0' совпадает с множеством вершин графа G_0 $(V_0' = V_0)$;
2) ребра в графе G_0' не ориентированы; множество ребер графа включает лишь те ребра из G_0, которые содержат права *Take* или *Grant* $(E_0' = E_0 \setminus E_0^{\alpha}$, где E_0^{α} — множество ребер графа G_0, не содержащих прав *Take* или *Grant*);
3) все ребра графа G_0' имеют одинаковый вес.

Для графа G_0' применим алгоритм Дейкстры для поиска кратчайшего пути в графе. Кратчайший путь между двумя указанными вершинами, найденный алгоритмом Дейкстры в графе G_0', будет являться tg-путем между этими вершинами в графе G_0.

Оценка времени работы алгоритма Дейкстры в зависимости от числа вершин исходного графа будет $O(N^2)$ [6], где N — число вершин. Для построения графа G'_0 потребуется также $O(N^2)$ операций.

Для произвольного графа доступов необходимо найти *острова* в нем, а также *мосты*, их *начальный* и *конечный пролеты* [2–4]. Для поиска островов воспользуемся алгоритмом Флойда для поиска всех кратчайших путей [5].

Построим граф G_0^* по исходному графу доступов G_0 следующим образом:

1) множество вершин графа G_0^* совпадает с множеством вершин графа G_0 ($V_0^* = V_0$);
2) ребра в графе G_0^* не ориентированы; множество ребер графа включает лишь те ребра из G_0, для которых началом и концом дуги является вершина-субъект и которые содержат права *Take* или *Grant* ($E_0^* = E_0 \setminus (E_0^{s-o} \cup E_0^{\alpha})$, где E_0^{s-o} — множество ребер графа G_0, для которых начало и конец не являются субъектами; E_0^{α} — множество ребер графа G_0, не содержащих прав *Take* или *Grant*);
3) все ребра графа G_0^* имеют одинаковый вес.

Для графа G_0^* применим алгоритм Флойда. Алгоритм обнаружит кратчайшие пути между каждой парой вершин в графе G_0^*. Кратчайшие пути в графе G_0^* будут tg-путями, проходящими через вершины-субъекты в графе G_0. А сами вершины-субъекты, объединенные tg-путями, будут являться островами в графе G_0. Трудоемкость алгоритма Флойда — $O(N^3)$ [6]; граф G_0^* можно будет построить за $O(N^2)$ операций.

Зная критерии безопасности состояний информационной системы и способы проверки этих критериев, можно проанализировать исходное состояние системы на предмет наличия нежелательных доступов.

ЛИТЕРАТУРА

1. DoD 5200.28 – std. Department of Defense Trusted Computer System Evaluation Criteria. 1985.
2. *Lipton, Richard J.* A Linear Time Algorithm for Deciding Subject Security // J. ACM (Addison-Wesley). 1977. No. 3. P. 455–464.
3. *Девянин П. Н.* Модели безопасности компьютерных систем: учебное пособие для студентов высших учебных заведений. М.: Академия, 2005. 144 с.
4. *Гайдамакин Н. А.* Разграничения доступа к информации в компьютерных системах. Екатеринбург: Изд-во Урал. ун-та, 2003. 328 с.
5. *Майника Э.* Алгоритмы оптимизации на сетях и графах. М.: Мир, 1981. 324 с.
6. *Кормен Т., Лейзерсон Ч.* Алгоритмы: построение и анализ М.: МЦНМО, 2000. 960 с.

УДК 004.94

МОДЕЛИРОВАНИЕ УПРАВЛЕНИЯ ДОСТУПОМ И ИНФОРМАЦИОННЫМИ ПОТОКАМИ В ЭЛЕКТРОННЫХ ПОЧТОВЫХ СИСТЕМАХ[1]

К. А. Грищенко

В результате проведенного исследования распространенных электронных почтовых систем (ЭПС) получено описание реализованных в них механизмов управления

[1]Работа выполнена при поддержке гранта МД № 2.2010.10.