

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМОВ ПОИСКА НЕСАНКЦИОНИРОВАННЫХ ДОСТУПОВ

Д.М. Бречка

к.т.н., e-mail: dbrechkawork@yandex.ru

Р.А. Исин

студент, e-mail: newshriek@gmail.com

Факультет компьютерных наук, Омский государственный университет
им. Ф.М. Достоевского

Аннотация. Описана реализация алгоритмов поиска несанкционированных доступов для модели Take-Grant. Приводится внутренняя структура программных продуктов, реализующих алгоритмы, форматы входных и выходных данных, результаты тестирования производительности программных продуктов.

Ключевые слова: алгоритм, безопасность, Take-Grant, программирование, объекты, реализация..

Введение

В работах [1–4] предлагаются алгоритмы поиска tg-путей островов, мостов, начальных и конечных пролётов мостов в графе доступов. Для поиска tg-путей предлагается модифицированный алгоритм Дейкстры, для поиска островов — модифицированный алгоритм Флойда, для поиска мостов, начальных и конечных пролётов мостов предлагаются оригинальные алгоритмы, основанные на алгоритмах поиска в глубину и в ширину. Далее описана программная реализация и оптимизация алгоритмов поиска tg-путей, островов и мостов.

Для реализации алгоритмов поиска несанкционированных доступов было решено использовать объектно-ориентированный подход к программированию. В качестве языка программирования был выбран Java. Для разработки и отладки программного кода использовалась интегрированная среда разработки IntelliJ IDEA.

Для поиска основных структур в графе доступа было разработано три отдельных приложения, выполненных по общей схеме:

- 1) ввод данных;
- 2) создание графа доступов в памяти компьютера;
- 3) преобразование графа доступов;
- 4) применение соответствующего алгоритма;

5) вывод результатов работы.

Приложения различаются на этапах 3 и 4, так как для поиска различных структур в графе доступов необходимо различное преобразование исходного графа и применение различных алгоритмов.

Данные этапы в приложениях реализуются различными классами. Так поиск tg-пути реализован классом `Dijkstra.java`, поиск островов — классом `Floid.java`, для поиска мостов используется класс `BridgeSearcher.java`.

Помимо классов, реализующих основной функционал, в программных продуктах также используются вспомогательные классы:

1. `Node.java` — описывает узел графа доступов.
2. `Link.java` — описывает связь между вершинами графа.
3. `Input.java` — используется для реализации ввода данных в программу.
4. `Main.java` — служит для связи классов в единую структуру и вывода результатов работы.

Вспомогательные классы одинаковы для всех трёх приложений. Ввод данных в программу организуется через файл формата `csv`, содержащий матричное представление графа доступов. Данный файл должен быть подготовлен заранее вручную или с помощью специальных программ. Формат матрицы приведён на рисунке 1 (крайний левый столбец добавлен на рисунок для ясности, в файле отсутствует). В первой строке файла записываются все вершины графа, во второй — определяется тип вершины. Последующие строки содержат матричное представление графа. По матрице можно определить направление дуги: начальная вершина записана в строке слева, конечная — сверху. Вывод результатов работы приложения осуществляется в консоль.

В случае поиска tg-путей, мостов и пролётов мостов на вход приложению также передаются названия вершин, между которыми отыскивается соответствующая структура. Названия вершин передаются через параметры командной строки. В качестве результата работы приложения выдают:

1. В случае поиска островов — матрицу достижимостей, по которой можно делать вывод о наличии или отсутствии островов в графе.
2. В случае поиска tg-путей — длину пути, а также матрицу весов для алгоритма Дейкстры, по которой можно восстановить сам путь, либо сообщение о том, что путь не найден.
3. В случае поиска мостов и пролётов мостов — выводится список вершин, входящих в мост (пролёт), либо сообщение об отсутствии моста (пролёта).

Анализ производительности программных продуктов проводился на компьютере со следующими основными характеристиками:

1. Процессор: Intel Core i5-3210M 2.50Ghz.

```

      10
      A  B  C  D  E  F  G  H  I  J
A  S  O  S  O  O  O  O  S  O  S
B  *  t  0  rw 0  0  0  0  0  0
C  0  *  0  g  0  0  0  0  0  0
D  0  r  *  0  0  0  0  0  0  0
E  0  0  0  *  0  t  0  0  g  g
F  0  0  0  g  *  0  0  0  0  0
G  0  0  0  g  w  *  0  0  0  0
H  0  0  rw 0  0  0  *  0  0  0
I  0  0  0  0  g  t  0  rw *  0
J  0  0  0  0  0  0  0  t  0  *

```

Рис. 1. Формат входного файла для анализа безопасности компьютерной системы. * — полный набор прав; 0 — нет дуги.

2. ОЗУ: 8 Гб.

3. ОС: Windows 8 x64.

Результаты тестирования приведены в таблице 1 в виде зависимости времени работы каждого алгоритма от количества вершин в исходном графе.

Таблица 1. Результаты тестирования эффективности работы алгоритмов поиска несанкционированных доступов

Количество вершин	18	30	60	120	512	612	712	810	1024
Поиск tg-путей (мс)	1	2	2	4	16	20	29	36	40
Поиск островов (мс)	1	3	6	12	326	703	1556	2711	5475
Поиск мостов (мс)	1	1	4	33	4942	9939	25858	41176	77997

Как видно из таблицы 3, производительность алгоритмов имеет экспоненциальную зависимость от количества вершин, что соответствует теоретическим оценкам, приведённым в [1–4].

ЛИТЕРАТУРА

1. Бречка Д.М. Алгоритмы анализа безопасности состояний компьютерной системы для модели Take-Grant // Математические структуры и моделирование. 2009. Вып. 20. С. 160–172.
2. Бречка Д.М. Анализ возможности доступа в модели Take-Grant // В мире научных открытий. 2010. № 4 (10). Часть 4. С. 11–13.
3. Проблемы обработки и защиты информации. Книга 1. Модели политик безопасности компьютерных систем / Белим С.В., Белим С.Ю., Бречка Д.М. и др.; под редакцией С.В. Белима. Омск: ООО «ООО Полиграфический центр КАН», 2010. 164 с.
4. Бречка Д.М. Алгоритм поиска мостов типа \vec{t}^* и \overleftarrow{t}^* в графе доступов для дискреционной модели безопасности Take-Grant. // Математические структуры и моделирование. 2011. Вып. 23. С. 99–104.

THE PROGRAM REALIZATION OF ALGORITHMS FOR AN UNAUTHORIZED ACCESS SEARCHING**D.M. Brechka**

Ph.D. (Eng.), e-mail: dbrechkawork@yandex.ru

R.A. Isin

Postgraduate Student, e-mail: newshriek@gmail.com

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

Abstract. The article describes the realization of algorithms for an unauthorized access searching in Take-Grant model. The internal structure of software products that implement algorithms, input and output data formats, results of software performance testing is described.

Keywords: algorithm, security, Take-Grant, programming objects implementation.