Математические структуры и моделирование
2015. № 4(36). С. 129–133

УДК 65.012.8

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ СИСТЕМЫ РАЗГРАНИЧЕНИЯ ДОСТУПА ХАРРИСОНА-РУЗЗО-УЛЬМАНА

Д.М. Бречка
к.т.н., доцент кафедры кибернетики, e-mail: dbrechkawork@yandex.ru
В.В. Зубова
студентка, e-mail: zubovaleria60@gmail.com

Омский государственный университет им. Ф.М. Достоевского

Аннотация. Статья посвящена разработке компьютерной модели системы разграничения доступа Харрисона-Руззо-Ульмана. Такая модель необходима для изучения принципов работы систем с дискреционным разграничением доступа.

Ключевые слова: безопасность, HRU, компьютерные системы, программная реализация, объект, матрица доступа, разделение доступа.

Введение

Разделение доступа является одним из основных механизмов защиты информации в современных компьютерных системах. Для формального описания правил разделения доступа часто используются модели разделения доступа. Одной из таких моделей является модель Харрисона-Руззо-Ульмана (Harrison-Ruzzo-Ulman, HRU). Данная модель относится к дискреционным моделям разделения доступа [1].

В работах [2–7] была показана возможность применения модели HRU для анализа безопасности современных операционных систем. Работы [8, 9] показывают возможность составления матрицы доступов в операционных системах, матрица доступов является одним из составных элементов модели HRU.

Целью данной работы является программная реализация системы HRU. Программная реализация необходима для моделирования систем, работающих согласно принципам HRU, что позволит более глубоко изучать вопросы безопасности компьютерных систем.

1. Краткое описание модели HRU

Модель HRU использует субъект-объектный подход к моделированию безопасности компьютерной системы. Основными компонентами HRU являются:

- множество субъектов S — множество активных сущностей системы;
- множество объектов O — множество пассивных сущностей системы, при этом $S \subseteq O$;

- множество прав доступа R — множество действий, которые субъекты могут совершать над объектами;
- матрица доступа M — таблица, в строках которой расположены все субъекты системы, в столбцах — объекты, а в ячейках — соответствующие права доступа.

Функционирование системы рассматривается как последовательное изменение матрицы доступов. При этом над матрицей доступа возможно производить следующие 6 примитивных действий:

- добавить право r в матрицу доступа;
- удалить право r из матрицы доступа;
- добавить субъект;
- добавить объект;
- удалить субъект;
- удалить объект.

Примитивные действия объединяются в команды HRU. Таким образом, функционирование компьютерной системы расматривается как последовательное применение команд HRU к матрице доступа.

2. Программная реализация модели HRU

Для реализации программной модели был выбран язык C#. Данный язык реализует объектно-ориентированную парадигму программирования, что позволяет составить удобную для разработки и анализа модель системы.

Для моделирования основных сущностей системы создадим три класса: $Subject, Objects, Monitor$. Класс $Subjects$ представляет собой структуру, содержащую в себе набор полей и методов, необходимых для задания субъекта. А класс $Objects$ — для объекта соответственно. Класс $Monitor$ содержит в себе матрицу доступа M, реализованную как двумерный массив типа $char$, строки которого соответствуют субъектам, а столбцы — объектам доступа. При этом, учитывая, что множество S является подмножеством O, в столбцах матрицы M также содержатся и субъекты доступа.

В данной программе класс $Monitor$ является основным, так как в нем отслеживаются запросы на изменение матрицы доступов и непосредственно производятся эти изменения. Диаграмма классов данной программы представлена на рисунке 1.

Для простоты и удобства обработки элементов системы в классе Monitor реализованы две структуры $Dictionary$: $subjects$ и $objects$. Каждый экземпляр $Dictionary$ имеет два параметра: ключ (key) и значение ($value$). В нашем случае параметр $value$ будет нести в себе номер, название элемента $subjects$ или $objects$, а параметр key — идентификатор данного элемента.

Для моделирования примитивных действий с матрицей доступа каждому действию сопоставлен метод класса $Monitor$. Рассмотрим подробнее их реализацию.

Метод $createS(int)$. Данный метод реализует добавление нового субъекта в матрицу доступа M. В первую очередь, происходит проверка того, что субъект

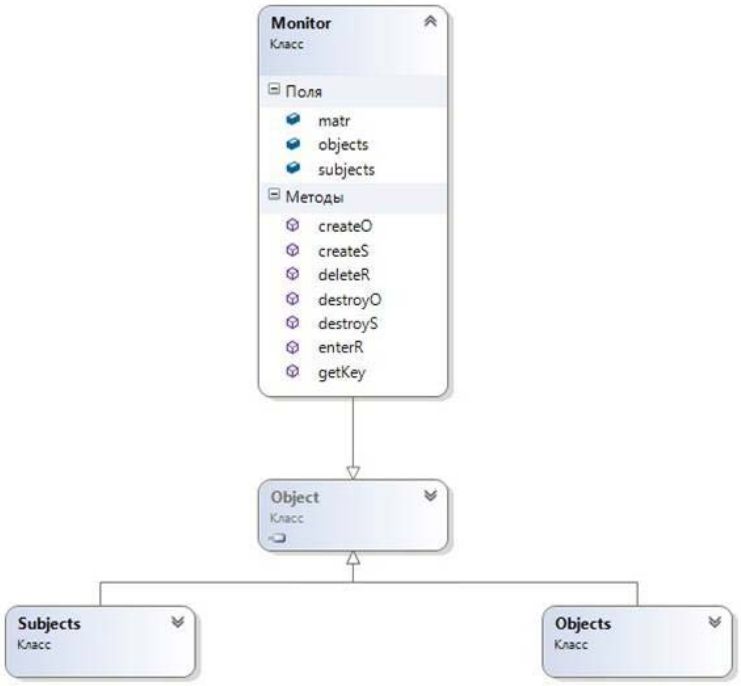

Рис. 1. Диаграмма классов

с данным номером ещё не был создан в системе. Если условие выполнено, то в словарь субъектов добавляется новый элемент, значение параметра $value$ которого равно номеру субъекта. Номер субъекта посредством интерфейса программы задаёт пользователь (рис. 2). В качестве значения параметра key выбирается количество уже созданных субъектов в словаре. После вызова метода в матрицу доступов добавляются новая строка и столбец с номером $subjects.Key$ и названием $subjects.Value$.

Метод $createO(int)$. Реализация данного метода аналогична предыдущему, однако, отличие в том, что для работы с объектами используется словарь $objects$. В результате выполнения в матрицу доступов добавляется только новый столбец с названием $objects.Value$ и идентификатором $objects.Key$.

Метод $destroyS(int)$ — метод удаления субъекта из системы и из матрицы доступа M. В начале проверяется, существует ли субъект с номером, заданным пользователем, в словаре $subjects$. Проверка выполняется по ключу. С помощью метода $While()$ класса $Dictionary$, по заданному пользователем значению $value$, возвращается параметр key. Если условие выполнено, то, используя метод $Dictionary.Remove()$, субъект удаляется из словаря $subjects$. Затем из матрицы доступа удаляются строка и столбец с параметром key, который был получен на предыдущем шаге.

Метод $destroyO(int)$ — метод, аналогичный предыдущему. В отличие от $destroyS()$ здесь рассматривается словарь объектов $objects$, а из матрицы до-

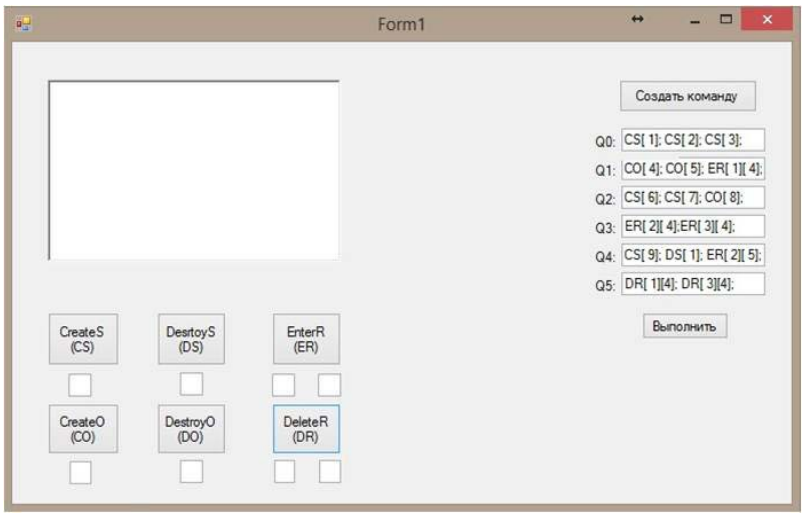

Рис. 2. Интерфейс программы

ступа удаляется только столбец с идентификатором объекта.

Метод $enterR(int, int)$. Условием для выполнения метода $enterR()$ является существование в матрице доступа строки и столбца с номерами, заданными пользователем. Пусть субъект имеет номер i, а объект — номер j. Если условие выполнено, то в матрице на пересечении данных строки и столбца ставится некоторый символ $'r'$. Это символизирует, что субъект i теперь имеет право r на объект j.

Метод $deleteR(int, int)$ — метод удаления права r из матрицы доступа M. Пользователь вводит номер i-ой строки и j-го столбца. В методе осуществлена проверка — есть ли в системе субъект и объект с такими номерами и существует ли у субъекта i право r над объектом j. Если условие выполнено, то право r удаляется из матрицы доступа M.

Набор примитивных операторов образует команду HRU. Последовательное выполнение таких команд переводит систему из состояния Q_0 в новое состояние Q_1. Таким образом, реализовав в классе $Monitor$ методы, которые имитируют работу примитивных операторов модели Харрисона-Руззо-Ульмана, можно перевести систему в новое состояние, изменив при этом матрицу доступа M.

Заключение

Результатом проделанной работы является реализация программной модели компьютерной системы, работающей согласно принципам HRU. Тестирование модели показало корректность работы. В дальнейшем планируется использовать данную модель для детального изучения безопасных систем HRU, таких как монотонные и монооперационные системы, а также монооперационных систем в базисе [3–5].

Литература

1. Девянин П.Н. Модели безопасности компьютерных систем: учебное пособие для студентов высших учебных заведений. М. : Издательский центр «Академия», 2005. 143 с.
2. Бречка Д.М., Белим С.В. Исследование безопасности компьютерных систем в модели дискреционного разделения доступа HRU // Математические структуры и моделирование. 2009. Вып. 19. С. 97–103.
3. Бречка Д.М., Белим С.В. Базисный подход в модели безопасности HRU // Проблемы информационной безопасности. Компьютерные системы. 2010. Вып. 2. С. 18–23.
4. Бречка Д.М., Белим С.В. Расширение класса безопасных систем в модели HRU // В мире научных открытий. 2010. № 4(10). Часть 4. С. 9–11.
5. Бречка Д.М., Белим С.В. Классы безопасности в модели HRU // Безопасность информационных технологий. 2010. Вып. 3. С. 26–31.
6. Бречка Д.М., Белим С.В. Исследование безопасности дискреционного разделения доступа в ОС Windows // Математические структуры и моделирование. 2011. Вып. 22. С. 121–130.
7. Проблемы обработки и защиты информации. Книга 1. Модели политик безопасности компьютерных систем / С.В. Белим [и др.]. Омск : ООО «Полиграфический центр КАН», 2010. 164 с.
8. Бречка Д.М., Сыргий Е.В. Формирование матрицы доступов на основе внутренних структур операционной системы LINUX // Материалы I междунар. науч.-практ. конф. «Информационная безопасность в свете Стратегии Казахстан-2050», Астана, 12 сентября 2013. С. 506–511.
9. Бречка Д.М., Сыргий Е.В. Система составления матрицы доступов запущенных процессов в операционной системе Windows // Вопросы защиты информации. 2014. Вып. 3(106) С. 17–24.

COMPUTER MODEL OF HARRISON-RUZZO-ULMAN ACCESS CONTROL SYSTEM

D.M. Brechka
Ph.D.(Eng.), Associate Professor, e-mail: dbrechkawork@yandex.ru
V.V. Zubova
Student, e-mail: zubovaleria60@gmail.com

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

Abstract. The article describes development of a computer model for Harrison-Ruzzo-Ulman access control system. This model is necessary to study the principles of systems with discretionary access control.

Keywords: security, HRU, computer system, implementation, object, access matrix, access control.