Определение 6. Пусть $G_0 = (S_0, E_0, P_0, F_0, H_{E0}, class_0)$ — состояние системы $\Sigma(G^*, OP)$, в котором существуют сущности $x, y \in E_0$. Определим предикат $can\_write\_time(x, y, G_0)$, который будет истинным тогда и только тогда, когда существуют состояния $G_1, \ldots, G_N = (S_N, E_N, P_N, F_N, H_{EN}, class_N)$ и правила преобразования состояний $op_1, \ldots, op_N$, такие, что $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \ldots \vdash_{op_N} G_N$ и $(x, y, write_t) \in F_N$, где $N \geqslant 0$.

Замечание 1. Заметим, что для проверки истинности предикатов $can\_share(u, e, \alpha_r, G_0)$, $can\_write\_memory(x, y, G_0)$ и $can\_write\_time(x, y, G_0)$ в соответствии с определением требуется учесть все траектории функционирования КС, что не осуществимо на практике.

В связи с замечанием 1 представляется целесообразным сформулировать и обосновать алгоритмы проверки истинности предикатов can_share, can_write_memory и can_write_time. Такие алгоритмы реализуют преобразование начального состояния КС в его замыкание и позволяют проверить истинность предикатов для всех пользователей, сущностей и прав доступа одновременно. В работе вводятся определения замыканий ДП-модели КС $SELinux$ ($time$-замыкания и $memory$-замыкания), предлагаются и обосновываются алгоритмы их построения.

Предлагается также метод применения построенной модели на практике для проверки возможности получения права доступа и реализации информационного потока в КС $SELinux$. Метод состоит из двух основных этапов. Первый этап — это построение начального состояния предложенной ДП-модели КС $SELinux$ по набору конфигурационных файлов КС. Второй этап — это построение $time$-замыкания полученного состояния и интерпретация полученных результатов с точки зрения КС $SELinux$. На входе метод имеет весь набор необходимых конфигурационных файлов КС $SELinux$, а на выходе — ответ на вопрос, возможны ли получение заданного права доступа или реализация заданного информационного потока.

УДК 004.94

ОСОБЕННОСТИ РАЗРАБОТКИ ДП-МОДЕЛЕЙ СЕТЕВОГО УПРАВЛЕНИЯ ДОСТУПОМ[1]

Д. Н. Колегов

Работа посвящена особенностям построения ДП-моделей компьютерных систем (КС), реализующих сетевое управление доступом. Такие модели будем называть сокращенно СУД ДП-моделями, а при их описании используем основные определения и обозначения из [1].

Механизмы сетевого управления доступом в современных КС, как правило, обладают следующими свойствами, затрудняющими применение элементов и средств существующих ДП-моделей для их описания и исследования:

— распределенностью компонентов управления доступом и их сетевым взаимодействием;
— динамическим управлением доступом субъектов к сущностям на основе правил доступа и, как следствие, предоставлением субъектам различных прав доступа в зависимости от истинности условий того или иного правила доступа;

[1]Работа выполнена в рамках реализации ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009–2013 гг. (гос. контракт № П1010).

— наличием различных правил управления доступом для одних и тех же субъектов доступа;
— принадлежностью сущностей одновременно нескольким иерархиям и использованием последних при определении и проверке правил управления доступом.

Для адекватного отражения этих особенностей КС с сетевым управлением доступом в ДП-моделях предлагается язык описания последних расширить следующим образом.

1. Вместо одной функции иерархии сущностей вводится множество $H = \{H_1, \ldots, H_n\}$, где $H_i : E \to 2^E$ для $i = 1, \ldots, n$ есть функция иерархии сущностей. Например, такой набор функций соответствует иерархиям типов, назначения и расположения узлов сетевой КС управления доступом Cisco Access Control Server.

2. В дополнение к сущностям, параметрически ассоциированным с субъектом, вводятся множества сущностей, параметрически ассоциированных с субъектом, или ПАС-множества. Сущности из ПАС-множества $Z \subseteq E$ называются параметрически ассоциированными с субъектом $s \in S$ в состоянии G, если данные в этих сущностях позволяют идентифицировать вид преобразования данных, реализуемого субъектом s в этом или последующих состояниях КС. Наличие у субъекта нескольких ПАС-множеств является типичной ситуацией в современных КС. Например, одно из ПАС-множеств может содержать сущность-пароль и IP-адрес в базе данных сетевой системы управления доступом КС, а второе — сущность-пароль в базе данных программного обеспечения, реализующего удаленный доступ. Другим примером наличия нескольких ПАС-множеств является реализация в операционной системе механизмов двухфакторной аутентификации. В этом случае первое ПАС-множество содержит сущности, являющиеся частями разделенного пароля, одна часть которого может храниться на аппаратном носителе, другая — в базе данных; второе содержит сущности, являющиеся сегментами виртуальной памяти ОС.

Предполагается, что если субъект реализовал информационные потоки по памяти от всех сущностей из ПАС-множества другого субъекта к себе, то первый субъект получает право доступа владения ко второму субъекту. В соответствии с этим предположением вместо правила $know(x, y, z)$ вводится правило $know(x, y, Z)$.

3. Дополнительно к правам доступа, традиционно рассматриваемым в ДП-моделях, вводятся права доступа, специфичные для конкретной КС сетевого управления доступом. Примерами таких прав доступа являются $access_r$ — право доступа к сущности сетевой КС и $control_r$ — право конфигурирования сущностей сетевой КС. При добавлении значительного числа новых прав доступа выполняется агрегирование прав доступа — замена нескольких прав доступа одним с более высоким уровнем абстракции.

В некоторых сетевых КС, например в Cisco Secure Access Control Server, можно явно разрешить или запретить субъектам выполнение определенных команд в рамках сессии конфигурирования сущности-узла КС. Формальное описание данного механизма возможно с использованием элементов ролевого управления доступом совместно с агрегированием прав доступа.

4. Для формального описания элементов управления доступом на основе правил доступа и функций иерархии сущностей вводится множество учетных записей U, от имени которых субъекты реализуют доступ к сущностям КС, и множество векторов доступа V, описывающих права доступа некоторой учетной записи $u \in U$ с сущности-контейнера (узла) $c \in C$ к сущности $e \in E$ и определяемых следующим образом.

Пусть имеются: множество прав доступа R, учетная запись $u \in U$, сущность $e \in E$ и контейнеры $c_1, \ldots, c_n \in C$, такие, что $u < c_i$ для $i = 1, \ldots, n$. Пусть также есть функция

$f : U \times E \times C \to 2^R$, описывающая права доступа учетной записи $u \in U$ к сущности $e \in E$ при инициализации сессии с контейнера $c \in C$. Тогда вектором доступа учетной записи u к сущности e будем называть набор пар $(c_1, f(u, e, c_1)), \ldots, (c_n, f(u, e, c_n))$.

5. В соответствии с этим определением к правилам преобразования состояний добавляется правило $create\_session\_remote(s, u, c, e)$, описывающее создание сессии удаленного доступа субъектом $s \in S$ с правами доступа учетной записи $u \in U$ к сущности $e \in E$ и назначение субъекту s в рамках этой сессии прав доступа учетной записи u в зависимости от ребра (u, e, v) в графе доступа и контейнера c, с которого порождена сессия субъектом s.

С использованим этих расширений языка ДП-моделей базовая СУД ДП-модель может быть разработана по стандартной схеме построения ДП-моделей в [1].

ЛИТЕРАТУРА

1. *Девянин П. Н.* Модели безопасности компьютерных систем. Управление доступом и информационными потоками. Учеб. пособие для вузов. М.: Горячая линия-Телеком, 2011.

УДК 004.75

МОДЕЛЬ БЕЗОПАСНОСТИ КРОСС-ПЛАТФОРМЕННЫХ ВЕБ-СЕРВИСОВ ПОДДЕРЖКИ МУНИЦИПАЛЬНЫХ ЗАКУПОК

Д. Д. Кононов, С. В. Исаев

Целью работы является создание на основе RBAC [1] модели безопасности, учитывающей специфику веб-приложений. На базе этой модели разработаны программные средства для решения задач муниципального заказа администрации г. Красноярска, в том числе для проведения открытых электронных аукционов. Юридическая значимость действий пользователей обеспечивается использованием электронной цифровой подписи. Работа выполняется в рамках развития Автоматизированной системы проведения муниципальных заказов (АСП МЗ). Определяющими документами являются федеральные законы №1, 94 и 149.

В результате анализа предметной области была выбрана модель RBAC. В модель были внесены изменения, учитывающие особенности веб-приложений. К существующим понятиям «субъект» (subject), «роль» (role), «разрешение» (permission), «сессия» (session) были добавлены «токен» (token) и «запрос» (request).

Определение 1. *Токен* — набор атрибутов субъекта, позволяющих осуществить его аутентификацию в системе. Токеном является пара (имя, пароль) либо пара (сертификат ЭЦП, закрытый ключ ЭЦП).

Определение 2. *Запрос* — набор информации, пересылаемой клиентом серверу по протоколу HTTP. Запрос содержит набор заголовков, уникальный идентификатор ресурса, набор параметров имя/значение и тело запроса.

Запрос принадлежит сессии, в рамках одной сессии может выполняться несколько запросов. Понятия «запрос» и «разрешение» связаны отношением «многие-ко-многим». На множестве запросов вводится отношение включения.

Определение 3. Запрос A *включает* запрос B, если путь уникального идентификатора ресурса запроса A содержит путь уникального идентификатора ресурса запроса B с начальной позиции строки.