УДК 004.94

О МЕТОДАХ ПРЕДОТВРАЩЕНИЯ ЗАПРЕЩЁННЫХ ИНФОРМАЦИОННЫХ ПОТОКОВ ПО ВРЕМЕНИ НА ОСНОВЕ ПРОГРАММНОГО ИНТЕРФЕЙСА СОКЕТОВ

А. Н. Долгих

Анализ безопасности управления доступом и информационными потоками в компьютерных системах (КС) является одной из приоритетных задач компьютерной безопасности. Значительную сложность представляет идентификация и предотвращение информационных потоков по памяти и по времени, которые могут быть реализованы на основе разных механизмов и средств КС [1]. Различные виды информационных потоков рассмотрены в работах [2, 3].

Одним из известных средств реализации запрещённых информационных потоков по времени в операционных системах (ОС) является использование программного интерфейса сокетов.

Для реализации такого информационного потока используется тот факт, что если создан слушающий сокет одним субъектом-процессом ОС, то другой субъект-процесс не может создать слушающий сокет на этом же порту, и механизмами ядра ОС будет возвращена ошибка. Таким образом, по факту наличия ошибки при создании слушающего сокета возможна передача данных между различными субъектами-процессами ОС: если субъект-процесс S_1 хочет передать 1, то он создаёт слушающий сокет на некотором заданном порту, если 0 — не создаёт. Невозможность создания слушающего сокета на том же порту субъектом-процессом S_2 интерпретируется как передача значения 1, в противном случае — значения 0. При этом одним из необходимых условий передачи данных без ошибок является синхронизация действий взаимодействующих субъектов.

Данный информационный поток по времени реализован в ОС GNU/Linux на языке программирования C++. Скорость передачи данных в условиях, когда в ОС не запущено ресурсоёмких приложений и субъекты, не участвующие в реализации потока, не создают сокетов на заданном порту, составила 100 бит/с.

Известны следующие методы защиты от реализации запрещённого информационного потока по времени на основе сокетов. В ОС AsbestOS [2] субъектам запрещено создавать слушающий сокет на порту с определённым номером. В ответ на запрос о создании сокета ядро ОС возвращает непредсказуемый номер порта. Вводятся переменные окружения, в которых хранятся номера портов используемых в настоящее время сервисов. Следовательно, по факту существования или отсутствия переменной может быть реализован информационный поток по времени.

В работе [3] предложен метод, использующий блокирующий доступ доверенных субъектов к сущностям с целью зашумления скрытого канала передачи данных. На практике для применения предложенного метода необходимо идентифицировать предполагаемый запрещённый информационный поток в КС. Для решения этой задачи предлагается использовать механизмы регистрации событий: записывать идентификатор процесса, номер порта, на котором этот процесс создаёт слушающий сокет, а также продолжительность времени, в течение которого порт был заблокирован этим процессом. Если одно и то же событие наблюдается в течение некоторого промежутка времени, то применить блокирующий доступ к порту. С целью увеличения вероятности идентификации потока таким способом можно также регистрировать факт наличия передачи данных на этом порту.

Другим способом снижения пропускной способности рассматриваемого информационного потока является контроль частоты создания слушающих сокетов. Однако введение в ОС таких изменений может значительно снизить скорость взаимодействия процессов между собой с использованием интерфейса сокетов.

ЛИТЕРАТУРА

1. ГОСТ Р 53113-2008. Информационная технология. Защита информационных технологий и автоматизированных систем от угроз информационной безопасности, реализуемых с использованием скрытых каналов. Ч. 1. Общие положения. М.: Стандарты, 2008.
2. *Efstathopoulos P. O. and Krohn И. O.* Labels and Event Processes in the Asbestos Operating System // Proceedings of the SOOP'05. ACM, 2005.
3. *Девянин П. Н.* Анализ безопасности управления доступом и информационными потоками в компьютерных системах. М.: Радио, 2006. 176 с.

УДК 004.94

О ПОСТРОЕНИИ ИЕРАРХИЧЕСКОГО РОЛЕВОГО УПРАВЛЕНИЯ ДОСТУПОМ

Д. Н. Колегов

Рассматривается подход к построению ролевого управления доступом для компьютерных систем (КС) с иерархией сущностей, отражающей организационно-управленческие отношения.

В моделях ролевого управления доступом семейства *RBAC* [1], как и в других известных ролевых моделях и их расширениях [2, 3], не используются механизмы задания и проверки разрешённых прав доступов субъекта к сущности, учитывающие уровни иерархии сущностей КС.

В реальных КС, в которых одновременно могут работать сотни пользователей, структура ролей может быть очень сложной, а количество различных прав доступа значительным — проблема реализации и администрирования системы управления доступом является чрезвычайно важной задачей [4].

Для решения этой задачи строится расширение базовой модели *RBAC*, содержащее в дополнение к последней следующие положения:

— задана решётка уровней иерархии КС и для каждой сущности указан уровень иерархии;
— определено множество типов сущностей КС и для каждой сущности указан её тип;
— задано множество ролей, каждая из которых представляет собой некоторое множество прав доступа к сущностям определённого типа;
— каждый субъект обладает некоторым множеством разрешённых для данного субъекта ролей;
— субъект обладает правом доступа к сущности КС в том и только в том случае, если субъект обладает ролью, в множестве прав доступа которой имеется данное право доступа к сущности данного типа и уровень иерархии субъекта не меньше уровня иерархии сущности.

Использование введённых положений позволяет значительно эффективнее и гибче реализовать механизм ролевого управления доступом по сравнению с моделями *RBAC*.

На основе [4] опишем формальную структуру предлагаемой модели.

Основными элементами модели иерархического ролевого управления доступом, обозначаемой *RBAC-H*, являются: