функции – команды, которые поддерживает данная VM. Данные команды содержатся в таблице хендлеров. Зная описания каждой команды в таблице хендлеров мы знаем как работает VM.

Если VM использует таблицу хендлеров, то её правильное декодирование поможет сразу, в статическом режиме, получить хендлеры VM, а в дальнейшем – поставить точки останова в ней, что предоставит возможность снятия трассировок. Это крайне важный момент, который обеспечит быстрый анализ VM на низком уровне. Если удастся собрать трассировку выполнения VM, то можно построить некоторую модель поведения, которую можно использовать в качестве детектирования для вредоносной программы. Что, собственно, представляет собой таблица хендлеров и что понимается под словом хендлер?

VM – в первую очередь, это процессор, который имеет свой ограниченный набор инструкций, которые мы и называем хендлером: прочитать, записать, сложить и т.д. В подавляющим большинстве случаев этот размер = 256 (0-0xFF, 1 байт) [2]. Это размер по умолчанию для таблицы хендлеров. Процессор должен знать, точный адрес, где находится каждая инструкция, чтобы её использовать.

Любой исполняемый файл, защищенный VM, может рассматриваться вирусным аналитиком или исследователем не только, как упакованный разными упаковщиками исполняемых файлов, например, UPX. Из него получается следующую информация:

– общее количество VM, которые содержаться в файле;

– таблица хендлеров для каждой машины. Типы хендлеров. Характерная модель поведения VM, присущая именно этой программе;

– на основании таблицы хендлеров составляется процент покрытия кодом VM в соотношении с основным незашифрованным кодом программы.

Как минимум, эти метрики и данные используются в качестве детектирования похожих вредоносных файлов. А собственно, как максимум – возможность девиртуализации кода, защищенного VM.

В итоге имеем общую схему внутреннего устройства виртуальной машины (см. Рис. 3).

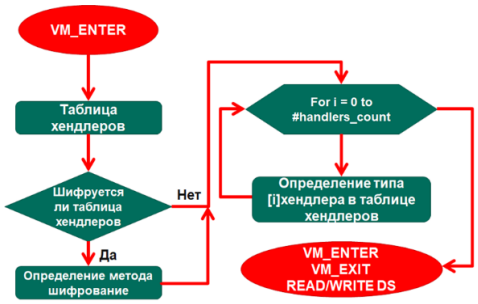

Рисунок 3 – Внутреннее устройство виртуальной машины

Таким образом, виртуальные машины могут рассматриваться как одно из эффективных и удобных средств защиты программного обеспечения от злонамеренных воздействий со стороны как вирусного ПО, так и других хакерских приемов.

Литература

1. Крепыш, 2011. Хабрахабр. https://habr.com/ru/sandbox/26302/.

2. 2019. VMProtect – Справочник исследователя программ. https://exelab.ru/faq/VMProtect.

3. eXeL@B, 2019 Реверс VMProtect. https://exelab.ru/f/index.php?action=vthread&forum=13&topic= 25221.

УДК 519.712.2

О ПРОБЛЕМЕ РЕАЛИЗАЦИИ ДИСПЕТЧЕРА ДОСТУПА К ДОКУМЕНТАМ

Лебедев Г.А., Родионов Д.Е., Лебедев А.Н.

Московский государственный технический университет имени Н.Э.Баумана
Москва, Российская Федерация

В любой автоматизированной системе в защищенном исполнении (АСЗИ) механизм управления доступом субъектов к объектам доступа выполняет основную роль в обеспечении конфиденциальности защищаемых данных. Реализация этого механизма обычно строится на концепции единого диспетчера доступа (ДД). Сущность этой концепции состоит в том, что диспетчер доступа выступает посредником-контролером при всех обращениях всех субъектов системы к ее объектам.

В качестве базовой платформы для использования в АСЗИ выступает операционная система Astra Linux Special Edition. Данная операционная система серифицированна в Российской Федерации для обработки информации ограниченного доступа, различных уровней конфиденциальности [1].

При разработке АСЗИ возникает задача реализации механизма разграничения доступа к объектам (документам), доступ к которым не может быть разграничен непосредственно ОС, так как в ОС объектами разграничения длоступа являются, в первую очередь, файлы. В этом случае возникает задача разработки защищенного приложения – ДД для реализации данных функ-

циональных возможностей [2]. Решение этой задачи, первым этапом требует проведение анализа существующих механизмов безопасности ОС для определения механизмов, которые могут быть использованы для основы ДД. На втором этапе выполнена разработка архитектуры приложения, реализующего функции диспетчера доступа.

Для реализации диспетчера доступа к электронным документам под управлением ОС Astra Linux SE невозможно использовать существующие механизмы разграничения доступа, поскольку «электронный документ» может состоять из набора следующих элементов:

– набор файлов (например, с телом документа и его версиями);

– набор метаданных (например, реквизиты документа, фактографические данные).

Набор файлов электронного документа должен храниться и обрабатываться непосредственно в файловой системе (файловом хранилище), а метаданные документа гораздо удобней хранить непосредственно в СУБД. Однако в этом случае для организации работоспособной системы разграничения доступа к таким электронным документам использовать отдельно только механизмы разграничения доступа ОС или только механизмы разграничения доступа СУБД не получится, так как нельзя полностью отобразить такой «электронный документ» только в объекты файловой системы или в только в объекты под управлением СУБД.

Таким образом, необходимо разработать некоторую новую сущность – диспетчера доступа, разграничивающий доступ к объектам доступа типа «документ» под управлением ОС Astra Linux.

ДД является промежуточным звеном между сервером системы и клиентскими приложениями, все соединения устанавливаются через него и им контролируются, кроме того, в ДД должен быть реализован весь функционал разграничения доступа к объектам защиты.

Рассмотрим основные функциональные задачи, возлагаемые на диспетчер досткпа.

Поддержка соединений клиента с сервером приложений. Все соединения между клиентскими приложениями и сервером системы устанавливаются с ДД. В свою очередь ДД соединяется с сервером приложений, открывая новые соединения с сервером для каждого установленного соединения с клиентскими рабочими местами.

Таким образом, ДД получает возможность анализировать весь поток данных между клиентом и сервером приложений, при этом он имеет возможность выяснять корректность всех передаваемых клиентскими приложениями запросов к серверу приложений с точки зрения установленной в системе политики безопасности и, в случае необходимости, может не пропускать некорректные запросы клиентских приложений или некорректные с точки зрения политики безопасности ответы сервера приложений.

При этом ДД сам формирует сообщения об ошибке в форматах, принятых для отвергнутых запросов, и передает их на клиентские приложения.

Получение от базовой ОС идентификационной информации о пользователях. Указанная задача является одной из основных задач для ДД. Извлекаемая им из контекста безопасности, полученного из именованного канала обмена данными, информация используется как для идентификации и аутентификации пользователей при их входе в систему, так и для выяснения различных характеристик (свойств) пользователей, учитываемых при работе системы, а именно, членство пользователей в различных группах, задаваемых в рамках системы управления учетными записями, наличие у них определенных административных привилегий и другие.

В базе данных учетных записей хранится список идентификаторов (SID) всех пользователей, которые имеют право работать с данным сервером, а также список идентификаторов (SID) всех учетных записей, имеющих права для запуска сервисов системы обмена документами.

При попытке соединения с сервером первоначально выполняется проверка наличия SID соединяющегося пользователя в этом списке.

Контроль доступа. Для реализации принципов дискреционного контроля доступа пользователей к электронным документам, в базе данных ДД хранится вся необходимая идентифицирующая информация о документах (объектах доступа) и списки доступа пользователей (субъектов доступа) к каждому из указанных документов.

Проверка допустимости операций осуществляется при получении с клиентского места каждого запроса на выдачу документа для чтения или записи (редактирования), а также при получении запросов пользователей на удаление документа. Дополнительно проверяется допустимость операции выдачи пользователю каждого файла, содержащегося в рамках данного электронного документа.

В базе данных ДД хранятся числовые значения, определяющие уровень доступа документа и его мандатные метки. Проверка допустимости выдачи документа конкретному пользователю производится непосредственно при получении его запроса на выполнение конкретного запрашиваемого действия с документом.

В базе данных диспетчера доступа хранятся все необходимые атрибуты доступа, проверка допустимости выполняется как при получении запроса на выдачу объекта, так и при отправке клиентскому приложению полученного от Сервера приложений пакета данных.

Протоколирование, оповещение и блокировки. ДД выполняет записи в протокол безопасности как самостоятельно, так и по команде от клиентских приложений. Самостоятельно протоколируются плановые действия такие, как вход пользователя в систему, изменение прав доступа к объектам, выдача объектов пользователям для просмотра и редактирования.

Также ДД протоколирует события, которые могут рассматриваться как попытки несанкционированного доступа, например, запрос объекта, не стоящего на контроле ДД. По команде с клиентских приложений протоколируются так называемые «события второй категории», например, отрицательный результат проверки электронной подписи пользователя под конкретным файлом, что может быть следствием искажения содержащегося в файле документа при его хранении или передаче по каналам связи.

Для событий, требующих немедленного вмешательства администратора безопасности, должна быть предусмотрена отправка ДД сообщения на АРМ аудитора системы (он же АРМ администратора безопасности ОС).

Должна быть предусмотрена также возможность блокировки документов в случаях, когда администратору безопасности необходимо прекратить работу пользователей системы с документом для расследования внештатных ситуаций (попытки несанкционированного доступа, некорректные действия сотрудников).

В этом случае документ блокируется по команде с АРМ администратора ОС или с клиентского рабочего места (автоматически, скрытно от пользователя), после чего этот документ не может быть выдан ни одному пользователю, кроме администратора безопасности. Разблокирование документа доступно только администраторам безопасности.

Алгоритм работы диспетчера доступа. Алгоритм функционирования диспетчера доступа к документам для операционной системы Astra Linux состоит в следующем:

– пользователь обращается к web-серверу с запросом на доступ;

– сервер проверяет результаты идентификации пользователя (в случае использования сервиса Astra Linux Directory и единого пространства пользователей) или запрашивает у пользователя логин и пароль для собственной автономной процедуры идентификации и аутентификации;

– в случае если пользователь успешно идентифицирован, web-сервер предоставляет пользователю список документов, иначе передается сообщение об отказе в доступе;

– пользователь выбирает документ из предоставленного списка документов;

– web-сервер передает запрос на доступ (в составе идентификатора документа, идентификатора пользователя, типа доступа) программному модулю диспетчера доступа;

– модуль диспетчера доступа проверяет установленные права доступа к документу для пользователя, по идентификатору документа из запроса и идентификатору пользователя из запроса;

– при разрешении на доступ модуль диспетчера доступа возвращает web-серверу карточку документа (список файлов и метаданные документа);

– при отказе в доступе модуль диспетчера доступа возвращает web-серверу сообщение об отказе в доступе.

Литература

1. П.В. Буренин, П.Н. Девянин, Е.В. Лебеденко, В.Г Проскурин, А.Н. Цибуля: Безопасность операционной системы специального назначения Astra Linux Special Edition, М. Учебное пособие для вузов / Под редакцией доктора тех. наук П.Н. Девянина.

2. А.А. Грушо, Э.А. Применко, Е.Е. Тимонина: Теоретические основы компьютерной безопасности, учебное пособие для студентов высших учебных заведений / Издат.центр «Академия», 2009.

УДК 519.712.2

ЭКСПЕРИМЕНТАЛЬНЫЕ ОЦЕНКИ МИНИМАЛЬНОЙ ДЛИНЫ РАЗДЕЛЯЮЩЕЙ ПОСЛЕДОВАТЕЛЬНОСТИ ЛЕЖАНДРА

Лебедев А.Н., Кокорин А.О., Савельев Е.Д.

Московский государственный технический университет имени Н. Э. Баумана
Москва, Российская Федерация

Введение. Для создания специализированных микроконтроллеров, реализующих оптимальным образом однонаправленные функции криптографических преобразований, составляющих базовые компоненты протокола выработки общего секрета Диффи – Хеллмана важно точно представлять на какой минимальной длине различных двоичных представлений чисел по модулю большого простого числа можно гарантировать однозначность такого представления чисел. Поэтому мы провели целый ряд экспериментальных исследований, позволяющих выдвинуть достаточно обоснованные гипотезы о характере поведения минимальной длины такого представления на основе последовательностей Лежандра.

Дадим определение последовательности Лежандра [1]. Пусть p – простое число. Целое число a называется квадратичным вычетом по модулю p,