электронный журнал
МОЛОДЕЖНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51038.

УДК 004.056.53

Основные аспекты защиты автоматизированных систем от угроз, связанных с эмуляцией HID-устройств

Полухин Я. И., *студент*
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Информационная безопасность»

Научный руководитель: Глинская Е.В., к.т.н, доцент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Информационная безопасность»
v.a.matveev@bmstu.ru

В современном мире, наверное, у каждого человека, хоть как-то связанного с вычислительными системами, имеется собственный флеш-накопитель, или попросту — флешка. Но далеко не каждый задумывается, какую угрозу может нести обыкновенная с виду флеш-карта, и уж тем более не задумываются о том, что существуют флешки, которые хоть и выглядят как таковые визуально, но сами по себе флешками и не являются. В данной работе будут рассмотрены подобные устройства, а также предложены способы защиты от такого рода девайсов.

Всем известно, что обмен данных между различными устройствами и флешкой происходит через универсальную последовательную шину (Universal Serial Bus), которая сокращенно называется USB. USB — это последовательный интерфейс передачи данных для среднескоростных и низкоскоростных периферийных устройств в вычислительной технике. Разработка спецификаций на шину USB производится в рамках международной некоммерческой организации USB Implementers Forum, объединяющей разработчиков и производителей оборудования с шиной USB. Если же внимательно изучить спецификации данного интерфейса можно обнаружить две всем известные, но, тем не менее, интересные особенности. Во-первых, через данный интерфейс может подключаться множество различных устройств: USB-накопители и карты памяти, модемы и сетевые карты, клавиатуры и мыши, называемые устройствами взаимодействия с человеком или HID-устройствами, и т.д. А во-вторых, с точки зрения USB-интерфейса все эти устройства являются доверенными. Действительно, при подключении все они без каких-либо разрешений опознаются системой, к ним подбираются драйвера и начинается полноценный процесс взаимодействия между устройством и системой.

Именно эти две особенности лежат в основе атаки, построенной на эмуляции различных USB-устройств. О возможности таких атак известно довольно давно, но во всеуслышание о них заговорили сравнительно недавно — на конференции DEFCON 18 в 2010 году, где IT-специалист Адриан Креншоу подробно описал использование HID-эмулирующих устройств для проверки безопасности систем. В своем выступлении Креншоу продемонстрировал возможность эмуляции клавиатуры и мыши, а также рассказал, как такой метод может применяться в тестировании на проникновение. Разберем основные моменты подобной атаки на системы.

На подготовительной стадии стоит задача создать устройство, которое должно отвечать двум главным требованиям: подключаться к системе через USB-интерфейс и уметь эмулировать устройства взаимодействия с человеком.

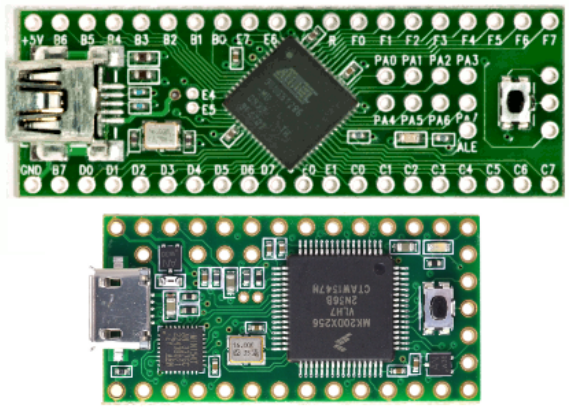

Рис. 1. Платы Teensy: Teensy ++ 2.0 (вверху), Teensy 3.1 (внизу)

Существует немало устройств, которые соответствуют вышеуказанным требованиям, но наиболее удобными для этого признаны платы Teensy (рисунок 1). Это программируемый микроконтроллер, который изначально идет вместе с полноценным USB-портом. Главным его плюсом является то, что для реализации эмулирующего устройства не требуется никакого физического вмешательства, а достаточно лишь написать соответствующую программную прошивку.

Но для проведения атаки этого мало. Есть еще один критерий, без которого эмулирующее устройство не может стать атакующим, — маскировка. Очевидно, что никто не будет подключать к системе неизвестную плату, а одним из этапов атаки является именно подключение. Из этого следуют еще два необходимых критерия: устройство должно выглядеть как обыкновенная флешка и иметь в своем составе рабочий флеш-накопитель.

Далее можно пойти двумя путями: вооружиться паяльником и собрать, так называемую, Peensy (от слияния Pentest + Teensy) и замаскировать ее или купить уже готовый вариант — USB Rubber Ducky Deluxe (рисунок 2).

Рис. 2. Peensy (слева), USB Rubber Ducky Deluxe (справа)

Еще одной необходимой составляющей такого устройства является программная прошивка, которая позволит выполнять необходимые атакующему действия. Изначально для этого требовался определенный набор знаний по программированию микроконтроллеров, но с тех пор многое изменилось. Появилось несколько программных пакетов, называемых тулкитами, цель которых автоматизировать процесс написания кода для Teensy. Пользователь такого пакета выбирает в нем устройство, которое он будет программировать, и устройство, которое хочет эмулировать, а также необходимую ему полезную нагрузку. Например, выбранным устройством является Teensy ++ 2.0, которое будет эмулировать клавиатуру, а в качестве полезной нагрузки можно выбрать «Скачать и выполнить», что подразумевает скачивание из интернета файла и его выполнение в системе. Всю работу тулкит делает самостоятельно, а на выходе получается файл с прошивкой для соответствующего микроконтроллера. Сейчас наиболее известными такими программными пакетами являются Social Engineering Toolkit, который является предустановленным в Kali-linux, а также Kautilya, написанный энтузиастом по имени Nikhil Mittal. Даже если не пользоваться подобными пакетами, в Интернете можно найти множество практических руководств, а также примеров подобных прошивок со всевозможными полезными нагрузками. Из вышесказанного можно сделать следующий вывод: для создания подобных девайсов не требуется никаких специальных знаний и навыков. Любой желающий может купить микроконтроллер, скачать прошивку для него и получить эмулирующее устройство, а, замаскировав его под флеш-карту, еще и атакующее.

Теперь рассмотрим в деталях процесс активной атаки с помощью вышеописанного устройства. Это необходимо для того, чтобы наглядно показать на каких шагах атаки возможно, а так же целесообразно, бороться с подобными угрозами безопасности системы. Целью атаки является полноценное выполнение устройством команд, указанных в прошивке, в атакуемой системе. Можно выделить четыре основных шага, которые будут встречаться в любой вариации подобной атаки:

1. Доставка атакующего устройства к автоматизированной системе (АС);

2. Подключение атакующего устройства к АС и его инициализация;
3. Ввод команд атакующим устройством;
4. Непосредственное выполнение введенных команд в системе.

Рассмотрим по порядку эти этапы. Устройство не может само по себе появиться в USB-разъеме автоматизированной системы. Между стадией создания устройства и его подключением оно как минимум было доставлено в помещение, где расположена атакуемая система, а так же кто-то должен подключить его к АС. Возможно два варианта развития событий: либо атакующий имеет доступ в помещение с АС и самостоятельно выполняет этот шаг атаки, либо не имеет, и тогда для выполнения этапа ему необходимо воспользоваться сотрудником с соответствующим доступом. Вариант с его собственным проникновением на территорию предприятия не рассматривается, так как обеспечение физической защиты предприятий является сложной темой, требующей отдельной проработки, и не сильно относящейся к данной атаке. Для выполнения второго варианта применяются навыки социальной инженерии. Например, можно просто подбросить в офис компании атакующие устройства и сыграть на человеческом любопытстве, или же притвориться коммивояжёром и в качестве рекламной акции предоставить атакуемой компании не только рекламные каталоги, но и несколько флешек. Вариантов может быть множество.

Итак, злоумышленник тем или иным образом добрался до автоматизированной системы. Как же будут развиваться события дальше и будут ли вообще? Не сложно придумать пример, в котором на данном моменте для атакующего все закончится. А именно, если у атакуемой АС нет USB-портов. Поэтому можно выделить ряд требований к автоматизированной системе для успешного выполнения этого шага атаки: наличие рабочих USB-портов, наличие операционной системы (ОС), способной работать с HID-устройствами по USB, наличие соответствующих драйверов в ОС. Но эти требования не являются проблемой для атакующего, так как все современные АС изначально соответствуют этим требованиям.

После того как устройство будет опознано операционной системой, начинается ввод различных команд, подготовленных атакующим, для выполнения в данной автоматизированной системе. Так как Teensy является полноценным эмулятором клавиатуры и мыши и даже опознается именно как USB-клавиатура/мышь, то дальше будут происходить обыкновенные нажатия клавиш, как будто бы их нажимает пользователь АС. И все введенные команды будут выполняться, так как система будет считать, что их вводит легитимный пользователь с легитимного устройства. А

выполнение этих команд, как было указано выше, и является конечной целью данной атаки.

Приведу пример реализации подобной атаки, опубликованный в 2012 году в журнале «Хакер»: «Я выполнял тест на проникновение для крупной индийской финансовой компании. Это был тест типа «черный ящик», то есть я был поставлен в условия настоящего злоумышленника без доступа к чему-либо. Серверы компании, доступные из интернета, были надежно защищены, и придраться там было не к чему. Что делать, непонятно. Тогда я пошел в их офис и сказал охранникам на входе, что я нашел несколько мышек и флешек, которые выпали у какого-то работника из сумки. Все устройства были протроянены Teensy и содержали различные пэйлоады. Через полчаса я получил первый шелл (скорее всего, это был компьютер в комнате охраны). К концу дня у меня уже были привилегии локального администратора на многих компьютерах — судя по всему, охранники передали устройства в IT-отдел. Из протроянных устройств 90% процентов были подключены хотя бы к одной системе, а 70% успешно выполнили заложенный в них пэйлоад. Клиент, вбухавший в безопасность кучу денег, был сильно удивлен и впечатлен подобным способом проникновения».

Именно так выглядит практическое применение данной атаки в тестировании на проникновение. Как можно заметить к такому вектору атак оказываются не готовы даже крупные компании, которые подходят к вопросам защиты информации очень тщательно. Поэтому теперь, когда были выявлены основные моменты атаки, необходимо определить, что может противопоставить атакующему защищающаяся сторона. Для этого еще раз пройдем по всем пяти шагам атаки, и попробуем выделить, какими же методами можно предотвратить или хотя бы минимизировать возможность проведения такого воздействия на систему.

На подготовительном этапе очень сложно что-то противопоставить злоумышленнику. Очевидно, что отследить и тем более предотвратить процесс создания HID-эмуляторов невозможно. Но гораздо хуже другое: невозможно даже усложнить процесс создания подобных устройств ни с одной из сторон: ни с практической, ни с юридической, ни с финансовой.

Говоря о практической стороне, выше уже было продемонстрировано, что реализовать данный девайс крайне просто. Вместо того, чтобы покупать микроконтроллер, припаивать к нему необходимое периферийное оборудование и маскировать его, есть возможность купить готовое устройство (USB Rubber Ducky Deluxe), причем изначально выглядящее как обыкновенная флешка. Вместо того, чтобы

изучать литературу по программированию микроконтроллеров и самому создавать прошивку для устройства, достаточно скачать специальный программный пакет, который выполнит эту работу самостоятельно. То есть, одно купленное устройство и скаченный тулкит позволяют подготовить атакующее устройство быстро, а главное с очень низкой трудозатратностью.

Если рассмотреть правовой аспект создания HID-эмуляторов, то легко заметить, что данное действие не нарушает никаких правовых актов. Наиболее подходящей кажется 273 статья УК РФ «Создание, использование и распространение вредоносных программ для ЭВМ». Но при ближайшем рассмотрении становится видно, что это не так. В данной статье УК присутствует такое определение вредоносной программы: «Вредоносная компьютерная программа — компьютерная программа либо иная компьютерная информация, заведомо предназначенная для несанкционированного уничтожения, блокирования, модификации, копирования компьютерной информации или нейтрализации средств защиты компьютерной информации». Ключевыми словами данного определения являются «компьютерная программа... заведомо предназначенная для...». Практически невозможно доказать, что подобное устройство или же его программная прошивка заведомо предназначены для нанесения ущерба. Например, подобные девайсы с подобными прошивками могут быть использованы системными администраторами для автоматизации своих служебных обязанностей. Единственным исключением является прошивка контроллера, которая в своем составе имеет ярко выраженный функционал, направленный на уничтожение информации системы или нейтрализации СЗИ. Но даже в таком случае, можно найти законные объяснения.

Также не существует механизмов влияния на финансовую сторону вопроса. Компании-изготовители микроконтроллеров имеют определенную ценовую политику, изменить которую невозможно. Таким образом, цена на плату Teensy ++ 2.0 составляет $24, на более свежую версию Teensy 3.1 — $19.80, а на уже готовый к атаке USB Rubber Ducky Deluxe — $42.99. В свою очередь тулкиты для написания прошивки распространяются энтузиастами абсолютно бесплатно. В результате цена данной атаки состоит только из цены на микроконтроллер, которая не является очень высокой, и которая может быть гораздо ниже цены потерянной в результате данной атаки информации.

Теперь рассмотрим этап доставки атакующего устройства. Как упоминалось выше, возможны два сценария атаки в зависимости от возможности доступа в помещение с автоматизированной системой. Стоит сразу упомянуть, что при составлении модели

нарушителей подобной АС, в обязательном порядке нужно учитывать как внутренних, так и внешних нарушителей. Но в обоих случаях наилучшим средством защиты будет являться комплекс организационных и физических мер безопасности. Разберем подробнее оба случая. В первом случае, когда атакующий имеет доступ в помещение, главным защитным средством должно являться свойство неотказуемости. Неотказуемость — способность удостоверять имевшее место действие или событие так, что эти события или действия не могли быть позже отвергнуты. В данном случае это свойство может реализовываться следующими способами: полное журналирование всех посещений помещения с АС, видеонаблюдение за самой системой. Также, чтобы минимизировать риск реальной атаки внутренним нарушителем, необходимо проводить контроль кандидатов на должности, связанных с подобными АС, и ограничивать доступ для неблагонадежных сотрудников к ней. В случае, когда роль доставщика устройства к автоматизированной системе выполняет жертва социальной инженерии, которая имеет доступ к АС, обнаружить данную атаку практически невозможно. Самым лучшим защитным механизмом в такой ситуации является высокая осведомленность каждым сотрудником базовых принципов информационной безопасности, в частности защиты от социальной инженерии.

Наиболее удобным этапом для предотвращения атаки является этап подключения к автоматизированной системе. Самым эффективным способом предотвращения атаки является полное отключение USB-портов АС. Это можно достичь как физическим отключением портов, так и программным. Но у этого подхода есть один существенный недостаток — эти порты могут требоваться для полноценной работы данной системы. А если учесть тот факт, что флеш-накопители — это один из самых используемых носителей информации, то отсутствие рабочих USB-портов может быть критичным. Менее глобальным является разрешение на подключение устройств по белому списку. Белый список подразумевает, что только разрешенные устройства могут быть подключены к АС. Но и этот метод не является полностью эффективным по двум причинам. Во-первых, как и при полном отключении, это может создать огромные неудобства при эксплуатации системы, а во-вторых, существует возможность выдать одно устройство за другое, что позволяет обходить черные и белые списки. По той же причине не являются эффективными современные антивирусы, которые определяют тип устройства по тем же признакам, по которым блокируют их. И в результате отличить Teensy от зарегистрированного в системе устройства может быть невозможно. Наиболее эффективным можно считать самый банальный способ: так как атакующее устройство

опознается системой как клавиатура или мышь, можно сделать простейшую «сигнализацию», которая выдает сообщение о подключении соответствующего HID-устройства. Пользователь видит, что было подключено устройство, которое он не подключал и немедленно реагирует на это. Единственным недостатком данного метода является то, что он эффективен только для внешних нарушителей.

Особняком стоит теоретический метод, который является улучшением вышеописанной «сигнализации» — проверка подлинности подключенной клавиатуры или мыши. Суть этого метода заключается в том, что после того как система полностью опознала новое устройство, нужно потребовать, чтобы это устройство сделало определенные действия для проверки своей подлинности. Например, при подключении клавиатуры система выдает специальное окно, которое невозможно закрыть иным путем кроме как, выполнив действия указанные в нем. В этом окне необходимо ввести с этой клавиатуры несколько символов, которые выбираются случайным образом, за небольшой промежуток времени. В случае верного текста окно пропадает, в случае неверного — срабатывает условная «сигнализация», блокирующая данный USB-порт. Такой метод имеет потенциальную эффективность, но до данного момента не был реализован.

На этапе ввода команд не существует реализованных средств для обнаружения и предотвращения атаки, но можно предложить теоретический метод, который при реализации позволит эффективно достичь заданной цели. Данный метод основан на поведенческом анализе действий системы, а именно на скорости ввода с клавиатуры. Суть данного метода заключается в следующем: на стадии обучения подсчитывается средняя скорость нажатий клавиш для данного пользователя, при подключении Teensy скорость ввода с клавиатуры резко возрастает и программа должным образом реагирует на такое изменение.

На последнем этапе уже существует ряд методов для противодействия атакующему, но на нем очень сложно найти средства для полной остановки атаки, поэтому все предложенные средства только уменьшают вероятность успеха атаки. Во-первых, для наибольшей минимизации несанкционированного воздействия необходимо пользоваться "золотым" правилом любого администратора безопасности: "Никто из рядовых пользователей не должен иметь прав администратора в защищаемой АС". Такое правило позволяет обезопасить систему даже от самых банальных команд таких как, например, *net user ... /add* — создание пользователя в системе Windows. Во-вторых, существуют некоторые способы ограничения выполнения программ в системе. Так в системе Windows предусмотрен подобный механизм защиты — UAC (User Account

Control, контроль учётных записей пользователей). В зависимости от учетной записи пользователя UAC требует подтверждения разрешения на выполнение программы. Но в случае с HID-эмуляторами это средство не очень эффективно, так как возможно автоматизировать процесс нажатие кнопки разрешения. Более эффективным способом ограничить выполнение программ является наличие грамотно составленных групповых политик. Еще одним потенциально эффективным способом является поведенческий анализ команд, введенных в командную строку или ее аналоги. Анализировать вводимые команды можно либо на предмет сходства с типичными для попыток воздействия на систему, либо наоборот — на предмет команд, которые не используются при типичном использовании данной АС. Понятно, что подобная мера усложнит жизнь администраторам, но зато увеличит вероятность обнаружения атак на систему.

Подводя итоги, еще раз рассмотрим потенциальную атаку с двух сторон: определим, что необходимо атакующему, для получения несанкционированного доступа к АС, и что может ему противопоставить защищающаяся сторона.

Для атакующего существует огромный набор всевозможных устройств и утилит для реализации данной атаки. Во-первых, это множество контроллеров, способных эмулировать HID-устройства. Наиболее популярными для этого являются плата Teensy и ее аналоги от компании Arduino, а также уже готовый для атаки вариант USB Rubber Ducky Deluxe, замаскированный под обыкновенную флешку. Во-вторых, разработано несколько тулкитов, которые позволяют атакующему не самостоятельно писать программу для микроконтроллера, а сами генерируют для него прошивку. Если рассматривать финансовый аспект данной атаки, то получаем следующее: цена данной атаки состоит только из цены на микроконтроллер. Суммируя вышесказанное, данный вектор атаки будет набирать популярность из-за своей низкой стоимости и трудозатратности, а также ввиду отсутствия необходимости в каких-либо специфических навыках для его реализации.

Если рассматривать данную атаку с точки зрения защиты автоматизированных систем, то можно сделать следующий вывод: на каждом из этапов активной атаки можно найти средство для противодействия. Для противодействия первым этапам, достаточно использовать физические меры защиты информации, например, контроль доступа к АС. Для всех остальных этапов можно использовать комплекс программных средств, который позволит успешно обнаруживать и предотвращать несанкционированное выполнение команд в системе. В нижеприведенной таблице сопоставлены основные этапы атаки с методами защиты автоматизированных систем.

Номер этапа	Название этапа атаки	Выбранные методы защиты на данном этапе атаки
1	Доставка атакующего устройства к АС	Контрольно-пропускная система прохода в организацию; Контроль доступа в помещение с АС; Журналирование входа в помещение с АС; Видеонаблюдение за АС; Обучение сотрудников основам ИБ.
2	Подключение атакующего устройства к АС и его инициализация	Отключение USB-портов АС (физическое или программное); Разрешение на подключение устройств только по белому списку; Антивирусная система защиты; Простейшая «сигнализация»; Проверка подлинности подключенной клавиатуры или мыши.
3	Ввод команд атакующим устройством	Поведенческий анализ работы системы, основанный на скорости нажатия клавиш клавиатуры.
4	Непосредственное выполнение введенных команд в системе	Контроль наличия прав администратора у рядовых пользователей; UAC; Групповые политики; Поведенческий анализ, вводимых команд.

В продолжение данной статьи будут рассмотрены практические аспекты реализации данного комплекса мер, позволяющего максимально надежно защитить систему от несанкционированного воздействия, организованного с помощью HID-эмулирующих устройств.

Список литературы

1. Adam Gordon. Official (ISC)2 Guide to the CISSP CBK. 4th Edition. Boca Raton: CRC Press, 2015. 1304 p.

2. Adrian Crenshaw. Programmable HID USB Keystroke Dongle: Using the Teensy as a pen testing device. Available at: http://www.irongeek.com/i.php?page=videos/phukd-defcon-18, accessed 12.05.2014.
3. Shon Harris. CISSP All-In-One Exam Guide. 6th Edition. New York: The McGraw-Hill Companies, 2012. 1305 p.
4. Джон Сноу. Создаем свой хардварный USB-троян // Хакер. 2012. №06(161). С. 18-23.