Математические структуры и моделирование
2016. № 2(38). С. 116–125

УДК 004.3+004.4+004.056

РАЗРАБОТКА АППАРАТНО-ПРОГРАММНОГО СРЕДСТВА ЗАЩИТЫ ОТ УЯЗВИМОСТИ BADUSB

Т.В. Вахний
к.ф.-м.н., доцент, e-mail: vahniytv@mail.ru
С.Ю. Кузьмин
студент, e-mail: sergkuz2@gmail.com

Омский государственный университет им. Ф.М. Достоевского

Аннотация. В статье описано создание аппаратно-программного модуля на основе печатных плат Arduino, который обнаруживает в USB-устройствах уязвимость BadUSB.

Ключевые слова: Защита информации, USB-устройства, уязвимость BadUSB, контроллеры USB, дескрипторы.

Введение

В повседневной жизни мы привыкли использовать планшеты, смартфоны, флешки, веб-камеры, гарнитуры, принтеры, сканеры и другие USB-устройства. Интерфейс USB стал одним из самых распространённых сегодня, поскольку обеспечивает удобную и быструю передачу данных. Работой этого интерфейса управляет USB-хост, который обнаруживает подключение и отключение USB-устройств, управляет передачей данных, обеспечивает питанием подключённые устройства. Однако производители не защищают USB-устройства от перепрошивки, а USB-хосты не проверяют их на подлинность. Класс хакерских атак, основанный на уязвимости USB-устройств, получил название BadUSB [1, 2]. Проведя реверс-инжиниринг конкретного устройства, можно создать и записать в него вредоносный код. Использование USB-устройства с модифицированной прошивкой очень опасно, т. к. эксплойт запускается в процессе инициализации устройства, а существующие антивирусные решения пока не могут сканировать служебную область памяти.

Прошить микроконтроллер USB-устройства в большинстве случаев можно прямо с компьютера через USB-разъём [1–4]. Записанный в прошивку устройства вредоносный код может, например, имитируя клавиатуру, произвести необходимые действия за пользователя на заражаемом компьютере. Или, имитируя сетевое устройство, изменить сетевые настройки таким образом, что пользователь будет просматривать интернет-сайты через подконтрольные злоумышленнику промежуточные серверы. Кроме того, имитируя USB-флешку, вредоносный код может загрузить и запустить на компьютере с включенным автозапуском вирусную программу. Такой вирус может скопировать себя и на

другие USB-устройства, подключённые в данный момент к компьютеру. Скомпрометированная система также может распространять вирус на остальные устройства, которые пользователь будет подключать к системе. Поскольку хосты не проверяют USB-устройства на подлинность, то это может повлечь за собой неконтролируемый рост заражённых аппаратных устройств, которые в настоящее время нечем проверять.

В данной статье описывается создание аппаратно-программного модуля на основе печатных плат Arduino, который обнаруживает в USB-устройствах уязвимость BadUSB. Для проверки работоспособности созданного аппаратно-программного модуля было реализовано USB-устройство с модифицированной прошивкой.

1. Общая характеристика USB-устройств

USB-интерфейс – это последовательный интерфейс передачи данных для периферийных устройств в вычислительной технике. К USB-портам подключаются мыши, клавиатуры, принтеры, сканеры, модемы, кардридеры, флэш-накопители, фотоаппараты, сотовые телефоны, плееры, жёсткие диски, оптические дисководы и др. Спецификация USB 2.0-3.0 является на сегодняшний день основным коммуникационным портом персональных компьютеров [3], а USB-порты являются своего рода стандартом для подключения внешних устройств, к которому стремятся все производители этих устройств.

Изначально в технологию USB закладывалась гибкость, универсальность, простота и удобство использования. В связи с этим в большинстве USB-устройств есть микросхема контроллера, которая выполняет много различных функций: от преобразования напряжений до связи с USB-хостом компьютера и обмена с ним различной информацией. Поскольку разнообразного USB-оборудования огромное количество, то производители контроллеров решили не делать каждому устройству по отдельной микросхеме контроллера, т. к. это неоправданно дорого. Они решили сделать всего лишь один контроллер, который будет совместим как с флэшками и принтерами, так и с другим оборудованием. Это было реализовано благодаря добавлению возможности прошивать внутреннюю память контроллера. В итоге получается, что микросхема контроллера, исходя из прошивки, которая в него записана, сообщает компьютеру, что именно к нему подключено, и далее компьютер передаёт эту информацию операционной системе. После этого происходит установка драйверов и с устройством становится возможным взаимодействовать.

Чтобы добиться всего этого, были использованы классы спецификаций, которые определяют тип и функциональность нового устройства. В составе операционной системы находятся драйверы классов, каждый из которых позволяет функционировать соответствующему классу устройств. Таким образом, любое USB-устройство, при наличии USB-разъёма, может взаимодействовать с компьютером. С одной стороны, это удобно для пользователей, т. к. нет необходимости искать, устанавливать самому драйвера на каждое USB-устройство, будь это клавиатура, мышь или флеш-накопитель. С другой стороны, одни и те же

разъёмы могут работать с разными классами устройств, а программно изменив этот класс, можно выдать одно устройство за другое.

2. Иерархия дескрипторов микроконтроллеров

Все USB-устройства имеют иерархию дескрипторов, которые описывают информацию для хоста: что это за устройство, кто его изготовил, какую версию USB поддерживает устройство, какими способами устройство может быть сконфигурировано, количество конечных точек и их типы и т. д. Наиболее общие дескрипторы USB следующие [3, 5]:

1) дескрипторы устройства;
2) дескрипторы конфигурации;
3) дескрипторы интерфейса;
4) дескрипторы конечной точки;
5) строковые дескрипторы.

USB-устройства могут иметь только один дескриптор устройства. Дескриптор устройства включает в себя такую информацию, как поддерживаемую устройством ревизию USB, PID (идентификатор продукта), VID (идентификатор производителя), используемые для загрузки соответствующего устройству драйвера, и количество возможных конфигураций устройства. Число конфигураций указывает, сколько имеется ответвлений по дескрипторам конфигурации.

Дескриптор конфигурации указывает величину потребляемой мощности от шины, питается устройство от собственного источника или от шины USB и количество интерфейсов, которые есть у конфигурации. Когда устройство проходит энумерацию, хост читает дескриптор устройства и принимает решение, какую конфигурацию применить. Хост может разрешить только какую-либо одну из конфигураций.

Дескриптор интерфейса можно рассматривать как заголовок или группирование конечных точек в функциональную группу, выполняющую единственную функцию устройства. Например, для многофункционального устройства факса/сканера/принтера дескриптор интерфейса 1 может описывать конечные точки функции факса, дескриптор интерфейса 2 может описывать функцию сканера, и дескриптор интерфейса 3 может описывать функцию принтера. В отличие от дескриптора конфигурации здесь нет ограничений на количество одновременно разрешённых интерфейсов. У устройства могут быть один или много интерфейсов, разрешённых одновременно.

3. Уязвимость BadUSB и атаки через USB-устройства

Любое взаимодействие USB-устройства с компьютером осуществляется с помощью микроконтроллера. Для того чтобы он мог осуществить операции, в его собственной служебной памяти хранится управляющий код. Простой доступ к данной памяти пользователь с помощью каких-либо программных

продуктов не имеет, более того, некоторым моделям контроллеров необходимо использование аппаратного программатора. В связи с распространённостью технологии USB многие производители для упрощения выполняют операцию перепрограммирования напрямую через интерфейс USB [2]. Обычная прошивка – это закрытый код, поэтому принято считать, что его изменение доступно только разработчику. Однако каждое USB-устройство включает в себя чип контроллера или, иными словами, собственный управляющий микрокомпьютер, который можно перепрограммировать с помощью небольшого физического и программного воздействия. Оригинальная прошивка чипа занимает меньший объем, чем доступный размер памяти для её хранения, что позволяет изменить микрокод прошивки контроллера USB-устройства, изменив класс спецификации. После этого оно будет выдавать себя за другое устройство. При этом антивирус не находит никаких следов вируса, т. к. он находится в прошивке устройства. В свою очередь, если компьютер заразился вирусом прошивальщика, то этот вирус может в дальнейшем прошивать контроллеры подключаемых к нему USB-устройств, внося в них дополнительный функционал.

Возможны различные варианты использования на практике злоумышленником уязвимости BadUSB [1, 2]. Основные из них следующие.

1. USB-устройство с модифицированной прошивкой может выдать себя за клавиатуру и начать отдавать команды от имени пользователя, под которым был выполнен вход в операционную систему. Если был выполнен вход от имени администратора, то устройство получает полный доступ ко всем возможностям операционной системы. По истечении некоторого времени такое устройство начинает отправлять последовательности нажатий клавиш. В результате злоумышленник от имени пользователя может, например, загрузить из интернета и запустить вредоносное программное обеспечение, отправить необходимые файлы злоумышленнику или запустить команду на удаление файлов с дисков. Также существует опасность заражения всех USB-устройств, подключаемых после, потому что нельзя исключить возможность заражения USB-контроллера, находящегося в компьютере. Существенным минусом данного вида атак является отсутствие доступа к информации на экране и, как следствие, отсутствие обратной связи на любые действия со стороны заражённого устройства. Например, злоумышленник не может определить как текущую раскладку клавиатуры, так и произведён ли вход в систему.

2. USB-устройство с модифицированной прошивкой представляется компьютеру жертвы сетевой картой, что позволяет осуществить подмену стандартных DNS-адресов и перенаправить весь трафик через свой сервер, получив возможность совершать атаку типа «человек посередине».

3. USB-устройство с достаточным местом для хранения вредоносного кода, например, флеш-накопитель, может определить момент включения компьютера и в момент определения BIOS'ом выдать на загрузку вирус для заражения операционной системы. Это становится возможным благодаря тому, что по поведению хоста при общении с USB-микроконтроллером возможно определить операционную систему хоста, в частности ОС Windows, Linux, MacOSX, а также BIOS. Администратор может и не узнать об этом, потому что вредонос-

ные файлы будут находиться уже в самой операционной системе до установки антивирусной программы.

4. USB-устройство с модифицированной прошивкой может использовать возможность повторной инициализации устройства. В этом случае, запускаясь в виртуальной машине, вирус заражает любое подключённое USB-устройство. Заражённая прошивка выполняет переинициализацию и представляется двумя независимыми устройствами: неким новым и тем, которое уже было подключено к виртуальной машине. Новое устройство будет автоматически подключено к хостовой операционной системе, а старое — обратно в виртуальную машину. Так может быть произведён выход за пределы виртуального окружения, т. е. осуществлён переход от клиентской до хостовой операционной системы.

Помимо этого можно отметить еще несколько сценариев возможных атак через USB-устройства [1]: сокрытие файлов от операционной системы и файловых менеджеров вместо удаления, изменение файлов при их записывании на флеш-память устройства, таким образом, данные модифицируются «на лету» и антивирусная программа проверяет «чистый» файл до его изменения, после на флеш-накопителе находится уже заражённый файл, возможна эмуляция дисплея с целью получить доступ к скрытой информации и др. При возможности эмулирования клавиатуры и скрытого хранения файлов нельзя исключать угрозу подмены системы BIOS.

Таким образом, компьютеры или подключённые USB-устройства, которые были заражены инфекцией BadUSB, невозможно вылечить и вернуть к первоначальному состоянию. Опасность угрозы BadUSB высока, потому что все основные чипы контроллеров USB-устройств не обладают никакой собственной защитой от их перепрограммирования. Антивирусам очень тяжело бороться с таким типом атак, потому что невозможно фильтровать USB-трафик и отличить – может устройство самостоятельно подавать такие команды или же это действие выполняет пользователь.

4. Средства защиты от уязвимости BadUSB

Ряд средств комплексной антивирусной защиты, таких как ESET Endpoint Antivirus, Kaspersky Endpoint Security, компонент «Родительский контроль» у Dr.Web AV-Desk, позволяют ограничивать доступ к сменным носителям и разрешать активацию согласно «белому списку». Однако в случае с BadUSB таких мер явно недостаточно, т. к. пользователь сам может разрешить подключение опасного устройства, ошибочно посчитав его безопасным. Эта мера защиты не спасёт компьютер от заражения, но может несколько обезопасить от некоторых разновидностей данной угрозы [1, 2].

Одним из возможных способов противостояния уязвимости BadUSB является подпись прошивки производителем оборудования и соответствующая проверка на стороне USB-хоста перед использованием устройства, что не предусмотрено текущей спецификацией USB. Специальное программное обеспечение для обнаружения вредоносных программ не проверяет прошивку и вряд ли будет обладать подобной функциональностью в ближайшем будущем. Разработчики

антивирусных решений в будущем будут вынуждены добавить отдельные модули для более гибкого дополнительного контроля над подключаемыми USB-устройствами [1, 2].

Другим решением проблемы может стать блокировка возможности перепрошивки USB-устройств самим производителем [1]. Для этого необходимо применить особый защитный механизм поверх прошивки, но внедрение новых средств безопасности повлечёт за собой полный пересмотр стандарта как такового. На реализацию этой задачи требуются существенные финансовые затраты и могут уйти годы, на протяжении которых обычные пользователи будут находиться в зоне риска. До тех пор, пока будет доступна возможность модифицировать прошивку USB-устройств, такие атаки, несомненно, будут представлять собой серьёзную угрозу.

5. Аппаратно-программный модуль для проверки USB-устройств

В качестве инструмента для проектирования аппаратно-программного модуля, который будет проверять прошивку USB-устройств, была выбрана платформа Arduino, предназначенная для «physical computing» с открытым программным кодом. Под торговой маркой Arduino выпускается несколько печатных плат с микроконтроллером и платы расширения. Микроконтроллеры для Arduino отличаются наличием предварительно прошитого в них загрузчика. С помощью этого загрузчика пользователь загружает свою программу в микроконтроллер без использования традиционных отдельных аппаратных программаторов. Загрузчик соединяется с компьютером через USB-интерфейс (если он есть на плате) или с помощью отдельного переходника UART-USB [4].

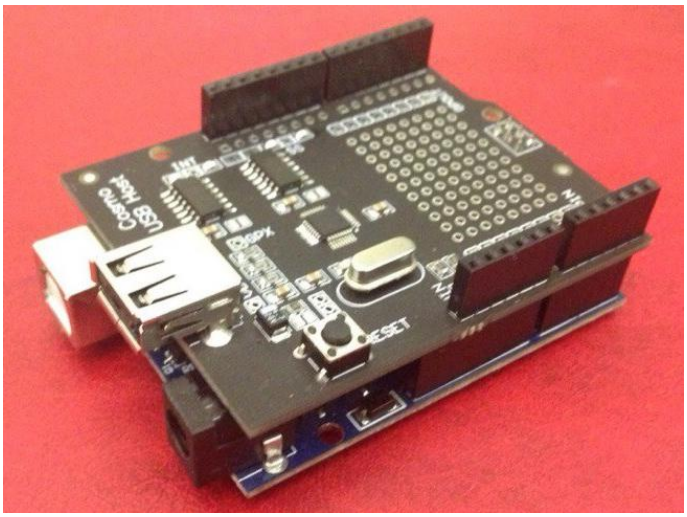

Рис. 1. Аппаратно-программный модуль для защиты от уязвимости BadUSB

Аппаратная часть разработанного модуля представляет собой набор смонтированных плат. Модуль был собран из печатной платы Arduino Duemilanove

и платы расширения Cosmo USB Shield (рис. 1). Для написания программной части аппаратно-программного модуля понадобилась среда разработки модуля Arduino IDE и среда разработки для принимающей программы Visual Studio.

Программный код для модуля был написан на языке программирования устройств Arduino, он основан на языке программирования C/C++. Для его написания понадобилась библиотека USB host libraries. Алгоритм работы модуля таков, что он обрабатывает дескрипторы подключённого устройства и отправляет их на компьютер через эмулированный com-port. Программное приложение для ОС Windows было написано на языке программирования C#, для написания кода понадобилась среда разработки Visual Studio с подключённым пространством имён System.IO.Port. Алгоритм работы программы таков, что она ищет изменения в дескрипторах и в ходе анализа полученных данных делает вывод об устройстве. Программа сверяет данные дескрипторов из разделов интерфейса и устройства, а именно идентификатор продукта, идентификатор производителя и код класса устройства.

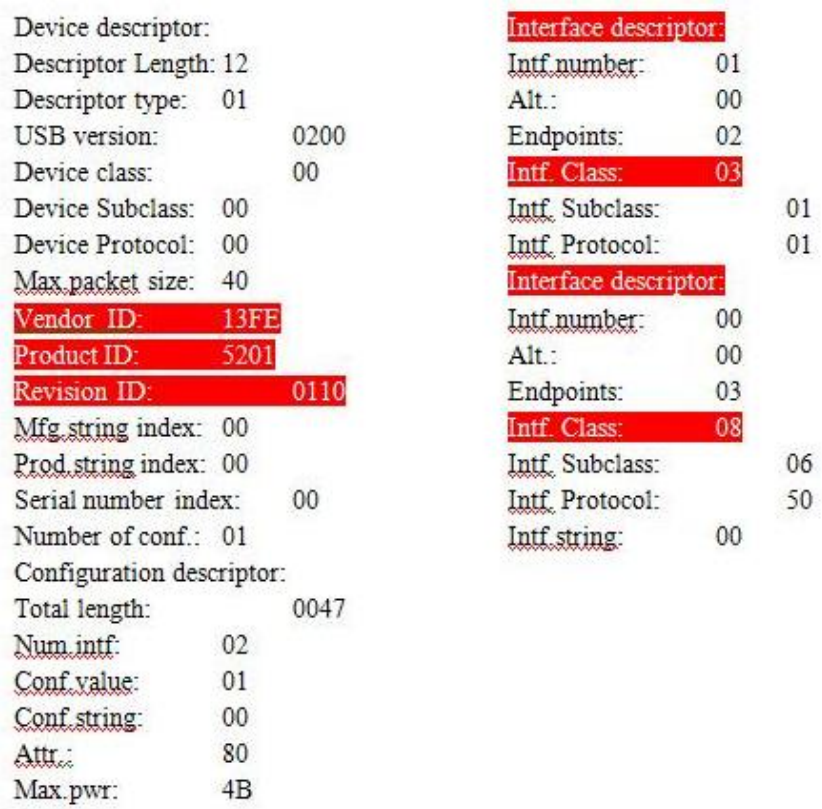

Рис. 2. Измененные дескрипторы USB-устройства

Как правило, у неоригинальных USB-устройств идентификаторы производителя и продукта отличаются от оригинала на один символ, либо им невозможно найти соответствие. У USB-устройства, которое выполняют одну функцию, должен быть только один дескриптор описывающий интерфейс. К примеру, для перепрошитого USB-flash устройства дескриптор интерфейса повторяется

(рис. 2), из них используется только тот, у которого поле Intf.number имеет значение 01, а дескриптор со значением поля Intf.number 00 не используется вовсе. Также у первого дескриптора интерфейса поле Intf.Class имеет значение 03, это означает, что класс данного устройства HID–устройство, но, на самом деле, это устройство имеет Intf.Class значение 08, что означает, что это Mass Storage устройство. Оригинальные настройки USB-устройств имеют правильные идентификаторы и правильное значение в поле Intf.Class (см. рис. 3).

Device descriptor:		
Descriptor Length:	12	
Descriptor type:	01	
USB version:		0210
Device class:		00
Device Subclass:	00	
Device Protocol:	00	
Max.packet size:	40	
Vendor ID:	0930	
Product ID:	6545	
Revision ID:		0110
Mfg.string index:	01	
Prod.string index:	02	
Serial number index:	03	
Number of conf.:	01	

Configuration descriptor:		
Total length:		0020
Num.intf:	01	
Conf.value:	01	
Conf.string:	00	
Attr.:	80	
Max.pwr:	4B	
Interface descriptor:		
Intf.number:	00	
Alt.:	00	
Endpoints:	02	
Intf. Class:	08	
Intf. Subclass:		06
Intf. Protocol:		50
Intf.string:	00	

Рис. 3. Оригинальные настройки дескрипторов USB-устройства

Для проверки работоспособности созданного аппаратно-программного модуля было реализовано USB-устройство с модифицированной прошивкой. С помощью программ и патчей была изменена память микроконтроллера флэшки, и компьютер стал воспринимать USB-накопитель как клавиатуру, которая выполняет действия от имени текущего пользователя и с его правами доступа. В качестве вредоносного программного обеспечения использовался скрипт, написанный на Rubberry Ducky, который открывал блокнот и печатал: «Hello World». После изменения прошивки с внедрённым в него скриптом, флэшка стала восприниматься компьютером как клавиатура и запускала на выполнение вредоносный код.

Для работы с созданным аппаратно-программным модулем необходимо подключить его к компьютеру посредством USB-кабеля или подать питание при помощи адаптера AC/DC или батареи. Рекомендуемый диапазон напряжения питания от 7 В до 12 В. При подключении к модулю немодифицированного USB-устройства приложение определяет, что прошивка в устройстве не изменялась и не несёт в себе угрозы от уязвимости BadUSB (см. рис. 4). Таким образом было проверено 20 USB-устройств, из которых только одно было модифицированным. С помощью разработанного аппаратно-программного модуля перепрошитое USB-устройство было обнаружено. На рис. 5 показано, что аппаратно-программный модуль определил модифицированное USB-устройство и выдал предупреждение.

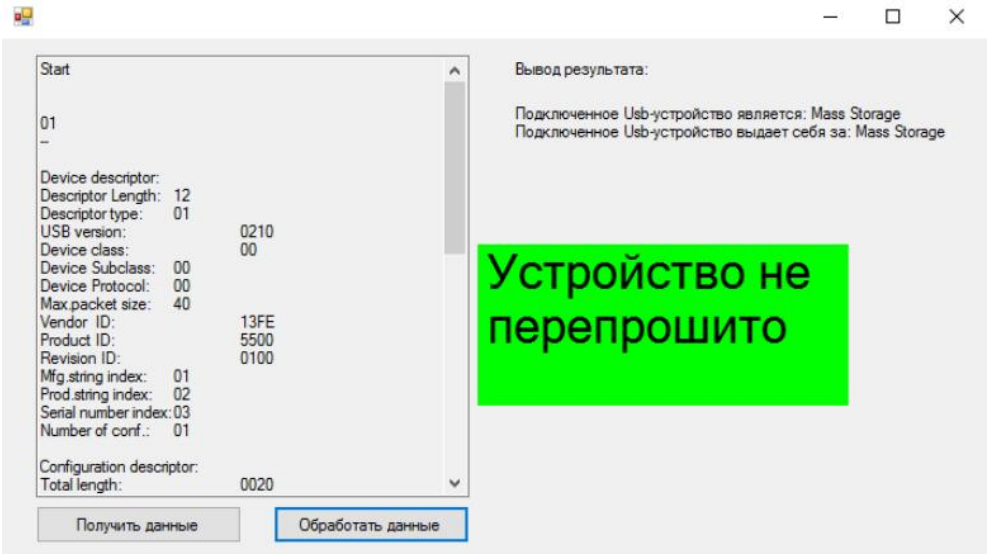

Рис. 4. Результат работы приложения с немодифицированным USB-устройством

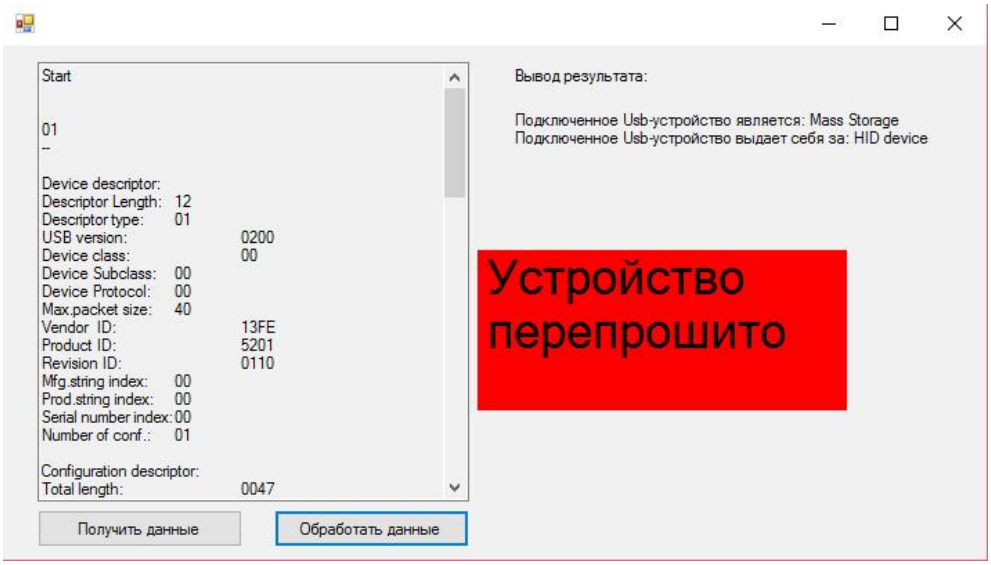

Рис. 5. Результат работы приложения с модифицированным USB-устройством

Таким образом, созданный аппаратно-программный модуль обнаруживает в USB-устройствах уязвимость BadUSB, поэтому его применение позволит повысить компьютерную безопасность. Он может быть полезен как для коммерческих, так и для персональных целей.

6. Заключение

Уязвимости BadUSB подвержены все устройства с незащищёнными USB-микроконтроллерами: флеш-накопители, вебкамеры, мышки, клавиатуры и т. д.

Данная уязвимость не требует особого программного обеспечения на компьютере жертвы и работает под любыми операционными системами, поддерживающими USB-HID устройства. Необходимость трудоёмкого реверс-инжиниринга каждого USB-устройства ограничивает этот класс атак. В настоящее время действенных методов защиты от данной угрозы не существует. В данной работе описано создание аппаратно-программного модуля, который обнаруживает уязвимость BadUSB в USB-устройствах. Выбор компонентов для создания данного модуля был основан на технических характеристиках печатных плат и их стоимости. Для проверки работоспособности созданного аппаратно-программного модуля было реализовано USB-устройство с модифицированной прошивкой.

Литература

1. Полежаев П.Н., Малахов А.К., Сагитов А.М. «Ахиллесова пята» USB-устройств: атака и защита // Философские проблемы информационных технологий и киберпространства. 2015. № 1(9). С. 106–117.
2. Васильков А. BadUSB – как новая атака реализована в разных устройствах. URL: `http://www.computerra.ru/108106/bad-usb-on-some-devices/` (Дата обращения: 12.12.2015).
3. Агуров П.В. Интерфейс USB. Практика использования и программирования. СПб. : БХВ-Петербург, 2004. 576 с.
4. Петин В.А. Проекты с использованием контроллера Arduino. СПб. : БХВ-Петербург, 2015. 448 с.
5. USB in NutShell – путеводитель по стандарту USB. URL: `http://microsin.net/programming/arm-working-with-usb/usb-in-a-nutshell-part2.html` (Дата обращения: 12.12.2015).

DEVELOPMENT OF HARDWARE-SOFTWARE MEANS FOR PROTECTION AGAINST BADUSB-VULNERABILITY

T.V. Vahniy
Ph.D. (Phys.-Math.), Associate Professor, e-mail: vahniytv@mail.ru
S.Yu. Kuzmin
Student, e-mail: sergkuz2@gmail.com

Dostoevsky Omsk State University

Abstract. The article describes a hardware-software module based on printed circuit boards Arduino. The module detects the BadUSB-vulnerability in USB-devices.

Keywords: data protection, USB-devices, BadUSB-vulnerability, USB-controllers, descriptors.