УДК 004.056

ИДЕНТИФИКАЦИЯ USB-УСТРОЙСТВ В СИСТЕМЕ АНАЛИЗА ПОЛЬЗОВАТЕЛЬСКИХ ПОТОКОВ ДАННЫХ

Варавка А.В., Цололо С.А.
ГВУЗ «Донецкий национальный технический университет»
antosha90@yandex.ru

Аннотация

Варавка А.В., Цололо С.А. Идентификация USB-устройств в системе анализа пользовательских потоков данных. *Выполнен анализ особенностей архитектуры, принципов работы и взаимодействия компонентов USB. Предложена система анализа и контроля пользовательских потоков данных с внешних устройств в рамках обеспечения информационной безопасности предприятия. Приведен способ идентификации USB-устройств в рамках предложенной системы.*

Введение

Шина USB (Universal Serial Bus – универсальная последовательная шина) является промышленным стандартом расширения архитектуры персонального компьютера, ориентированным на интеграцию с телефонией и устройствами бытовой электроники. Шина USB позволяет одновременно подключать последовательно до 127 устройств, которые могут быть как дополнительными компонентами рабочей станции (внешние накопители, устройства ввода, мобильные терминалы и прочее), так и хабами (узлами) – устройствами, через которые подключаются оконечные дополнительные компоненты [1].

В настоящее время одним из наиболее популярных в пользовательской среде способов применения шины USB является подключение внешних накопителей (флеш-брелоков). Данные устройства широко используются для хранения и переноса пользовательских данных, в числе которых может оказать и конфиденциальная информация предприятия. Поэтому с точки зрения обеспечения информационной безопасности предприятия *актуальной* является задача контроля, учета и анализа потоков данных, которые пользователи направляют с/на внешние USB-устройства.

Таким образом, авторами предлагается распределенная клиент-серверная система, которая позволяет вести централизованный контроль и учет пользовательских потоков данных с USB-устройств [2]. При этом записанные/считанные на рабочей станции данные анализируются по заданным критериям. В случае соответствия критериям фиксируется утечка данных, и клиентская часть системы отсылает на сервер подробную информацию о внешнем накопителе, пользователе и сетевом адресе рабочей станции, а также некоторую другую техническую информацию, достаточную для точной идентификации факта утечки.

В рамках предлагаемой системы авторами предлагается реализация функции идентификации USB-устройства в соответствии с его дескриптором. Набор получаемой при этом информации позволяет идентифицировать устройство. В дальнейшем эта информации в случае необходимости передается на сервер.

Далее рассмотрим особенности как общие принципы архитектуры шины USB, которые необходимо учитывать при разработке, так и особенности работы с дескрипторами устройств.

Архитектура шины USB

Архитектура и основные параметры шины USB определяются возложенными на нее задачами. Физическая топология шины USB, приведенная на рис. 1, имеет следующие основные особенности [3]:

- шина обеспечивает подключение USB-устройств к хосту USB;
- соединение устройств осуществляется по топологии многоярусной звезды;
- центром каждой звезды является хаб;
- каждый кабельный сегмент соединяет между собой две точки: хост с хабом или функцией, хаб с функцией или другим хабом.

Существуют следующие типы устройств USB.

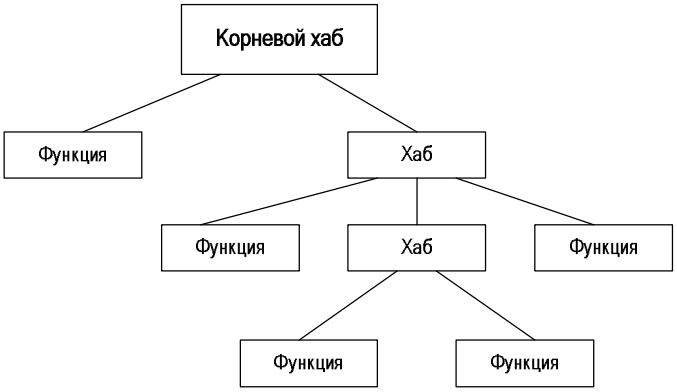

Рис. 1 – Физическая топология шины USB

Хост-контроллер (host controller) USB – это центральный контроллер, который входит в состав чипсета материнской платы и управляет работой всех устройств на шине USB. При этом на каждой шине USB допускается наличие только одного хоста. Типичная плата современного компьютера содержит несколько хост-контроллеров, каждый из которых управляет отдельной шиной USB.

Устройство (device) USB может быть хабом или функцией, при этом *хаб* (hub) – это устройство, которое обеспечивает дополнительные точки подключения к шине USB. Каждый хаб имеет один восходящий порт (Upstream Port), предназначенный для подключения к хабу верхнего уровня, и несколько нисходящих портов (Downstream Ports), предназначенных для подключения функций или хабов нижнего уровня. Хаб управляет работой нисходящих портов, осуществляет контроль подключения и отключения устройств.

Функция (function) USB – это периферийное устройство (ПУ) или отдельный блок ПУ, способный передавать и принимать информацию по шине USB.

Различают три уровня взаимодействия хоста с физическим устройством:

- на верхнем уровне (уровне функции) между собой взаимодействуют клиентская программа и функция;
- на среднем уровне (уровне устройства) взаимодействуют системное программное обеспечение и логическое устройство USB;
- на нижнем уровне (уровне интерфейса шины USB) хост-контроллер взаимодействует с USB-интерфейсом устройства.

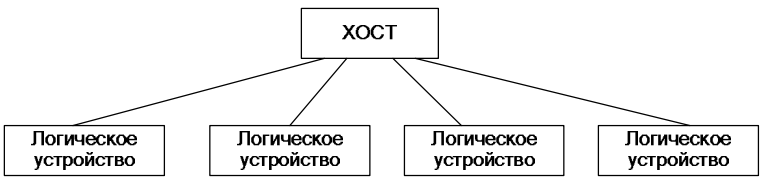

Рис. 2 – Логическая топология шины USB

Используемая на среднем уровне взаимодействия логическая топология шины USB (рис. 2) гораздо проще физической: хост обменивается информацией с логическими устройствами таким образом, что точка подключения устройства не

имеет значения, как если бы все устройства были подключены к корневому хабу.

Стандартные дескрипторы USB

В спецификации на шину USB указана группа дескрипторов, которые должны выдаваться устройствами USB в ответ на стандартные запросы. Структура таких дескрипторов стандартизирована, а в документации они именуются стандартными дескрипторами (standard descriptors) [4]. Далее рассмотрим основные из них.

Дескриптор устройства. Стандартный дескриптор устройства (tandard Device Descriptor) содержит основную информацию об устройстве USB. Структура стандартного дескриптора устройства показана в табл. 1.

Табл. 1 – Структура дескриптора устройства

Смещ.	Мнемоника	Размер	Описание
0	bLength	BYTE	Размер данного дескриптора в байтах
1	bDescriptorType	BYTE	Тип дескриптора (DEVICE)
2	bcdUSB	WORD	Номер версии спецификации USB
4	bDeviceClass	BYTE	Код класса устройства («0» – интерфейсы функционируют независимо, «FFh» – класс определяется производителем
5	bDeviceSubCIass	BYTE	Код подкласса устройства USB
6	bDeviceProtocol	BYTE	Код протокола USB
7	bMaxPacketSize	BYTE	Максимальный размер пакета для нулевой конечной точки (значения 8, 16, 32 и 64)
8	idVendor	WORD	Идентификатор производителя устройства
10	idProduct	WORD	Идентификатор продукта
12	bcdDevice	WORD	Номер версии устройства
14	iManufacturer	BYTE	Индекс дескриптора строки (производитель)
15	iProduct	BYTE	Индекс дескриптора строки (продукт)
16	iSerialNumber	BYTE	Индекс дескриптора строки (серийный номер)
17	bNumConfigurations	BYTE	Количество конфигураций

Некоторые поля стандартного дескриптора устройства традиционно содержат фиксированные значения и не несут информационной нагрузки. Код версии спецификации USB может принимать следующие значения: «0100h» – версия 1.0, «0110h» – версия 1.1, «0200h» – версия 2.0. Значение кода класса равно «9» для хабов и «0» для любых других устройств, поэтому определить тип устройства при помощи дескриптора устройства можно только в том случае, если оно является хабом, а идентификация других стандартных периферийных устройств выполняется по дескриптору интерфейса [3, 4].

Идентификатор изготовителя устройства, идентификатор продукта и номер версии устройства используются для точной идентификации устройства в рамках разрабатываемой системы. В случае необходимости они отправляются на сервер в виде строки-идентификатора. Значение последнего байта стандартного дескриптора устройства показывает, сколько различных вариантов конфигурации можно задать для данного устройства. Как правило, периферийные устройства, предназначенные для использования совместно с персональными компьютерами, имеют только один вариант конфигурации.

Дескриптор конфигурации. Стандартный дескриптор конфигурации (Standard Configuration Descriptor) содержит информацию об одной из возможных конфигураций устройства. Структура стандартного дескриптора конфигурации показана в табл. 2.

Устройство может иметь один или несколько дескрипторов конфигурации в соответствии с количеством возможных конфигураций, указанных в стандартном дескрипторе устройства.

Табл. 2 – Структура стандартного дескриптора конфигурации

Смещ.	Мнемоника	Размер	Описание
0	bLength	BYTE	Размер данного дескриптора в байтах
1	bDescriptorType	BYTE	Тип дескриптора (CONFIGURATION)
2	TotalLength	WORD	Общий размер данных в байтах
4	bNumInterfaces	BYTE	Количество интерфейсов конфигурации
5	bConfigurationValue	BYTE	Значения для установки заданной конфигурации
6	iConfiguration	BYTE	Индекс дескриптора строки конфигурацию
7	bmAttributes	BYTE	Характеристики конфигурации: биты «0-4» – резерв; бит 5 – признак пробуждения устройства по внешнему сигналу; бит 6 – источник питания, бит 7 – резерв.
8	MaxPower	BYTE	Код мощности, которую потребляет устройство

Каждая конфигурация описывается одним стандартным дескриптором. Размер стандартного дескриптора конфигурации всегда составляет 9 байт, а код типа дескриптора имеет значение «2». По запросу «Получить Стандартный дескриптор конфигурации» устройство выдает не только дескриптор конфигурации, но и все имеющиеся дескрипторы интерфейсов и конечных точек.

Идентификация USB-устройств

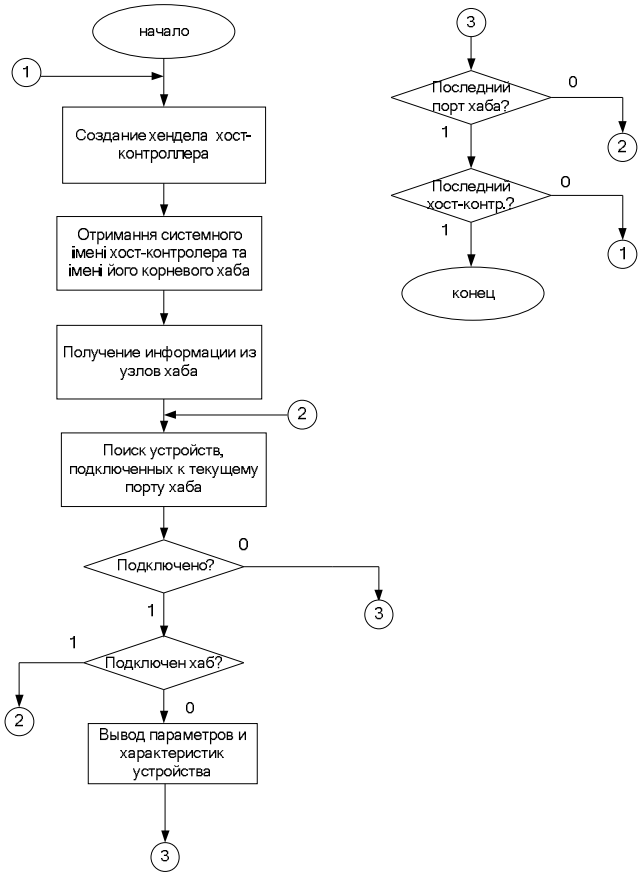

Рис. 3 – Алгоритм идентификации USB-устройств

В рамках системы анализа пользовательских потоков данных с/на USB-устройств выполнена разработка программы, с помощью которой можно определить параметры и характеристики устройств, которые подключены к шине USB рабочей станции. Алгоритм работы программы изображен на рис. 3.

Программа обнаруживает хост-контроллеры, которые установлены на рабочей станции, находит хабы и их порты, затем сканирует их на наличие подключенных устройств. Если устройство найдено, то формируются запросы на получение стандартных дескрипторов устройств. Сначала выполняется получение информации дескриптора устройства, потом отправляется запрос на дескриптор конфигурации. Размер дескриптора конфигурации всегда составляет 9 байт, а код типа дескриптора имеет значение «2». По запросу «Получить стандартный дескриптор конфигурации» каждое

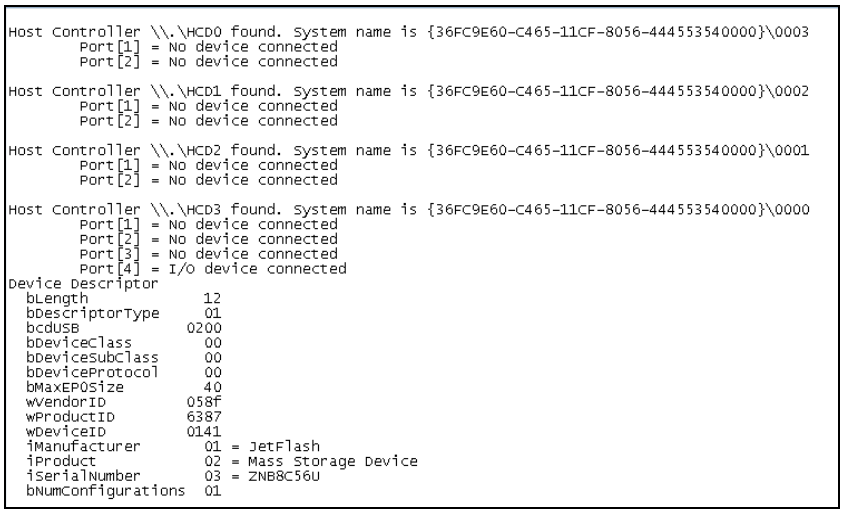

Рис. 4 – Результаты работы программы

устройство выдает не только дескриптор конфигурации, но и все имеющиеся дескрипторы интерфейсов и конечных точек. Поэтому, если по какой-то причине нельзя получить дескриптор конфигурации, информация дескрипторов строки и конечной точки интерфейса будет недоступной.

Программа является частью системы, поэтому реализация выполнена в виде консольного приложения. Результаты работы программы представлены на рис. 4.

Выводы

В результате анализа архитектуры и взаимодействия компонентов USB был предложена распределенная система, которая позволяет вести централизованный контроль и учет пользовательских потоков данных с USB-устройств. Реализация предложенной системы может быть использована на предприятиях как один из элементов комплекса обеспечения информационной безопасности.

В рамках предлагаемой системы реализована подпрограмма идентификации USB-устройства в соответствии с его дескриптором. Полученная информация, которая позволяет однозначно идентифицировать конкретное USB-устройство, в случает необходимости может быть сохранена на сервере.

В дальнейшем планируется разработка остальных элементов системы.

Литература

1. Агуров П. В. Интерфейс USB. Практика использования и программирования. – СПб: БХВ-Петербург,2004. – 576 с.
2. Варавка А.В., Цололо С.А., Демеш Н.С. Исследование алгоритма обеспечения информационной безопасности в компьютерных системах предприятий на базе интерфейса USB. – ГВУЗ ДонНТУ,2011 г., «Информатика и компьютерные технологии 2011».
3. Скотт Мюллер. Модернизация и ремонт ПК (17 издание) – М.: «Вильямс», 2007. – С. 1016-1026.
4. Don Anderson. Universal Serial Bus System Architecture [электронный ресурс]. – Режим доступа: http://interface.centraltreasure.com/files/pdf/Hardware_USB_System_Architecture_pdf.pdf