Malicious USBs

A Netscylla whitepaper on the usage of cheap disposable USBs as exploit delivery devices.

Netscylla Cyber Security Ltd
GB 10571639

Address:
Telecom House,
125-135 Preston Road, Brighton,
East Sussex,
United Kingdom, BN1 6AF

Introduction

Hackers and malicious developers are using more USB devices in an attempt to gain access to your computer(s). This is nothing new, and publicly available hardware kits have been available since 2008. Exploiting the trust relationship between a computer and its peripherals; hackers can program a USB device that can be covertly transformed into an attack tool. A malicious USB device could be used to load malware onto your computer, or can trick a computers network interface to redirect some internet traffic to an attacker controlled website.

BadUSB

Turning common devices bad

USB devices are frequently connected to computers, whether it's a keyboard, webcam, or even a printer. The USB standard caters for different device classes that all share the same connector design. With the ease of availability and the low expense of a USB device; attackers are turning genuine devices into disposable exploit delivery mechanisms.

USB manufacturers often make available the development software to program their devices, this enables developers to create or program more than one type of USB device class. Depending on the manufacturer some development environments are open source and freely available.

Development software usually includes an Application Programming Interface (API), where the barebones of functional code already exits. Developers with basic programming experience can easily program a device to talk to a computer using the Human Interface Device (HID) protocol and perform a given action on the connected computer. Advanced developers can add or even change the functionality of the code, to cause unintended actions on the connected computer.

Comparison of available devices

The table below is a selection of programmable USB devices on today's market; they range on functionality, price and ease of use. The Hak5 USB Rubber Ducky is preferred with the less technical user's due to a simple scripting language, as opposed to the Teensy that requires the attacker to have a deeper development background.

Table 1: Summary of devices available on the open market

Device	Capabilities	Ease of Use	Average Cost ($ USD)
Phison Mass Storage Drive	Mass Storage	Difficult	$10-20
Teensy	HID Emulation Mass Storage	Easy	$16-20
USB Rubber Ducky	HID Emulation Mass Storage	Very Easy	$40
USB Armoury	HID Emulation Mass Storage Network Emulation	Moderate	$100

Typical Attacks

Attack Categorisation

There are typically three main types of attack:

- Keyboard emulation
- Composite device emulation
- Network device emulation

Keyboard Emulation

The most common freely available code on the web is the HID keyboard emulation. A person can program a USB device to emulate a keyboard, or more specifically a human typing at a keyboard. The USB device can be programmed to execute multiple key presses and combinations; within the computers Operating System (OS) this allows for command execution under and context of a logged in user.

This type of attack has also been demonstrated to brute-force PINs on some Android devices within a 24-hour period.

Figure 1: A typical USB keyboard

Composite Device Emulation

More experienced developers have been able to utilise the USB APIs, to create composite devices; a device that can emulate both a keyboard and mass-storage (drives). This type of attack essentially has brought the return of Auto-run attacks that were prevalent a few years ago. As soon as a user plugs in the device, the device will execute a program installed on the removable drive.

Depending on the nature of the executed program, malware could be introduced to the OS, or specific files could be copied to the drive for data exfiltration.

Figure 2: USB 'Ducky' acts as both keyboard and mass storage

Network Device Emulation

By emulating a networking adapter, it is possible to fool the OS that it is connected to a network. The USB device can subvert the Domain Name Server (DNS) address for resolving hostnames on the internet to one of the programmers choosing. This change remains while the computer is powered on, even if the USB device is removed.

By redirecting the victims DNS address to one of the attackers choosing, they can:

1. Record the internet history of the victim;
2. Change the intended destination of the Victim E.g. Redirect them to a web server with a web-based exploit serving additional malware.
3. Perform a Man-In-the-Middle attack, thus recording all of the Users internet traffic.

Figure 3: A typical USB ethernet adapter

Solutions & Bypass Attempts

Defending against malicious USB devices in software

Internet security Vendors now include 'Device Control' software within their enterprise solutions, additionally bespoke 'Device Control' software is available from specialist Vendors.

Device Control software depending on its configuration allows System Administrations to effectively black-list or white-list known USB devices. This software can limit what devices are allowed to connect to the machine and additionally limit what functions the device can perform.

Bypass #1: Altering a USB device's identity

However, USB hackers are capable of changing the Vendor and Product Identifiers (VID & PID) of a given device. The VID and PID are typically used by the OS to identify the device so that the right device driver can be loaded into memory; ready to talk to the newly inserted device. If hackers are able to obtain the correct information regarding the various types and models of equipment within your environment. Hackers can easily program the USB to mimic the VID and PID of an allowed device; thus, subverting some of the levels of protection installed on the server.

Figure 4:Cases are interchangeable, to appear more genuine

Defending against malicious USB devices physically

Some internet Vendors sell USB drive locks, a small device that fits within the computers port, simply turn the key and the device blocks the USB port from being used – this protection mechanism can get rather expensive when applied to a number of machines within a given companies estate.

Other System Administrators have resorted to epoxying the USB ports, but this is rather a permanent solution that can be difficult to undo. If IT equipment is leased, this would cause additional problems from the owner of the computers.

Bypass #2: Using a USB Hub

Usually not all ports are capable of being blocked. If users need a USB keyboard and mouse, then these devices can be unplugged to deliver the payload. Or an attacker can simply insert a USB hub into the host computer thus increasing the number of available ports.

Sometimes, if a computers case can be opened and the internals exposed. There may be a useable USB port hidden away inside the external case.

Defending against malicious USB devices with the Operating System

Modern OS's can allow a fine control over physical devices; this is usually very technical and can be difficult to manage successfully. The Windows OS allows Vista and above the capability to protect itself against undesired USB drives, the control is very granular and you can block different types of device classes. Open source operating systems (Linux based) also have a very similar method of controlling external devices through device-rules.

Bypass #3: Operating System Configuration & Permissions

When Operating Systems are locked down correctly, this last scenario is very difficult to bypass. Sometimes Bypass #1 may work if the security rules are slightly lax. If malware has infiltrated the OS by other means and has high enough permissions these protections can be disabled.

Conclusion

Despite having good anti-virus and anti-malware controls, your organisation may still be under threat from attack via innocuous USB devices. Recent improvements in Data Loss Prevention (DLP) software and security policies can help mitigate against most of these threats, but these solutions are not a magic bullet to preventing every single attack. Resourceful Hackers (and would be attackers) have already proved that they can potentially bypass these security mechanisms if they gain enough knowledge or understanding on what type of devices are permitted and by whom in the organisation.

USB attacks are often used by professional hackers and consultants in the following engagements:

- Social Engineering
- Red-Teaming
- Data Loss Prevention Assessments
- Bespoke USB Dead Drops

Appendix: References

The following references were used in relation to this paper:

Name	Description	URL
BadUSB	Paper concerning malicious USB flash drives	**https://srlabs.de/badusb/**
Hak5 USB Rubber Ducky	A programmable and customisable USB product.	**http://www.usbrubberducky.com**
Hak5 Android & USB Rubber Ducky	Demonstration that mobile devices are additional vulnerable to USB attacks	**http://revision3.com/hak5/update-with-the-rubber-duck/**
Psychson	Open source toolkit to convert regular flash drives into BadUSB devices.	**https://github.com/adamcaudill/Psychson**
Teensy PHUKD	Programming API to facilitate malicious USB behaviour	**http://www.irongeek.com/i.php?page=security/programmable-hid-usb-keystroke-dongle**
USB Armoury	An advanced programmable USB product.	**http://www.inversepath.com/usbarmory**
USB IDs	Registry of assigned Vendor and Product Identifiers used in USB classes.	**http://usb-ids.gowdy.us/read/UD/05ac**
USBdriveBy	An open source USB exploitation framework	**http://samy.pl/usbdriveby/**
Windows Vista+ USB Defences	A technical walkthrough on the native defences of the Microsoft Windows registry	**http://www.irongeek.com/i.php?page=security/locking-down-windows-vista-and-windows-7-against-malicious-usb-devices**