HONEYWELL FORGE

HONEYWELL CYBERSECURITY REPORT:

USB HARDWARE ATTACK PLATFORMS

USB Threats are More than Malware

Report

LEGAL DISCLAMER

This report is not an endorsement of any USB attack tools or techniques, nor is it an endorsement of any of the USB attack platforms and other devices mentioned within this report. The information is made available in order to provide a better understanding of the USB threat landscape and industry best practices for better risk management.

Honeywell does not endorse or recommend any third-party products, services, tools referenced in the document. Honeywell does not endorse or recommend the use of commercial penetration testing tools for any unintended or malicious purpose. Guidance and recommendations expressed herein represent industry best practices and are not a guaranteed protection against cybersecurity threats.

USB THREATS ARE MORE THAN MALWARE

Honeywell USB Threat Reports focus on the malware detected and blocked by our Secure Media Exchange (SMX) USB security solution. These reports provide valuable insight into how industrial and OT environments are exposed to malware, specifically originating from USB removable media.

These reports show the prevalence and potency of malicious files transferred via USB removable media, and that USB media remains a top vector into most OT environments, including industrial automation and control, buildings control and most aspects of critical manufacturing.[1] However, they purposefully do not discuss a different kind of USB threat: malicious USB devices that intentionally manipulate or misuse the USB protocol itself to cause harm. According to Ben-Gurion University's Cyber Security Research Center (CSRC), USB-born malware represent a small subset of the overall breadth of possible USB attacks – 20% of the attacks identified in the paper.[2] The majority of these attacks didn't involve traditional malware at all, using the USB protocol itself instead of relying on USB storage to transfer malicious files. These alternate attacks were classified as 'programmable microcontrollers' and 'USB peripherals that have either been maliciously reprogrammed or are capable of malicious use.' [3] It's no secret that malicious USB devices, including commercially available pen testing tools such as Rubber Duckies, Bash Bunnies and O.MG cables are prevalent, or that BadUSB and BadUSB v2 remain a threat, but how common are these malicious devices and how capable are they? To increase awareness of the remaining attacks, this report focuses on malicious devices rather than USB-born malware.

USB-based attacks are either intentionally programmed microcontrollers or maliciously reprogrammed USB Peripherals. While some attacks leveraging USB Peripherals, that are not reprogrammed, tend to rely on file-based malware, there are also attacks that leverage legitimate USB device classes for malicious purposes and these are discussed here as well.[4] This includes using USB devices to create backdoors, steal data, update drivers, etc.

Malicious USB devices date as far back as the early 2000s gaining momentum in 2010 with the introduction of the infamous Rubber Ducky. Some of these legacy platforms will be discussed further here as they are either still actively in development, have introduced new features, and/or are widely used today.

The USB standard is open and powerful and can easily be manipulated to produce robust USB Attack Platforms (UAPs) with a variety of capabilities.

In the past few years, UAPs have become increasingly popular. Many are easily built with instructions, material lists and source code available for DIY development. Many are available commercially offering complete ready-to-use UAPs for purchase.

The available platforms fall roughly into three categories:

1. Plug-and-deploy UAPs which are designed to execute an attack upon insertion.
2. Remote attack UAPs which leverage wireless communication paths to establish command and control capabilities. These devices are primarily designed to be implanted in a target network and then managed remotely.
3. UAP Appliances which are more comprehensive toolkits that may interface a target via USB and/or other means, and typically contain robust exploitation toolsets.

These devices could be deployed as plug-and-deploy or Remote UAPs but have additional general-purpose penetration testing capabilities that are best suited for direct, interactive use.

PLUG-AND-DEPLOY USB ATTACK PLATFORMS

Plug-and-deploy devices, or hot-plugs, are perhaps the simplest type of malicious USB device designed to perform a pre-defined attack upon insertion into a USB interface. These devices require someone either an attacker with physical access to the target system(s) or an unwary insider tricked by the malicious USB device to plug the device into a computer. Having a third party install the device presents a low risk to the attacker but is not always reliable. In an often-cited research project conducted by Google and the Universities of Michigan and Illinois at Urbana Champaign, 45%–98% of dropped devices were connected, with some connections occurring in as little as 6 minutes.[5]

However, even when the plug is guaranteed, the deploy still requires a combination of planning, timing and luck to be successful. Making the deployment of one or more payloads conditional based on host environmental variables, keystroke pattern matching, date/time or other conditions can make them more effective and harder to detect, but advanced reconnaissance will still be required to make these UAPs truly effective.

Plug-and-deploy devices tend to focus on HID attacks. While they appear to be one thing, e.g., a USB thumb drive, they identify themselves to the USB host controller as a keyboard or mouse and proceed to execute a keystroke injection attack or mouse-jacking attack.

RUBBER DUCKY

One of the first USB HID attack platforms to go mainstream was the Rubber Ducky. Introduced in 2010 by hak5, Duckies remain a favorite among pen-testers today. HID attacks are crafted using Ducky Script, a simple scripting language that allows any sequence of character strings, certain special keys (like Alt or the Windows key) and delays. The scripts are loaded onto a micro-SD card which is then inserted into the Rubber Ducky itself. When the Rubber Ducky is plugged into a computer, it will mount as a USB HID and start to type. Ducky Script got an upgrade with the introduction of the Key Croc (discussed below) to version 2 and was renamed QUACK in honor of the Rubber Ducky. QUACK leverages a Bash interpreter to enable much more sophisticated scripting.

Rubber Duckies are the same footprint as the average USB thumb drive and can easily be disguised as a benevolent storage device simply by snapping on a plastic cover.

“Duckies” have inspired many DIY variations of HID penetration esting tools, including the popular Raspberry Pi (Raspiducky[6]) and Digispark (duck2spark[7]). Arguably, most HID penetration testing tools owe at least some degree of homage to the original Rubber Ducky.

BASH BUNNY

Also the work of hak5, the Bash Bunny expands on the capabilities of the Rubber Ducky making it easier to load HID payloads through an integrated arming mode allowing two separate payloads to be loaded at once, all using a small toggle switch on the side of the device. This added flexibility required extra space and an LED to provide the user with needed feedback, so the Bash Bunny, though more powerful than a Rubber Ducky, is not as covert. While it would be easy to sneak one into a facility, it is unlikely that a reasonable person could be fooled into thinking it was just a normal USB drive.

USB CHAOS DRIVE

A plug-and-deploy platform that does not leverage HID attacks is the Chaos Drive. Introduced at Defcon 27 in 2019, the platform, built on a BeagleBoard computer, possesses multiple USB storage devices in one and presents different storage to the host controller based on a few different user-defined parameters. This is useful for covert movement of files which in turn is useful for injecting malware or exfiltrating files both with the ability to bypass detection by anti-malware or data loss protection controls.

For example, the Chaos Drive might present a perfectly clean drive to a USB scanning station, but then present a different unscanned drive to a target host. The second drive is never scanned, so it could contain a cornucopia of malware. Similarly, the platform can bypass DLP controls by writing files to the hidden drive instead of, or in addition to, the normally presented drive.[8]

Why are USB HID attacks so effective?

USB Human Interface Device, or HID, attacks are popular because they are highly effective. A malicious USB device presenting itself as a keyboard can directly inject keystrokes as if the attacker were sitting at the local keyboard. HIDs leverage the broad adoption of the USB standard and the nature of Windows to create attacks that are efficient, evasive and effective.

EFFICIENT

Keystrokes can be injected at high speed (the Rubber Ducky advertises 1,000 words per minute) to quickly inject payloads.

- Use keyboard shortcuts to control all aspects of the operating system or any installed applications
  Type commands directly into the command line
- Leverage scripting capabilities to directly execute complex malicious code.

EVASIVE

HID attacks are difficult to detect with security monitoring toolsand can leave a small forensic fingerprint as well.

- Bypass endpoint security controls by interacting directly with authorized applications
- Avoid detection by human users by running in hidden consoles and/or during idle time periods
- By spoofing legitimate peripherals, HIDs are difficult to detect by host monitoring controls

EFFECTIVE

Keyboards are the most common and powerful means for a user making HID attacks extremely versatile and effective.

- Keyboards are required input methods for most systems and are therefore almost always enabled by policy
- The keyboard is an extension of the currently logged-in user, so the keyboard has the same access to keyboard shortcuts,scripting and the command line
- Armed with user credentials, an attacker can use a HID to authenticate with escalated privileges on locked PCs

REMOTE USB ATTACK PLATFORMS

Remote USB Attack platforms are typically designed to be deployed as implants: plugged into a host within a target environment and left there indefinitely. Having remote access to a USB device provides significant advantages to an attacker. As Rogan Dawes describes in his 2016 DefCon session on remote physical attacks, the goal of these devices is, "... to create a stealthy bi-directional channel between the host and device with remote connectivity via 3G/Wi-Fi/Bluetooth and offload the complexity to our hardware leaving a small simple stub to run on the host."[9] Unlike Plug-and-Deploy tools, having an implant with built-in wireless communications provides a covert backdoor without using the target's network at all, bypassing network security and monitoring tools and thereby avoiding detection.

Some implants can be deployed in-line between a legitimate USB device and a host, typically a keyboard, enabling USB man-in-the-middle attacks. In its most basic form, the USB keylogger has been around for some time and is a prime example of this platform in action: small in-line devices capture keystrokes and store the contents onto a flash drive or microSD card for later retrieval. As we've seen with other types of USB attacks, inline devices have grown much more capable taking keylogging to a new level of sophistication. After all, to get a device deployed in-line requires some skullduggery, serious social engineering, or manipulation of the supply chain (see "making implants harder to detect" below). An attacker first has to sneak the device into a facility, install it onto a target system and get it back out. Now, with remote access and management via WiFi or cellular, the rewards of a successful implant are much greater and the risk is much lower; there is no need to physically retrieve the device once it's been deployed because exfiltration occurs over a covert network path.

KEY CROC

Hak5's latest entry into the HID attack arena is the Key Croc, and like other hak5 tools, they've now made in-line keylogging and HID attacks easy. Payloads can be initiated based on a variety of pattern matches (including regex support). For example, waiting for the target user to type 'ctrl-alt-delete', then capturing keystrokes up until the user presses 'enter' in order to grab user credentials to be used in subsequent attacks.

The Key Croc is pre-built and designed for ease of use, including built-in manageability using hak5's Cloud C2 command and control application. Useful features like covert remote access and automatic device spoofing make it difficult to detect. It's built on Debian and allows users to install additional tools extending the platform even further with little effort.

KEYSWEEPER

The Keysweeper is a "stealthy Arduino-based device, camouflaged as a functioning USB wall charger, that wirelessly and passively sniffs, decrypts, logs and reports back (over GSM) all keystrokes from any Microsoft wireless keyboard in the vicinity."[10] The keysweeper doesn't leverage the USB standard or USB interfaces, so technically it's not a USB based attack at all, however, it's mentioned here as an example of how USB devices, including cables, wall chargers, speakers, et. al. are so ubiquitous that one more isn't likely to be noticed. Therefore, impersonating such devices remains an effective tactic.

KEYVILBOARD

The Keyvilboard is a pre-built device with similar functionality to the Uber HID (see below), but available with either WiFi or 2G connectivity built-in. Like the Uber HID, the Keyvilboard is an in-line programable computer with two USB interfaces, an integrated WiFi access point and remote management and control. Key differentiators include its pre-built structure easily concealed inside a black plastic enclosure and significantly more covert than a home-soldered board with exposed wires, unlike the Arduino used by Uber HID, and the cellular option which removes a huge potential barrier. Hidden WiFi access points are very effective, but it means that the attacker needs to be within WiFi range. Especially for WiFi antennae built into small system-on-chip platforms, this can mean a very short range - as close as a few dozen meters. Using cellular, remote access is restricted only by cellular connectivity.[11]

MALTRONICS KEYLOGGER

Less feature-heavy than the Croc but significantly more covert is the Maltronics keylogger. It's purpose-built for covert keylogging, with remote management and an intuitive user interface. Available in an easily concealable solder-it-yourself version, it can easily be integrated into existing USB keyboards. If an attacker knows the make and model of keyboard used by the target, it only takes a few solder points to create a nearly undetectable manipulated version of the original to be re-introduced in to the supply chain.[12]

O.MG CABLE / DEMONSEED EDU

The O.MG cable is an extremely covert remote attack appliance. Hidden inside a cable that is nearly indistinguishable from an OEM USB charging cable is a tiny but powerful system-on-chip with WiFi, USB emulation, HID capability, and rudimentary command and control capabilities, available in a variety of flavors, with USB Type A, USB Type C, and Apple lightening connectors. In addition to the O.MG Cable, a less powerful version is available as an educational kit called the DemonSeed EDU, so not only is there a powerful and covert offensive USB platform out there, but there's an inexpensive kit available to educate others on how to expand and enhance that platform.[13]

O.MG KEYLOGGER

Launched at Defcon 28 Safe Mode, the O.MG keylogger promises to deliver the perfect balance between powerful keylogging and remote keystroke injections in the same covert form factors of its predecessor.[14] These cables are virtually indistinguishable from authentic cables, so if an attacker manages to get one of these cables in place, it's highly unlikely the user would ever notice.

POISONTAP

While Poinsontap can certainly be listed as a plug-and-deploy device, it's unique, comparatively, as it includes a persistent backdoor that will continue to work long after it's unplugged, qualifying it as a remote UAP. Based on the Raspberry Pi Zero, it too, is inexpensive to build and small enough to be easily concealable, making it a good option for an implant device. Because of its capabilities, which include hijacking internet traffic, stealing cookies and session data, establishing outbound websockets, DNS rebinding and other advanced techniques,[15] it was tempting to classify it as a UAP appliance, although it's clearly not intended for use as a penetration toolset.

UBER HID

The Uber HID is an example of an inline keylogger with keystroke injection and other advanced features. It's built on an Arduino atmega32u4 requiring only a handful of additional parts, some downloadable firmware, and some basic soldering skills making the Uber HID extremely accessible to DIY pen-testers and hackers. With a built-in access point, an attacker can connect to it remotely via WiFi to capture and view recorded keystrokes, and/or execute HID attacks.[16]

WHID 31337 ("ELITE")

The WHID 31337 is another remote attack platform with a GSM/2G enabled device with support for remote HID injection, mousejacking, RF replay attacks, audio surveillance, location tracking and more, all over covert wireless connections that, like other wireless platforms, can bypass air gaps and network defenses. The hardware is custom designed, making it slightly less accessible, but there are instructions for various modifications as well as instructional videos to support the DIY crowd.[17]

INPUT FROM THE EXPERTS

"I have been using keyloggers in my red teaming since 'Keygrabber PS/2' was a thing ... Currently, I am using O.MG cable and have written multiple payloads which are unique at its own. My [preferred] combo would be using the O.MG Cable with the ScreenCrab, together with my cellphone and a Raspberry Pi during a red team engagement. This will enable me to have a look at the screen and type whatever payload in an isolated network. The Pi Zero can also be used for keystroke injection (using RaspiDucky)."

Talib Nadeem Usmani
Lead Penetration Tester
Honeywell Cybersecurity Center of Excellence, Dubai

MULTITOOL USB ATTACK PLATFORMS

Certain USB attack platforms offer such comprehensive capabilities that they can only be described as USB multi-tools. Multitool UAPs include more than USB attacks, up to and including the full implementation of entire penetration testing platforms such as Kali Linux, or even other specialized non-USB hardware hacking capabilities. These platforms have the tools necessary for a penetration tester or an attacker to perform many – if not all – of the various stages in a typical cyber kill chain. Tools for enumeration, intrusion, exploitation, etc. are all at the user's fingertips alongside the universal serial bus connectivity that defines a UAP.

P4WNP1 A.L.O.A.

The P4WNP1 A.L.O.A., short for A Little Offensive Appliance, is an application framework created by MaMe82, which "... turns a Raspberry Pi Zero W into a flexible, low-cost platform for pen testing, red teaming and physical engagements". The ALOA has the ability to emulate network, serial, storage and HID devices, and includes a scripting language for HID attacks called HID Script. While it could be easily classified as a remote attack platform, the ALOA is built on the Kali Linux PI Foundation image, making the full extent of the Kali toolset available as well, so it definitely qualifies as an attack platform. it's built a hardware platform that costs only a few dollars, it's extremely viable; and because it measures only 66mm x 30.5mm, it is easily concealable and effortless to carry in a pocket.[18]

KALI NETHUNTER

NetHunter is a robust penetration testing platform by Offensive Security. It's basically a Kali Linux build for Nexus and OnePlus mobile devices with support wireless attacks such as 802.11 frame injections. Because NetHunter contains the entire Kali Linux toolset, it is extremely versatile, and because it can connect to a target as a USB device (via a cable), it supports HID attacks, BadUSB man-in-the-middle attacks, etc.[19]

USB ARMORY (MK II)

The USB Armory is a mobile computer built on an open source hardware design from F-Secure Foundry. While any bootable USB can be used to compromise a target computer, the USB Armory Mk II is a full computer platform in the form factor of a USB drive. It includes full USB device emulation as well as additional USB interfaces making this a powerful UAP, as well as an ideal platform for penetration testing, low level USB security testing and more. While Kali did not support the Mk II at the time of this writing, it does support the USB Armory Mk I. [20]

FLIPPER ZERO

The Flipper Zero is the only device discussed here that hasn't been fully released; it is currently an active Kickstarter project with an expected completion in 2021. The Flipper is also the only UAP to include a digital pet, in the form of a "Cyber Dolphin" Tamagotchi. Despite the cute appearance, the Flipper Zero is a powerful multitool UAP built for manipulating wireless access control systems. The Flipper has built in support for reading and emulating RFID cards and iButton keys, as well as sub-1GHz radio and IR transceivers with learning to enable the remote control of gates, garage doors, etc. If that's not enough, Bluetooth and NFC support makes this one dangerous dolphin. Of course, it wouldn't be a USB Attack Platform if it didn't have the ability to launch USB attacks: in this case, the provided USB-C port can be used in "BadUSB Mode" to emulate USB device and perform USB fuzzing.[21] A true multitool, the Flipper Zero is also extensible, with exposed GPIO interfaces to support add on hardware, and a modular plug-in system to support additional programs.

INPUT FROM THE EXPERTS

"The readily available components highlighted here should bring to life the reality of portable device threats; the significance has been backed up by numerous industry reports. This serves as a reminder that our perimeters are constituted by more than a firewall, and that it includes all things which traverse the boundary between external and internal, notably USB devices. Air-gapping does not preclude cyberattacks, but it does corner determined adversaries into choosing portable devices as the next best approach. Portable USB devices, exposed ports, and wireless are all part of perimeter protection and must be defended as such."

Eddie Wade
Honeywell Cybersecurity Center of Excellence Singapore

COMMON MALICIOUS USB ATTACKS

The attacks that can be performed by malicious USB devices are numerous and varied. Virtually any payload can be delivered in a variety of ways, leveraging the various capabilities of the USB standard in combination with powerful, embedded computing platforms. While it's impossible to list every possible attack, the following examples are provided to convey the breadth and severity of the USB threat. All examples described below are readily available on the surface web and/or commercially available via legal online shops.

AIRGAP BYPASS

Many critical network infrastructures are air-gapped, meaning they are disconnected from outside networks. However, most air-gapped networks are not entirely disconnected. To bypass an airgap, an attacker typically relies on finding and leveraging these existing authorized network connections and facing the additional challenges of evading detection by whatever network security controls are in place. From inside the air gap, it becomes easier to establish an outbound connection that can then be used for remote access, command and control. However, when a device with wireless capability is planted inside an airgap, a completely separate network can be established that bypasses the target's own network entirely. Using USB to emulate ethernet, the target computer becomes dual-homed to both networks, establishing a covert wireless connection directly into the target's air-gapped network

BIT BANGING

Bit banging refers to the use of software to generate signals typically handled by dedicated hardware. In the context of a USB attack, it is useful for device emulation, especially in smaller embedded devices where dedicated USB hardware components are unavailable.

CLIPBOARD MANIPULATION

The clipboard is a common part of daily computer use allowing objects to be copied and pasted in a variety of contexts. USB HID devices are able to leverage the clipboard by injecting common keyboard shortcuts, e.g., Ctrl-A, Ctrl-C and Ctrl-V to 'Select All', 'Copy', and 'Paste', respectively. The clipboard is therefore useful to an attacker for both data exfiltration, and a fast, covert method of transferring larger scripts or payloads.

CREDENTIAL THEFT

USB devices can capture credentials in several ways. The most obvious is through keylogging performed buy an in-line device. However, more sophisticated methods exist. By emulating a network interface, a USB device can redirect all network traffic to itself. This technique allows the device to force authentication to networked resources and attempt to capture the hashes used for authentication. Because most browsers operate even in the background, these attacks are possible on locked workstations.

DENIAL OF SERVICE

Denial of Service attacks from malicious USB devices could take several forms. A HID attack could be used to disable a necessary service, remove a user account, delete files or otherwise obstruct normal operation on the target computer. Additionally, the USB interface itself can be DoS'd by forcing the target's root hub to reset. This will cause all USB devices to disconnect, including any internal system components that are connected via USB (as is the case with many laptops and certain embedded devices). A network Denial of Service could be performed by leveraging a USB Ethernet interface to flood the host, or to cause the host to flood other connected networks. Using the same technique described in Credential Theft (above), an emulated Ethernet interface could redirect all network communications to itself and either redirect or drop them. While a USB DoS attack is limited to the host that the USB is inserted into, it could be devastating if the USB device is attached to a critical server, especially if the device is covert enough to avoid visible detection.

DRIVER EXPLOITATION

The USB Standard provides a process for which a device controller presents itself to a host controller, to, among other things, identify its device class, subclass, and interfaces. The host controller, in turn, uses this information to load the appropriate driver(s) to enable plug-and-lay operation of the device. When an appropriate driver is not available, the operating system of the host will prompt the user to locate and install the appropriate driver. This process may be used by an attacker to load legitimate drivers that possess known vulnerabilities, even if the driver's intended device isn't attached so that the vulnerability can then be exploited and used as a kernel-level attack vector to infect the host. This process can also be used to trick an unsuspecting user into installing a malicious driver crafted by the attacker: when the user inserts the malicious USB device, it tells the host controller that it requires a vendor-specific driver. Conveniently, the USB device can also present a mass storage interface that contains the malicious driver. The further addition of HID manipulation can select and install the driver.

DROPPERS

USB devices are excellent at dropping payloads. A USB mass storage device (a.k.a., a normal thumb drive) is designed to store files, and once connected to a target computer transferring those files to the host is possible in several ways. If Autorun is enabled, or if systems are unpatched and vulnerable to .LNK exploits, malware can move laterally onto the target on its own. Of course, using HID functionality a malicious USB platform could first enable Autrorun, or simply copy and paste the file to the host. By emulating Ethernet, networks can be redirected to malicious sites, open backdoors or enable network shares.

ENUMERATION

Because USB devices can attach as network interfaces, file storage and Human I/O devices, they are well suited for various types of reconnaissance and enumeration. Network enumeration, account enumeration (both local and network accounts), identifying applications and services, etc. is trivial when executed via a locally attached USB platform with advanced toolsets (e.g., Kali). In addition, USB devices can determine additional hardware capabilities through the USB standard device identification process: for example, by cycling through additional USB device types, it is possible to determine if applicable drivers for specialized peripherals are installed.

EXFILTRATION

Exfiltration via malicious USB devices can be done via file storage or network transfer. In the case of file storage, files can be copied to a standard USB Mass Storage interface and retrieved physically. The files can be copied to hidden storage interfaces to avoid Host-run DLP software from monitoring the USB device's filesystem in order to evade detection. However, it is becoming increasingly common to have remote capabilities on malicious devices allowing exfiltrated data to be transferred directly to a remote command and control server.

HID ATTACKS

Human Interface Device attacks, or keystroke injection attacks, use standard keyboard and mouse interactions to type/click within the target computer's interface. HID attacks provide the same capabilities to an attacker as if they were sitting at the target computer typing directly. If used with an administrative account, this could include almost anything: reformatting a drive; creating a user account; deleting files, directories or entire drives; installing or uninstalling applications; editing registry setting, etc. Most applications can be manipulated to a large extent, if not entirely, using keyboard shortcuts and hotkeys, allowing cleverly crafted HID attacks to interact with applications in a very precise manner. Typically, however, an HID is best leveraged to open the way for more sophisticated exploits: creating rogue accounts, or establishing a reverse shell, for example.

HID ATTACKS (REMOTE)

Combining HID attacks with remote monitoring and control overcomes the issues presented by traditional plug-and-deploy HID attacks. The attacker can interact directly with the target system and directly monitor the results, making the attack process adaptable and interactive, and therefore much more reliable and effective.

HID ATTACKS (DYNAMIC)

When Keystroke injection is combined with other UAP capabilities, a HID attack can be triggered based on a variety of conditions. If keylogging is supported (e.g., from an inline implant), specific HID payloads can be configured in response to legitimate keystrokes typed by an actual user. In-line USB devices are also uniquely positioned to both block legitimate keystrokes and/or inject new ones. This opens up the opportunity to force a user into typing something unintended. Examples could be as benign as intentionally causing embarrassing typos, or they could be very serious: for example, injecting characters when typing a password in order to force an account lock-out, or typing 'Windows-L' to lock the screen and force the user to re-authenticate, so that credentials can be captured by a keylogger. Other types of reconnaissance can enable contextual triggers: deploying the HID payload only if a certain user is logged in, a certain application is installed or certain devices are reachable on the network, etc.

JAMMING

Various wireless capabilities on malicious USB devices can be used to spew signals at a desired frequency to effectively jam wireless communications within range. Malicious applications include jamming GPS to prevent tracking, jamming WiFi to block wireless Ethernet communications or RF jamming to block two-way radio/walkie talkie communications.

LOCATION TRACKING

A UAP with GPS capability can be used to provide an attacker with the physical location of the malicious device. For Plug-and-Deploy UAPs carried by an unwitting third party, this can be useful to determine if the device is deployed in the correct target and also for device retrieval. Obvious location tracking is also useful for surveillance, especially when combined with audio/visual surveillance capabilities.

MOUSEJACKING

Mousejacking refers to the interception of wireless mouse transmissions and leveraging associated vulnerabilities to inject keystrokes. Because most non-Bluetooth wireless mouse systems are unencrypted, the communications can be easily intercepted and new commands can be easily injected. In addition, many systems fail to verify that the commands sent are for the same device subtype that is sending the packet, meaning once a session is intercepted the attacker is able to wirelessly inject keystroke on a target PC without any physical interaction.

REMOTE ACCESS

Remote access can be achieved in a variety of ways. A USB HID device can establish a PowerShell session and type scripts purely using keystroke injections. Similarly, malicious code can be written via raw HID injections, and then executed via a simpler PowerShell script. These injections could be used to change network settings, disable the target's firewall, connect to a rogue access point, etc. The ways to establish network connections with local access are numerous, and most, if not all, can be implemented via keystroke injection. Using the appropriate combination of USB device classes, a UAP can connect to an existing network, create out of band networks, emulate networks, or even establish a serial connection.

RF CAPTURE & REPLAY

USB wireless devices can be used to capture various RF communications and replay them. Almost any wireless technology can be monitored and/or manipulated, including security gates, door access cards, Bluetooth devices and more.

SURVEILLANCE

UAPs that leverage USB audio or video device classes can act as microphones or video cameras. Especially when combined with remote capabilities, a UAP implant can be an effective tool for surveillance activities.

SECURITY CONTROLS TO HELP PROTECT AGAINST UAPS

There are several methods available to help protect against malicious USB devices. The simplest method is to leverage existing controls such as Windows group policies to prevent the installation of drivers for specific USB device classes. Commercial USB device filtering solutions improve upon this model, providing more granular control. Finally, Trusted Response User Substantiation Technology (TRUST) provides improved device identification and highly granular control, including CAPTCHA based authorization. There are also hardware-based solutions available, including the Malicious Cable Detector by O.MG – the same group behind the O.MG Cable and the O.MG Keylogger.

WINDOWS GROUP POLICY (GPO)

GPO policies allow control over USB device types and subtypes, preventing the system from installing the necessary drivers to utilize that type of device. This is powerful but heavy-handed, prone to implementation issues, and extremely simple to bypass. GPO prevents the installation of devices, not the use of devices, so it's important that existing devices are first removed, otherwise the new policy will have no effect. GPO is also an all-or-none control: if you need to use a keyboard, you have to enable the installation of keyboards; there is no finer level of control. Because most computers require a keyboard, this leaves GPO prone to HID attacks and it has been repeatedly shown how a UAP, connected via USB, can be used to disable or change GPO settings to allow a broader range of USB attacks.

USB DEVICE FILTERS (VID/PID BASED DEVICE CONTROL)

There are several commercial solutions that help to further protect against UAPs by providing kernel-level filters based on USB device identifiers, including Vendor ID (VID), Product ID (PID), Serial Number, Device Class, or combinations thereof.

These solutions can be effective, as they provide much more granular control over the types of devices allowed, and they enforce the use of devices rather than the installation of devices. However, they share a common weakness in that they depend upon the USB standard's established USB device identification and classification protocols. Because USB devices self-identify themselves to host controllers, it is trivial for an attack to spoof any given device. For inline devices, this spoofing can be performed automatically, as is the case with the Key Croc, described earlier.

MALICIOUS CABLE DETECTOR (HARDWARE DETECTOR BY O.MG)

The Malicious Cable Detector is an in-line hardware, similar in design to the various in-line implants discussed above. By plugging the detector between a suspect cable and any USB host interface, the detector continuously analyzes the cable and provides user feedback via an alert light, or blocks the cable outright. The Malicious Cable Detector can also be modified (it is based on the Arduino IDE).[22]

TRUST (INTELLIGENT DEVICE IDENTIFICATION AND CONTROL)

TRUST (Trusted Response User Substantiation Technology) can be used to enforce who can use certain devices on specific nodes, enforcing the use of everything from keyboards to cell phones. It uses intelligent device inspection rather than accepting a USB device's self-identification providing a highly effective method of protecting against spoofed devices and UAPs. TRUST requires human validation using a Captcha system to prevent UAPs from connecting programmatically. Combined with granular user authorization controls, TRUST provides a less disruptive, yet more secure solution to USB device connectivity.

SECURITY IMPLICATIONS AND RECOMMENDED MITIGATION

The threat posed by UAPs is enough to warrant special consideration when developing cybersecurity policies, procedures and controls. While nothing can provide guaranteed protection, the following recommended mitigations may help to improve your organization's defenses against these types of cyber threats and are based on industry best practices.

SECURITY IMPLICATION	RECOMMENDED MITIGATION
Usb Devices Are Capable Of Impersonating (Spoofing) And Emulating A Diverse Range Of Device Types	On computers requiring frequent use of USB devices, implement controls that can validate device class and subtype and protect against unauthorized devices. At a minimum, configure your group policies to limit usage of certain device types. However, be aware that UAPs can spoof allowed device types to easily bypass these policies. Implementing stronger USB security controls, e.g., TRUST, will protect against most known malicious USB devices.
Malicious Usb Devices Are Capable Of Initiating Multi-Vector Attacks	A defense in depth strategy is required to protect against the various attack methods and vectors introduced by UAPs: monitor wireless spectrums for unknown or unauthorized SSIDs; monitor legitimate networks for new or unknown devices; and implement controls to enforce USB device policies (by type, subtype, etc.) at all endpoints. This must be considered in addition to traditional perimeter controls such as firewalls and USB media malware scanning.
Many Uaps Are Capable Of Evading Traditional Detection	Implement USB-specific security controls, such as TRUST, to detect and identify all USB attached devices. However, because UAPs are adept at evading certain types of detection, always implement a defense-in-depth strategy consisting of multiple detection methods
Uaps Are Much More Capable When Remotely Managed	Monitor wireless spectrums for unauthorized or rogue access points and perform regular assessments to detect new access points that may otherwise avoid detection. Because of the risk of cellular access, consider jamming or blocking wireless communications in extremely critical areas, using faraday protections or similar methods. However, understand that this type of heavy-handed approach will also interfere with legitimate wireless communicates.
Openly Accessible Usb Interfaces Are Susceptible To Quick And Effective Hot Plug Attacks	Again, implement USB-specific security controls, e.g., TRUST, where possible. On especially critical systems and/or on systems where software-based security controls cannot be installed, use physical USB interface locks to prevent users from inserting USB devices.
Usb Implant Devices Are Easily Concealed And Difficult To Detect	Perform regular audits to identify any new hardware. Network monitoring may be able to identify some rogue devices, but physical inspection will be required to locate devices using integrated access points or other rogue networks. Examine supply chain policies and procedures to identify weaknesses that could enable malicious or manipulated devices to be introduced via the supply chain.
Hid Attacks Are Effective Because They Inherit The Privileges Of The Currently Logged In User	Implement least-privilege account policies, and re-architect systems that rely on active administrative accounts if necessary. For systems unable to comply with the tenets of least-privilege user accounts, implement additional security controls and consider isolating these systems on the network and implementing strict USB device policies.

REFERENCES

1. Honeywell Global Analysis Research and Defense team. Honeywell Industrial USB Threat Report, 2020. Honeywell Inc. June 2020. http://hwll.co/sbc4f
2. Nir Nissim *, Ran Yahalom, Yuval Elovici. USB-based attacks. Malware Lab, Cyber Security Research Center, Ben-Gurion University of the Negev. Beer-Sheva, Israel. 2017
3. Ibid
4. Ibid
5. Matthew Tischer, Zakir Durumeric, Sam Foster, Sunny Duan, Alec Mori, Elie Bursztein, Michael Bailey. Users Really Do Plug in USB Drives They Find. University of Illinois, Urbana Champaign, University of Michigan, Google, Inc. 2016.
6. Hackaday. Raspiducky. (Document on the internet, https://hackaday.io/project/20842-raspiducky)
7. Tomas C. Low-cost USB Rubber Ducky pen-test tool for $3 using Digispark and Duck2Spark. (Document on the internet, 2018, https://hackernoon.com/low-cost-usb-rubber-ducky-pen-test-tool-for-3-using-digispark-and-duck2spark-5d59afc1910)
8. Mike Rich. Chaos Drive, because USB is still too trustworthy. (Paper presented at DefCon 27, Las Vegas Nevada, 2019)
9. Rogan Dawes, Dominic White. Remote Physical Access Attacks. DefCon 24. (paper presented at Las Vegas, Nevada, 2016)
10. Samy Kamkar. Keysweeper. (Document on the internet, https://github.com/samyk/keysweeper)
11. Luc Pluimakers, Matthijs Vogel. Keyvilboard. (Document on the Internet, 2019, https://keyvilboard.nl/en/)
12. Maltronics.com. WiFi KeyLoggers. (Document on the Internet, 2020, https://maltronics.com/collections/wifi-keyloggers)
13. MG. O.MG Cable. December. (Document on the Internet, 2019, https://mg.lol/blog/omg-cable/)
14. MG. O.MG Keylogger. (Document on the Internet, 2020, https://mg.lol/blog/keylogger-cable/)
15. Samy Kamkar. PoisonTap - siphons cookies, exposes internal router & installs web backdoor on locked computers. (Document on the Internet, 2020, https://samy.pl/poisontap/)
16. João Pedro Dias. UBERHid: Wifi Keylogger and HID Injector. (Document on the Internet, 2019, https://jpdias.me/infosec/hardware/2019/12/26/uberhid.html)
17. WHID (We Hack In Disguise). WHID-Injector, v1.31, WHID 31337. (Document on the Internet, 2020, https://github.com/whid-injector/whid-31337)
18. Rogan Dawes. P4wnP1 ALOA. (Document on the Internet, 2020, https://github.com/RoganDawes/P4wnP1_aloa)
19. Offensive Security. Kali Linux NetHunter. (Document on the Internet, 2020, https://www.kali.org/kali-linux-nethunter/)
20. F-Secure Foundry. USB Armory MkII. (Document on the Internet, 2020, https://github.com/inversepath/usbarmory/wiki)
21. Pavel Zhovner. Flipper Zero: Tamagotchi for Hackers. (Document on the Internet, 2020,https://flipperzero.one/zero)
22. O.MG. Malicious Cable Detector by O.MG. (Document on the Internet, 2020, https://shop.hak5.org/products/malicious-cable-detector-by-o-mg)

For more on TRUST and how Honeywell can help to protect you from USB threats please go to www.becybersecure.com

Honeywell Connected Enterprise
715 Peachtree Street NE
Atlanta, Georgia 30308
www.honeywell.com

Honeywell Forge Connect | Rev | 10/ 2020

THE
FUTURE
IS
WHAT
WE
MAKE IT

Honeywell