2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)

Spyduino: Arduino as a HID exploiting the BadUSB vulnerability

Evangelos Karystinos
Department of Informatics
University of Pireaus
Piraeus, Greece
p07176@students.cs.unipi.gr

Antonios Andreatos
Div. of Computer Engineering and Information Science
Hellenic Air Force Academy
Dekeleia, Attica, Greece
antonios.andreatos@hafa.haf.gr

Christos Douligeris
Department of Informatics
University of Piraeus
Piraeus, Greece
cdoulig@unipi.gr

***Abstract*—BadUSB is a critical vulnerability which has not yet been successfully addressed. BadUSB attacks are based on reprogramming the firmware of a USB device. This paper presents Spyduino, a properly programmed Arduino appearing as a Human Interface Device (HID), which can operate in most of the common operating systems (OSs). Spyduino exploits the BadUSB vulnerability in order to gain access to sensitive data and send information to the cloud via FTP. In this implementation, Spyduino is embedded in a USB keyboard and sends sensitive OS and user information to an FTP server without user permission. Various countermeasures are discussed and potential extensions are presented.**

***Keywords*—Spyduino, BadUSB, Human Interface Device, HID, IoT, Arduino, passive FTP mode, DFU mode.**

I. Introduction

Human Interface Devices (abbreviated as HID) are common electronic peripherals which facilitate human-computer interaction. Numerous HID devices exist, the most common being keyboards, mice, pointing devices, as well as speakers, webcams and headsets [1]. HID are frequently considered innocent. Nevertheless, even these devices can be exploited.

BadUSB is a Universal Serial Bus (USB) device whose firmware has been modified [2], [3]. From then on, the original USB device acts maliciously, according to the way programmed by the attacker [4], for example, by installing backdoors, key loggers or password sniffers. One type of BadUSB attack is for the device to spoof the OS simulating another common USB device above suspicion (usually a HID) in order to avoid being detected. BadUSB malicious products are being sold massively.

Arduino is suited for developing custom BadUSB products and for testing possible exploits. As a physical system, an Arduino device connected to the Internet and to cloud-based applications belongs to the Internet of Things (IoT) products [5].

This paper presents Spyduino, an Arduino UNO with reprogrammed firmware simulating a keyboard, which can be hidden in a normal USB keyboard. After the malicious keyboard is plugged-in, a pre-written script is executed and keystrokes are simulated, performing a set of malicious actions. Spyduino is a product using a low cost board with no additional hardware [6]. The UNO version was selected because it is the most commonly available Arduino board [7]. UNO does not have the special features of other versions, like Arduino Leonardo that has keyboard human interface device (HID) libraries, but it includes a separate microcontroller for USB serial communication.

Monk pointed out that Arduino is suited for various types of projects [8], including a keyboard prank, and described methods for programming it in many ways to work as various devices, apart from the original Arduino project. Spyduino is an Arduino UNO hidden inside a common USB keyboard. At a predefined time, the Arduino can start issuing keystrokes corresponding to a predefined script, harvesting useful information and sending it to a predefined FTP cloud server. Third party users can connect to this cloud server to collect this information without any user awareness or permission.

The Arduino sketch can be created via the Arduino IDE, [9], which is a free platform. Installing extra libraries is not required in this project. The Arduino UNO does not use special features like other Arduino boards, so any version of the platform is valid for this project.

The firmware of Spyduino has been upgraded using Flip, a free tool distributed at the Atmel online site [10].

After programming Spyduino, there is no need for new drivers when connecting to a terminal. Universal HID keyboard drivers are being used, which are normally pre-installed in almost every personal computer's Operating System (OS).

II. Related Work

A survey on USB-based attacks can be found in [4] and [11]. The proposed system belongs to category 'A' - Programmable microcontrollers. Similar products are Rubber Ducky [4], [12] and Evilduino [13].

Fig. 1. Rubber Ducky

Most of the related devices use similar descriptors to emulate or simulate a keyboard or a mouse, sending keystrokes or executing data from an embedded SD card, trying to infiltrate the user's interface, being a faker of the legitimate user. These devices are either commercial products, selling massively online, or they are used as system

2325-2944/19/$31.00 ©2019 IEEE
DOI 10.1109/DCOSS.2019.00066

penetration test tools by researchers and security software developers.

Rubber Ducky (Fig. 1) is a commercial ransomware threat developed in 2010 with a primary aim to encrypt user files by acting as a keyboard with pre-entered keystrokes [4], [11], [14]. It works on every operating system that recognizes a USB stick as the main input device keyboard. The most probable scenario is that the attacker will offer a PIN code to decrypt the files in exchange for money. The extremely low cost of the Rubber Ducky USB stick makes it very attractive to a variety of attackers [15].

Rubber Ducky supports a simple scripting language that enables an attacker to craft payloads capable of changing system settings, opening back doors, retrieving data, initiating reverse shells, or basically anything that can be achieved with physical access – all of which are automated and can be executed in a matter of seconds [4].

Malduino (Fig. 2) is an Arduino microcontroller programmed to emulate a keyboard, sending keystrokes to a computer [16]. It is sold online and its scripts are written in DuckyScript [17]. There are two versions of Malduino. The Lite version stores a script on its onboard memory, but it is limited to 32KB. There exists an online script converter that can change the code that the user has written to a script language that Malduino understands. The converted script can be uploaded using the usual Arduino IDE. After that process, there is a hardware switch that makes it ready for using as a keyboard. The Elite version is more advanced compared to the Lite one. There is a additional microSD capability, where a maximum of 16 scripts can be stored. Dip switches onboard define which script is going to be used when it is used as a keyboard.

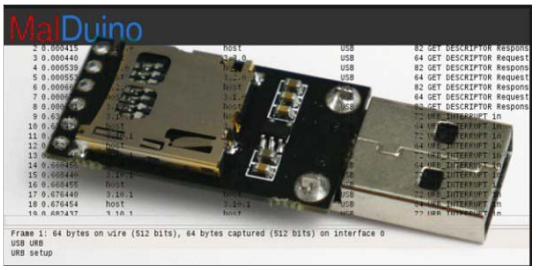

Fig. 2. Malduino Elite

Evilduino uses an Arduino microcontroller, reprograms it and injects malicious keyboard and mouse strokes on the attacked computer [11]. Evilduino can also emulate a keyboard but it is immediately detected as a new device apart from a HID device, because its descriptors have not been overwritten for use only as a keyboard. There is also the need for the user to upload new sketches, depending of the needs of the user's goals. So it is immediately blocked in organizations prohibiting the connection of USB devices except of HID.

The differences between the aforementioned systems and Spyduino are the following:

- It is hidden inside a normal USB keyboard, so it cannot be visually suspected. The related projects described before are evident, thus, they can be easily detected.
- It works in Windows and Linux.
- It does not need an extra USB slot.
- It uses FTP in order to transfer the files.
- It does not create new user accounts (which may be detected).

Spyduino can not be easily programmed by an average user, since it needs special knowledge for upgrading the Arduino UNO firmware in order to be able to upload a new sketch. Even if that is done, further knowledge is required for upgrading again with custom keyboard firmware so that it rolls back to a Spyduino.

III. Programming the Spyduino

The Spyduino's Arduino UNO is programmed using Atmel's Flip software [10]. In order to re-program the firmware, the Arduino should first get into the Device Firmware Upgrade (DFU) mode [18]. The DFU mode is an operational mode that gives the programmer the opportunity to rewrite the descriptors of the device. The UNO board which is used in this project can get into this mode by short-circuiting the pins as shown in Fig. 3.

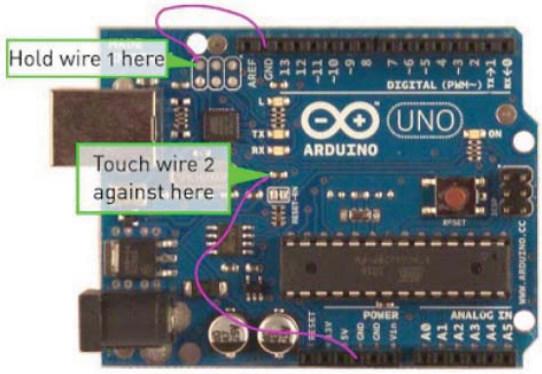

Fig. 3. Enabling DFU mode on Arduino UNO

The spying script is written in shell language (DOS batch language or PowerShell for Windows, bash for Linux). The spying script performs the following functions:

- In case of root privileges, it disables the Firewall.
- It sends specific system and user files via the passive FTP mode to the cloud.
- It re-enables the Firewall.

Afterwards, the script is converted into a keystroke sequence which is then programmed in Arduino using the DFU mode. The programmed Arduino is then hidden in a USB keyboard and connected to the keyboard USB wires.

One should note that DFU is simultaneously a capability and a potential vulnerability. Vendors from all over the world use this feature in order to reduce the cost of supporting their products. When a bug is found after the product has been released, or in cases that a customer asks for support for a malfunctioning device, the vendor releases a new firmware for the device and asks consumers to run a DFU, having made simple the procedure for anyone to use. DFU is an operational mode that forbids any other activities from the device while in this mode. The Arduino sketch must be loaded before modifying the firmware. After upgrading the firmware, there is no way to upload a sketch, or use the

device as an Arduino in any way. The change of the USB descriptors makes the device a HID keyboard.

IV. Spyduino Features

Spyduino has the following features:

- It is programmed to operate on any targeted OS.
- It does not create new user accounts and leaves minimal log traces.
- In default-installed Windows and Linux-based OSs, it gains root privileges. This can be done by bypassing tools like the UAC Windows tool or by editing the Linux iptables.
- It uses the passive FTP mode (Fig. 4) in order to avoid being detected [19]. In the passive mode, the client sends a PASV command from a random high numbered port to port 21 of the cloud server. The server responds with the specific number of the ephemeral port for the client to connect. Most firewalls allow inbound traffic from the client side that initiated transfer.
- It is hidden inside a normal keyboard. Since it is not restrained to specific keyboard vendors, it can be hidden to any keyboard used by the target machine.
- The whole spying operation is completed in less than a minute.

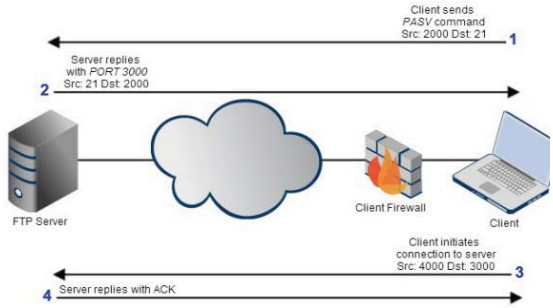

Fig. 4. Passive FTP mode

The predefined script can issue scripts for multiple OSs, by refreshing sessions after the end of a possible OS attack. For example, after terminating the command prompt at Windows it can start a script that works on Linux. In this case, a previous script that was designed to work on Windows will just have no effects on Linux. It will only result on keystrokes with no meaning, that will seem random before the bash script starts running.

V. Spyduino Operation

When the Spyduino keyboard is connected, the Arduino takes control after a predefined time delay. It opens a CMD terminal with administrator privileges (Fig. 5) and then:

- If there is an active UAC account, it approves the initiation of the command prompt.
- It disables the Firewall. This is done using the netsh command in Windows. In Linux versions, the iptables can be modified if there is such a need.
- It connects to the FTP cloud server in passive mode.
- It uploads selected files to the FTP server (Fig. 6).
- It terminates the connection and re-enables the Firewall.
- Finally, it terminates the command line session.

Fig. 5. Command prompt keystrokes script example operation

For Windows 10 target systems, at the start of the command prompt session Spyduino copies the path of the working directory which is the user's account. It does not matter if it is an administrator or another user, Spyduino targets this user's personal data.

In this folder, personal data, such as contacts, links, documents and pictures, are stored. Every targeted object can be uploaded to the FTP cloud server. Furthermore, there are many interesting files inside the 'windows' directory. Uploading files from the 'windows' directory sometimes requires to change the permissions or the owner of some folders. This can be programmed to happen automatically using the Windows Powershell.

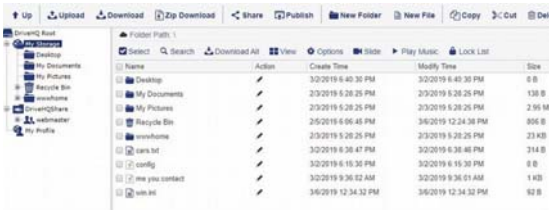

Fig. 6. FTP cloud service with intercepted files

The embedded in the OS file compression utilities can be used for joining many files in one, in order to facilitate uploading. For example, each contact is stored in a separate file. If we try to upload multiple files, a considerable amount of time might be needed. In this case, compressing all these files into one is very useful, so that only one file will be uploaded.

VI. Countermeasures

In order to limit the occurences of Spyduino and similar BadUSB attacks, there are some measures that have to be taken. It is generally difficult to completely avoid such an attack, especially if the attacker knows the kind of security that is used by the system. If the attack is successful, it is already too late to take any action. The system that has been compromised should be taken away from the network and be totally checked for malicious hardware or injected software. In some cases, when a computer is not connected to the internet, Spyduino can be used for encrypting important files and leaving traces for contacting the attacker, so that the attacker can ask for ransom. That is the reason companies should be very careful and prepared for this kind of attacks.

In case of such an event, it is probably impossible to restore the damage done in the system.

Duckhunt is a free Windows application, created specifically for this kind of exploits [20]. Duckhunt runs as a background process and continuously monitors the key typing speeds. It also blocks all the HID if it detects unusual speeds. But if the Spyduino programmer has set a small delay between keystrokes, Duckhunt will not take any action.

Administrator rights and folder permissions on Windows 10 can be modified with a registry edit [12], [13]. Thus, there is a need to ask for the administrator password for critical operations. This will limit the likelihood of having malicious actions in folders that are critical for the OS, like for example, c:\windows\system32.

Disabling the USB autorun feature in Windows is recommended. Antivirus and protection software should always be up-to-date. The number of trusted users which use a computer should be limited and access should be restricted to other people. Some additional guidelines follow.

- The employees of a company should not share computers or USB devices.
- In case a keyboard was used in an unauthorized computer, it should be treated as non-trusted and get checked for potential security breaches.
- The supply of USB keyboards and, generally, of USB peripherals should be allowed from trusted vendors only.
- The use of non-upgradable USB devices adds an extra level of protection.
- The restriction of adding new peripherals before having them checked by the security department.

A general protection from this kind of attack would be a physical protection [10], so that future unauthorized device connections are prevented. There are special products for that reason in the market, like the Lindy port blockers [21]. They are small cartridges that fit in the USB ports and cannot be easily extracted without a special tool. Nevertheless, in the Spyduino case, this measure will probably not work, as it is a camouflaged keyboard which is supposed to be connected for normal operation.

It is recommended to use a device control application that will monitor the use of all the devices connected to a system [12]. This feature has some advantages, like working in many OSs and being able to make a policy for a company or to disable USB ports. On the other hand, this software usually makes applications run much slower and it can be difficult to configure. Furthermore, this increases the related costs and creates a need for permanent support for this feature. Finally, it is not sure that it will protect a system in every occasion because, in the occasion of Spyduino, it will ask for a new device twice when the keyboard gets connected. If the user does not get suspicious and accepts the request, the attack will go on unattended.

Computers should be video recorded in a 24/7 base, so that no unauthorized personnel should get access to a computer terminal, even at night. The potential attackers are not only programmers or people from the computer science department. The original attacker could hand over a Spyduino to a person from the cleaning personnel, who could easily switch a normal keyboard with a Spyduino at the night shift. There is no other way to prevent such an attack, apart from the video recording of the computer.

Systems with no internet connection cannot be threatened by Spyduino, since its major feature requires uploading to the cloud FTP. Moreover, most BadUSB attacks have this flaw. This could be fixed by providing extra features to the Spyduino which are proposed in the next section.

VII. Future extensions

Spyduino is a project full of potential for future expansions. Many advanced Arduino boards using special libraries with specific capabilities may significantly enhance Spyduino possibilities. Moreover, Arduino shields can provide Spyduino with extra features. One example would be a Wi-Fi shield with a predefined configuration as Access Point where the attacker could connect and manipulate Spyduino using real-time remote control, issuing commands and extracting data from a computer.

In order to vastly extend the minimum distance between the attacker and Spyduino, a GSM shield could be used. Thus,, Spyduino could be online, uploading files and information without the need of the victim-computer to be connected to the Internet. Even more capable, would be a Spyduino that could monitor the network traffic in a private network, sending the capture file to the attacker via the GSM-internet shield for further analysis.

One could also use a small version of Arduino, like the Nano, hidden inside other USB devices, like a headset, a speaker or a mouse. Even a USB hub could be a Spyduino.

Spyduino is designed for computer systems but could also be used for mobile devices. A small-sized Arduino board like Nano, programmed for Android or Apple OS and inserted in a mobile USB peripheral, could work the same way as in the computer systems. Such a device could be a docking station, a sound station or even a charger, which are perfect in terms of size for the Arduino Nano. Android systems have minimum protection against such attacks, because the security system of smartphones and tablets is designed to provide a minimal interference to the normal average user operations.

Further future plans involve enhancing the capabilities of Spyduino by intercepting audio clips from the microphone or video clips from the web camera. We also plan to run the script when the user is absent, by detecting user activity.

VIII. Conclusion

In this paper, we have presented Spyduino, an Arduino UNO programmed to issue pre - programmed keystrokes corresponding to a specific script exploiting specific user and OS files.

Spyduino cannot be detected because it is hidden in a common USB keyboard. When the keyboard is plugged in a USB port, only the keyboard is detected. The key features of Spyduino are the following:

- Very low cost.
- Can execute keystrokes at superfast speeds, much faster than humans.
- Can run powerful scripts.
- Can be modified to be used in various OSs or in a variety of User Interfaces.

- It is hidden inside a normal keyboard with no extra cables.
- It is difficult for the user to suspect a hardware hack; usual software measures do not affect Spyduino.
- It can be programmed to run at a specific time, or in a loop or after every computer restart with a certain pre-specified delay.

Various counter-measures have also been presented, as well as future extensions which can make Spyduino even more powerful.

REFERENCES

[1] https://www.techopedia.com/definition/19781/human-interface-device-hid

[2] K. Nohl and J. Lell, "BadUSB - On accessories that turn evil", presented at BlackHat 2014. Available online from: https://srlabs.de/wp-content/uploads/2014/07/SRLabs-BadUSB-BlackHat-v1.pdf.

[3] A. Caudill and B. Wilson,"Making BadUSB work for you", presented at Derbycon 4.0. Available online from: https://youtu.be/xcsxeJz3bII.

[4] N. Nissim, R. Yahalom, and Y. Elovici, "USB-based attacks", Computers & Security, vol. 70, pp. 675–688, August 2017.

[5] P. Desai, Python Programming for Arduino (chapter 9). Packt Publishing, 2015.

[6] S. Monk, 30 Arduino Projects for the evil genious, Second Edition. McGraw-Hill Education, 2013.

[7] https://store.arduino.cc/arduino-uno-rev3

[8] J. A. Langbridge, Arduino Sketches: Tools and Techniques for Programming Wizardry. Wiley, 2015.

[9] F. Perea, Arduino Essentials (chapter 2). Packt Publishing, 2015.

[10] http://domoticx.com/firmware-atmel-flip/

[11] C. Cimpanu, Here's a List of 29 Different Types of USB Attacks. https://www.bleepingcomputer.com/news/security/heres-a-list-of-29-different-types-of-USB-attacks/

[12] U. Shafique and S.B. Zahur, Towards Protection Against a USB Device Whose Firmware Has Been Compromised or Turned as 'BadUSB'. In: Arai K., Bhatia R. (eds), Advances in Information and Communication. FICC 2019. Lecture Notes in Networks and Systems, vol 70. Springer, Cham, 2020.

[13] http://forum.notebookreview.com/threads/29-types-of-USB-attacks-and-how-to-stay-safe-from-them.816068/

[14] K. Savage, P. Coogan, and H. Lau, The evolution of ransomware, Symantec, August 2015. Available online from: https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/the-evolution-of-ransomware.pdf.

[15] https://shop.riftrecon.com/collections/all/ducky

[16] K. Opasiak and W. Mazurczyk, "(In)Secure Android Debugging: Security Analysis and Lessons Learned", Computers & Security, vol. 82, pp. 80-98, May 2019.

[17] https://github.com/hak5darren/USB-Rubber Ducky/wiki/Duckyscript.

[18] https://www.arduino.cc/en/Hacking/DFUProgramming8U2

[19] E. D. Zwicky, S. Cooper and D. B. Chapman, Building Internet Firewalls, Second Edition. O' Reilly Media, June 2000.

[20] https://hackaday.com/2016/10/28/duckhunting-stopping-RubberDucky-attacks/

[21] https://www.cio.com/article/3192878/9-essential-tools-for-the-security-conscious-mobile-worker.html#slide4.