ISSN XXXX XXXX © 2017 IJESC

Research Article Volume 7 Issue No.7

A Review of Detection of USB Malware

Dr. Sunil Sikka[1], Utpal Srivastva[2], Rashika Sharma[3]
Associate Professor[1], Assistant Professor[2], M.Tech Student[3]
Department of Computer Science & Engineering
Amity University Haryana, India

Abstract:
USB device is a very common device now days and because of its user friendly nature and ease. It is used by massive group of people. The hackers are taking advantage of this technology and planning a malware inside the usb device. It is also one of generally focus technologies by hackers. The usb with a malware is known as BAD USB; generally the malware is present in the firmware of the usb so it remains undetectable. As the firmware of the device is not scanned by the system and other applications. This flaw is the serious risk to our operation system as the malware can be undetected and it can come easy into our operating system (OS), without even the knowledge of the users.

Keywords: Bad USB, Malware, HID (Human Interface Device), Firmware, Operating system (OS), USB, CMD

I. INTRODUCTION

Hackers can easily plant a malware in the usb device and that usb with a malware is known as **"Bad USB".** Recently security researchers have disclosed this vulnerability and hackers are taking full advantage of this by planting a malware in the USB device.[L] The malware remains undetected as it is present in the firmware of the usb device and generally the firmware is not been scanned. This keeps our operating system (OS) in a danger & we are prone to risk because it stays undetected and can come in the OS without the knowledge of the user. These attacks are happening because devices such as keyboard, printers, web cam etc are termed as safe in our systems by default. That's means that these devices can not harm the OS and they are in safe category and they also have access to the system. In these attacks the usb devices are behaving to be a keyboard (they function like a keyboard) .the sign of this type of attack are like a notification that a keyboard is been attached or CMD window will get open automatically. It can just a fraction of second and disappears as the duration is so less that the user is not aware of them or if the user is aware still its vey late to stop the threat.[A] The hackers have created a malware called a BAD USB that can be used to completely take control of the pc that means just by transferring a small file of malware , all the data and system file can be at risk and anything can happen to pc , its completely a hackers call what he wants to do with the system [L] The security researchers say's that there is no fix for this kind of attack. The only solution is to stop sharing your usb devices. In many organizations use of usb devices is strictly banned because of the same issue we are talking about. In this attack of bad usb the USB device registers as many different types of devices. For example a USB flash drive could register itself as both a storage device and a keyboard, enabling the ability to inject malicious scripts. This functionality is present in the Rubber Ducky penetration test tool [16], which is now available for public sale. Unfortunately, because USB device firmware cannot be scanned by the host, antivirus software is not positioned to detect or defend against this attack. This problem is not just limited to flash drives: any device that communicates over USB is susceptible to this attack. We observe that the root cause of the Bad USB attack is a lack of access control within the enumeration phase of the USB protocol.

1.1 BUILDING A BAD USB DEVICE

1. Creating the test environment using virtual environment with window operating system.[L]
2. Gathering and installing the tools & applications required to make BadUSB device[L]

3. Tool used are [L]:

- DriveCom
- Duck encode.jar
- Embed-Payload
- Malware Script
- Custom Firmware
- Injector
- SDCC (Small Device C Compiler)
- JRE (Java Runtime Environment)

All the above mentioned steps include tools, applications and files are required to successfully build a Bad USB device.[L]

1.2 Impact of Bad USB Attack

- Enter keystrokes: Mimic like a keyboard and issue commands on behalf of the user.
- Make changes in file**:** Modifies the files, add or delete the files.
- Affect Internet activity: Can take control of your system and can send emails on your behalf and reset you're save passwords.

1.3 Challenges of Bad USB

- The malware lies in the firmware of the USB device, but not in the flash memory so the attack. This makes it difficult to locate the attack code and remains undetected.
- Even after scanning the device this malware code is not recognized.

- An anti-virus scan or a reformat are general ways of dealing with such threats. But With Bad USB however, the threat is Undetectable, since it is not present in the flash memory of the drive but at the firmware of the device that controls how the device operates. USB controller chips' firmware gives no protection from reprogramming and there is no effective way to detect a corrupted USB device, this is due to "Malware scanners cannot access the firmware running in the USB devices. We cannot make out that the firmware of the USB devies is modified or not.

2. SAFETY MEASURES.

The team of security researchers Adam Caudill & Brandon Wilson and Karsten Nohl & Jacob Lell have disclosed the flaw in the USB and has given detailed explanation about the USB firmware initialization process and how the malware remains undetected & executes itself is one of the major aspects [L]. Researcher Nohl has given a brief summary on the process of USB firmware initialization took place this will help us in understanding exactly on what stage does the malware trigger itself and come in action. The explanation is given below.[L] The USB devices have controller chip inside them. This chip is responsible for giving details to the operating system that what type of function USB device has to offer. The controller chip has a firmware embedded that starts the registration process when the device is attached to the operating system. All USB devices which are connected to the operating system have to go through a registration process before the end user can use it [L]. The procedures for the registration are [L]:-

1. Registration request & temporary address response:
The USB device when connected sends a registration request to the OS to this the OS replies with a temporary address [L].
2. Send Descriptor: when the USB device receives the temporary address sends descriptor information to the OS of what type of device is connected. E.g. webcam, keyboard etc.
Depending upon the information the OS install the appropriate drivers [L].
3. Set Configuration: After the drivers are installed successfully the proper configuration is required so that the USB device can work effectively.[L]
4. Normal operation: once the proper installations and configuration process is complete the device can start to function as its normal operations.[L]
5. Re-Register: till the time the USB is connected to the OS, it can re- register itself at any point of time without needing the users permission. That means it can re-do the entire process of registration mentioned above as many times it wants and that to without the permission of the user and every time the OS will provide it a new temporary address[L].

This ability of USB devices to re-register itself at any given point of time is what the hackers are taking advantage of and they are targeting this flaw of the USB for their own profit.

2.2 Existing Methods

A very limited number of counter measures are there for this newly found threat exists. BAD USB is considered as one of the worst IT flaw ever found. Researchers now have recommended this basis and simples that can keep our OS safe. However, there are non-technical controls, which are as follows.

- **Create a Blacklist**
- **Avoid automatic USB installation**
- **Disable inactive USB ports**

1. Create a Blacklist:

2. Avoid automatic USB installation: Be default, In Windows automatically drivers are been installed for the devices that are connected to the computer. However, we do not want Windows to automatically install the drivers, for that we can use the method listed below to avoid that.

- step 1: Change device installation setting
- Click Start, type devices and printers in the search box, and then click Devices and Printers.
- All the devices connected to the computer are listed, including monitor, keyboard, mouse, printer, and so on. Under **Devices**, right-click the icon for the computer, and then click **Device installation settings**.
- A new window pops up asking you whether you want Windows to download driver software.

Click to select **No, let me choose what to do**, select **never install driver software from Windows update**, and then click **Save Changes**.

- Click **yes** when you are prompted for confirmation.

3. DISABLE INACTIVE USB PORTS:

Sometimes we need to disable the inactive ports to safeguard our OS from getting trapped. This is very useful as it prevent copying of vital data. Disabling USB ports also keeps a network from becoming infected viruses and malwares.

How to disable the USB ports

1. Access the Run box or the Search box in the Windows Start Menu.
2. Open the Windows Registry by typing **regedit** in the Run or Search box.
3. In the Registry editor, open the below folder.
4. In the USBSTOR, locate the **Start** DWORD and double-click "Start" to edit the value. Change the Value data to "**4**" and then click Ok.
5. Once completed, close the Registry editor. Windows will no longer start the USB device when detected.

4. CONCLUSION AND FUTURE SCOPE

We discussed about BadUSB devices' known and unknown threats which the user is unaware of. As the BadUSB code is present in the firmware rather than a mass storage part, it is difficult to scan the firmware and detect whether the firmware is modified or not. A big thanks to the security researchers who made this flaw public and now public is aware of it and now it's a challenge for the manufacture to come up with good solution to fight against this attack of BadUSB. Hence till now this vulnerability remains unsolved.[Q] However there are few precautions we can take to safeguard our system but these are technical protections. Therefore after lot of research we have come to the conclusion that the existing methods are sufficient enough to provide the safety protection to our system .we suggest that the user should try not to trust the devices which does not belong to them. this will help in keeping the system safe and secure [Q].

5. REFERENCES

[1].Jakob Lell, Karsten Nohl, "BadUSB - On Accessories that Turn Evil", BlackHat USA, Location: Las Vegas, NV, August 2014.

[2].Adam Caudill, Brandon Wilson, "Making Bad USB work for you", Derbycon, Location: Unknown, September 2014.

[3].NullByte. "How to make your own bad usb". Internet: http://nullbyte.wonderhowto.com/how-to/make-your-own-bad-usb-0165419/, November. 2014 [February. 1, 2016].

[4].Pentesting Shop. "Make your own bad usb device". Internet: http://www.pentestingshop.com/pentesting/make-your-own-usb-rubberducky-using-a-normal-us b-stick/, Publish date unknown. [February. 2, 2016].

[5].David Kierznowski, MSc (Royal Holloway, 2015) Keith Mayes, ISG, Royal Holloway" BadUSB 2.0: Exploring USB Man-In-The-Middle Attacks1"

[6].Bertin IT, CNIM Group."Bad usb unpatchable flaw". http:/ /www.bertin-it.com.

[7].Steven J. Vaughan-Nichols" Big, bad USB security problems ahead" file:///C:/Users/Dell/ Desktop/bad% 20usb%20 amity /Bad USB_- Big-bad-USB-security-problems-ahead-_-ZDNet % 20(1).pdf.

[8].Manish Kumar Sahu et al," A Review of Malware Detection Based on Pattern Matching Technique" International Journal of Computer Science and Information Technologies, Vol.

[9].A. N. Magdum, Dr. Y. M. Pati" A Secure Data Transfer Algorithm for USB Mass Storage Devices to Protect Documents". International Journal of Emerging Engineering Research and Technology Volume 2, Issue 4, July 2014, PP 78-84 ISSN 2349-4395 (Print) & ISSN 2349-4409.

[10].Sarat Kompalli, Intel Corporation" Using Existing Hardware Services for Malware Detection" 2014 IEEE Security and Privacy Workshops.

[11].R. Bhakte et al," Security Controls for Monitored Use of USB Devices Based on the NIST Risk Management Framework" Information Systems Assurance Management Concordia University of Edmonton.

[12].Moritz Jodeit at all," USB Device Drivers: A Stepping Stone in to your Kernel".

[13].J. Zico Kolter, Marcus A. Maloof" Learning to Detect and Classify Malicious Executables in the Wild" Journal of Machine Learning Research 7 (2006) 2721-2744 Submitted 3/06; Revised 9/06; Published 12/06.

[14].Ang Cui, Michael Costello and Salvatore J. Stolfo" When Firmware Modi cations Attack: A Case Study of Embedded Exploitation". Department of Computer Science Columbia University New York, US {ang, costello, sal}@cs.columbia.edu.

[15].E.Gandotra et al," Detecting and Classifying Morphed Malwares: A Survey" International Journal of Computer Applications (0975 – 8887) Volume 122 – No.10, July 2015.

[16].Myung-gu Kangv et al," USBWall: A Novel Security Mechanism to Protect Against Maliciously Reprogrammed USB Devices" 201