Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 5, 1090-1100
2024
Publisher: Learning Gate
DOI: 10.55214/25768484.v8i5.1809
© 2024 by the author; licensee Learning Gate

Forensic analysis of bad USB attacks: A methodology for detecting and mitigating malicious USB device activities

Bandr Siraj Fakiha
Department of Medical Health Services, Faculty of Health Sciences, Umm Al-Qura University, Saudi Arabia;
bsfakiha@uqu.edu.sa (B. S.F.).

Abstract: BadUSB is one of the most dangerous cybersecurity threats, given that it uses the firmware of USB devices to perform various undetectable actions with numerous tools. This research aims to evaluate the efficiency of different forensic approaches, such as signature-based detection, behavioral analysis, and the machine learning (ML) approach, in detecting and analyzing BadUSB attacks. Experiments were conducted with preconfigured USB peripherals to perform keystroke injection, data exfiltration, malware delivery, and network traffic manipulation. The analysis shows that the behavioral analysis and the ML-based methods show high detection accuracy and low false positives. Machine learning detection is the most efficient method. Behavioral analysis had higher accuracy in detecting abnormal device behavior but had a longer detection time than the ML methods. This research beneficently addresses the issues and challenges in the field of digital forensics and calls for further improvement in the detection methods. It proposes ways to implement these methods within the existing cybersecurity models. Future studies should focus on the best approaches to fine-tune these techniques, diversify datasets for machine learning detection methods, and advance methodologies in forensics to accommodate new generations of technologies like the Internet of Things and cloud systems.

Keywords: *BadUSB attacks, Behavioral analysis, Digital forensics, Cybersecurity, Machine learning.*

1. Introduction

1.1. Overview of BadUSB Attacks in the Modern Cyber Environment

BadUSB attacks are complex and typically dangerous kinds of cyberattacks that take advantage of the trust people put in USB devices like cables and chargers. As opposed to typical malware, which exists inside files or applications, BadUSB focuses on the device's firmware [1]. The firmware is the basic code that determines the operations of the USB device within the host and has become vulnerable to hacker exploitation since the advent of this hacking technique [2]. By altering this firmware, an attacker gets the USB device to act in a given way, potentially keylogging, hijacking browser connections, injecting code, or opening a backdoor into a system. Such actions are performed covertly and are very hard to distinguish from legitimate operations using conventional anti-viral tools since the infected files are never stored on the device in the first place.

For this reason, a look into the contemporary threat posed by BadUSB in today's advanced world of cybercrime holds great potential to improve security, considering that Universal Serial Bus (USB) devices are some of the most common sharing tools in the modern world [3]. Many types of peripherals, including USB flash drives, keyboards, external hard drives, and others, are utilized within nearly every business and organization. This makes them an ideal gateway for attackers to penetrate critical systems and go around established security measures normally meant to protect against network or software-related attacks [4]. BadUSB attacks are also especially devastating when

© 2024 by the author; licensee Learning Gate
* Correspondence: bsfakiha@uqu.edu.sa

delivered in air-gapped conditions, where physical USB devices are the only method of information exchange due to network isolation from the Internet [5]. Thus, attackers who have leveraged the BadUSB vulnerabilities can easily infiltrate highly confidential systems. **Figure 1** below shows a typical air-gapped condition.

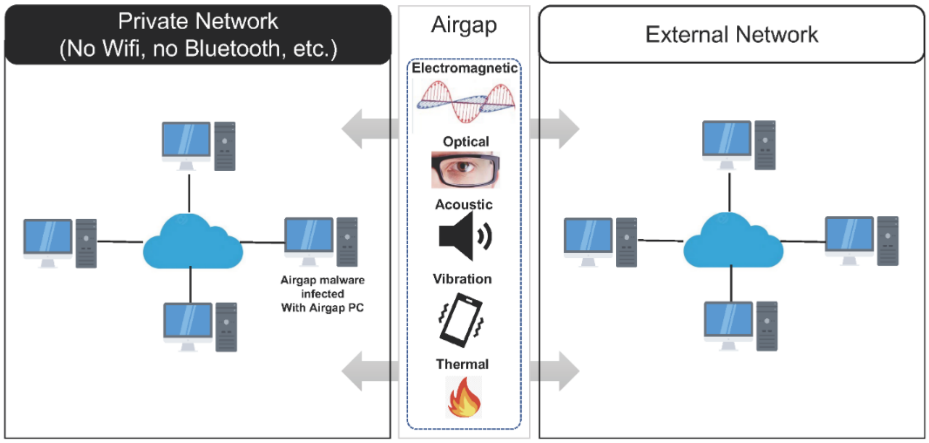

Figure 1.
A typical air-gapped network environment.

The Stuxnet worm is a real-world example of a USB-based attack not necessarily linked to BadUSB. This malware attacked the Iranian nuclear facilities through a few infected USB devices [6]. This 2010 computer worm was designed to take over specific programmable industrial control systems and cause malfunctions in the equipment running the nuclear systems, all while feeding false information to the systems monitors, indicating the equipment is operating as intended [7]. The incident showed that USB exploits have the capability to cause catastrophic consequences if leveraged by ill-intending hackers. The Stuxnet virus replicated as a USB drive and infected computers with malware. Typical BadUSB exploits, however, encompass targeted attacks on the device itself [8]. This makes detecting and preventing attacks such as BadUSB even more challenging, hence the need to conduct research to solve this issue.

1.2. Challenges Emerging from BadUSB Attacks

Typically, BadUSB attacks are difficult to prevent and mitigate by traditional security methods, and most digital forensic analysis approaches [9]. Legacy anti-malware and antivirus applications are centered on certain file patterns, malicious or virus code strings, or abhorrent execution of software programs [10]. BadUSB malware works at the firmware level and never uses the software or executable files detected by these tools [11]. It encompasses designed malware that looks like a normal USB device in the eyes of the user and the SECMAL software, but it secretly does dangerous executions.

Another challenge is that, in most systems, logging or tracking mechanisms for USB firmware activities are not implemented. Once an infected USB device is inserted into a machine, it can perform numerous malicious activities that are hard to detect on the system, especially if the attack looks like ordinary hardware like a keyboard or a network adapter. This is because digital forensics investigators work mainly on system logs, network traffic, or files to trace the source of an attack [12]. In the case of BadUSB attacks, these traditional forensic methods provide no results because the firmware changes made do not create easily identifiable traces. Therefore, there is no way or little proof of how the attack occurred or what was done maliciously if there are no efficient detection channels in place for forensic investigators to discover.

Moreover, since BadUSB attacks can change USB devices to mimic several types of equipment, like a keyboard injecting keystroke or a NIC redirecting traffic, such attacks pose issues that typical forensics utilities are ill-equipped to solve [13]. Its polyfunctional character makes BadUSB tricky to identify in the first place. One needs to use highly specific tools that are not universal for analysts and IT security practitioners.

1.2. Research gap

The absence of adequate forensic approaches to identifying and investigating BadUSB occurrences is a challenge in the current security systems. A lot of the current generation of forensic tools have been designed and work at the macro level and, therefore, cannot easily detect firmware changes in USB devices [14] because they are more suitable for tasks such as file creation time, modification time, and possibly many network activities. Also, there needs to be more automatable and trustworthy methods for distinguishing between the normal behavior of USB devices and malware-tampered firmware. There are works on identifying malicious USB activities, behavior analysis, or machine learning. However, these studies are limited, and most do not explore the possible solutions to this cyber threat. Also, there are few standard measures for assessing the effectiveness of forensic methods to detect BadUSB attacks. The lack of a solid, scientifically proven way to identify, evaluate, and prevent BadUSB often puts many organizations at severe risk of attack [15]. From the perspective of digital forensic investigators, this gap implies that tasks such as attributing an attack, identifying affected devices, or reconstructing an attack timeline and sequence are close to impossible using existing tools and methodologies [16]. This research seeks to fill this gap by developing and presenting a new forensic model that would be useful in identifying BadUSB attacks and the subsequent analysis of these attacks.

1.3. Objectives and hypothesis of the research

The main goal of this work is to outline and compare the forensic approaches for identifying and analyzing the BadUSB attack. This will be achieved by constructing several prototypes of the BadUSB attack and comparing the ability of the different forensic approaches, including signature detection, behavioral analysis, and ML, to identify the attacks. This research hypothesizes that conventional forensic approaches are not very effective in detecting and analyzing BadUSB attacks, but using behavioral analysis and machine learning improves the detection rate of the attack and its accuracy. Specifically, the research posits that machine learning models can improve detectability because it is possible to distinguish between malicious and non-malicious USB devices based on pattern recognition.

2. Literature Review

2.1. Review of existing literature

USB-based attacks have been identified for a long time as one of the threats in cybersecurity, with various research showing how USB media have been involved in compromising both individual and corporate networks. The BadUSB attack is one of the most dangerous evolutions in USB-based threats and was first brought to the public domain by two different security researchers, namely Karsten Nohl and Jakob Lell, in 2014 [17]. They presented possible exploits if USB devices are processed at the firmware level, and none of the existing security products could detect these attacks. This attack differs from previous threats, such as the USB-based viruses that inserted malicious files inside a USB drive. However, BadUSB works in the firmware plane, which means that it manages the actions of the device. Several researchers have subsequently looked into BadUSB and how these attacks are implemented on the technical front. Ray and Apala (2017), for instance, suggested that the actual problem with these attacks revolves around the trust-based infrastructural model that is seen as a part of the Universal Serial Bus (USB) [18]. Much like in the case of the pseudo-random number generator, USB devices are presumed to behave in a manner that is consistent with their function (e.g., a keyboard being a keyboard), and thus, it is almost impossible to differentiate between a genuine electronic device and a device with modified firmware.

DOI: 10.55214/25768484.v8i5.1809
© 2024 by the author; licensee Learning Gate

Several efforts have been made to find ways to contain the detection and prevention of BadUSB, but they encounter several challenges. As pointed out by Dobiasch et al. (2018), this is because BadUSB cannot be detected by the signature-based detection methods normally employed in antivirus software because these are not briefed in any executable code with a signature [19]. Tian et al. (2018) noted the greatest challenge is the similarity between appropriate and malicious USB peripherals [20]. For instance, a BadUSB attack might employ a USB device that imitates, for example, a keyboard and performs key logging, which is impossible to differentiate from actual keystrokes on a higher level. Another important issue is that there are currently limited possibilities of identifying or determining BadUSB attacks with the help of common forensic analysis tools. Almost all security products target detecting software-based threats without possessing the ability to inspect firmware-level changes of the BadUSB attack. Consequently, identifying attacks of the BadUSB type remains a challenge beyond traditional cybersecurity best practices.

2.2. Overview of Current Forensic Techniques

Analysis of USB devices has been one of the most significant applications of digital forensic investigations for quite some time. Traditional analysis is carried out by monitoring the USB device activities in system logs, registry keys, and other file system artifacts [21]. For instance, Windows OS keeps some records of those USB devices that have been connected to a user's machine, including things such as the device ID, serial number, and even the time the device was connected. This data may prove useful in such a case as it aids in reconstructing scenarios in an investigation, thus identifying the time and location of the use of a USB device. However, the landscape of forensic analysis is continuously changing due to the rising complexity of USB-based exploits, including BadUSB [22]. Existing approaches in digital investigations for USB instances may be sufficient in identifying routine USB interactions, but they lack the capacity to identify firmware changes.

The second approach includes the analysis of the actual behavior of USB devices connected to the system in real time (behavioral analysis). This method has been effective in detecting malicious activities by detecting behavior changes in the device. For instance, where a USB device sends many commands to the system in quick succession, as is typical with keystroke injection attacks, such a feature will be flagged as suspicious. The challenge in behavioral analysis is to differentiate between a security threat and a user's normal activity [23]. For example, it may be a person typing quickly using a USB keyboard and typing multiple keystrokes per second, which is not necessarily an attack. In the last defense technique, the use of machine learning methods in identifying USB-based attacks has also emerged recently. Such methods include designing algorithms trained and tested on huge volumes of data of normal and abnormal USB device activities. Mittal et al. (2021) explained in a study that machine learning can be employed to detect unauthorized activity on a USB device which, for instance, engages in operations that are not expected of it, like fundamentally changing system files in a USB drive [24]. However, as to the disadvantages, the application of machine learning is vulnerable to a high number of false positives since many legitimate actions USB devices can mimic an attack [25].

2.3. Limitations in Existing Methods

Most present-day forensic applications are centered on identifying the software communications between USB devices and the host, which include log files, registry entries, and file systems. However, attacks function on the firmware level, which is undetectable to the usual forensic tools. A firmware has no logs created by the operating system, so no modifications in the firmware can be logged, hence posing a major security threat. This research seeks to fill this gap by exploring new methods that can be used to identify firmware changes in USB devices. The problem with employing signature-based detection for countering BadUSB attacks is that it successfully works only as protection against traditional types of malwares. According to Vouteva et al. (2015), BadUSB attacks remain undetected because they do not create malware signatures or malicious files [26]. Therefore, this research will consider the following promising approaches that do not operate based on signature-based detection, behavioral analysis, and machine learning. Wang et al. (2021) also designate that current machine-learning models need big datasets to learn the distinctions between normal and malicious device activities [27]. However, due to the high versatility of legitimate USB device activities, it is challenging to build reliable training sets. This research will add to such models by garnering better information about USB behaviors through enhanced data gathering and analysis of the aforementioned BadUSB behaviors at the firmware level.

3. Methodology

3.1. Experimental Design

I contacted Trend Micro Inc., an American-Japanese cyber security software company, to determine the efficiency of the different forensic approaches for identifying and analyzing BadUSB. With the help of this company's cybersecurity specialists, I conducted an experiment at the cybersecurity lab. The experiment was designed in a laboratory setup created to depict an actual situation where USB devices can be used for malicious purposes. The experiment was conducted to compare the effectiveness of various forensic approaches to detect modified USB devices that may perform malicious activities.

3.1.1. Hardware Setup

Three commercially available desktops with generic configurations were used as target systems. To mimic different conditions, some of these computers had different operating systems installed on them: Windows 10 and Linux Ubuntu 20. 04, macOS Monterey. This variation would compare the effectiveness of various forensic tools depending on the OS they were used on. Additionally, ten USB devices were selected for the experiment; these are: USB flash drive, USB keyboard, and USB network adapter, among others. These devices were infected with BadUSB firmware to mimic the attacks. Every USB device contained different types of malicious firmware, which was programmed to do a particular action, such as keystroke injection, data theft, and malware distribution. Other recommended forensic hardware used were the USB write blockers to ensure that the modified USB equipment does not modify the information in the system when connected for forensic analysis and the external hard drives for storage of the forensic data.

3.1.2. Software Tools

Forensic analysis- Tools like Autopsy and FTK Imager were employed to conduct forensic analysis.

Data logging—For these purposes, the open-source packet capturing tool USBPcap was employed to intercept all USB inter(pair)face communication with the target computers. This tool recorded the USB device descriptors, firmware, and command sequences executed.

Behavioral monitoring tools—Wireshark was used to monitor and study the behavioral patterns of USB devices, capturing the traffic that the USB network adapters generated on the network.

3.1.3. Preparation of USB Devices with BadUSB Firmware

In this experiment, all the USB devices were preprogrammed by installing the BadUSB firmware tools, the USB Rubber Ducky. The USB devices were coded in the analysis to execute several nefarious activities for every attack scenario. The preparation steps involved:

Keystroke injection—The keyboards were physically changed to function like human interface devices. This enabled them to execute scripted keystrokes as soon as the USB was connected to the specific stations.

Data exfiltration—Malware contained in USB flash drives was configured to copy files from the target PCs to a given host server. This attack was designed to mimic a data exfiltration scenario in which files were taken without the user's permission.

Malware delivery—A selection of manipulated USB devices was presented to execute harmful programs. When inserted, these devices looked like ordinary adapters but installed dangerous malware into their system. This included ransomware and remote access trojans, better known as RATs.

Network traffic manipulation—Other exploits involved tampering with the USB network adapters to reroute traffic through the bad proxies. This mimicked an attack whereby an attacker can intercept messages and modify them to launch man-in-the-middle (MitM) attacks.

3.2. Procedure

The 4-step procedure shown in Figure 2 below presents the simulated attack scenarios performed on the target systems and the data collection process aimed at assessing the analyzed forensic methods.

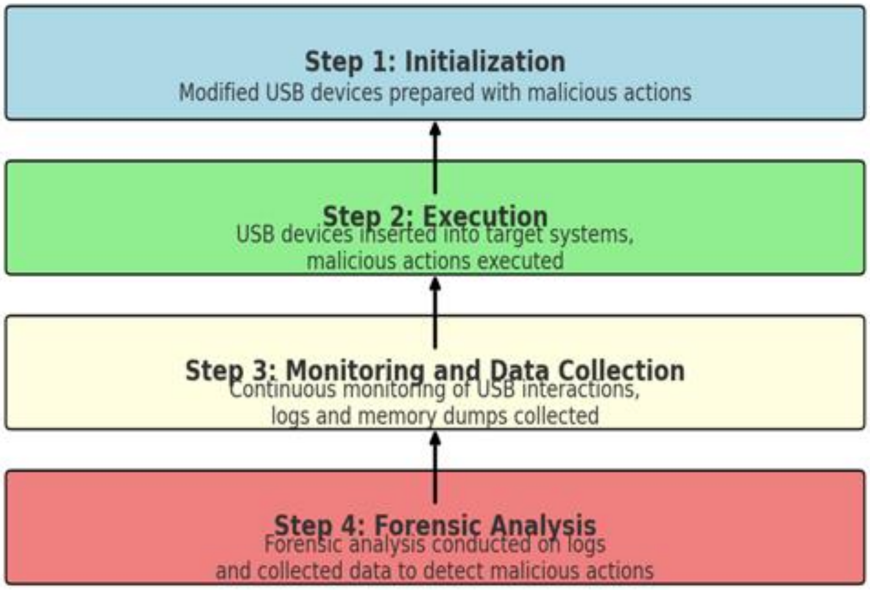
Step 1: Initialization
Modified USB devices prepared with malicious actions
Step 2: Execution
USB devices inserted into target systems,
malicious actions executed
Step 3: Monitoring and Data Collection
Continuous monitoring of USB interactions,
logs and memory dumps collected
Step 4: Forensic Analysis
Forensic analysis conducted on logs
and collected data to detect malicious actions

Figure 2.
The data collection process aimed at assessing the analyzed forensic methods.

The experiment commenced with the first step (Initialization), where USB Rubber Ducky was introduced with virulent firmware to mimic some attacks like keystroke injection attacks, data leakage attacks, malware delivery attacks, and network traffic attacks. In the second step (Execution), these modified USB devices are connected to the Windows, Linux, and macOS target systems to execute the preprogrammed attacks. The third step (monitoring and data collection) was performed by logging all USB interactions, which included communication capture with the help of USBPcap, system event monitoring with the help of Sysmon, and network activity monitoring with the help of Wireshark. This information includes memory dumps and USB activity logs obtained during the attacks. In the fourth step (forensic analysis), the copied logs and memory dumps were analyzed using Autopsy and FTK Imager for any signs of malicious activity, particularly any abnormality in the devices' behavior implicated in the BadUSB attack.

3.3. Evaluation Metrics

The experiment findings were recorded in a table (Table 1) to assess the efficiency of forensic methods. A second table (Table 2) was also filled with data regarding the number of false positives and average detection time for each forensic method. This information will compare the speed and accuracy of the mentioned techniques. The efficiency of these techniques was based on several key criteria: detection rate, false positives rate, and time of detection. The positive detection rate evaluated the effectiveness of every approach in properly identifying compromised USB peripherals and firmware updates. False positives were determined by how every approach identified the normal operation of the USB device and then flagged it as malware. Lastly, the detection time, which implies the time taken by different methods to detect malicious activities besides analyzing them in case of real-time threat intelligence, offered an effective evaluation of the reliability of the given method in time-bound situations.

4. Results

4.1. Experimental Findings

Findings from the experiment were recorded in Table 1 below, which assesses the efficiency of forensic methods by showing the success rate of each detection method for different attack types.

Table 1.
The success rate of different forensic methods

Type of BadUSB attack	Signature-based detection (Success rate)	Behavioral analysis method success rate	The ML-based method success rate
Keystroke injection	80%	92%	95%
Data Exfiltration	70%	87%	92%
Malware delivery	80%	88%	89%
Network traffic manipulation	80%	90%	94%
Unauthorized execution	70%	85%	90%

According to Table 1 findings, signature-based detection was relatively effective, with detection rates that fell within 70% to 80 % per endpoint. It was most effective against keystroke injection, malware delivery, and network traffic manipulation cases, which was 80% successful. However, it was less effective in identifying data exfiltration percentage, which is 70%, and unauthorized execution, which was also 70%, thus pointing to the weaknesses associated with signature detection for firmware-level attacks.

The detection rates with BA and ML-based methods were much higher than the results of the previous method. Behavioral analysis proved to be successful for all types of attacks, with an overall success rate of 85 - 92 % and much higher for keystroke injection (92%) and network traffic manipulation attacks (90%). The success rates ranged between 89% and 95%, with the highest score in keystroke injection (95%) and data exfiltration (92%) detection, where ML provided the best results in all the examined scenarios. These results indicate that though signature-based detection techniques can give some level of protection, other advanced methods, such as behavioral analysis and ML-based approaches, have higher precision in detecting BadUSB attacks. The ML-based method is clearly the most effective in handling sophisticated or stealthy actions like unauthorized execution and data theft. The graph in **Figure 3** below visually represents the average success rate of the three detection methods. ML-based detection method showed the highest average success rate of 92%. BA indicated an 88.4% success rate, while signature-based detection indicated an average of 77%.

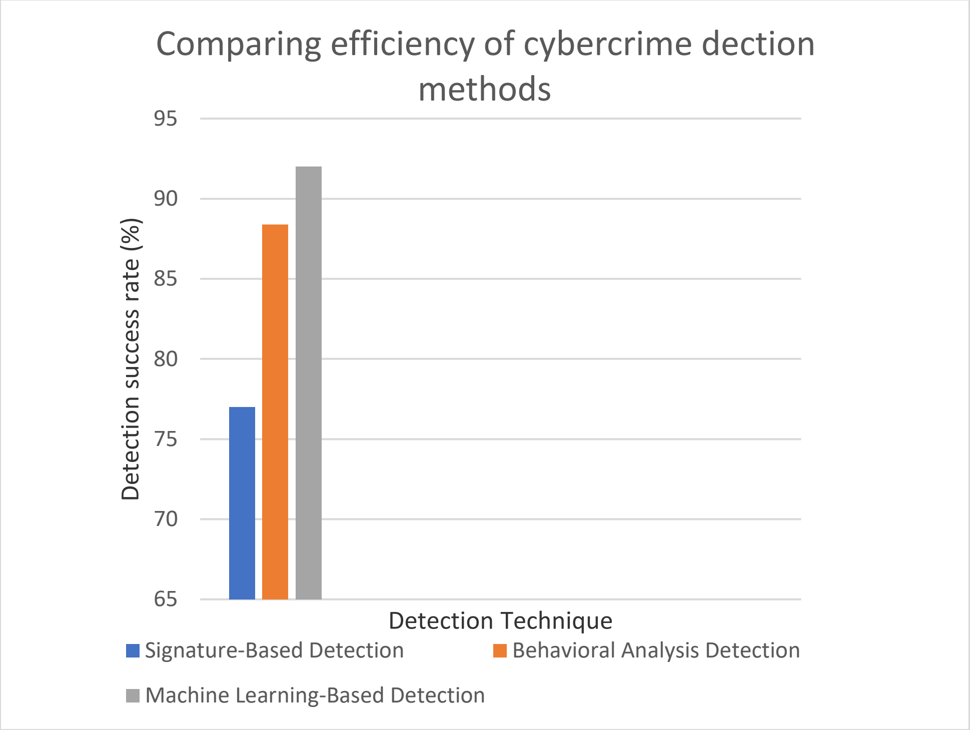

Figure 3.
A comparative bar graph showing the efficiency of different detection methods.

Table 2's Findings highlight the number of false positives and the average time to detect each attack.

Table 2.
Number of false positives and the average time taken to detect an attack.

Forensic technique	False positives	Average detection time
Signature-based method	6	13
Behavioral analysis	4	20
ML-based detection	2	18

The values highlighted in the table showed significant disparities in the performance of the forensic techniques in terms of false positives and total detection time. Signature-based detection created the highest number of false alarms, while on average, it took 13 seconds to detect an abnormality, making it relatively quick but inaccurate. Though generating 4 false positives, the behavioral analysis took the longest time to detect malicious behavior, an average of 20 seconds, indicating the tradeoff between accuracy and time complexity of monitoring device behavior. The study results show that the ML-based detection had the optimal, or the lowest, false positive rate of 2 and a relatively fast average time to detect the attack, around 18 seconds, which makes this detection technique both accurate and effective compared to the other methods.

5. Discussion

5.1. Interpretation of the Results

As depicted in the experiment results above, the signature-based detection approach is inadequate in identifying BadUSB attacks. Thus, despite achieving an overall detection rate of 70-80% across attack types and methods, signature-based techniques miss firmware-level modifications, which previous studies like Shafique et al. have shown to be ineffective when detecting BadUSB-type threats.

© 2024 by the author; licensee Learning Gate

As expected, the behavioral analysis and ML methods showed better results and were up to 95% successful in some attacks. Comparing the results of false positive rates and detecting accuracy of conventional methods and the proposed ones based on ML methods indicates that the ML-based methods have a higher potential with a detection rate of 89-95%. Indeed, these findings highlight the importance of transitioning the current cybersecurity approaches to much more advanced and ML-based ones to detect more complex USB threats.

5.2. Implications and Recommendations

According to the studies, cybersecurity personnel and the concerned organizations must embrace sophisticated modes of investigation to counter the new trends of USB attacks. Given the proliferation of USB devices in various industry sectors and the advancement of attack methods, organizations should enhance detection systems with ML-enabled systems. For instance, the data gained from behavioral analysis and machine learning-based methods, together with the low false positive rate (4 and 2 correspondingly) indicate the direction for improvement of cybersecurity teams' work regarding incident detection without high false alarm rates. As such, machine learning models can be trained with various datasets of USB behaviors designed to detect abnormalities at the firmware level. This can be done via endpoint detection and response systems such as Trend Micro or Crowdstrike, which contain machine learning-enabled threat identification. Using all these techniques, along with the signature-based ones, will form a layered, multi-faceted defense against the USB-based threats.

5.3. Potential limitations and areas for further research

This study employed the experimentation method as the primary data collection approach. Like any other scientific study, it might have potential limitations that could have introduced bias to the findings above. First, the simulated attack scenarios used were realistic and depicted a real-world scenario. However, they may not have encompassed all the possible variations of the real-world systems. Also, as described in the experiment, the target systems are restricted to certain operation systems, namely Windows, Linux, and Mac OS, while the hardware configurations remain constant throughout all the trials. This also brings out bias in the outcome since numerous variations are available in real-life networks. Further, pre-selection of the malware and attack types may have affected the detection performance of the forensic techniques in the experiment. In real-world circumstances, the adversaries can employ other more complex or different procedures that were not simulated in this study, and, therefore, the validity of the findings may only apply to some situations.

Another limitation is that the USB devices used in the experiment were modified with the use of an available freely opened tool, the USB Rubber Ducky. This tool may not capture the likeness of the malware that may be designed uniquely and particularly by a professional cybercriminal. Also, the machine learning models employed for analysis in the study had been trained with a small data set, and they may not work in more general or other conditions. This study's limitations should guide future studies, involving bigger datasets and more diverse samples to enhance the applicability of machine learning-based forensic approaches.

6. Conclusion

This study's findings show that signature-based detection techniques are limited when it comes to detecting more advanced forms of USB-based threats such as BadUSB. Pattern-matching approaches, which involve the detection of known malware, fell short of detecting firmware level changes characteristic of BadUSB. On the other hand, behavioral analysis and ML-based approaches were identified to have more potential to detect these new and advanced threats with greater accuracy by analyzing the abnormal behavior of a device and by learning from the patterns. These observations agree with the recent realization that more sophisticated intrusion methods are being used to access systems, making the traditional forensic strategies inadequate for present-day use in forensic analyses, as Neuner and Sebastian suggest [29].

To the best of the authors' knowledge, this research advances the knowledge of Digital Forensic research by establishing that it is possible to identify BadUSB attacks using advanced forensic analysis like Machine Learning detection. The results extend Rabbani [30] and Vanjire's works [31], which embrace several machine learning and behavior-based detections toward eradicating firmware-level threats. In this way, the results of this research support the need for using more advanced techniques in cyber forensics that are well-equipped to handle the advanced forms of attacks. Therefore, the findings of this study provide clear guidance to cybersecurity professionals and organizations on how to improve the existing defenses against USB-based threats.

It is crucial to note that more research is needed to improve and fine-tune the techniques highlighted in this study. Another area that needs enhancement is the creation of broader and more diverse datasets to train machine learning because, presently, such machine learning models are inadequate in handling different types of attacks. Besides, future researchers need to look for ways of improving these techniques to enhance accuracy and make them applicable to various environments and different types of attacks.

Copyright:

© 2024 by the authors. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

References

[1] Shafique, U., & Zahur, S. B. (2020). Towards Protection Against a USB Device Whose Firmware Has Been Compromised or Turned as 'BadUSB.' In Advances in Information and Communication: Proceedings of the 2019 Future of Information and Communication Conference (FICC), Volume 2 (pp. 975-987). Springer International Publishing.

[2] Feng, X., Zhu, X., Han, Q. L., Zhou, W., Wen, S., & Xiang, Y. (2022). Detecting vulnerability on IoT device firmware: A survey. IEEE/CAA Journal of Automatica Sinica, 10(1), 25-41.

[3] West, R., Golchin, A., & Navarro, A. (2023). Real-Time USB Networking and Device I/O. ACM Transactions on Embedded Computing Systems, 22(4), 1-38.

[4] Erdin, E., Aksu, H., Uluagac, S., Vai, M., & Akkaya, K. (2018, October). OS independent and hardware-assisted insider threat detection and prevention framework. In MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM) (pp. 926-932). IEEE.

[5] Benadjila, R., Renard, M., Trebuchet, P., Thierry, P., Michelizza, A., & Lefaure, J. (2018). Wookey: USB devices strike back. Proceedings of SSTIC.

[6] Datta, P. M., & Acton, T. (2023). Did a USB drive disrupt a nuclear program? A Defense in Depth (DiD) teaching case. Journal of Information Technology Teaching Cases, 20438869231200284.

[7] Nissim, N., Yahalom, R., & Elovici, Y. (2017). USB-based attacks. Computers & Security, 70, 675-688.

[8] Griscioli, F., Pizzonia, M., & Sacchetti, M. (2016, December). USBCheckIn: Preventing BadUSB attacks by forcing human-device interaction. In 2016 14th Annual Conference on Privacy, Security and Trust (PST) (pp. 493-496). IEEE.

[9] Lu, H., Wu, Y., Li, S., Lin, Y., Zhang, C., & Zhang, F. (2021, May). Badusb-c: Revisiting BadUSB with type-c. In 2021 IEEE Security and Privacy Workshops (SPW) (pp. 327-338). IEEE.

[10] Sumitra, S. (2022). An Analysis of Cybersecurity for Business Enterprises.

[11] Hernandez, G., Fowze, F., Tian, D., Yavuz, T., & Butler, K. R. (2017, October). Firmusb: Vetting USB device firmware using domain-informed symbolic execution. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (pp. 2245-2262).

[12] Yaacoub, J. P. A., Noura, H. N., Salman, O., & Chehab, A. (2021). Digital forensics vs. Anti-digital forensics: Techniques, limitations and recommendations. arXiv preprint arXiv:2103.17028.

[13] Mueller, T., Zimmer, E., & de Nittis, L. (2019, August). Using context and provenance to defend against USB-borne attacks. In Proceedings of the 14th International Conference on Availability, Reliability and Security (pp. 1-9).

[14] Villalba, L. J. G., Orozco, A. L. S., Vivar, A. L., Vega, E. A. A., & Kim, T. H. (2018). Ransomware automatic data acquisition tool. IEEE Access, 6, 55043-55052.

[15] Dobiasch, N. (2019). Identification and analysis of forensic artifacts of file-less malware abusing Power Shell.

[16] Casino, F., Dasaklis, T. K., Spathoulas, G. P., Anagnostopoulos, M., Ghosal, A., Borocz, I., ... & Patsakis, C. (2022). Research trends, challenges, and emerging topics in digital forensics: A review of reviews. IEEE Access, 10, 25464-25493.

[17] Blanchet, S. (2018). BadUSB is the threat hidden in ordinary objects. Technical report, Bertin Technologies.

[18] Ray, A. (2017). On Pre-deployment Assessment and Security Bootstrapping of Industrial Communication Networks (Doctoral dissertation, Mälardalens högskola).

[19] Dobiasch, N. (2019). Identification and analysis of forensic artifacts of file-less malware abusing Power Shell.

[20] Tian, D. J., Bates, A., & Butler, K. (2015, December). Defending against malicious USB firmware with GoodUSB. In Proceedings of the 31st Annual Computer Security Applications Conference (pp. 261-270).

[21] Bajahzar, M., & Mishra, S. (2023). Cloud Forensic Artifacts: Digital Forensics Registry Artifacts discovered from Cloud Storage Application. International Journal of Computing and Digital Systems, 14(1), 1-xx.

[22] Rodriguez, M. (2023). Exploring the Landscape of Operating System Forensics: An In-Depth Evaluation. International Journal of Creative Research in Computer Technology and Design, 5(5).

[23] Öğütçü, G., Testik, Ö. M., & Chouseinoglou, O. (2016). Analysis of personal information security behavior and awareness. Computers & Security, 56, 83-93.

[24] Mittal, A., & Garg, U. (2022, October). A review for insider threats detection using machine learning. In AIP Conference Proceedings (Vol. 2555, No. 1). AIP Publishing.

[25] Chakraborty, S., Krishna, R., Ding, Y., & Ray, B. (2021). Deep learning-based vulnerability detection: Are we there yet? IEEE Transactions on Software Engineering, 48(9), 3280-3296.

[26] Vouteva, S., Verbij, R., & Roos, J. (2015). Feasibility and deployment of bad USB. University of Amsterdam, System and Network Engineering Master Research Project.

[27] Wang, Z., Fok, K. W., & Thing, V. L. (2022). Machine learning for encrypted malicious traffic detection: Approaches, datasets, and comparative study. Computers & Security, 113, 102542.

[28] Shafique, U., & Zahur, S. B. (2020). Towards Protection Against a USB Device Whose Firmware Has Been Compromised or Turned as 'BadUSB.' In Advances in Information and Communication: Proceedings of the 2019 Future of Information and Communication Conference (FICC), Volume 2 (pp. 975-987). Springer International Publishing.

[29] Neuner, S. (2017). Bad things happen through USB (Doctoral dissertation, Wien).

[30] Rabbani, M., Wang, Y., Khoshkangini, R., Jelodar, H., Zhao, R., Bagheri Baba Ahmadi, S., & Ayobi, S. (2021). A review on machine learning approaches for network malicious behavior detection in emerging technologies. Entropy, 23(5), 529.

[31] Vanjire, S., & Lakshmi, M. (2021, September). Behavior-based malware detection system approach for mobile security using machine learning. In 2021 International Conference on Artificial Intelligence and Machine Vision (AIMV) (pp. 1-4). IEEE.

DOI: 10.55214/25768484.v8i5.1809
© 2024 by the author; licensee Learning Gate