Embedded Malware Techniques for PoS

Micah Flack
Beacom College of Computer and Cyber Sciences
Dakota State University (DSU)
Madison, South Dakota
micah.flack@trojans.dsu.edu

***Abstract*—This is a literature review for embedded malware techniques and methods for cashier systems. Platform/operating-system attacks are reviewed as well as those made specifically for POS systems. The goal is to review current methods and speculate future research.**

Keywords—malware; cashier systems; point of sale; pos; vulnerabilities; human input devices; hid; usb; serial devices.

I. Introduction

This paper will discuss a literature review for embedded malware techniques and methods used against cashier systems or point-of-sale (POS) systems. A lot of research has been done in this field over the years, however, it is not always disseminated as POS research. Instead, it will be researched as linux malware or embedded systems vulnerability research. In some cases, because there is a shift from dedicated POS systems running embedded linux towards Android based systems, the platforms targeted and the relevant attacks changes as well. The goal is to compare current research to see what types of deployment is being used by the attackers (e.g., hardware or software), specifically, whether they are cloning serial devices or other input devices used by the sales system.

II. Methodology

The methods used for finding papers and then identifying their relevance and currency was straightforward. All papers used were limited to IEEE X-PLORE [1], Google Scholar [2], arXiv [3], and PapersWithCode [4]. Typically, during this process the use of Google Scholar is limited because it tends to pull search results from sites that only publish whitepapers. Whether or not these papers have been formally reviewed, submitted to a conference, or a journal can be difficult to verify. So, the use of Google Scholar was limited to topical research and not for formally gathering current papers.

PapersWithCode is a great resource for finding current papers with GitHub repositories. The papers tend to be more concise and focus on the specifics of how their tool or techniques function. Although, the researchable topics are limited.

Another resource typically used in this process is Malpedia [5] by Fraunhofer FKIE; this resource, however, was not used for this paper. When looking for technical analyses for a specific strain or type of malware family, this is the best resource. All of the articles are tagged by their attributable family names (e.g, APT-XX, Sandworm, Mirai, etc…). The articles are also listed chronologically which makes it easier to see how analysts came to different conclusions about the function of the malware as time passes and new data arrives.

Once a suitable batch of papers have been identified, the researcher reviews the abstract. If there aren't too many grammatical errors or incorrect statements/assumptions made, the research will then review the introduction and methodology. Here it is easier to see the maturity of the research as well as identify how novel the contributions claimed are. At this stage it can also be useful to take note of where researchers made mistakes that led to inaccuracies later-on. It might even suggest possible future research because of the limited scope.

With the final papers chosen for the literary review, the final details about each paper are collected. The required details are as follows (but not limited to):

- The number of papers that cite it.
- The goals and contributions claimed.
- Any flaws and/or weaknesses.
- Overall remarks and impression of the research.

The researcher will then review each of the papers and prepare a neutral abstract. Additionally, an abstract is also prepared by an artificial intelligence (AI) assistant, Claude 2 from Anthropic. [6] talks about the capabilities and limitations of LLMs like ChatGPT. The inputs provided by the AI are used purely for demonstrative purposes and to assist the researcher in recognizing under-represented information.

III. Reviews

The following papers have either made novel contributions that expanded the research topic, provide background information about current state of the research, or they are contributions made towards other topics that can be co-opted towards this one.

A. Mobile payment security, threats, and challenges [7]

This paper has been cited 42 times within IEEE Xplore, and Google Scholar reports the number of citations as 135. The paper demonstrates that the number of pages does not indicate success since it was published in the 2016 Second International Conference on Mobile and Secure Services (MobiSecServ). Section five could have used more data and citations for each of the subheadings. The content is concise,

but it has enough information to inform any reader of the current state of mobile payment security.

1) Individual Synopsis:

The paper discusses the desired security services in a mobile payment system, such as authentication, confidentiality, integrity, nonrepudiation, and availability. A mobile payment processing model is provided to describe how transactions are processed across different systems. The focus is more for smartphone payment systems like PayPal, Alipay, WeChat Wallet, and Samsung Pay rather than the hardware based systems this paper is interested in. However, they do discuss some of the attacks that EMV chip cards face when using a system like Stripe or Square Register. They also examine the security mechanisms that are currently used in mobile payment systems, such as fingerprint, username/password, multi-factor authentication, SSL/TLS, and secure element.

The researchers go on to identify and analyze the main threats to mobile payment security, such as attacks against the above security mechanisms. And suggests some security measures for mobile payment users and service providers to mitigate the risks. Some highlights of the security challenges that mobile payment systems face are malware detection, multi-factor authentication, data breach prevention, and fraud detection and prevention.

Although this paper is not as technically dense as the others, nor does it describe a novel attack method, it is helpful for establishing an understanding of current payment system security. Arguably, the paper lacks technical depth when discussing how attacks are initiated or how data is exfiltrated. Anyone seeking to thoroughly understand each of the topics shown in the paper would need to review their references.

2) Artificial Intelligence Synopsis:

The paper provides an overview of mobile payment systems, their security mechanisms, threats, and challenges.It introduces five types of mobile payment systems: payment at POS, payment as POS, payment platforms, independent systems, and direct carrier billing. Desired security services for mobile payment include authentication, access control, confidentiality, integrity, non-repudiation, and availability.

Current security mechanisms include fingerprint authentication, passwords, multi-factor authentication, SSL/TLS encryption, and secure elements. However, mobile payment faces threats like malware stealing credentials, SSL/TLS vulnerabilities exposing data, and data breaches compromising sensitive information.

Major security challenges highlighted are malware detection, multi-factor authentication when devices are lost/stolen, preventing data breaches, and fraud detection/prevention. To mitigate risks, the paper advises users to take precautions like strong passwords, caution with emails/SMS, and avoiding untrusted WiFi. Service providers must validate certificates, protect payment apps, and prevent backend data breaches. Overall, the paper provides a comprehensive overview of the landscape, issues, and recommendations surrounding mobile payment security.

B. Understanding Linux Malware [8]

The paper has 82 citations within IEEE Xplore and Google Scholar shows 213 citations. The contents themselves are not necessarily novel, however, it does provide a contemporary survey of the methods used, types of platforms, and other generally informative data. The biggest shortcoming of the paper is that the researchers made no mention of embedded linux systems such as Yocto or scheduler based operating-systems such as RTOS/FreeRTOS.

1) Individual Synopsis:

The authors claim to present the first comprehensive study of Linux malware, which targets embedded devices and IoT systems; despite neglecting other heavily used systems such as Yocto and RTOS. There were several challenges and techniques involved in collecting, analyzing, and understanding the Linux malware samples used for their research.

They also discussed the difficulties of dealing with Linux malware, such as the diversity of target environments, architectures, libraries, and operating systems, the prevalence of static linking, the lack of previous studies, and the need to support different privilege levels depending on the type of environment used for analysis.

The analysis pipeline describes a process which consists of file and metadata analysis, static analysis, dynamic analysis, and data collection. They also detail the tools and methods they used to overcome the challenges and extract useful information from their malware corpus.

2) Artificial Intelligence Synopsis:

The AI assistant failed to provide input on this paper, throwing the error: “failed to fetch”.

C. WIGHT: Wired Ghost Touch Attack on Capacitive Touchscreens [9]

The paper has zero citations on IEEE Xplore, but Google Scholar shows three from arXiv, ACM, and Oxford Archive. The paper is relatively young and cybersecurity is a niche field, so it’s expected that there are not many citations yet. The authors claimed they could send manipulated touch inputs via an Android charging cable, regardless of USB data-blocker. Their attacks averaged 85.9% to 100% depending on the device used. This type of attack is different from the other cited papers in this review, however, it is novel and implicates any other device using capacitive touch screens. The authors mention that the method is limited by the construction of the screen. Meaning, at most the injected ghost touches will appear along a random vertical or horizontal line of the screen.

1) Individual Synopsis:

In this paper the researchers proposed a method for injecting ghost touches into Android devices using a capacitive touch screen. They claim that it is the first paper to present this type of wired attack with several vectors besides touch interference. The methods used are very thoroughly documented with many figures that aid unfamiliar researchers.

They demonstrated many capabilities ranging from auto allowing malicious bluetooth connections to three types of touch attacks: injection attacks that create ghost touches without users touching the screen, alteration attacks that change the detected legitimate touch position, and denial-of-service (DoS) attacks that prevent the device from identifying legitimate touches. All of these had varying degrees of success, with one attack succeeding 89.5% to 100% of the time depending on the device used.

In conclusion, their evaluation found that 6 smartphones, 1 tablet, 2 touch screen panels, 6 power adapters, and 13 charging cables were capable of demonstrating these attacks. Each of the phones and tablets used were all commonly found as well. However, the overall usability of the attacks were limited in nature by the composition of the touchscreens.

2) Artificial Intelligence Synopsis:

The paper presents WIGHT, the first wired ghost touch attack against capacitive touchscreens exploiting the charging cable. The attack injects malicious signals into the ground (GND) line of the USB charging cable. These common-mode noises get converted into differential-mode noises inside the device due to asymmetric touchscreen circuits. The resulting noises interfere with the capacitance measurement and cause ghost touches.

WIGHT achieves three types of attacks - injection, alteration, and denial of service. Injection attacks create ghost touches when the screen is idle. Alteration attacks change the position of touches when the user taps the screen. Denial of service attacks disable touch sensing. The attacks are effective even across power adapters.

Experiments on 9 devices show WIGHT can inject ghost touches along chosen TX lines with 93% success rate. Alteration attacks achieve 66% success in changing touch positions. Denial of service succeeds in disabling touch interactions. WIGHT is fast with ghost touches appearing within 0.5-1 second. The attacks pose a real threat with the ability to surreptitiously approve Bluetooth connections or download malware. Proposed mitigations include common-mode chokes, touch characteristic detection, and authenticated charging stations.

In summary, the paper demonstrates a novel threat against touch screens by maliciously exploiting the USB charging cable. The attacks highlight vulnerabilities in seemingly innocuous charging ports that can be leveraged by attackers in public charging stations. WIGHT calls for rethinking the security of touch interfaces and USB trust boundaries.

D. Design of printer data senseless acquisition device [10]

This paper has zero citations on IEEE Xplore as well as Google Scholar. Like the previous research, this paper was published in 2022 at the China Automation Congress (CAC). Anyone unfamiliar with the topic might dismiss the paper as not being novel, however, their proposed model has greater functionality than the average BadUSB attack. Their model would allow an attacker to mask their device bidirectionally. Controlling the inputs/outputs between the host and the peripheral device. This implementation is interesting because it is closer to a man-in-the-middle attack and allows the attacker greater control over the communications between the legitimate host and device.

1) Individual Synopsis:

The paper presents the design of a printer data acquisition device for traditional shopping malls. The device can be connected in series between the cash register and the printer, and can transparently transmit the print data to the upper-level network board for data network transmission or SD card temporary storage. The device can also adaptively simulate the cash register or the printer according to the connection mode, without affecting the printing process or any other interoperability.

The device uses STM32F103C8T6 and MT7628AN as the core chips of the data acquisition board and the network board, respectively. The authors describe the hardware design, software design, and experimental verification of the device developed, and they show that it can effectively collect any data transferred between the host cashier system and the printer. This is like a man-in-the-middle version of the BadUSB attack because you can control how both physical devices (the host and the keyboard) see each other. It would be interesting to see how this attack method might be used against smart keyboards or keyboards with programmable profiles to setup rgb lights and macros.

It would have been better if the authors provided a resource to see the configurations and scripts they used for further replication of the study. It would have also aided some of the claims for future work or to better compare against previous studies and existing tools. In the conclusion they also hypothesized future work that could be done to expand the research; such as data leakage revealed by their device during communication.

2) Artificial Intelligence Synopsis:

The paper presents the design of a printer data acquisition device to help shopping malls audit sales data reported by merchants. Traditional shopping malls are struggling due to the rise of e-commerce. Massive sales data generated by shopping malls could help reform them if properly acquired. However, independent cashier systems of merchants make unified data acquisition difficult.

The proposed solution is a plug-and-play device connected between the cash register and printer. It adaptively simulates the printer for the cash register and vice versa, enabling transparent data transmission without affecting printing. The hardware uses an STM32 MCU and MT7620 router chip. Software includes USB host/device modes, UART, and SPI communication. Experiments verified successful data acquisition and network transmission without impacting printing.

The device provides an effective way to collect merchant sales data. It is compatible with USB and serial printers. Future work includes optimizing parallel communication for higher throughput and addressing data security issues. Overall, the paper presents a novel printer data acquisition solution to

help shopping malls utilize sales data and optimize merchant types.

E. *Spyduino: Arduino as a HID Exploiting the BadUSB Vulnerability [11]*

Spydruino has 2 citations within IEEE Xplore and 11 citations on Google Scholar. It was presented in 2019 at the 15th International Conference on Distributed Computing in Sensor Systems (DCOSS). It provides a background of existing devices and a demonstration of how their solution is different. The Spydruino they developed is limited to physical connections, but the researchers suggested expanding connectivity with a GSM shield for greater coverage. The device is also limited in terms of what each attacker develops for it. Either the device is simply a more advanced keylogger, or it will be able to record audio inputs and outputs. It depends on what the attacker will have developed beforehand.

1) *Individual Synopsis:*

In this paper the researchers created a device using an Arduino as a human input device (HID) using the BadUSB vulnerability. The researchers claim that the device is able to gain access to sensitive data and then exfiltrate it outside of the network via FTP. The content of the research and the methods used are not novel, but their combination of existing methods and tools is.

They discussed the existing work in the field, such as a HAK5 Rubber Ducky, Malduino, or Evilduino. However, their Arduino differs because the device has greater interoperability and customization depending on the target. For instance, they highlighted using a GSM shield as future work to expand the research. Adding the device would allow for wireless communications with the target.

This paper has particular merit because it describes the serial device emulation process as well as how to create custom tooling like exfiltration or socket manipulation for keylogging and key injection. This can be leveraged for future development of HID devices that either capture leaked USB data or injection.

2) *Artificial Intelligence Synopsis:*

The paper introduces Spyduino, an Arduino microcontroller that has been reprogrammed to emulate a keyboard and exploit the BadUSB vulnerability. BadUSB refers to USB devices like keyboards that have had their firmware maliciously altered to enable attacks. By reprogramming an Arduino to act as a Human Interface Device (HID), Spyduino can infiltrate a computer system and appear as a normal keyboard while secretly executing pre-programmed malicious actions.

Once plugged in, Spyduino gains administrative privileges on the target computer. It then disables the firewall and connects to a pre-defined FTP server in the cloud to exfiltrate sensitive files and information. Spyduino uses a passive FTP connection to avoid detection by firewalls and network monitoring. The entire process from initial connection to data exfiltration happens very quickly, usually in under a minute.

A key advantage of Spyduino is that it is hidden inside a normal-looking keyboard enclosure, making it almost impossible to visually identify as malicious. Furthermore, since the attack operates at the firmware level, usual antivirus and malware detection methods are ineffective. The authors recommend physical security measures like port blocking as well as careful monitoring of device connections to mitigate BadUSB vulnerabilities like Spyduino.

In conclusion, Spyduino demonstrates how easily USB devices can be repurposed to mount virtually undetectable attacks on computer systems by exploiting the BadUSB vulnerability at a firmware level. The paper provides an overview of how a simple and inexpensive Arduino board can be transformed into a covert attack tool using this technique.

IV. Conclusion

Each of the papers provides crucial information about the development and state of attacks for PoS systems or Linux. There are also greater implications for techniques such as WIGHT [9] against other platforms than Android that use capacitive touch interfaces. The conclusions from [10] and [11] also suggest compatibility for future work by combining their designs to adhoc the data and functionality of the intended device. For example, could a researcher create a HID dummy and master device to manipulate data before it arrives at the intended peripheral device? Or, could the researcher modify the device firmware to expand its capabilities more than intended? In conclusion, [7] shows there is a continued interest in payment system attacks and [9]–[11] provide examples of current software/hardware attacks.

References

[1] "IEEE Xplore." https://ieeexplore.ieee.org/Xplore/home.jsp (accessed Sep. 07, 2023).

[2] "Google Scholar." https://scholar.google.com/ (accessed Sep. 07, 2023).

[3] "arXiv.org e-Print archive." https://arxiv.org/ (accessed Sep. 07, 2023).

[4] "Papers with Code - The latest in Machine Learning." https://paperswithcode.com/ (accessed Sep. 07, 2023).

[5] "Malpedia (Fraunhofer FKIE)." https://malpedia.caad.fkie.fraunhofer.de/ (accessed Sep. 07, 2023).

[6] M. N.-U.-R. Chowdhury and A. Haque, "ChatGPT: Its Applications and Limitations," in *2023 3rd International Conference on Intelligent Technologies (CONIT)*, Jun. 2023, pp. 1–7. doi: 10.1109/CONIT59222.2023.10205621.

[7] Y. Wang, C. Hahn, and K. Sutrave, "Mobile payment security, threats, and challenges," in *2016 Second International Conference on Mobile and Secure Services (MobiSecServ)*, Feb. 2016, pp. 1–5. doi: 10.1109/MOBISECSERV.2016.7440226.

[8] E. Cozzi, M. Graziano, Y. Fratantonio, and D. Balzarotti, "Understanding Linux Malware," in *2018 IEEE Symposium on Security and Privacy (SP)*, May 2018, pp. 161–175. doi: 10.1109/SP.2018.00054.

[9] Y. Jiang *et al.*, "WIGHT: Wired Ghost Touch Attack on Capacitive Touchscreens," in *2022 IEEE Symposium on Security and Privacy (SP)*, May 2022, pp. 984–1001. doi: 10.1109/SP46214.2022.9833740.

[10] C. Peng, P. Zhimin, L. Hongsheng, Z. Ming, and H. Kaifa, "Design of printer data senseless acquisition device," in *2022 China Automation Congress (CAC)*, Nov. 2022, pp. 793–796. doi: 10.1109/CAC57257.2022.10055971.

[11] E. Karystinos, A. Andreatos, and C. Douligeris, "Spyduino: Arduino as a HID Exploiting the BadUSB Vulnerability," in *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, May 2019, pp. 279–283. doi: 10.1109/DCOSS.2019.00066.