PRINTSHOP: ASSESSING OS AND CAPABILITIES OF SERIAL PRINT DEVICES

Research Proposal

Doctor of Philosophy

in

Cyber Operations

September 21, 2023

By

Micah Flack

Beacom College of Computer and Cyber Sciences

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Introduction

1.1 Overview

According to the Federal Trade Commission (FTC), there were 37,932 reports of credit card fraud in 2012 and 87,451 reports in 2022. This marks an increase of credit card payment fraud by an estimated, 30.5%. By comparison, since 2020, there has been a 14.6% increase in credit card related fraud. Which does not include the millions of other fraud reports the FTC receives every year. In 2022 alone, there were around 5.1 million fraud, identity theft, and miscellaneous reports in total [1], [2]. The statistics for these reports stresses how crucial the security of payment systems are, both physical and online. And, the need to secure them grows every year.

This research primarily focuses on physical point-of-sale (PoS) systems or terminals and their hardware (serial accessories), rather than online solutions. For instance, not mobile payment apps like Venmo, CashApp, Zelle, or Paypal [3]. There are many reasons, but the types of systems being targeted varies greatly in terms of the hardware and software supported, as well as, how the transactions are handled with the payment processor.

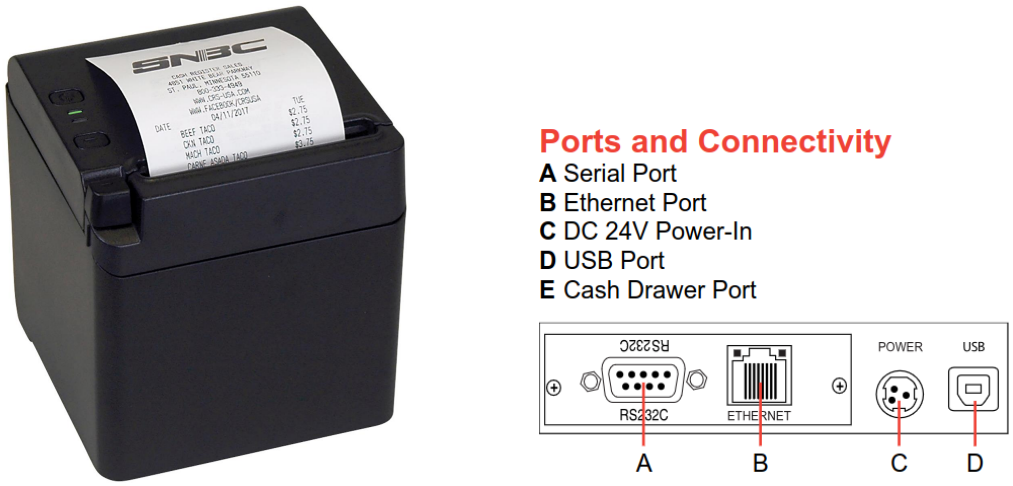
Figure 1.1 shows us two similar looking point-of-sale systems, albeit one is much older looking. However, the operating system and required hardware is very different. Typically, unless you have the Square provided terminal, their software/client is installed onto an Android or iOS device and connected to a Square compatible card reader [4]. Whereas, the SurePoS, NCR, or other common EFTPoS system will run a proprietary OS derived from Windows or Linux [5]. Furthermore, these PoS tend to require some form of printing receipts as record keeping for the business owner and customer. And these devices also vary in terms of processing capabilities and operating system.

(a) Square PoS (b) SurePoS

Figure 1.1: Comparison of common PoS systems

For instance, a common thermal printer seen with PoS systems, integrated with fuel pumps, or other industrial control equipment, is the SNBC BTP-S80 thermal printer [6], [7]. There are multiple versions of the device with support for Bluetooth, USB only, or combination of USB/Serial/Ethernet. The bluetooth hardware is provided over an accessory 25-pin serial connection, with more I/O as a serial connection via RS232C connector and USB Type-B. It has driver support for various platforms: Android, iOS, Windows, Linux, and MacOS. The most interesting aspects are the processor, an Arm Cortex M4 clocked at 3.54MHz, and the operating system, a proprietary version of FreeRTOS. The system architecture is Armv7E-M with JTAG/SWD hardware debugging support [8], [9].

By default, the printer has enough headroom to process ESC/POS commands for printing paper and a webserver for debugging or general diagnostics. In theory, the uncompromised device could be flashed with modified firmware to act as a decoy and human-input-device (HID) against the host PoS. In this paper, we propose exploring the processing capabilities and extensibility of FreeRTOS to act as a dual HID clone and printer for continued research.



(a) BTP-S80 (b) Printer I/O

Figure 1.2: SNBC BTP-S80

The goal of this research is to further establish academic works in regards to embedded printer devices testing and security. This area is loosely documented within academia and only mentioned vaguely in relation to statistical reports or applied research using entirely different environments. Through this research we hope to apply gainful conclusions towards the development of an embedded environment for vulnerability assessment, penetration testing, and hardware-to-software interoperability against device hosts. Some examples of how the research could be applied in the future vary: BadUSB/BashBunny [10], JuiceShop [11], DVWA [12], or Webgoat [13]; no such work exists for embedded systems within the point-of-sale or serial printer context.

Related Works

2.1 RTOS: Software and Security

[14] introduces several embedded kernels and discusses their differences in regard to developing a secure mass storage device. For this research, we are primarily interested in RTOS-like kernels because of existing support for a sample device like the SNBC BTP-S80 printer. However, the paper criticizes such operating systems because their "real-time driven design is barely compatible with the overhead produced by security mechanisms." For many applications, there is a trade off with RTOS where performance is the main criteria and security is not a priority. [15] introduces several common RTOS and discusses their security issues. Notably, most RTOS are susceptible to code injection, cryptography inefficiency, unprotected shared memory, priority inversion, denial of service attacks, privilege escalation, and inter-process communication vulnerabilities. Depending on the MPU (microprocessor unit), the vendor has hardware protections like Intel SGX or Arm Trust Zone. These are all areas that can be used for pivoting onto the device, especially shared memory and privilege escalation. If the target device firmware is outdated (or, even libraries used by the firmware) and there are known CVEs that can be repeatedly exploited, persistence mechanisms are not a requirement to gain routine access.

2.2 Embedded Firmware Patching

Typically, updating the firmware for a device or even delivering patches requires a complete shutdown and hardware debug access (if supported). In some cases, the reflashing is unsupported through the operating system or bootloader and the flash memory needs to be reprogrammed. [16] describes a method for hotpatching downstream RTOS devices without needing to shutdown or reboot. Any changes made are permanent and as effec-

tive as traditional delivery methods. RapidPatch was capable of patching over 90% of vulnerabilities for the affected device, only needing at least 64KB or more memory and 64 MHz MCU clock. This appears to be an effective method for attackers to sideload client or server implants without risking detection.

2.3 BadUSB-like Devices

BadUSB is a well-known and documented attack vector. One of the most popular hacker tools is built-on the concept [10]. However, there are some limitations:

- Precision of attacks is limited since scripts or effects are typically deployed blind. There is no knowledge of the user environment nor ability to interact with functional user interface mechanisms (e.g., a mouse clicking a button).
- Limited to the USB 2.0 standard. Meaning, no support for video adapters like HDMI, DisplayPort, or PowerDelivery like with USB 3.0.
- There are existing methods for limiting USB access from the host, such as GoodUSB [17].

GoodUSB supports the Linux USB stack, so another solution would be required for Windows systems or RTOS. This all depends on the environment of the connected host, the PoS system. It is entirely possible that the PoS could have software like Crowdstrike Falcon deployed, which would monitor system behavior and mass storage device access [18]. Although the experiment environment will not use such software, it is an important distinction to make.

In [19], they describe several attacks at each of the applicable layers to USB attacks: the human, application, transport, and physical layers. These attacks would typically require some human element for deployment, but that is not the focus of the research (e.g., social engineering versus hardware hacking). Whereas the physical layer could

allow signal eavesdropping or injection. This could enable a modified printer to overvolt the host (USBKiller [20]) to cause physical damage or perform other side-channel attacks [21]. Either of those methods would require investigating the device hardware to determine what level of control the bootloader or operating system has over power delivery.

2.4 Summary

As demonstrated by the previous works, vulnerability assessment of an embedded device is a well documented process. However, the extent that a serial thermal printer (e.g., Figure 1.2) can be maliciously expanded through a modified FreeRTOS image, while supporting original functionality, has not. And, given success in the assessment, it could suggest room for continual and improved research.

Proposed Research

3.1 Research Objectives

The goal of this research is to understand the hardware and software capabilities of serial print devices. Whether the hardware can support adding unintended functionality at the application and physical layers. And, with what we know about the USB standard and developing real time operating systems, can that functionality be used to create a dual purpose device?

3.2 Research Questions

The research questions this study aims to answer are as follows:

- **RQ1:** What is the baseline or minimum hardware these devices are running?
- **RQ2:** What software is being used on these devices? OS, libraries...
- **RQ3:** Can the software/firmware be modified? FreeRTOS/ReconOS/VXWorks.
- **RQ4:** If so, how much can be modified in memory? Is manually reflashing possible?
- **RQ5:** Assuming reflashing is possible, can the original OS keep original functions and be used as a HID clone or hub?

3.3 Methodology

There are several parts to the methodology of the proposed research. First, technical information and datasheets must be collected for each of the identified devices. Then, device capabilities will be verified before beginning teardown and flash recovery. During

the disassembly, each component will be documented and further technical information will be gathered from respective manufacturers. The format for presenting the collected data is described later in section 3.3.3.

3.3.1 Research Approach

For this research, the quantitative approach and survey research will be used [22], [23]. Because the goal of the research is to gather and examine, point-in-time, data across a sampled population of serial printer devices. By using quantitative survey research, it is possible to evaluate which devices are vulnerable to the attacks hypothesized, as well as, which devices are the most eligible for future design artifact research (i.e., creation of modified OS for HID cloning).

3.3.2 Cross-Sectional Survey

Using cross-sectional surveys [23] has multiple benefits. It can be used to represent data as it is taken, rather than over a long period of time. The study method also focuses on providing summaries that describe the patterns and context between collected data, and how it relates to the research questions.

3.3.3 Data Collection Process

The data collection process begins with gathering technical specifications from device manufacturers. Typically, these contain information about the capabilities of the intended device functions. For a printer, this could contain information ranging from hardware specifications (e.g., CPU, architecture, memory) to things like printed pages per minute. This information forms the baseline for the device survey. Afterwards, further specifications will be gathered for components as each device is disassembled and examined.

Roughly, the types and format of gathered device specifications will appear as follows (e.g., SNBC BTP-S80 is used here):

Specifications	
Max print speed	120mm (Two-Color), 150mm (Grayscale), 250mm (Mono)
Printing method	Direct Thermal
Paper roll type	9 x 7, 82.5 x 80 x 57.5mm
Bar code support	UPC-A, UPC-E, EAN8, EAN13, Code39, Code93, CODE128, CODABAR, ITF, PDF417, QR Code, Maxicode
Printer interpreter	ESC/POS
Interfaces	Serial+USB+Ethernet USB+Parallel USB+Serial USB+Bluetooth USB+WiFi USB Only
Supported OS	32-bit (Windows XP/2000/POSReady) 64bit (Windows XP/Server 2012) 32/64bit (Windows 10/8.1/8/7/Server 2008/Server 2003/Vista) Other (Linux/OPOS/BYJavaPOS Windows/BYJavaPOS Linux)
Development Kit	Android, iOS
Data Buffer	Receive Buffer RAM: 64KB RAM Bitmap: 128KB Flash Bitmap: 512KB
Power Supply	AC 100 $\sim$ 240V, 50/60 Hz Adapter
Current/Power Usage	2.0A / 60W
Safety and EMI	FCC/UL

Table 3.1: Device specifications for SNBC BTP-S80

Following the previous example, the next step in the data collection process would be identifying the SoC. In the event that there is no beforehand knowledge, the SoC can be identified by comparing gathered datasheets during the components discovery. This is easily accomplished using an online service like FindChips [24]. The expected type and format for SoCs is described by Figure 3.2.

The process for gathering flash/memory chip specifications is similar; identify serial number and manufacturer, then find the component datasheet. Gathering the pin layouts and format is useful for later stages, should manual flash recovery be needed. The expected format for memory chips can be seen at Figure 3.3.

Specifications	
Architecture	32-bit ARM
Platform	ARM Cortex-M3
Frequency	80-MHz, 100DMIPS performance
Memory	128KB single-cycle Flash memory 64KB single-cycle SRAM
Firmware	Internal ROM loaded with StellarisWare
Advanced Comm. Interfaces	UART, SSI, I2C, I2S, CAN
Debug Interfaces	JTAG, SWD
Package format	100-pin LQFP 108-ball pin BGA

Table 3.2: SoC technical specs example using Stellaris LM3S2793 Microcontroller

Specifications	
Single power supply operation	2.7 to 3.6V
Software Features	SPI Bus Compatible Serial Interface
Memory architecture	Uniform 64KB sectors 256 byte page size
Programming	Page programming (up to 256 bytes) Operations are page-by-page basis Accelerated mode via 9V W#/ACC pin Quad page programming
Erase commands	Bulk erase function Sector erase for 64KB sectors Sub-sector erase for 4KB and 8KB sectors
Protections	W#/ACC pin used with Status Register Bits to protect specified memory regions andconfigure parts as read-only One time programmable area for permanent and secure identification
Package format	16-pin SO 8-contact WSON 24-ball BGA, 5x5 pin config 24 ball BGA, 6x6 pin config

Table 3.3: Memory specifications example using Infineon Technologies S25FL064P [25]

A final report will be created detailing each of these tables for the devices and their identified core components. Operating system features and protections will be loosely summarized for each device, there is not set reporting format or requirements. Using the final report will aid in the process of designing an artifact for future research and testing.

3.3.4 Hardware Assessment

NIST SP 800-115 [26] provides general guidelines for performing information security testing and assessment, however, there is little information regarding hardware reverse engineering and firmware analysis. Their guidelines are aimed more towards single/multi-tasking operating systems like Windows or Unix-like, those where network logging and listener agents is feasible. For the targeted devices in this research proposal, a different approach is needed that evaluates hardware protections of the SoC and flash memory.

Analysis of device components, once disassembled, requires using a hardware debugger tool with the correct interface. The majority of the targeted devices are expected to use joint test action group (JTAG) or single wire debugging (SWD). By referring to the manufacturer datasheet for a given SoC, it is possible to identify the pin layout for serial debugging access.

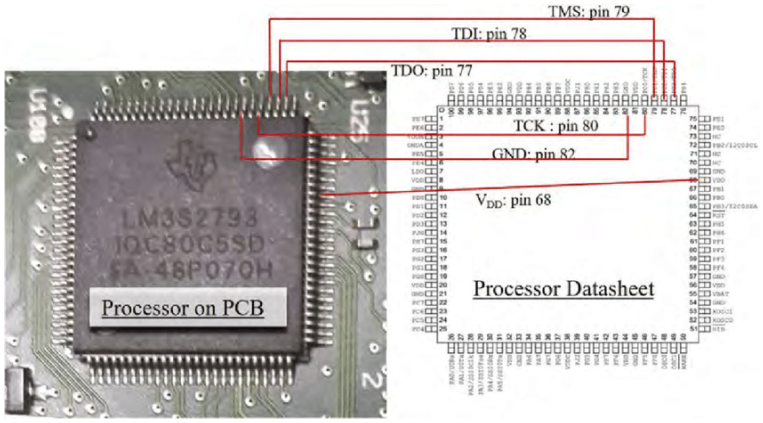


Figure 3.1: JTAG pin out example for Texas Instruments LM3S2793

Figure 3.1 is an example showing what the physical SoC looks like on a PCB compared to the pin layout described in the datasheet. The dot in the top left of the SoC denotes the beginning of the pin layout. Counting in a counter-clockwise method indicates the pin number and the associated functions. For instance, to access the JTAG debug interface on the LM3S2793:

- TDO: pin 77

- TDI: pin 78
- TMS: pin 79
- TCK: pin 80
- GND: pin 82
- V_{DD}: pin 68

Using this information, a device like the JTAGULATOR [27] can be connected and enumerate or verify pin layouts as described. Ball joint SoCs require a different process and are much harder to debug if there is no visible header available on the board. Once an interface is connected, if debugger access is not disabled, the researcher can interact with the bootloader to further investigate enabled protections and recover flash storage.

If the JTAG is disabled, the researcher will then attempt to recover flash manually using a device like the Segger J-Link [28]. The Segger has pre-defined and existing support for working with flash memory and flash breakpoints, whereas using OpenOCD with the JTAGULATOR would require time crafting custom configurations. Assuming there are no access protections to the flash memory, the researcher can begin performing firmware analysis to identify the operating system or potential vulnerabilities. Documenting the size and address range of memory regions is a key part of the process.

Timeline

The timeline for the research proposal is divided into four parts: review, surveying, disassembly, and writing. The dates provided are rough estimates and will vary as the project progresses. Each part is described as follows (refer to Table 4.1):

- **Review**: Review the proposal before beginning the research process to familiarize with the defined methodology, processes, and scope of project.
- **Surveying**: Gather the necessary technical data and acquire the devices.
- **Disassembly**: Each device is torn down, components identified, and documented.
- **Writing**: All data collected, pictures taken, and documents created will be gathered to create a formal report as dictated by the research purpose.

Task	Duration	Start Date	End Date
Review	7 days	06 Nov 23	13 Nov 23
Reread proposal paper	1 days	06 Nov 23	07 Nov 23
Review each related works	3 days	07 Nov 23	10 Nov 23
Take notes	1 day	10 Nov 23	11 Nov 23
Prepare work environment	2 day	11 Nov 23	13 Nov 23
Surveying	7 days	13 Nov 23	20 Nov 23
Gather list of most popular printers and acquire	4 days	14 Nov 23	17 Nov 23
Collect technical datasheets and specifications	2 days	18 Nov 23	19 Nov 23
Verify I/O and document before teardown	1 days	20 Nov 23	20 Nov 23
Disassembly	12 days	20 Nov 23	03 Dec 23
Disassemble each printer	3 days	20 Nov 23	22 Nov 23
Document interior/exterior of devices	1 days	23 Nov 23	23 Nov 23
Identify device components	3 days	24 Nov 23	26 Nov 23
Document interior/exterior of devices	3 days	27 Nov 23	29 Nov 23
Attempt hardware debug	1 days	30 Nov 23	30 Nov 23
Identify software/hardware protections	3 days	01 Dec 23	03 Dec 23
Writing	3 days	04 Dec 23	06 Dec 23

Table 4.1: Research lifecycle

Conclusion

The goal of this research is to assess different serial printers for their equipped hardware for input/output, architecture, memory type/capacity, processor; software for operating system, network stack library (e.g., think Treck TCP vulnerabilities or FreeRTOS specific), featured capabilities (e.g., intended operations like ESC/POS commands to printed paper over serial), and security protections. Then, with the information that has been gathered, make several determinations: what hardware/software protections can be relaxed, if modified how can firmware be pushed to the device, can memory be removed and reflashed, does the device have hardware debugging, is the hardware debugging enabled, is the native operating system open source, can we modify the operating system and save versions with debug symbol data, is the flash storage on the device enough for multiple functions (e.g., webserver and printing or BadUSB and printing).

Although the research itself is not entirely novel or critical to furthering the field, it proposes an original combination of theory and applied research to PoS security. Using a quantitative approach with cross-sectional surveys for a small sample of printer devices, this research will provide a final report to aide future research and development of a design artifact. Specifically, using the surveyed data to aide the research of a modified RTOS, or similar OS, to create a BadUSB-like device.

References

[1] "Consumer Sentinel Network Data Book for January - December 2011," Federal Trade Commission. (Oct. 22, 2023), [Online]. Available: https://www.ftc.gov/reports/consumer-sentinel-network-data-book-january-december-2011 (visited on 10/23/2023).

[2] C. FortheSentinel, "Consumer Sentinel Network Data Book 2022," 2022.

[3] 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.

[4] J. Ondrus and K. Lyytinen, "Mobile Payments Market: Towards Another Clash of the Titans?" In *2011 10th International Conference on Mobile Business*, Jun. 2011, pp. 166–172. DOI: 10.1109/ICMB.2011.41. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6047067?casa_token=jp6ioVlqPjQAAAAA:L69Yx3rjP2tvbnS5zWF8-eMm8sUNjXR586jm2QV6hLH8T2MyuueyIFCeMDqjhNWtUh-_2ERWgQ (visited on 10/23/2023).

[5] S. T. Ebimobowei, Z. Enebraye Peter, and Y. Pual, "THE ROLE OF SOFTWARE IN A CASHLESS ECONOMY (CASE STUDY NIGERIA)," *International Journal of Research -GRANTHAALAYAH*, vol. 6, no. 1, pp. 177–186, Jan. 31, 2018, ISSN: 2350-0530, 2394-3629. DOI: 10.29121/granthaalayah.v6.i1.2018.1607. [Online]. Available: https://www.granthaalayahpublication.org/journals/index.php/granthaalayah/article/view/IJRG17_A11_812 (visited on 10/23/2023).

[6] "SNBC BTP-S80 Thermal Printer - Black Cabinet (USB/Serial/Ethernet)," CRS Inc. (), [Online]. Available: https://www.crs-usa.com/products/snbc-btp-s80-thermal-printer-black-(usb-serial-ethernet) (visited on 10/23/2023).

[7] "SNBC New Beiyang-Intelligent Micro-Super, Smart Express Cabinet, Barcode Label Printer, Ticket Printer_Electronics_Receipt/Log Printer, Barcode/Label Printer, Special Scanning Products, Mixed Print Scanning Products, Smart Express Cabinet, Smart Micro-Super_New Beiyang specializes in the development, production, sales and service of intelligent print identification and system integration products. It provides leading products and complete, one-stop application solutions for various industries around the world. It is the only core design in the industry in the country through independent innovation. Manufacturing technology and forming a large-scale production enterprise." (), [Online]. Available: https://www.snbc.com.cn/ (visited on 10/23/2023).

[8] "Cortex-M4." (), [Online]. Available: https://developer.arm.com/Processors/Cortex-M4 (visited on 10/23/2023).

[9] "FreeRTOS - Market leading RTOS (Real Time Operating System) for embedded systems with Internet of Things extensions," FreeRTOS. (), [Online]. Available: https://www.freertos.org/index.html (visited on 10/23/2023).

[10] Hak5. "Bash Bunny," Hak5. (), [Online]. Available: https://shop.hak5.org/products/bash-bunny (visited on 10/23/2023).

[11] "OWASP Juice Shop — OWASP Foundation." (), [Online]. Available: https://owasp.org/www-project-juice-shop/ (visited on 10/23/2023).

[12] R. Wood, *DAMN VULNERABLE WEB APPLICATION*, Oct. 22, 2023. [Online]. Available: https://github.com/digininja/DVWA (visited on 10/23/2023).

[13] "OWASP WebGoat — OWASP Foundation." (), [Online]. Available: https://owasp.org/www-project-webgoat/ (visited on 10/23/2023).

[14] R. Benadjila, M. Renard, P. Trebuchet, P. Thierry, and A. Michelizza, "Wookey : Usb devices strike back," 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:199552896.

[15] W. D. Yu, D. Baheti, and J. Wai, "Real-Time Operating System Security,"

[16] Y. He, Z. Zou, K. Sun, *et al.*, "{RapidPatch}: Firmware Hotpatching for {Real-Time} Embedded Devices," presented at the 31st USENIX Security Symposium (USENIX Security 22), 2022, pp. 2225–2242, ISBN: 978-1-939133-31-1. [Online]. Available: https://www.usenix.org/conference/usenixsecurity22/presentation/he-yi (visited on 10/24/2023).

[17] D. J. Tian, A. Bates, and K. Butler, "Defending Against Malicious USB Firmware with GoodUSB," in *Proceedings of the 31st Annual Computer Security Applications Conference*, ser. ACSAC '15, New York, NY, USA: Association for Computing Machinery, Dec. 7, 2015, pp. 261–270, ISBN: 978-1-4503-3682-6. DOI: 10.1145/2818000.2818040. [Online]. Available: https://doi.org/10.1145/2818000.2818040 (visited on 10/24/2023).

[18] J. Backer, "Sdn-controlled isolation orchestration to support end-user autonomy," Ph.D. dissertation, WORCESTER POLYTECHNIC INSTITUTE, 2021.

[19] J. Tian, N. Scaife, D. Kumar, M. Bailey, A. Bates, and K. Butler, "SoK: "Plug & Pray" Today – Understanding USB Insecurity in Versions 1 Through C," in *2018 IEEE Symposium on Security and Privacy (SP)*, May 2018, pp. 1032–1047. DOI:

10.1109/SP.2018.00037. [Online]. Available: https://ieeexplore.ieee.org/document/8418652 (visited on 11/05/2023).

[20] "USB Kill devices for pentesting & law-enforcement," USBKill. (), [Online]. Available: https://usbkill.com/ (visited on 11/05/2023).

[21] K. Sridhar, S. Prasad, L. Punitha, and S. Karunakaran, "EMI issues of universal serial bus and solutions," in *8th International Conference on Electromagnetic Interference and Compatibility*, Dec. 2003, pp. 97–100. DOI: 10.1109/ICEMIC.2003.237887. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1287775 (visited on 11/05/2023).

[22] R. Babbie, *The Basics of Social Research*. Cengage Learning, 2017, ISBN: 978-1-305-58586-7. [Online]. Available: https://books.google.com/books?id=O1SXzwEACAAJ.

[23] J. Creswell and J. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE Publications, 2017, ISBN: 978-1-5063-8671-3. [Online]. Available: https://books.google.com/books?id=KGNADwAAQBAJ.

[24] "Findchips: Electronic Part Search." (), [Online]. Available: https://www.findchips.com/ (visited on 11/06/2023).

[25] "S25FL064P Series NOR Flash Datasheets – Mouser." (), [Online]. Available: https://www.mouser.com/c/ds/semiconductors/memory-ics/nor-flash/?series=S25FL064P (visited on 11/06/2023).

[26] "NIST SP 800-115," *NIST*, [Online]. Available: https://www.nist.gov/privacy-framework/nist-sp-800-115 (visited on 11/06/2023).

[27] *JTAGulator*, Grand Idea Studio, Nov. 5, 2023. [Online]. Available: https://github.com/grandideastudio/jtagulator (visited on 11/06/2023).

[28] "SEGGER J-Link debug probes." (), [Online]. Available: https://www.segger.com/products/debug-probes/j-link/ (visited on 11/06/2023).