International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Unveiling Vulnerabilities in Serial Printer Point - of - Sale Systems: A Hardware Analysis Approach

Santosh Kumar Kande

Email: *kandesantosh9[at]gmail.com*

Abstract: ***With the escalating rates of credit card fraud, securing payment systems, particularly physical point - of - sale (PoS) terminals, is imperative. This paper presents a comprehensive hardware - focused investigation into the security vulnerabilities of serial printer PoS systems, a crucial but often overlooked aspect of payment security. Utilizing a quantitative survey research approach, the study delves into these systems hardware and software configurations, addressing key research questions concerning their baseline hardware, software capabilities, and susceptibility to modification. The study identifies potential entry points for exploitation through meticulous disassembly and component analysis and examines the feasibility of firmware modification. Furthermore, the research sheds light on the critical role of device manufacturers in understanding and fortifying the security features of peripheral devices. The findings underscore the need for heightened vigilance and proactive measures to safeguard PoS systems against emerging threats in the digital landscape.***

Keywords: serial devices, thermal printer, PoS, ICS, badusb, hardware hacking, embedded devices.

1. Introduction

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 a 30.5% increase in credit card payment fraud compared to 2012. By comparison, since 2020, there has been a 14.6% increase in credit - card - related fraud. This excludes the millions of other fraud reports that the FTC receives every year. In 2022 alone, there were around 5.1 million fraud, identity theft, and miscellaneous reports in total [13, 6]. The statistics for these reports stress how crucial the security of payment systems is, 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, this does not include mobile payment apps such as Venmo, CashApp, Zelle, or PayPal [18]. There are many reasons, but the types of systems being targeted vary greatly in terms of the hardware and software supported, as well as, how the transactions are handled with the payment processor.

Serial devices pose a significant threat to the PoS security landscape and industrial control systems due to the limited security features of the devices. Because these devices implement no host monitoring, minimal hardware protections, dubious component suppliers, and unchecked communication with their host. Furthermore, the manufacturer's origin, supplied components, firmware, and supporting libraries are separate areas of research that could provide further scrutiny into where these devices are deployed and in what environments.

1.1 Research Objectives

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 used on these devices? OS, libraries. . .
- RQ3: Can the software/firmware be modified? FreeRTOS, ReconOS, and 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 an HID clone or hub?

Manufacturers must understand the hardware and software capabilities of their peripheral devices. During the design process, they should ask whether the hardware can support adding unintended functionality at the application and physical layers. With what I know about the USB standard and developing real - time operating systems, can that functionality be used to recreate a dual - purpose device? This research answers all of the questions presented.

2. Methodology

There are several parts to the research methodology. First, technical information and datasheets were collected for each identified device. Then, device capabilities were verified before beginning device teardown and flash recovery. During the disassembly, each component was documented and further technical information was gathered from respective manufacturers. The format for presenting the collected data is described later in section 2.3.

2.1 Research Approach

For this research, the quantitative approach and survey research was used [11, 12]. Because the goal of the research was to gather and examine, point - in - time, data across a sampled population of serial printer devices. By using quantitative survey research, it was 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).

2.2 Cross - Sectional Survey

Using cross - sectional surveys [12] has multiple benefits. It can represent data as it is taken rather than over a long period. The study method also provides summaries describing the patterns and context between collected data and how it relates to the research questions.

2.3 Data Collection Process

The data collection process began 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 printed pages per minute. This information forms the baseline for the device survey. Afterward, further specifications were gathered for components as each device was disassembled and examined. Roughly, the types and format of gathered device specifications appear as shown in Table 1 (e. g., SNBC BTP - S80 is used here).

Table 1: Device specifications for SNBC BTP - S80

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 ~ 240V, 50/60 Hz Adapter
Current/Power Usage	2.0A / 60W
Safety and EMI	FCC/UL

Following the previous example, the next step in the data collection process would be identifying the SoC. The SoC can be identified without prior knowledge by comparing gathered datasheets during the discovery of the components. This is accomplished using an online service like FindChips or AllDataSheets [3]. The process for gathering flash/memory chip specifications was similar: identify the serial number and manufacturer, then find the component datasheet. Gathering the pin layouts and format is also useful for later stages, should the manual flash be recovered.

2.4 Hardware Assessment

NIST SP 800 - 115 [5] 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 are feasible. For the targeted devices in this research proposal, a different approach is needed to evaluate the hardware protections of the SoC and flash memory.

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

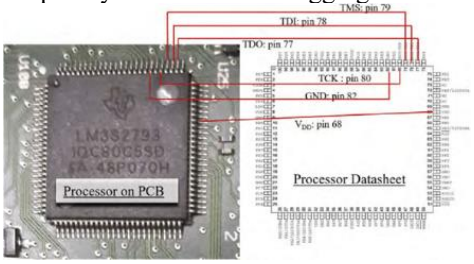

Figure 1: JTAG pin out example for Texas Instruments LM3S2793

Figure 1 shows 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
- VDD: pin 68

Using this information, a device like the JTAGULATOR [7] 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 no visible header is 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 attempt to recover flash manually using a device like the Segger J - Link [4]. The Segger has predefined and existing support for working with flash memory and breakpoints. Using OpenOCD with the JTAGULATOR would require time to craft custom configurations. Assuming no access protections exist to the flash memory, the researcher could begin performing firmware analysis to identify the operating system or potential vulnerabilities. Documenting the size and address range of memory regions was a key part of the process.

3. Related Works

3.1 RTOS: Software and Security

Introduces several embedded kernels and discusses their differences regarding developing a secure mass storage device. I am primarily interested in RTOS - like kernels for this research 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 largely incompatible with the overhead produced by security mechanisms. ” For many applications, there is a trade - off with RTOS where performance is the main criterion and security is not a priority. [19] 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 areas 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.

3.2 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. [15] describes a method for hot - patching downstream RTOS devices without needing to shut down or reboot. Any changes made are permanent and as effective as traditional delivery methods. Rapid Patch was capable of patching over 90% of vulnerabilities for the affected device, only needing at least 64KB or more memory and a 64 MHz MCU clock. This is an effective method for attackers to sideload client or server implants without risking detection.

3.3 BadUSB - like Devices

BadUSB is a well - known and documented attack vector. One of the most popular hacker tools is built on the concept [14]. 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. This means there is no support for video adapters like HDMI, DisplayPort, or PowerDelivery like 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. Although the experiment environment will not use such software, it is an important distinction.

4. Results

I had to adjust the scope of the research due to time constraints and under - budgeting. Initially I have aimed for a small sample population of three to five of the most common, accessible serial printers. However, for this study, I focused solely on the SNBC BTP - S80 serial printer.

4.1 Device Disassembly

The SNBC BTP - S80 is a common thermal printer used for providing receipts for PoS systems and immediate reporting for industrial control systems (ICS).

Figure 2: SNBC BTP - S80

This model features three buttons on the front face of the printer. Starting from the top: paper roll release, auto - feeding, and power button.

I can see some of the available I/O from the device’s rear. There are several ways to interface with the thermal printer. The host device can connect using an internet address via the RJ45 ethernet connector or over serial using the USB type - b and RS232. I can also see a screw on either side of the expansion card containing the RJ45 jack and RS232 connector. The manufacturer’s website shows this slot is

interchangeable and can provide different functionality depending on the end - user's needs.
Removing each of the screws allows us to slide the chassis out of the outer shell. Here, I can see the motherboard (leftmost) and the expansion cards (top - side, right of the motherboard). The expansion cards are divided into two parts, as shown in Figure 5.

The left expansion card allows the device user to swap networking stacks. In this configuration, it features the RJ45 Ethernet and RS232 serial connector. The right expansion card provides USB connectivity and power via the barrel plug connector.

The disassembly is completed after carefully disconnecting each cable, noting their respective connectors, and preparing the boards for component identification. Although there are plenty of components within the device, my concern is only the motherboard and two expansion cards. I am only interested in researching the components used for processing and storing data.

4.2 Identified Components

Component identification and analysis is divided into two parts. The first is an analysis of the motherboard, and the second is the serial expansion card. Analysis of the power delivery and USB type - b expansion card is unnecessary since it has no controllers nor any flash to analyze. In some cases, these might still be used for fuzzing and debugging because labeled connectors are provided. However, this can be ignored with most single - wire debugging tools (e. g., Jtagulator or Bus Pirate).

The list of motherboard components, as shown in Figure 5:
- NXP 32 - bit ARM Cortex - M4 MCU, LPC4078FET208
- ESMT DRAM, M12L16161A - 7T

The list of expansion components, as shown in Figure 6:
- SIPEX RS - 232 Transceiver, SP2209E
- ASIX Ethernet Controller, AX88796C

All of these parts can be identified by reading the labeled markings. Typically, there is a manufacturer logo, production date (e. g., YYMM), lot number, and part number. Using this information, I cross - referenced a part database to validate the functionality of the component's package (e. g., QFP versus BGA types). The records should tell us whether the component is the main processing unit, flash storage, or a microcontroller. Refer to Table XX for relevant specifications regarding each component. This information aids the firmware analysis section as well as future research.

Table 2: NXP 32 - bit ARM Cortex - M4 MCU, LPC4078FET208 [8]

Specifications	
Architecture	32 - bit ARM
Platform	ARM Cortex - M4
Frequency	120 - MHz performance
Memory	512 kB flash program memory 96kB SRAM data memory 4032 bytes of EEPROM data memory
Features	External Memory Controller (EMC) General Purpose DMA Controller Quadrature Encoder Interface CRC calculation engine Code Read Protection (CRP)
Advanced Comm. Interfaces	UART, SPIFI, I2C, USB/OTG, Ethernet, SSP
Debug Interfaces	JTAG, SWD
Power	Single 3.3V (2.4V to 3.6V)
Package format	208 - ball, TFBGA208

Table 3: ESMT DRAM, M12L16161A - 7T [9]

Specifications	
Single power supply operation	3.3V
Software Features	CMOS Technology Synchronous high data rate DRAM LVTTL compatible with multiplexed address. Burst Read Single - bit Write operation
Memory architecture	Dual banks operation 16, 777, 216 bits organized as 2 x 524, 288 words by 16 bits
Package format	50 - pin, TSOP

Table 4: SIPEX High ESD Dual Port RS - 232 Transceiver, SP2209E [10]

Specifications	
Single power supply operation	12V Operates with +3V or +5V logic as well as standby
Comm. Interfaces	Two serial ports, Six drivers, and 10 receivers. One receiver on each port for active standby. 460kbps minimum data rate
Features	LapLink Compatible Low EMI Emissions (EN55022) Fast Transient Burst (EFI) Immunity (EN61000 - 4 - 2) Pin compatible with ADM2209E
Package format	38 - pin, TSSOP

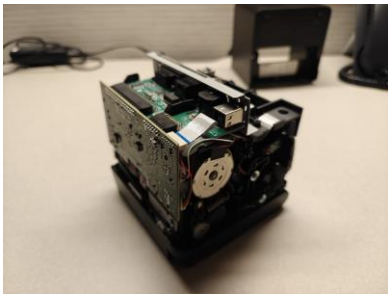

Figure 3: SNBC BTP - S80 inner chassis

4.3 Technical Resources

All technical data and specifications sheets were procured through AllDataSheet or FindChips [1, 3]. There is a litany of other sources that can be used. There are no empirical reasons for using these sources instead of others; they were purely motivated by personal preference. Regardless of whichever datasheet aggregator is used, it should not harm the replicability of the research.

4.4 Firmware Analysis

As shown in Section 2.4 Hardware Assessment, I can determine the correct pin layout for interfacing with the debug headers using the manufacturer datasheet. Whether or not this process is successful largely depends on the type of

component packaging. In the previous example, where I demonstrated reading the datasheet in relation to the pin diagram, it was possible to attach a debugger to the interface without a header prepared by the manufacturer on the device PCB. This is not possible for the SNBC BTP - S80.

Figure 5 and the corresponding datasheet show that the MCU uses a 208 - ball thin and fine ball grid array (TFBGA) package. Unless I used decapsulation or delamination, it is impossible to access these pins without risking damage to the device [16]. The manufacturers of the BTP - S80 were kind enough to provide a JTAG debug header, as shown and labeled in Figure 5.

However, the exact pin layout is not provided. This can be verified manually via a multimeter and logic analyzer while referencing the manufacturer's documentation. The process can be time consuming for those unfamiliar, and a tool such as the Jtagulator [7] is recommended for enumerating the correct configuration.

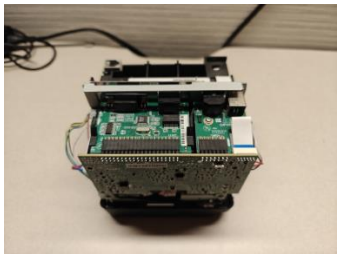

Figure 4: SNBC BTP - S80 expansion cards

The research can connect up to 24 channels or pins with the ground and voltage reference pins validated. The tool will then check every permutation of the channels until an available header is identified. Using this information, I can continue with the Jtagulator or another USB debugger to complete the investigation. Figure 7 provides a clearly labeled diagram of the JTAG layout.

In Section 4.2 Identified Components, I discovered that the MCU used by the thermal printer is the NXP LPC4078FET208. This controller has a code read protection (CRP) feature that allows the manufacturer to disable access to the JTAG interface. With this enabled it prevents hardware debugging, firmware recovery, and overall physical interactions. In this instance, SNBC failed to enable the CRP feature for their devices before distributing them. Because of this, I can gain access to the device, pull the memory, and reverse engineer the firmware using tools such as Ghidra.

Using a J - Link Segger, a USB serial debugging device, I was able to pull all active memory stored in the thermal printer's EEPROM. The total memory dump is 659, 460 KiB (645MiB). Opening the memory dump using a disassembler like Ghidra allows us to begin examining the functions and data.

Upon inspecting the strings for the dump, I can immediately see that the thermal printer is running a modified version of RT - Thread RTOS, version "FV1.040.00. " This is an open - source operating system that is community developed and accessible to any party. With this information, it is possible to begin recreating our own debuggable version of the operating system for vulnerability analysis or extended features (e. g., BadUSB/MITM).

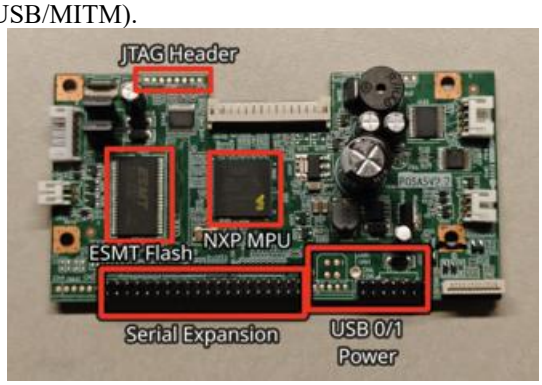

Figure 5: Motherboard components

I can also analyze the remaining strings to see indications of a php web server if the networking interfaces are enabled; remember, these devices have Bluetooth, wireless, and ethernet support. The library used by this device is the WebNet v1.0.0 software package developed for RT - Thread devices. In particular, the webserver supports the following features:

- HTTP 1.0/1.1 and CGI function
- AUTH basic authentication function
- ASP variable substitution function
- SSI file embedding function
- INDEX directory file display function
- ALIAS alias access function
- file upload function
- cache function
- In addition, there were other libraries spotted within the memory dump:
- Epson "7.00 ESC/POS", the printer command interpreter

Records showed that the BTP - S80 and BTP - 2002NP share the same firmware. A boolean value within the management software can control which profile is used. There is also an option to choose the default profile or printer mode. These settings, the WebNet library, and the Epson ESC/POS print interpreter are potential areas for vulnerability research. Especially if the printer is connected to the network and the web server is running, it might provide a form of persistence or entry into the device.

- "Printer Mode"
- "Default Printer Mode\n"
- "The Printer Mode will be BTP - 2002NP Mode\n"

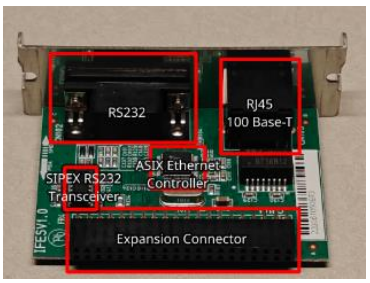

Figure 6: Expansion card components

5. Conclusion

In conclusion, the research demonstrates four of the five objectives. I identified a baseline of the hardware and operating system used by the serial printer. I identified the

specific version of the operating system, any supporting libraries, and their versions (i. e., RT - Thread RTOS). I could also clearly demonstrate that the manufacturers have enabled minimal security protections despite the hardware supporting them. It is especially noteworthy that the operating system has minimal protections for isolating memory across processes other than CRC checksums. It is also possible, as demonstrated, to reflash the memory utilizing the MCU hardware debug interfaces.

Although the SNBC BTP - S80 is a fairly common serial printer used across financial sectors and industrial control, the research would have been better represented with the cross - examination against several other devices. Lastly, another goal of the research was to use the gathered data to support future research proposals into a design artifact for implementing BadUSB - like concepts on peripheral serial devices.

Table 5: ASIX Low - Power SPI or Non - PCI Ethernet Controller, AX88796C [2]

Specifications	
Power	Variable voltage I/O (1.8/2.5/3.3V) Programmable driving strength (8/16mA)
Software Features	SPI slave interface for CPU w/ SPI master Interrupt pin with programmable timer IPv4/IPv6 checksum offload engine VLAN match filter. IEEE 802.3/802.3u standards for 10Base - T/100Base - TX
Memory	8/16 - bit SRAM - like host interface Supports Slave - DMA to minimize CPU overhead. Burst mode read and write access over SRAM - like interface Embeds 14KB SRAM packet buffers. Supports optional EEPROM interface for MAC storage
Package format	64 - pin, LQFP

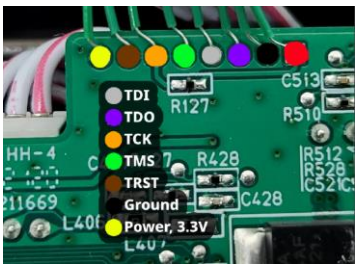

Figure 7: SNBC BTP - S80 Labeled JTAG Header

References

[1] ALLDATASHEET. COM - Electronic Parts Datasheet Search. https: //www.alldatasheet. com/
[2] AX88796C Datasheet https: //www.alldatasheet. com/datasheet - pdf/pdf/502430/ASIX/AX88796C. html
[3] Findchips: Electronic Part Search. https: //www.findchips. com/
[4] SEGGER J - Link debug probes. https: //www.segger. com/products/debug - probes/j - link/
[5] NIST SP 800 - 115. NIST (Jan 2020).
[6] Consumer Sentinel Network Data Book for January - December 2011. https: //www.ftc. gov/reports/consumer - sentinel - network - data - book - january - december - 2011
[7] JTAGulator. Grand Idea Studio (Nov 2023)
[8] LPC4078FET208 AllDataSheet. https: //www.alldatasheet. com/datasheet - pdf/pdf/467560/NXP/LPC4078FET208. html
[9] M12L16161A AllDataSheet. https: //www.alldatasheet. com/datasheet - pdf/pdf/82125/ESMT/M12L16161A - 7T. html
[10] alldatasheet. com: SP2209E Datasheet (PDF) - Sipex Corporation. https: //www.alldatasheet. com/datasheet - pdf/pdf/45906/SIPEX/SP2209E. html
[11] Babbie, R.: The Basics of Social Research. Cengage Learning (2017)
[12] Creswell, J., Creswell, J.: Research Design: Qualitative, Quantitative, and Mixed Methods Approaches. SAGE Publications (2017)
[13] FortheSentinel, C.: Consumer Sentinel Network Data Book 2022 (2022)
[14] Hak5: Bash Bunny. https: //shop. hak5. org/products/bash - bunny
[15] He, Y., Zou, Z., Sun, K., Liu, Z., Xu, K., Wang, Q., Shen, C., Wang, Z., Li, Q.: {RapidPatch}: Firmware Hotpatching for {Real - Time} Embedded Devices. In: 31st USENIX Security Symposium (USENIX Security 22). pp.2225–2242 (2022) https: //www.usenix. org/conference/usenixsecurity22/presentation/he - yi
[16] S´anchez Lopez, S.: Contribution to the study of electronic decapsulation (Jun 2015) https: //www.ub. edu/portal/documents/1749702/4543976/Sanchez_Lo pez_Sergi_summary. pdf/0a767707 - 1296 - 418c - a152 - 9ade116ab5ce
[17] Tian, D. J., Bates, A., Butler, K.: Defending Against Malicious USB Firmware with GoodUSB. In: Proceedings of the 31st Annual Computer Security Applications Conference. pp.261–270. ACSAC '15, Association for Computing Machinery, New York, NY, USA (Dec 2015). https: //doi. org/10.1145/2818000.2818040
[18] Wang, Y., Hahn, C., Sutrave, K.: Mobile payment security, threats, and challenges. In: 2016 Second International Conference on Mobile and Secure Services (MobiSec - Serv). pp.1–5 (Feb 2016). https: //doi. org/10.1109/MOBISECSERV.2016.7440226
[19] Yu, W. D., Baheti, D., Wai, J.: Real - Time Operating System Security