PwnBoxer

A network security device intended to provide small businesses and residential networks an automated auditing service to help rectify network vulnerabilities.

Group 22 Authors:

Franco Mezzarapa - Computer Engineering
Gugulethu Sigogo - Electrical Engineering
Abdiel Marcano - Computer Engineering
Robert Lee - Computer Engineering

Review Committee:

Dr. Mike Borowczak - ECE - Associate Professor
Dr. Nazanin Rahnavard - ECE - Professor
Dr. Justin Phelps - ECE - Adjunct Professor

Advisor:

Dr. Saleem Sahawneh - ECE - Lecturer

Contents

1 Introduction

The PwnBoxer is a hardware network auditing device intended for home and small businesses, allowing consumers to achieve affordable automated network auditing with feedback reporting for remediation of basic to intermediate network vulnerabilities. The PwnBoxer is intended to run through an automated workflow that audits the networks to which it is connected using readily available penetration testing tools to audit and generate a report. This tool also allows for technician connection into the network, making it a viable option for Internet Service Providers (ISPs) and Managed Service Providers (MSPs) to provide as a service, in which they can lend the device to do continuous penetration testing and, when needed, have a technician connect and continue the audit manually.

1.1 Motivation

With the rapid advancement of technology, including Artificial Intelligence (AI), new cyber threats and vulnerabilities are continually being discovered. This necessitates ongoing evaluations of organizational weaknesses at both the network and service levels. For small enterprises and residential users, however, conducting vulnerability assessments can be challenging due to financial and technical limitations. Traditionally, network audits require skilled professionals to identify the attack surface of the infrastructure and recommend appropriate mitigations. The proposed solution is designed not only for residential and small business users but also for ISPs and MSPs. By automating the auditing process and enabling technician intervention, ISPs can reduce the cost of conducting assessments while offering a value-added service to their customers. Furthermore, residential and small business users benefit from a low upfront cost for the device and an optional subscription for off-site processing.

1.2 Background

1.2.1 Network Security

Organizations commonly deliver various services to customers, whether deployed onsite, hosted in the cloud, or otherwise non-technological in nature. These services depend on infrastructure to maintain operation. Many enterprises rely on on-site services such as database systems and Active Directory. The software applications running on servers are referred to as services, and the overall security posture of a server is significantly influenced by the specific services running and their respective configurations.

A primary source of software vulnerabilities arises from substandard code quality, particularly when services are developed in languages that allow direct memory manipulation or when user-interface handling follows poor programming practices. Services that accept user input are susceptible to memory-based attacks such as buffer overflows without sanitazation of inputs. Despite operating systems including numerous protections [1], many remote code execution (RCE) vulnerabilities persist, especially when they can be triggered without user authentication or from user-accessible interfaces. Once an attacker compromises a vulnerable service hosted on a server, they may escalate privileges to domain-

administrator levels and pivot laterally to compromise additional services within the network.

Another significant source of data breaches arises due to misconfigurations in network services, which allow adversaries to exploit insecure or unintended functionalities. Common examples include permitting unlimited authentication attempts without rate limiting, enabling brute-force or credential-stuffing attacks, and insecure file upload mechanisms that fail to validate input, potentially leading to arbitrary code execution or malware injection. Other frequent misconfigurations involve leaving default credentials enabled, exposing administrative interfaces to the public Internet, or failing to enforce proper access control policies. Misconfigurations consistently rank among the top causes of security incidents, as they create low-effort entry points for attackers compared to more complex exploitation [2]. Standardized configuration requirements, continuous monitoring, and adherence to security baselines are crucial for mitigating these risks.

Users and Employees are often easy targets for social engineering attacks due to publicly available information about them, weak authentication credentials, or habitual reuse of passwords exposed in prior breaches. Attackers exploit these weaknesses using techniques such as credential stuffing, leveraging leaked credential databases and wordlists such as rockyou.txt to automate logins into servers and services.

1.2.2 WiFi

Wireless networks, while offering flexibility and convenience, introduce a distinct set of vulnerabilities compared to wired infrastructures. A primary weakness lies in outdated or insecure encryption protocols such as WEP and WPA, which are still deployed in some environments despite being deprecated. Attackers can exploit these weak encryption schemes to intercept and decrypt network traffic using relatively inexpensive hardware and widely available software tools. Even WPA2, once considered robust, is susceptible to key reinstallation attacks (KRACK), enabling adversaries to replay packets, decrypt traffic, and potentially inject malicious data [3].

Rogue Access points and Evil Twin attacks are also of concern; adversaries can set up fake wireless networks that mimic legitimate SSIDs. Users who connect to these access points risk exposing credentials or sensitive traffic. Additionally, poor access control measures, such as guest networks without segmentation, can allow lateral movement into production networks. Preventative strategies include adopting WPA3 with modern encryption, deploying wireless intrusion detection systems, segmenting guest traffic, and enforcing strong authentication mechanisms such as 802.1X.

1.2.3 Penetration Testing

Penetration testing is commonly employed to identify vulnerabilities within a target system by an individual, team, or external firm. However, traditional penetration tests are costly and must be repeated periodically to account for architectural changes and evolving security threats. This is addressed through Penetration Testing as a Service (PTaaS), which offers a subscription-based model that combines automated testing with human intervention. PTaaS

platforms often integrate directly into development pipelines, enabling continuous vulnerability assessments and faster remediation feedback. Many services provide dashboards for vulnerability tracking and compliance reporting, making the remediation process more actionable for the customers.

2 Goals

The goals defined here are broad and represent the overall outlook and interpretation of our project. Basic goals constitute the fundamental ideas for what tasks PwnBoxer is intended to perform. Advanced goals are improved functionalities that we intend to make for PwnBoxer in this design, and stretch goals are functionalities for PwnBoxer which would be beneficial to the project, but are beyond the scope of our implementation.

2.1 Basic Goals

- Create a compact and portable system capable of auditing networks and detecting vulnerabilities through an automated pipeline.
- Generate comprehensive reports that summarize the overall security posture and identified vulnerabilities.
- Collect input from peripherals connected to the MCU and transfer the data to the compute module for processing.
- Perform network security computations on the compute module and communicate results back to the MCU for user output.
- Enable direct technician or developer access to the compute module through external peripherals.
- Provide system condition feedback through integrated LED indicators and audio alerts.

2.2 Advanced Goals

- Implement an animated status bar to display progress during network audits.
- Support customizable audit durations based on user requirements.
- Enable simultaneous auditing of multiple networks.
- Allow technicians and advanced users to integrate custom vulnerability tests.

2.3 Stretch Goals

- Incorporate battery bypass functionality, enabling components to draw power directly from AC input to extend battery life.

- Develop a web-based security dashboard that displays test results, trends, and recommendations for the end user.

3 Objectives

The objectives defined here are specific and represent technical and measurable deliverables that will be applied to the project. Basic objectives denote deliverables that are essential requirements for successful system operation. Advanced objectives are aspects in which the system will be significantly improved upon implementation, and stretch goals are ideal specifications that may be implemented if resources permit.

3.1 Basic Objectives

- Execute industry-standard security tools such as Nmap and OpenVAS to audit small business or residential networks.
- Generate vulnerability assessment reports using a large language model (LLM).
- Transmit generated reports to the user via email, with the email address collected prior to the audit.
- Provide a touchscreen interface for collecting network information and email addresses as user input and displaying audit progress updates.
- Utilize LEDs to convey system conditions such as charging status, network audit progress, and device health.

3.2 Advanced Objectives

- Enable remote technician or user support via a secure reverse shell connection to the compute module.
- Implement SPI communication between the compute module and MCU to display pipeline stage information to the end user.
- Support concurrent auditing of distinct networks (e.g., Ethernet and Wi-Fi) when permitted by the user.

3.3 Stretch Objectives

- Add API-based integration to support custom vulnerability tests defined by the technician or end user.
- Implement automated power management to reroute power directly to components when the battery reaches a predefined threshold.
- Host a secure, authenticated web dashboard on the compute module using Apache to present historical performance metrics and comparative results.

4 Existing Products

4.1 Bonelli Systems Automated Penetration Testing

Bonelli Systems offers an automated penetration testing platform that provides real-time network vulnerability assessments and actionable reporting [4]. It is a cloud-based service suitable for small to medium-sized businesses that require continuous security testing. Like PwnBoxer, Bonelli Systems provides automated vulnerability assessments and delivers reports to help users remediate security issues. The differences lie in the deployment and delivery model: Bonelli Systems operates entirely as a cloud service requiring ongoing subscriptions, whereas PwnBoxer is a portable hardware device with optional on-site technician support. Additionally, PwnBoxer is designed for periodic assessments and usability by non-expert users, whereas Bonelli focuses on continuous automated testing.

4.2 Kali Linux on Raspberry Pi

Kali Linux is a widely adopted penetration testing distribution that can be deployed on lightweight single-board computers, such as the Raspberry Pi [5]. This configuration transforms an inexpensive device into a portable penetration testing platform with hundreds of pre-installed tools covering web, network, wireless, and cryptographic security assessments. Operation is highly flexible but requires manual setup of the Raspberry Pi, installation of Kali Linux, and ongoing system administration by the user. User interaction is almost entirely command-line and GUI-driven, depending on the selected tools, making it best suited for technically proficient security professionals, researchers, and enthusiasts. While this approach shares PwnBoxer's small hardware form factor and portability, it lacks PwnBoxer's built-in automation, integrated reporting, and technician support, placing the burden of customization and tool management on the user.

4.3 LAN Turtle by Hak5

The Hak5 LAN Turtle is a covert USB Ethernet adapter designed for stealthy penetration testing and remote access [6]. Once connected to a target network via USB, it operates by establishing a hidden communication channel that allows penetration testers or red teamers to remotely access internal systems. Its capabilities include network sniffing, man-in-the-middle attacks, credential harvesting, and persistent remote shells, making it a versatile and powerful tool for covert operations. Interaction with the LAN Turtle requires command-line configuration and scripting, which assumes a high level of technical expertise from the user. The intended audience primarily consists of professional penetration testers and red team operators who value stealth and persistent access. Like PwnBoxer, it is hardware-based and portable; however, its design philosophy emphasizes covert deployment and deep customization rather than user-friendly automation and reporting.

4.4 MiniPwner by Hacker Warehouse

MiniPwner is a compact penetration testing device built on OpenWrt, typically utilizing a TP-Link MR3040 router and configured to run tools such as Metasploit and Aircrack-ng. It is battery-powered, enabling portable network assessments and providing flexibility for penetration testers to deploy in a variety of environments [7]. Similar to the PwnBoxer, MiniPwner is hardware-based and allows for automated vulnerability testing, including assessments of wireless networks. However, MiniPwner is primarily a DIY device that requires technical expertise to configure and lacks integrated reporting or technician support features. Its focus is more on stealth and portability rather than on providing a fully automated workflow for users with limited technical knowledge.

4.5 Penetrator by SecPoint

SecPoint's Penetrator is a comprehensive vulnerability scanning and penetration testing appliance available as a physical device or virtual machine. It performs automated scanning across internal and external networks, identifying vulnerabilities such as SQL injection, cross-site scripting, and misconfigurations. It also generates detailed reports with remediation suggestions [8]. Like PwnBoxer, Penetrator automates network assessments and produces actionable reports for remediation. However, Penetrator is primarily designed for enterprise environments, is less portable than PwnBoxer, and offers a wider range of scanning profiles that may require technical expertise to operate. PwnBoxer, by contrast, balances automation with ease of use for small businesses and residential users and integrates optional technician intervention and subscription-based off-site processing.

4.6 Pwn Plug by Pwnie Express

The Pwn Plug, developed by Pwnie Express, is a penetration testing device disguised as a power adapter that plugs into an Ethernet port to provide remote access to networks. It is widely used by professional penetration testers for enterprise-level assessments and allows the execution of various penetration testing tools [9]. Similar to the PwnBoxer, it provides hardware-based access for network vulnerability assessments and supports a range of penetration testing tools. The key differences are that the Pwn Plug emphasizes stealth and covert deployment, is designed for professional security experts, and does not provide user-friendly reporting or optional technician intervention like PwnBoxer.

4.7 WiFi Pineapple by Hak5

The WiFi Pineapple is a specialized wireless auditing platform designed for penetration testing of WiFi networks [10]. It streamlines and automates many common wireless attacks, including rogue access point creation, credential harvesting, client deauthentication, and traffic interception. The device provides a web-based interface for configuration, making it more user-friendly than command-line-only tools while still allowing advanced customization through downloadable modules. Its primary audience is penetration testers, red teamers, and IT security professionals focused on wireless security assessments. Com-

pared to PwnBoxer, the WiFi Pineapple automates significant parts of wireless exploitation but is limited in scope to WiFi networks. PwnBoxer, by contrast, supports broader wired and wireless network auditing with integrated reporting and optional technician assistance, making it more accessible to non-expert users while serving a wider range of use cases.

5 Required Specifications

5.1 Overall System Specifications

Maximum Audit Completion Time	4 Hours
Power Consumption	15W
Battery Capacity	3000mAh
Onboard Data Storage	32GB
Onboard Memory Size	8GB
Weight	2 lbs
Size	0.1335m Diagonally
Battery Life	30 Minutes
Display	3.5” TFT with Touch
AI Model	Gemma3n

5.2 Component Specifications

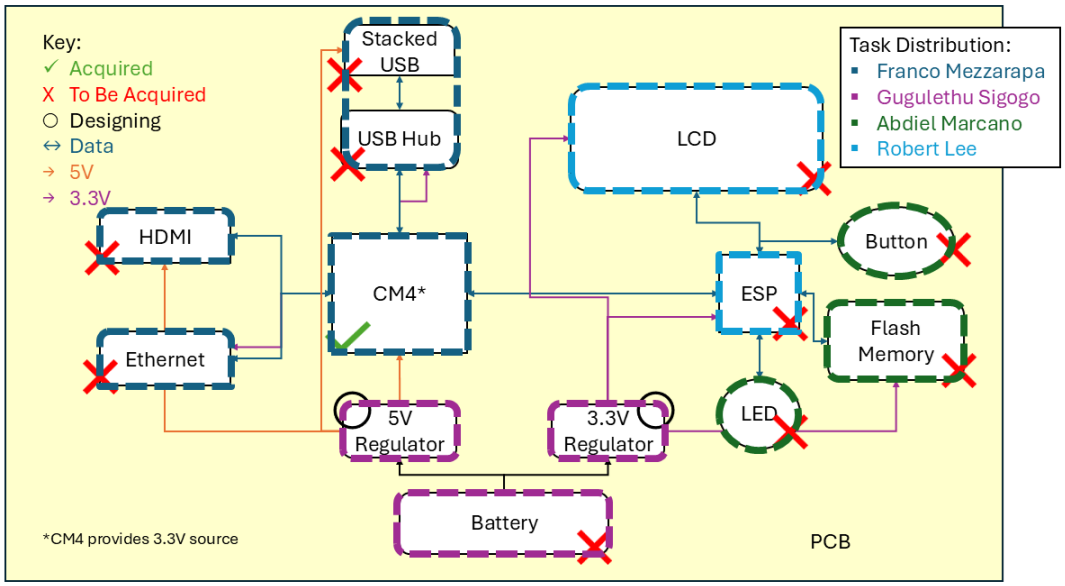
5.2.1 Raspberry Pi Compute Module 4 Specifications

GPIO Pins	28 Pins
Standard Clock Frequency	1.5 GHz
Supply Voltage	5V
Onboard Memory	8GB
Onboard Storage	32GB
Communication Standards	UART, I2C, SPI, Ethernet, PCIe, USB 2.0, HDMI, Wi-Fi

5.2.2 ESP32 Specifications

GPIO Pins	45 Pins
Maximum Clock Frequency	240MHz
Supply Voltage	3.3V
Onboard Memory	512KB SRAM
Communication Standards	UART, I2C, SPI, Wi-Fi, Bluetooth

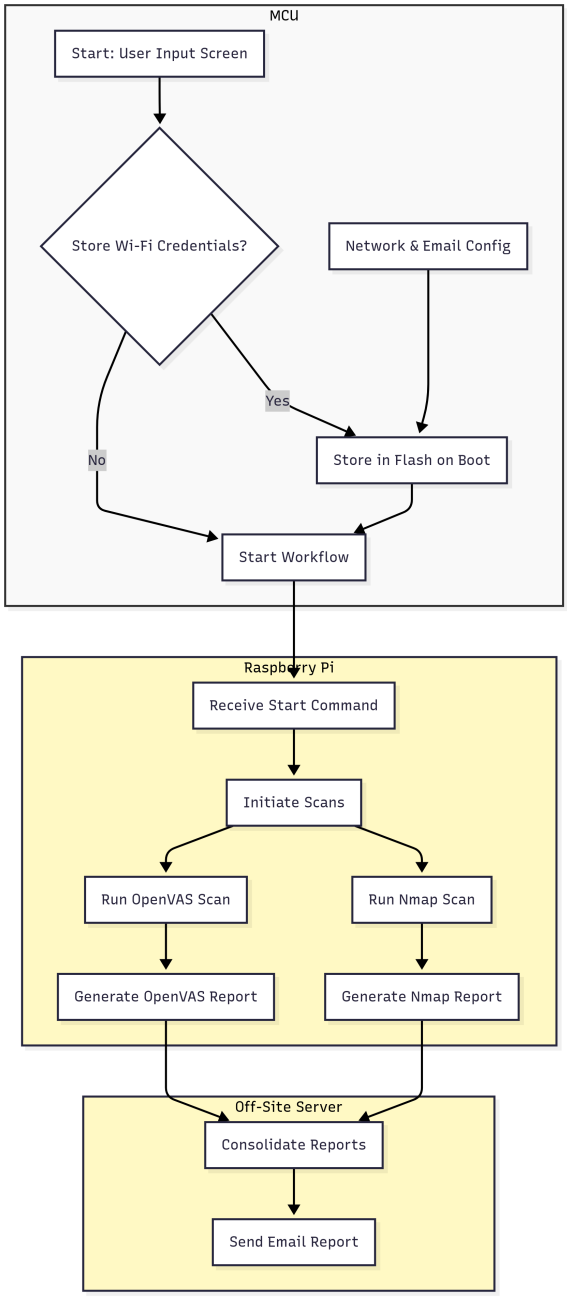
6 Hardware Block Diagram



Hardware block diagram of the PwnBoxer hardware network auditing system, illustrating component interconnections, power regulation, and data flow, including the CM4 module, ESP controller, regulators, peripherals, and communication interfaces.

Figure 6.1 Hardware Block Diagram by Authors

7 Software Flowchart

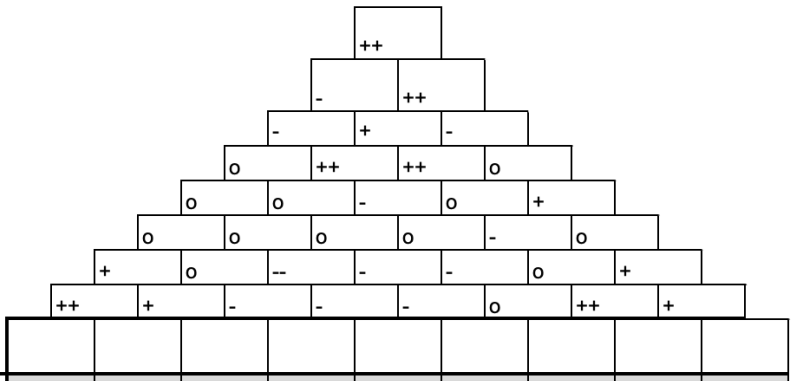


The Software Flowchart outlines the process the system follows for automated vulnerability assessment, including user configuration via the MCU, parallel scan execution using Nmap and OpenVAS on the Raspberry Pi, and a final reporting mechanism handled by an off-site server.

Figure 7.1 Software Flowchart by Authors

8 House of Quality

Strong Positive	++
Positive	+
No Correlation	o
Negative	-
Strong Negative	--



Design Requirements / Customer Requirements		Size	Weight	Battery Life	Task Completion Time	Power Consumption	Report Complexity	Manufacturability	Maintainability	Cost
		-	-	+	-	-	-	+	+	-
Ease of Use	+	++	++	o	++	o	o	o	o	o
Portability	+	++	++	++	o	++	o	o	+	+
Audit Accuracy	+	o	o	o	-	o	--	o	o	o
Customizability	+	-	o	o	o	o	-	-	-	-
Longevity	+	o	o	+	o	++	o	+	++	o
Retail Cost	-	+	+	-	o	-	o	++	++	+
Ease of Report Dissemination	+	o	o	o	+	o	++	o	o	o

The House of Quality provides a visual aid representing engineering tradeoffs that may be made during design and production. The column on the top represents the engineering specifications that can be altered while under consideration. The column to the left represents the specifications that are most important to the customer. The matrices above and between the columns signify the correlations among each of the items individually.

Figure 8.1 House of Quality Diagram by Authors

9 Administrative Content

9.1 Budget

Our team has capped the project budget at $1000, with an expected initial cost of about $215 for core components. The largest expenses will come from the Raspberry Pi Compute

Module 4 and IO board, along with the TFT capacitive touchscreen display. The ESP32-S3 microcontroller, 16MB SPI flash memory, and supporting peripherals such as the speaker, status LEDs, and pushbuttons are relatively inexpensive, costing only a few dollars each. Additional funds are allocated toward power supplies, wiring, and prototyping accessories. By keeping the upfront hardware cost low, we retain a significant buffer of over $700 for replacements, PCB fabrication, or unforeseen design changes during development.

9.2 Bill of Materials

Table 9.2.1 Bill of Materials

Component	*Part Name*	*Quantity*	*Unit Price*	*Total Price*
Computer Engineer				
Raspberry Pi Compute Module 4	CM4 (2GB RAM, 16GB eMMC, WiFi)	1	$75.00	$75.00
CM4 IO Board	Waveshare IO Board Lite	1	$45.00	$45.00
Microcontroller	ESP32-S3-WROOM-1	1	$8.00	$8.00
Flash Memory	Winbond 25Q128JVSIQ (16MB SPI Flash)	1	$2.00	$2.00
Display	4.0” TFT LCD w/ Capacitive Touch (SPI, ribbon cable)	1	$45.00	$45.00
USB HUB	USB2512B-I/M2	1	$2.38	$2.38
Electrical Engineer				
Speaker	12mm Piezo Buzzer / Beeper	1	$2.50	$2.50
LED Status Indicator	RGB LED (3mm/5mm) + resistor	2	$0.50	$1.00
Pushbutton Switch	SPST tactile button	2	$0.25	$0.50
Power Supply	5V 3A USB-C Adapter	1	$10.00	$10.00
Stacked USB	72309-7043BLF	1	$2.68	$2.68
Ethernet	1-1734264-1	1	$1.03	$1.03
Micro SD Card Reader	MEM2067-02-180-00-A	1	$1.06	$1.06
Miscellaneous	Wiring, headers, mounting hardware	–	$25.00	$25.00
Total Cost				
Computer Engineer		$177.38		
Electrical Engineer		$43.77		
Grand Total		**$221.15**		

As shown in the table, the project’s total cost is well within its budget. This leaves a surplus of $221.15 from our $1000 goal, providing a healthy margin for prototyping and replacement parts.

9.3 Senior Design Milestones

Table 9.3.1 Senior Design Milestones

Milestone Number	*Milestone Description*	*Start Date*	*Anticipated End Date*	*Dependencies and Requirements*
1	Project Idea Finalization	08/18/25	09/09/25	Advisory Approval (Sahawneh)
2	Finalizing Design Choices	08/18/25	09/12/25	Confirming Design Requirements and Researching EE and CpE Compatibility
3	Divide and Conquer Document Finalization	08/28/25	09/06/25	Obtaining Professor Sahawneh's Meeting Availability
4	Divide and Conquer Meeting	09/09/25 12:30-1:00	09/09/25	Complete Divide and Conquer Documentation / Review Committee Form
5	Committee Formation / Meeting	08/28/25	09/05/25	Email Professors for Committee Availability
6	Midterm Report Draft	09/30/25	10/14/25	Communicate Deadlines with Team and Split Micro-Objectives
7	Midterm Report Draft Revision	10/14/25	10/31/25	Reference Document Guidelines and Revise
8	Final Report	10/20/25	11/06/25	Review and Approval of Team Parameters
9	Mini Demo Video	10/27/25	11/06/25	Reference Hardware Milestones for Part Integration

Table 9.3.2 Senior Design Hardware Milestones

Hardware Milestone	*Start Date*	*Expected Acquisition Date*	*Dependencies and Requirements*
ESP32-S3 Development Board Bring-Up	08/18/25	08/25/25	Development board delivery, programming environment setup
Connect ESP32-S3 to TFT Display	08/25/25	09/05/25	TFT display delivery, SPI wiring, verify graphics output
Interface External SPI Flash with ESP32	09/05/25	09/12/25	ESP32 SPI bus functional, connect 16MB Winbond flash
Breadboard Initial Prototype	09/12/25	09/20/25	ESP32 + display + flash tested, wiring diagram completed
Establish Communication with RPi CM4	09/20/25	09/30/25	Raspberry Pi CM4 environment prepared, wired connection to ESP32
Finalize PCB Design	10/01/25	10/10/25	Breadboard prototype validated, schematic/layout reviewed
Order PCB and Supporting Components	10/11/25	10/15/25	PCB design finalized, BOM approved
Hardware/Software Integration Testing	TBD SD2	TBD SD2	N/A
Final Prototype Assembly	TBD SD2	TBD SD2	N/A

10 Appendices

10.1 Appendix A - Citations

References

[1] G. Elbaz, "Code execution vulnerability: Impact, causes, and 8 defensive measures," https://www.oligo.security/academy/code-execution-vulnerability-impact-causes-and-8-defensive-measures, 2025, accessed: 2025-09-03.

[2] IBM X-Force, "Ibm x-force 2025 threat intelligence index," https://www.ibm.com/thought-leadership/institute-business-value/en-us/report/2025-threat-intelligence-index, Apr. 2025, accessed: 2025-09-03.

[3] M. Vanhoef and F. Piessens, "Key reinstallation attacks: Forcing nonce reuse in WPA2," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. Dallas, TX, USA: ACM, 2017, pp. 1313–1328.

[4] BonelliSystems, "Automated penetration testing," Bonelli Systems Website, 2025, https://bonellisystems.com/automated-penetration-testing/.

[5] O. Team, "Kali Linux on a Raspberry Pi (A/B+/2) with Disk Encryption — offsec.com," https://www.offsec.com/blog/raspberry-pi-luks-disk-encryption/?utm$_source = chatgpt.com, [Accessed 12 - 09 - 2025]$.

[6] Hak5, "Lan turtle: Covert systems administration and penetration testing tool," Hak5 Website, 2025, https://shop.hak5.org/products/lan-turtle.

[7] N. Adamou, "Minipwner github repository," GitHub Repository, 2020, https://github.com/nicholasadamou/minipwner.

[8] SecPoint, "Penetrator: Vulnerability scanning appliance," SecPoint Website, 2025, https://www.secpoint.com/penetrator.html.

[9] W. McGrew, "Pwn the pwn plug: Analyzing and counter-attacking attacker-implanted devices," DEF CON 21 Presentation, 2013, https://www.defcon.org/images/defcon-21/dc-21-presentations/McGrew/DEFCON-21-McGrew-Pwn-The-Pwn-Plug.pdf.

[10] Hak5, "WiFi Pineapple — shop.hak5.org," https://shop.hak5.org/products/wifi-pineapple, [Accessed 12-09-2025].

10.2 Appendix B - Copyright Permissions

10.3 Appendix C - etc.