

IOT DEVICE SECURITY: A PRACTICAL APPROACH TO HARDWARE VULNERABILITY ANALYSIS

Author: Aman Sharma

Year: 2025

This document is a public technical version of my academic project report. Institutional identifiers and private data have been removed for privacy reasons.

“All tests were conducted on personally owned devices for academic research and responsible disclosure purposes.”

ABSTRACT

The rapid growth of IoT has led to widespread security vulnerabilities due to weak firmware protection and flawed cryptographic implementations. This project analyzes the security of TP-Link TL-WR845N and D-Link DWR-116 routers through hardware testing, firmware extraction, and reverse engineering.

Hardware identification and serial interface communication were conducted to analyze the devices, followed by firmware extraction and vulnerability assessment. The TP-Link TL-WR845N firmware contained a hardcoded Data Encryption Standard key, enabling attackers to decrypt, modify, and re-encrypt it, leading to potential backdoor installation. The D-Link DWR-116 stored root credentials with weak password hashing, making it vulnerable to credential theft and privilege escalation.

This research highlights how insecure firmware compromises network security and underscores the need for stronger cryptographic protections, secure credential storage, and improved security measures in IoT devices. It contributes to IoT security by demonstrating exploitation techniques, proof-of-concept attacks, and mitigation strategies.

TABLE OF CONTENTS

| Chapter No. | Title | Page No. |
|-------------|--|----------|
| | ABSTRACT | vi |
| | TABLE OF CONTENTS | vii |
| | LIST OF FIGURES | x |
| | LIST OF TABLES | xi |
| | LIST OF ACRONYMS | xii |
| 1 | INTRODUCTION | 1 |
| | 1.1 Introduction | 1 |
| | 1.2 Background | 1 |
| | 1.3 Motivation | 2 |
| | 1.4 Problem Statement | 2 |
| | 1.5 Objectives of the Study | 2 |
| | 1.6 Methodology Overview | 3 |
| | 1.7 Scope of the Project | 4 |
| | 1.8 Contributions of the Research | 4 |
| | 1.9 Organization of the Thesis | 5 |
| 2 | LITERATURE SURVEY | 6 |
| | 2.1 Introduction | 6 |
| | 2.2 Review of Relevant Works | 6 |
| | 2.3 Summary | 8 |
| 3 | PROJECT DESCRIPTION | 9 |
| | 3.1 Hardware and software requirements | 9 |

| | | |
|----------|--|-----------|
| | 3.1.1 Hardware Tools | 9 |
| | 3.1.2 Software Tools | 9 |
| | 3.1.3 Specific Device Setup | 9 |
| | 3.2 Modules | 9 |
| | 3.2.1 Hardware Identification | 10 |
| | 3.2.2 Serial Interface Communication | 10 |
| | 3.2.3 Firmware Extraction | 10 |
| | 3.2.4 Reverse Engineering Firmware | 12 |
| 4 | EXPERIMENTAL SETUP | 14 |
| | 4.1 Hardware Setup | 14 |
| | 4.2 Hardware Interfaces Analyzed | 18 |
| | 4.3 Software Tools Used | 20 |
| 5 | EXPERIMENTATION | 21 |
| | 5.1 Device Selection and Setup | 21 |
| | 5.2 UART Interface Identification | 22 |
| | 5.3 Baud Rate Detection using Logic Analyzer | 22 |
| | 5.4 Firmware Extraction using Flashrom | 23 |
| | 5.5 Firmware Reverse Engineering | 23 |
| | 5.6 Sensitive Data and Vulnerability Discovery | 24 |
| | 5.7 Summary of Experimentation | 24 |
| 6 | RESEARCH FINDINGS | 25 |
| | 6.1 TP-Link TL-WR845N | 25 |
| | 6.1.1 UART Serial Interface Analysis | 25 |
| | 6.1.2 Security Vulnerabilities | 26 |

| | | |
|----------|---|-----------|
| | 6.2 D-Link DWR-116 | 30 |
| | 6.2.1 UART Serial Interface Analysis | 30 |
| | 6.2.2 Security Vulnerabilities | 31 |
| 7 | RESULTS AND DISCUSSION | 32 |
| | 7.1 Result | 32 |
| | 7.2 Mitigation Strategies & Recommendations | 32 |
| | 7.3 Summary of Findings | 33 |
| 8 | CONCLUSION | 34 |
| | 8.1 Conclusion | 34 |
| | 8.2 Contributions of the Study | 34 |
| | 8.3 Future Research Directions | 36 |
| | REFERENCES | 37 |
| | APPENDIX | 43 |

LIST OF FIGURES

| Figure No. | Figure Name | Page No. |
|------------|--|----------|
| 4.1 | TP-Link TL-WR845N UART Interface | 18 |
| 4.2 | TP-Link TL-WR845N SPI Flash | 18 |
| 4.3 | D-Link DWR-116 UART Interface | 19 |
| 4.4 | D-Link DWR-116 SPI Flash | 19 |
| 6.1 | TP-Link TL-WR845N - UART Connection I | 25 |
| 6.2 | TP-Link TL-WR845N - UART Connection II | 26 |
| 6.3 | TP-Link TL-WR845N - Extracted Weak Password Hashes | 27 |
| 6.4 | TP-Link TL-WR845N - Cracked Password Hash | 27 |
| 6.5 | TP-Link TL-WR845N: dm_decryptFile() Function | 28 |
| 6.6 | TP-Link TL-WR845N - Hardcoded DES key | 29 |
| 6.7 | TP-Link TL-WR845N - decrypted config File | 29 |
| 6.8 | D-Link DWR-116 - UART Connection I | 30 |
| 6.9 | D-Link DWR-116 - UART Connection II | 30 |
| 6.10 | D-Link DWR-116 - Extracted weak Hashes | 31 |

LIST OF TABLES

| Table No. | Table Name | Page No. |
|------------------|--|-----------------|
| 4.1 | TP-Link TL-WR845N - General Specifications | 14 |
| 4.2 | TP-Link TL-WR845N - Chipset and Hardware Details | 15 |
| 4.3 | TP-Link TL-WR845N - Firmware Details | 15 |
| 4.4 | D-Link DWR-116 - General Specifications | 16 |
| 4.5 | D-Link DWR-116 - Chipset and Hardware Details | 17 |
| 4.6 | D-Link DWR-116 - Firmware Details | 17 |

LIST OF ACRONYMS

| | |
|------|---|
| C2 | Command and Control |
| CVE | Common Vulnerabilities and Exposures |
| DES | Data Encryption Standard |
| GND | Ground |
| IoT | Internet of Things |
| OS | Operating System |
| RCE | Remote Code Execution |
| RX | Receive |
| SoC | System on Chip |
| SPI | Serial Peripheral Interface |
| SSH | Secure Shell |
| TX | Transmit |
| UART | Universal Asynchronous Receiver-Transmitter |

Chapter 1

INTRODUCTION

1.1 Introduction

The expanding implementation of Internet of Things (IoT) devices has introduced significant security challenges, especially in consumer networking equipment such as routers. These devices become an important entry point for networking communications. Out there are many of them with varying vulnerabilities, such as weak cryptographic algorithms, hard-coded usernames and passwords, and firmware security mechanisms that just are not up to the mark. Exploiting such vulnerabilities leads to critical severity attacks, ranging from unauthorized access to network compromise and beyond. This entire project includes security assessment of two very cheap and consumer accessible routers - TP-Link TL-WR845N and D-Link DWR-116 - identity of the hardware, serial interface communication, firmware extraction, and reverse engineering, before identifying exfiltration avenues. Analyzing the firmware as well as the hardware designs of such devices would reveal latent security flaws, which an attacker may exploit to elevate his privilege for various types of attacks such as unauthorized access or privilege escalation.

1.2 Background

In modern times, IoT is an indispensable reality, with almost every device in a typical home operating on the internet. Routers serve as the primary agent of connectivity between home or enterprise networks and the internet. They are the entry point that allows internet traffic access to the network, thus making it a potentially rich target for all sorts of hacker.

Unfortunately, the product being offered to consumers is poorly matched in security configuration because nearly all manufacturers build routers with low-cost and performance as the priority. Hence, most of these routers are left extremely unprotected from any or most cyber-attacks. Such manufacturing cost-saving measures give rise to firmware with hardcoded credentials, being filled with outmoded encryption algorithms and unfit access control mechanisms, allowing for very large attack surfaces for an attacker. The absence of necessary security practices such as firmware

integrity checks, reliance on obsolete cryptographic standards, and improper credential management are some of the common flaws that exposes routers to serious risks

1.3 Motivation

The growing dependence on IoT devices demands the strong need for stronger security practices in their deployment and development. Routers, being important to network infrastructure, are essential in preserving secure communication.

Given the widespread vulnerabilities found in consumer grade routers, there is a pressing requirement to investigate the security posture of these devices. This research is motivated by the goal of conducting an in-depth security analysis on two widely used consumer routers, TP-Link TL-WR845N and D-Link DWR-116. The objective is to find potential vulnerabilities and provide mitigation strategies that can help improve the security posture of IoT ecosystems.

1.4 Problem Statement

Many consumer routers suffer from weak firmware security due to improper cryptographic implementations and insecure storage of sensitive data. Attackers can exploit these vulnerabilities to:

- Gain unauthorized access to the device.
- Modify firmware to install backdoors or malicious payloads.
- Extract credentials stored in an insecure format.

This study aims to analyze the TP-Link TL-WR845N and D-Link DWR-116 routers, focusing on firmware security, cryptographic weaknesses, and hardware-level vulnerabilities

1.5 Objectives of the Study

This study is focused on assessing the security posture and security vulnerabilities of consumer routers through:

1. **Hardware Security Testing** which includes Identifying and analyzing hardware components to find possible attack vectors.

2. **Serial Interface Communication** which includes Establishing communication with the router via UART to access bootloader options.
3. **Firmware Extraction** which includes Extracting firmware off the flash rom through various techniques, such as SPI flashing and UART communication.
4. **Reverse Engineering** which includes Analyzing the firmware to detect hardcoded credentials, weak cryptographic implementations, and insecure authentication mechanisms.
5. **Security Risk Evaluation** which includes Identifying the impact of discovered vulnerabilities and suggesting mitigation strategies.

1.6 Methodology Overview

The security evaluation of TP-Link TL-WR845N and D-Link DWR-116 follows a systematic approach:

1. Hardware Identification & Serial Interface Communication

- Identifying hardware components, such as SPI flash chips and UART interfaces.
- Establishing communication with the router via UART serial connection interface to interact with the bootloader of the router.

2. Firmware Extraction

- Dumping the firmware from flash memory using SPI programming tools.
- Extracting firmware using UART-based techniques for analysis.

3. Reverse Engineering & Security Analysis

- **Static analysis** which includes examining extracted firmware files for hardcoded credentials, cryptographic keys, and hidden backdoors.
- **Dynamic analysis** which includes running the firmware in an emulated environment to test for vulnerabilities.

- **Cryptographic assessment** which includes checking for weak encryption schemes, such as outdated hashing algorithms.

4. Vulnerability Identification & Exploitation

- Searching for hardcoded credentials, outdated authentication mechanisms, and improper encryption practices.
- Analyzing how these vulnerabilities can be exploited in real-world attack scenarios.

5. Mitigation Strategies & Recommendations

- Proposing firmware security best practices to rectify ified weaknesses.
- Suggesting improvements in credential storage and cryptographic implementations.

1.7 Scope of the Project

This project is focused on the security assessment of consumer grade routers functionality.

The scope includes:

- Analyzing firmware security flaws in consumer-grade routers.
- Identifying weak cryptographic implementations in authentication and data storage.
- Examining hardware communication interfaces for potential exploitation.
- Providing security recommendations for router manufacturers.

1.8 Contributions of the Research

This research makes significant contributions to the field of IoT security by:

- Identifying a hardcoded DES encryption key in the TP-Link TL-WR845N, which allows attackers to decrypt, modify, and re-encrypt firmware, enabling potential backdoor installations.
- Uncovering weak password hashing mechanisms (MD5-based \$1\$ and NTLM) in the D-Link DWR-116, making root credentials susceptible to cracking and privilege escalation attacks.

- Demonstrating practical firmware extraction and reverse engineering techniques for identifying security flaws in IoT devices.
- Providing security recommendations to mitigate firmware vulnerabilities and enhance IoT security.

These findings emphasize the importance of firmware security in consumer routers and highlight the need for stronger cryptographic protections to prevent cyberattacks on IoT devices.

1.9 Organization of the Thesis

The structure of this thesis is designed to provide a clear and logical progression of the research work. **Chapter 2** presents a detailed literature survey, reviewing previous research on embedded firmware and hardware security, and summarizing the progression of relevant techniques and tools.

Chapter 3 outlines the project description, listing the hardware and software tools used, and explaining the core modules implemented for device identification, serial communication, firmware extraction, and reverse engineering. **Chapter 4** discusses the experimental setup, detailing both the hardware configurations and the software environments utilized.

Chapter 5 details the experimentation and analysis that was done on the devices and outline the command and tools that were used. **Chapter 6** focuses on the practical findings from the analysis of two IoT devices—TP-Link TL-WR845N and D-Link DWR-116—highlighting UART interface interactions and discovered vulnerabilities.

Chapter 7 presents the results, recommended mitigation strategies, and a summary of the key findings. Finally, **Chapter 8** concludes the thesis with a discussion on outcomes, contributions of the study, and future research.

Chapter 2

LITERATURE SURVEY

2.1 Introduction

This literature survey reviews research works and contributing factors for embedded systems and firmware security reverse engineering, vulnerability detection, and hardware hacking techniques. The studies give insight into the present challenges and solutions regarding IoT and embedded device security.

2.2 Review of Relevant Works

The field of security for embedded systems as well as firmware analysis has gained ground in recent times. The pioneering efforts by Andrei Costin [1] and Cui et al. [2] brought early vulnerabilities with firmware and offered foundational concepts in embedded exploitation. The paper had been laid from foundation to further research into firmware manipulation, attack vectors, and reverse engineering practices. Complementing this, Fyrbiak and Strauß [3] gave quite a detailed survey on hardware reverse engineering pointing out the technical and ethical issues with the manipulation of hardware-software interactions in embedded systems.

The knowledge of the physical access interfaces is still a requirement for device-level compromise. Mohieldin S. [4] offered a practical introduction to JTAG-a primary point of exploitation and debugging-framing it within a more general technique of hardware security research. Vasile et al. [5] added a classification of the methods of firmware extraction from IoT devices, in invasive and non-invasive ways. Similarly, Hou et al. [6] studied new techniques for hybrid vulnerability detection by static and dynamic analysis, improving the confidence on the security assessments of embedded firmware.

We have moved a step ahead down the path of automated and organized detection of vulnerabilities. It was David et al. [7] who drew up the plan for FirmUp-a static detection tool directed toward localized reasoning for recurring firmware vulnerabilities. An analysis framework

about IoT smart cameras was revealed by Alharbi and Aspinall [8], demonstrating how those ignored configurations on consumer hardware would potentially set a stage for severe vulnerabilities associated with this technology. Product vulnerabilities were cited: reverse engineering and backdooring of router firmware illustrate to what extent the shoddy firmware defenses can lend.

From the academic and systematic view, Chandy [10] pushed hardware hacking to be included in the cybersecurity curriculum for secure hardware design practices. Ahmid and Kazar [11] did a full study of IoT Security threats and suggested some strategic countermeasures as further support to this. Similarly, Lee and Lee [12] conducted a systematic mapping report that looks into newly emerging firmware-level vulnerabilities and turns IoT security-related research toward embedded environments.

Nadir et al. [13] introduced a classification in 2022 to organize the burgeoning field and group concrete firmware vulnerabilities and analysis methods for neophytes as well as researchers to grasp easily. Hemram et al. [14] also enlarged the framework by presenting some detection mechanisms in both embedded and IoT environments. Works by Cao et al. [17] and Huang & Sheng [22] focused particularly hard on certain hardware interfaces, such as USART, which are the most common interfaces used for debugging and extracting data.

Research on hardware has been, in 2023, repositioned as one of the major security frontiers. A paper in IJNRD [18] analyzed hardware hacking aspects and risk quite bleakly, reiterating its importance in vulnerability finding. Your Puschner and Paar [19] equally proposed that effort take place toward development of system security and logic security evaluation in tandem, fostering a comprehensive look into the security architectures for IoT devices.

AI has gotten into the firmware security domain, just like anything else. Ye et al. [21] used macro-linguistic models to conduct taint analysis and improve the detection of command injection flaws in Linux-based firmware. This showed the new frontier of using machine learning in security automation. Further research work by Qiu [23] extended the use of reverse engineering techniques to automotive ECUs, affirming that firmware security is diverse across domains.

Kuzminykh et al. [20] discussed the relation between encryption and performance impact on devices-evaluation of algorithm choices in IoT systems and their influence on energy consumption and life span. Hossain et al. [24] highlighted this complex affliction in their review of IoT-edge convergence, as they suggested an all-inclusive approach to security in interconnected environments. Finally, Zhou et al. [25] revisited firmware emulation as a highly scalable method for fuzz testing and described it as a vital instrument for high-throughput repeatability in vulnerability assessment across modern IoT firmware ecosystems.

2.3 Summary

It further discusses the present-day trends in automation and artificial intelligence methods for detection on the fertile grounds of evolution embedded system security research and firmware analysis from past vulnerability disclosures. The first pioneering efforts into publicizing vulnerabilities in embedded firmware and methodologies for reverse engineering were produced through the basic research of pioneers like Andrei Costin and Cui et al. This work initiated further studies on Fyrbiak and Strauß, who exegited about the hardware-software interaction and ethical problems relating to reverse engineering.

This foundational work was reinforced with practical knowledge on hardware interfaces like JTAG and UART, which are critical in device-level compromise. Holieldin S. and Cao et al. illustrated how these interfaces afford entry for debugging, interception of data, and low-level manipulations on a device. On the other hand, the move towards automation and accuracy in firmware testing is typified by tools like FirmUp and hybrid vulnerability analysis models by Hou et al.

The up-and-coming trends portend quite a bit in terms of growing specific uses in environment security and AI-endowed detection methods. Some key missions that really symbolize the clear forward movement of the research impact aimed at scaling vulnerability discovery are Ye et al.'s application of LLM to taint analysis and Zhou et al. on firmware emulation for fuzzing. Together, these efforts underscore a collective movement toward building secure, transparent, and resilient embedded systems that can withstand evolving threat landscapes.

Chapter 3

PROJECT DESCRIPTION

3.1 Hardware and software requirements

3.1.1 Hardware Tools

- Logic analyzers for signal capture.
- UART debuggers for accessing debug ports.
- A Digital Multimeter.
- CH341a SPI programmer.

3.1.2 Software Tools

- Ghidra for reverse engineering firmware.
- Binwalk for extracting the various partitions from a data file.
- dd tool is used to carve specific parts of a file.
- Flashrom is used for firmware extraction.
- Openssl to decrypt the encrypted files.
- Picocom to interact with the UART interface.

3.1.3 Specific Device Setup

- TP-Link TL-WR845N router for practical testing.
- D-Link router DWR-116 router for practical testing.

3.2 Modules

The following modules outline the end-to-end workflow used to analyze and assess the routers' security:

3.2.1 Hardware Identification

This module involves the physical inspection of the router's printed circuit board (PCB) to identify key components:

- Identifying flash memory chips (typically SPI NOR Flash).
- Locating UART (Universal Asynchronous Receiver/Transmitter) ports used for serial communication.
- Identifying main microcontroller or SoC (System on Chip) which manages router functions.

By understanding the board layout and components, the groundwork is laid for low-level interaction and firmware access.

3.2.2 Serial Interface Communication

After identifying the UART ports, serial communication was established using USB-to-TTL converters:

- Connection was made via standard 3.3V UART (TX, RX, GND) pins.
- Baud rate and port configuration were adjusted to match the router's default UART settings.
- Access to the U-Boot bootloader or Linux shell was achieved, allowing for interaction with the firmware in a pre-boot or diagnostic mode.

This provided insights into the boot process and enabled command-line access to explore or extract system files.

3.2.3 Firmware Extraction

The extraction of the firmware from the physical IoT device is a critical step in security analysis. It allows for inspection and testing without any restrictions imposed by the vendor of the IoT devices which in this case are routers. This stage required a combination of hardware interfacing and specialized tools to access the device's memory and extract its firmware image.

• **UART-Based Dumping**

One of the primary methods used for firmware extraction is **Universal Asynchronous Receiver/Transmitter (UART)**. UART provides a serial communication channel often exposed on debugging headers within the device. The process included:

- **Identifying UART Pins**

- Using tools such as a multimeter, logic analyzer, or automatic UART detectors, the TX (transmit), RX (receive), and GND (ground) pins were located on the device's circuit board.

- **Accessing Bootloader or Shell**

- After connecting the UART interface to a USB-to-serial adapter, the device was powered on and communication was established using Picocom. In this instance, interrupting the bootloader allowed access to a limited shell.

- **Memory Dumping via Bootloader Commands**

- Commands like md, dump, or cat /dev/mtdX were used to read memory sections (e.g., flash partitions) and send their contents to a host machine over the serial connection.

- **Challenges**

- Some devices had disabled UART interfaces or locked bootloaders, requiring fallback to lower-level methods.

• **SPI Flash Programming**

When UART access was restricted meaning the extraction of any form of data is not possible, **Serial Peripheral Interface (SPI) flash programming** is used as an alternative. This method involved direct access to the flash memory chip on the device's PCB:

- **Locating the Flash Chip**

- Flash memory chips were identified on the PCB using chip markings.

- **Interfacing Tools**

- Hardware programmer like the **CH341A** was used to interface directly with the SPI flash.
- **Firmware Dumping**
 - Tools such as **Flashrom** were used to read the memory contents from the flash chip and store them as binary firmware images.

This extraction process laid the foundation for reverse engineering, allowing a detailed inspection of the device's architecture and potential security flaws.

3.2.4 Reverse Engineering Firmware

The goal of this module is to analyze firmware obtained from routers to unveil possible vulnerabilities that are at the disposal of malicious actors. Static, dynamic, and cryptographic analyses are the three pillars of a firm, and the combination of all three provides the most comprehensive view of firmware security.

- **Static Analysis**

Static analysis was performed to analyse the firmware without executing it. This involved unpacking the firmware using tools such as **Binwalk**, **Firmware Mod Kit**, and **Ghidra**. The main objective was to analyze the internal file systems and binary executables for common security misconfigurations and vulnerabilities.

The following key areas were examined:

- **Hardcoded Credentials**
 - Credentials embedded directly into configuration files or binaries were identified. These could be exploited for unauthorized access if the device is deployed with default settings.
- **Configuration Files**
 - Files like `/etc/passwd`, startup scripts, and web server configuration files were inspected for improper access controls and unsafe defaults.

• **Dynamic Analysis**

The dynamic analysis of the firmware has been carried out over emulation on the specific device's environment to observe real-time behavior of firmware. The emulation tools QEMU was used here as environment hardware and software emulations for the device. This would have enabled to:

- **Authentication Testing:** Login mechanisms, especially through web interfaces and telnet/SSH logins, would undergo tests for weaknesses such as credential bypasses or logic flaws.
- **Service Interaction:** Network services such ssh and http server are running in the router while interacting with them at runtime while analysis has been done to reveal vulnerabilities.
- **Log and Runtime Behavior Monitoring:** The information from kernel output, daemon behaviors, and system logs are captured, logged, and analyzed for insecure operations which static analysis could not uncover.

• **Cryptographic Analysis**

On the other hand, the strengths and implementations of the cryptographic algorithms present in the firmware were examined. The analysis revealed certain disturbing practices:

- **Weak Hash Functions**
 - A review revealed many instances where user passwords were stored with weak and insecure algorithms such as MD5 (\$1\$) and NTLM, which are known to be collision and brute-force attack risks.
- **Hardcoded Cryptographic Keys**
 - Some binaries contain hardcoded DES keys or Base64-encoded credentials, further diminishing the security of sensitive data.
- **Use of Insecure Protocols**
 - Some services have been configured to use obsolete SSL/TLS versions and, in some cases, even plaintext channels, thus exposing sensitive information to interception.

Chapter 4

EXPERIMENTAL SETUP

4.1 Hardware Setup

In-depth hardware and firmware specifics of the two IoT devices of the TP-Link TL-WR845N and D-Link DWR-116 project have been discussed here. The technical insights into these devices will help outline the possible access points required for hardware interfacing and firmware extraction.

- TP-Link TL-WR845N: General specifications, chipset, and firmware details.
- **General Specifications:**

Table 4.1: TP-Link TL-WR845N - General Specifications

| Specification | Details |
|------------------------|---------------------------------------|
| Model | TP-Link TL-WR845N |
| Type | Wireless Router |
| Wi-Fi Standards | IEEE 802.11b/g/n (2.4 GHz only) |
| Frequency Band | 2.4 GHz |
| Wireless Speed | Up to 300 Mbps |
| LAN Ports | 4 × 10/100 Mbps LAN Ports |
| WAN Ports | 1 × 10/100 Mbps WAN Port |
| Antenna | 3 × Fixed 5dBi Antennas |
| ‘Button(s) | WPS/Reset Button, Power On/Off Button |
| Power Supply | 9V DC / 0.6A |

- **Chipset and Hardware Details:**

Without knowledge of the internal hardware, it would be extremely difficult to perform firmware extraction and reverse engineering activities. The table below explains some of the most important hardware components of a router, including CPU, memory, and switch chip:

Table 4.2: TP-Link TL-WR845N - Chipset and Hardware Details

| Component | Details |
|--------------|--|
| CPU/SoC | MediaTek MT7628N |
| RAM | 64 MB DDR1 |
| Flash Memory | 8 MB SPI Flash (e.g., MXIC MX25L6406E) |
| Switch Chip | Integrated in MT7628N SoC |

- **Firmware Details:**

TP-Link provides proprietary firmware for its router, which can be configured via a web-based GUI website with default credentials. The following are templates for the firmware format and bootloader type:

Table 4.3: TP-Link TL-WR845N - Firmware Details

| Detail | Information |
|--------------------|---|
| Firmware Type | Proprietary TP-Link firmware |
| Firmware Interface | Web-based GUI (accessible at 192.168.0.1) |
| Login Default | Username: admin, Password: admin |
| Firmware Format | .bin file (when upgrading manually) |
| Bootloader | U-Boot |

- D-Link DWR-116: Hardware specifications and firmware version details.

D-Link DWR-116 is a multi-WAN wireless router that has WAN and LAN ports and can support USB-based 3G/4G dongles.

The router is meant for providing redundancy to the network and has moderate throughput, which means that it can satisfy the performance requirement for some basic networking jobs.

- **General Specifications:**

The general specifications of this device are as follows: -

Table 4.4: D-Link DWR-116 - General Specifications

| Specification | Details |
|------------------------|---|
| Model | D-Link DWR-116 |
| Type | Wireless N300 Multi-WAN Router |
| Wi-Fi Standards | IEEE 802.11b/g/n (2.4 GHz) |
| Frequency Band | 2.4 GHz |
| Wireless Speed | Up to 300 Mbps |
| LAN Ports | 4 × 10/100 Mbps LAN Ports |
| WAN Ports | 1 × 10/100 Mbps WAN Port |
| USB Ports | 1 × USB 2.0 (for connecting a 3G/4G LTE dongle) |
| Antenna | 2 × External Fixed Antennas |
| Buttons | WPS Button, Reset Button, Power Button |
| Power Supply | 12V / 1A |

- **Chipset and Hardware Details:**

The internal hardware of the D-Link DWR-116 is built around low-power network devices with a Ralink SoC. Specifics of the DWR-116 chipset and memory configuration are listed in the specifications table below:

Table 4.5: D-Link DWR-116 - Chipset and Hardware Details

| Component | Details |
|----------------|---------------------------------------|
| CPU/SoC | Ralink RT3352 (MIPS 24Kc, 400 MHz) |
| RAM | 32 MB DDR2 |
| Flash Memory | 8 MB SPI NOR Flash |
| USB Controller | Integrated in SoC |
| Switch Chip | Integrated (no dedicated switch chip) |

- **Firmware Details:**

Firmware is proprietary, as with most TP-Link devices, and the end-user interface is browser-based. The functional aspects of the firmware and default credentials are presented below for accessing:

Table 4.6: D-Link DWR-116 - Firmware Details

| Detail | Information |
|--------------------|---|
| Firmware Type | D-Link proprietary firmware |
| Firmware Interface | Web-based GUI (default: 192.168.0.1) |
| Login Default | Username: admin, Password: <i>blank</i> |
| Firmware Format | .bin file |
| Bootloader | U-Boot |

4.2 Hardware Interfaces Analyzed

Evaluation of the hardware interfaces, which include the UART and the Flash ROM, is achieved.

UART is hardware communication protocol for sending as well as receiving data throughout asynchronous serial communication between two devices. Simply stated, the system receives a parallel data from a microcontroller and converts that into the serial form

Flash ROM is non-volatile memory that stores data during the off state of the power. It is a common storage where firmware, bootloaders, and other configuration data are kept in embedded systems and networking devices, such as routers.

- TP-Link TL-WR845N

- UART



Figure 4.1 : TP-Link TL-WR845N UART Interface

- SPI Flash



Figure 4.2 : TP-Link TL-WR845N SPI Flash

- D-Link DWR-116
- UART

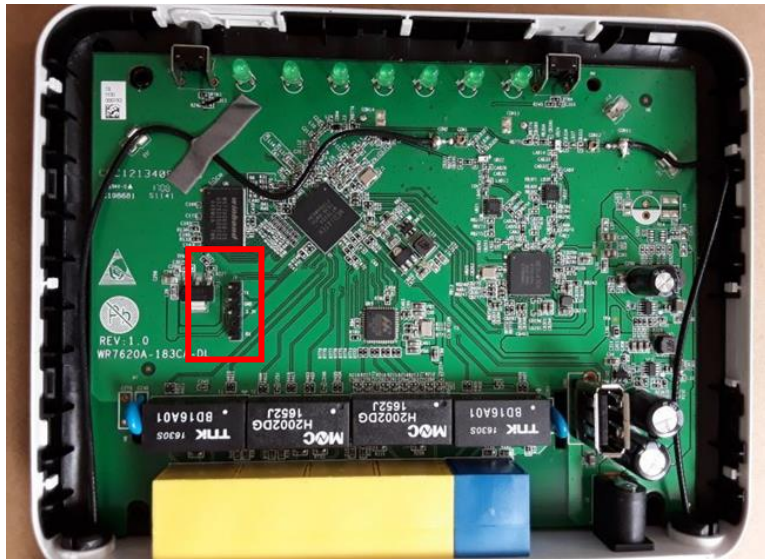


Figure 4.3 : D-Link DWR-116 UART Interface

- SPI Flash

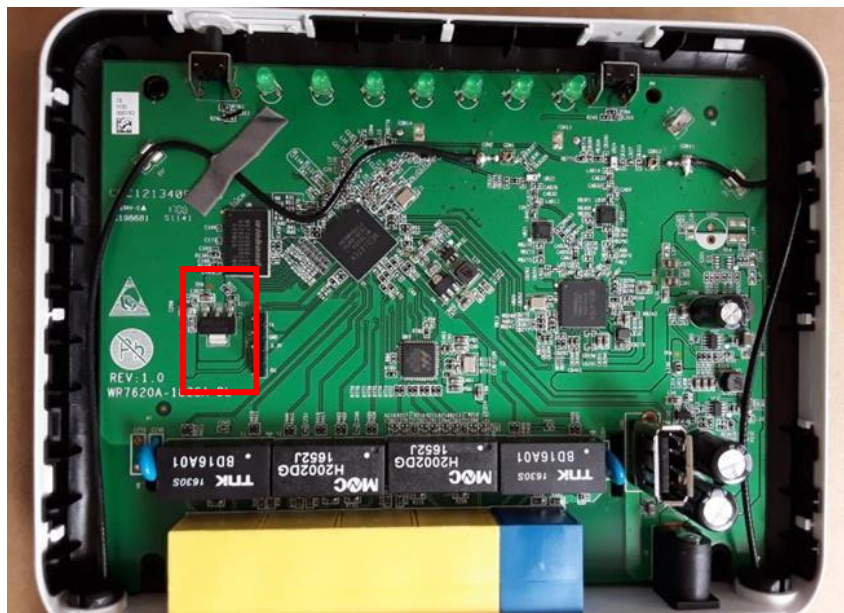


Figure 4.4 : D-Link DWR-116 SPI Flash

4.3 Software Tools Used

○ Firmware Extraction & Analysis

- **Binwalk** – A firmware analysis tool used for extracting and identifying embedded files and file systems within firmware images. It plays a critical role in understanding the firmware's structure and locating important data such as file systems and executables.
- **dd & Strings** – dd is used for low-level copying and extraction of data from binary dumps, while strings are employed to locate readable plaintext information like hardcoded credentials and paths within binaries.
- **Flashrom** – A command-line utility used to detect, read, write, and verify flash ROM chips. It is essential to safely dumping firmware contents from the router's flash memory to prevent corruption of the extracted data.
- **Screen** – A terminal emulator that connects to UART interface and makes it possible to communicate with the board. It was used to monitor the boot process and interact with the router's console during live debugging and testing.

○ Reverse Engineering & Debugging

- **Ghidra** – this is a tool developed by NSA for the analysis of low-level code. This was used to disassemble binaries used in the firmware of the routers to identify program logic, function calls and hardcoded data.
- **PulseView** – this is a graphical utility that can capture digital data and visualize them. It was used to visualize the UART transmissions and to calculate baud rate of the router.

○ Cryptography & Password Analysis

- **Hashcat** – Hashcat is a password brute forcer utility that was used to crack or decode the extracted hashed credentials from the firmware.
- **OpenSSL** – It was used to decrypt the encrypted configuration files that were used by the firmware of the router and to detect the usage of outdated algorithms

Chapter 5

EXPERIMENTATION

This chapter lists the technical procedures that were undertaken for security analysis on both routers, TP-Link TL-WR845N and D-Link DWR-116. Following the step-by-step methodology followed for security analysis, possible vulnerabilities were identified. The entire project was aimed at determining the security profile of both devices as observed through practical experiment and analysis.

The setup process included identification of the UART interface via hardware identification, establishment of a serial connection, and the final measurement of its baud rate via logic analyzer. Successful communication then led to firmware extraction using Flashrom. Reversing and analyzing the extracted firmware revealed some hidden information like hard-coded credentials and possible configuration mistakes.

Thereafter, the following sections provide a detailed account of the processes for each stage.

5.1 Device Selection and Setup

The TP-Link TL-WR845N and D-Link DWR-116 were the routers chosen for this study since they are both extensively used consumer-grade devices that are well-liked in homes and small offices due to their affordable pricing. These models were selected for practical IoT security study due to their affordability, accessibility, and public interest.

Each device was disassembled to analyse the internal printed circuit board. Precautions such as use of an anti-static wrist strap and handling the board in a grounded environment such as on a silicon mat were taken to prevent electrostatic damage. Tools such as a precision screwdriver set, magnifying lens, multimeter, and soldering tools, which enabled inspection and connectivity testing in a detailed fashion.

5.2 UART Interface Identification

UART (Universal Asynchronous Receiver/Transmitter) is a serial communication protocol often exposed on embedded boards of IoT devices for debugging and development purposes. Identifying UART access points is important for obtaining console level access to the router's operating system.

The identification began with inspection of the circuit board to locate header pins or test pads. Using a multimeter, voltage levels were measured to distinguish between power pin, ground pin (GND), transmit pin (TX), and receive pin (RX) pins.

Once UART pins were identified and confirmed, a USB to UART adapter, CP2102 module, was used to interface with the board of the host machine.

Various baud rates were tested using serial terminal application Minicom to detect valid communication. However, to determine the correct baud rate, a logic analyzer was used in the next stage.

5.3 Baud Rate Detection using Logic Analyzer

A **logic analyzer** is used to capture and analyze serial communication signals from the identified UART pin headers. This was done to precisely calculate the baud rate.

The logic analyzer was connected to the TX pin and ground, and the boot sequence was triggered by powering on the device. Captured waveforms were analyzed using software like PulseView. By measuring the duration of signal pulses and bit intervals, the baud rate was estimated. In both router models, baud rates such as **115200** were observed.

Once the correct baud rate was identified, it enabled successful access to the router's serial console, often providing low-level system logs and sometimes even shell access.

5.4 Firmware Extraction using Flashrom

With UART access established, the next phase involved extracting the router firmware directly from the onboard SPI flash memory chip using the **Flashrom** utility. This approach provides a raw binary dump of the firmware for detailed offline analysis.

First, the flash chip was identified based on chip markings, and datasheets were consulted for pinout confirmation. A SOIC-8 test clip was used to make a non-invasive connection to the flash memory. The clip was wired to a CH341A programmer and connected to the host system running Linux.

Using the Flashrom tool, the following command was used:

```
flashrom -p ch341a_spi -r router_firmware.bin
```

Verification was performed to ensure the firmware dump was valid and not corrupted, using hash comparison and examining entropy to confirm structure of the dumped firmware.

5.5 Firmware Reverse Engineering

The extracted firmware image was subjected to reverse engineering using a combination of static analysis tools. The first step involved scanning the firmware using **Binwalk**, which helped identify embedded file systems such as **SquashFS** and **JFFS2**.

The file systems were extracted using:

```
binwalk -eM router_firmware.bin
```

from the extracted root file system, sensitive files such as `/etc/passwd`, `/etc/shadow`, and binaries were identified. Tools like `strings`, `hexdump`, and **Ghidra** were used to perform further static analysis on ELF binaries, identify function calls, API calls and logical vulnerabilities in executable code

Sensitive data and files such as hardcoded credentials, API keys, debug modes, and misconfigured implementation of functionalities were found during this analysis.

5.6 Sensitive Data and Vulnerability Discovery

During the reverse engineering phase, several critical level security issues were discovered:

- **Hardcoded credentials** in configuration and binary files
- Presence of **unencrypted private keys**
- Open **Telnet/SSH access** via UART without authentication
- Debugging scripts and test interfaces left in production firmware

These findings indicate a significant lack of security practices. Such issues can be exploited by attackers in multiple ways to gain unauthorized access, escalate privileges, or inject malicious firmware into the device.

Findings were cross referenced with various public vulnerability databases such as CVE and Exploit-DB to identify whether similar vulnerabilities had been reported or exploited previously.

5.7 Summary of Experimentation

The experimentation process followed a consistent methodology for both router models, enabling detailed analysis of hardware level and firmware level security.

The table below summarizes key observations:

Table 5.1 : Summary of Experimentation

| Step | Tools Used | Findings (Both Routers) |
|---------------------|--------------------|--|
| UART Identification | Multimeter, CP2102 | UART console available, accessible interface |
| Baud Rate Detection | Logic Analyzer | 115200 baud |
| Firmware Dump | Flashrom, CH341A | Complete firmware extracted |
| Reverse Engineering | Binwalk, Ghidra | Sensitive data, hardcoded credentials found |

Chapter 6

RESEARCH FINDINGS

This section details the vulnerabilities discovered in the TP-Link TL-WR845N and D-Link DWR-116 routers through firmware analysis. The analysis uncovered weak cryptographic practices, hardcoded secrets, and insecure credential storage mechanisms. Screenshots and tool outputs are included to support each finding.

6.1 TP-Link TL-WR845N

6.1.1 UART Serial Interface Analysis

Upon connecting to the UART interface, the following results were obtained.

```
picocom /dev/ttyUSB0 --b 115200
picocom v3.1
port is          : /dev/ttyUSB0
flowcontrol      : none
baudrate is      : 115200
parity is        : none
databits are     : 8
stopbits are     : 1
escape is        : C-a
local echo is    : no
noinit is        : no
noreset is       : no
hangup is        : no
nolock is        : no
send_cmd is      : sz -vv
receive_cmd is   : rz -vv -E
imap is          :
omap is          :
emap is          : crcrlf,delbs,
logfile is       : none
initstring       : none
exit after is    : not set
exit is          : no
Type [C-a] [C-h] to see available commands
Terminal ready
...
```

Figure 6.1 : TP-Link TL-WR845N - UART Connection I

```

[04030805][8485000E][70700000][25253939][80252544]
Setting Cal Done
UBoot 1.1.3 (Dec 14 2020 18:36:31)
board: Ralink APSOC DRAM : 32 MB
relocate code Pointer at: Bifcese
Lash manufacture id: 1c, device id 70 16
Warning: un-recognized chip ID, please update bootloader!
=====
Ralink UBoot version: 4.3.0.
-----
ASIC 7628 MP (Ports<->None)
BRAM Component: 256 bits DC, width 16
BRAM bus: 16 bit
Total memory: 32 MBytes
Flash component: SPI Flash
Date:Dec 14 2020 Time:18:36:31
=====
cache: sets:512, ways:4, linesz:32,total:65536
cache: sets:256, ways:4, linesz:32 total:32768
=====The CPU freq 580 MKZ B estimate memory size 32 Mbytes
RESET MT7628 PHY!!!!!!
continue to starting system.
Disable switch phyptert...
1: System Boot system code via Flash. (@xbc010000)
Elo_boots:argc 2, addr=@xbc@10000
Booting image at bc010000...
Uncompressing Kernel Image ... CK
No initrd
Transferring control to Linux (at address 8000c150) ...
Giving linux sessize in D, 32
Starting kernel...
INUX started...
THIS IS ASIC
Linux version 2.6.36 (jenkins@mobile-System) (gcc version 4.6.3
(Buildroot 2012.11.1)) #1 Mon Dec 14 18:39:18 CST 2020
...

```

Figure 6.2 : TP-Link TL-WR845N - UART Connection II

6.1.2 Security Vulnerabilities

Weak Password Hashing Algorithms (MD5-based \$1\$, NTLM)

- **Observation:** Upon extracting the firmware and inspecting files such as /etc/passwd and configuration backups, it was discovered that TP-Link uses outdated hashing algorithms to store user passwords, specifically MD5-based \$1\$ hashes and NTLM-style hashes.

- **Tools Used:** binwalk, hashcat, john the ripper
- **Impact:** These hash types are considered cryptographically weak. They can be cracked in a short time using modern GPU-based password cracking tools.
- **Demonstration:**

```
admin:$1$$ic.dUsGpxNNJGeOm1dFio/:0:0:root:/:/bin/sh/
dropbear:x:500: 500: dropbear:/var/dropbear:/bin/sh
nobody:*:0:0: nobody:/:/bin/sh
```

Figure 6.3 : TP-Link TL-WR845N - Extracted Weak Password Hashes

```
root@kali:~# hashcat -a 0 -m 500 hash
/usr/share/wordlists/rockyou.txt

$1$$iC.dUsGpxNNJGeOm1dFio/:1234

Session.....: hashcat
Status.....: Cracked
Hash.Name.....: md5crypt, MD5 (Unix), Cisco-IOS $1$
(MD5)
Hash.Target.....: $1$$iC.dUsGpxNNJGeOm1dFio/
Time.Started.....: Wed Oct 28 14:10:47 2020 (1 sec)
Time.Estimated...: Wed Oct 28 14:10:48 2020 (0 secs)
Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 3279 H/s (11.38ms) @ Accel:256
Loops:125 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests
Progress.....: 3072/14344385 (0.02%)
Rejected.....: 0/3072 (0.00%)
Restore.Point....: 0/14344385 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:875-1000
Candidates.#1...: 123456 -> dangerous
```

Figure 6.4 : TP-Link TL-WR845N - Cracked Password Hash

- **Severity:** High – Enables offline password recovery attacks and potential unauthorized administrative access.

Hardcoded DES Key Exploit

- **Observation:** Through static analysis of the firmware's binary files using strings and Ghidra, a hardcoded DES encryption key was found embedded directly in the source code used for encryption/decryption routines.
- **Location:** The DES key was identified within a binary in /bin/ or /etc/ and was used for encrypting configuration data, including admin credentials.
- **Impact:** This key allows any attacker who extracts the firmware to decrypt sensitive information without needing to brute-force anything. It effectively nullifies any confidentiality measures.
- **Demonstration:**

```
int dm_decryptFile(uint param_1,undefined4 param_2,uint
param_3,int param_4)
{
int iVar1;
undefined auStack_28 [8];
int local_20;
memcpy(auStack_28,&DAT_000c8270,8);
if (param_3<param_1){
}
cdbg_printf(8,"dm_decrypt File", 0xbcf,
"Buffer exceeded, decrypt buf size is %u, but dm file size is
%u",param_3,param_1); local_20 = 0;
else {
local_20 = cen des
MinDo(param_2,param_1,param_4,param_3,auStack_28,0);
iVar1 = local_20;
if (local_20 == 0) {
}
cdbg_printf(8,"dm_decrypt File", 0xbd6,"DES decrypt error\n");
else { do {
local_20 = iVar1;
if (((undefined *) (param_4 + local_20))[-1] != '\0') break;
iVar1 = local_20 +-1;
} while (local_20 != 0);
*(undefined *) (param_4 + local_20) = 0;
}}
return local_20;
```

Figure 6.5: TP-Link TL-WR845N: dm_decryptFile() Function

DAT_000c8270

| | | | | |
|----------|-----------|----|-----|---|
| 000c8270 | 47 | ?? | 47h | G |
| 000c8271 | 8d | ?? | 8Dh | |
| 000c8272 | a5 | ?? | A5h | |
| 000c8273 | 0f | ?? | 0Fh | |
| 000c8274 | f9 | ?? | F9h | |
| 000c8275 | e3 | ?? | E3h | |
| 000c8276 | d2 | ?? | D2h | |
| 000c8277 | cb | ?? | CBh | |

Figure 6.6: TP-Link TL-WR845N - Hardcoded DES key

The encryption key is “**478da50ff9e3d2cd**”.

Using the extracted key, the config files can be extracted.

```
<ManagementServer t-o r=P s=MANAGEMENT_SERVER_OBJ p=18>
  <EnableCWMP t=b r=W d=0 />
  <URL t s r=Wl-256/>
  <Username t=sr=W I=256 d=admin />
  <Password t=s r=W I-256 d=admin tp=1 />
  <PeriodicInformEnable t-b r=W d=0/>
  <PeriodicInformInterval tour-Wi-1 d=300/>
  <PeriodicInformTime t=d r=W/>
  <ParameterKey t-s r-R l-32 da-1 />
  <X TP ConnRegPort t-us r-W d=7547 h=1 />
  <X TP connRegPath t=s r=Wl=16 d=/tr069 h=1 />
  <ConnectionRequestURL t=s r=R I=64 fa=1 />
<ConnectionRequestUsername t=s r=W I-256 d=admin/>
<ConnectionRequestPassword t-s r=W I-256 d=admin tp=1 />
<UpgradesManaged t=b r=W />
<KickURL tes r=R I=64 />
<X TP Flag t=u r=W d=1 h=1 />
<X TP BoundIfName t-s r=Wl-16 h=1/>
</ManagementServer>
...
```

Figure 6.7: TP-Link TL-WR845N - decrypted config File

- **Severity:** High – Facilitates decryption of credentials and configuration files without authentication.

6.2 D-Link DWR-116

6.2.1 UART Serial Interface Analysis

Upon connecting to the UART interface, the following results were obtained.

```
picocom v1.7
port is       : /dev/ttyUSBO
flowcontrol   : none
baudrate is   : 115200
parity is     : none
databits are  : 8
escape is     : C-a
local echo is : no
noinit is     : no
noreset is    : no
nolock is     : no
send_cmd is   : Sz -VV
receive_cmd is : rz -vv
imap is       :
omap is       :
emap is       : crcrlf,delbs,
Terminal ready
...
```

Figure 6.8: D-Link DWR-116 - UART Connection I

```
...
init wlan, wsc_enable change to 3 ==>sys_reg_upnp_dev
<<<< start_wps_service (state=0)>>> [WR]rtl8192cd_set_wps_enable
1251:<===wps_start Done
Init WAN MAC address: EC:1A:59:3F:12:BD
set wan promisc mode disable
Init bridge...
rom_gut_pack_init GUI_PKG_INDEX_ADRS=bd1fffe8
rom_gut_pack_init g_pack_start_adr s=d5c12
rom_gui_pack_init g_pack_total_size=79396
compressFileHead content:
000000000000000000000000 7 93 960 78 0 1 0 0
rom_gut_pack_init g_g_gui_pack_head.len= 496534
rom_gut_pack_init g_gui_pack_head.page_num=78
rom_gut_pack_init g_gui_pack_head.flag=1
Init success!
Success to launch watchdog with 60 seconds expire time and
action id as 1
...
```

Figure 6.9: D-Link DWR-116 - UART Connection II

6.2.2 Security Vulnerabilities

Weak Password Hashing Algorithms (MD5-based \$1\$, NTLM)

- **Observation:** Like the TP-Link router, the D-Link firmware was also found to be using weak password hashing mechanisms (MD5 \$1\$, NTLM) for administrative credentials stored in the system.
- **Tools Used:** binwalk, firmware-mod-kit, hashcat
- **Impact:** Attackers can perform offline cracking of these hashes to retrieve plaintext passwords. This makes the device susceptible to unauthorized access, especially if the same password is reused across networks.
- **Demonstration**

```
/etc/passwd
root:$1$N76hdwGfg11g0KdKbtyh21:0:0:root:/root:/bin/ash
nobody:$1$qRPK7m23GJusamGpoGLby/:99:99:nobody:/var/usb:/sbin/nologin
ftp:$1$qRPK7m23GJusamGpoGLby/:14:50:FTP
USER:/var/usb:/sbin/nologin

/etc/smbpasswd
root:0:F0D412BD764FFE81AAD3B435B51404EE:209C6174DA490CAEB422F3FA
5A7AE634: [U]: LCT-00000000: root
nobody:99:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXX: [u]:LCT-00000000:
```

Figure 6.10: D-Link DWR-116 - Extracted weak Hashes

Using Hashcat, the recovered credentials:

- /etc/passwd → **root:amittima**
- /etc/smbpasswd → **root:admin**
- **Severity:** High – Compromises password integrity and allows credential recovery.

Chapter 7

RESULTS AND DISCUSSION

7.1 Result

A practical analysis of the TP-Link TL-WR845N router and D-Link DWR-116 provided useful insights into hardware-level vulnerabilities and insecure firmware implementations. The two devices were successfully wired together via UART serial communication for firmware extraction and analysis. In the case of TP-Link, UART was used to access the bootloader such that it could perform an interaction with the internal operations of the device. An analysis of the firmware revealed hardcoded credentials and a lack of encryption of important configuration details. The D-Link router presented similar UART access; reverse engineering weakened memory protections and showed that it was running kernel versions that were outdated and full of known vulnerabilities.

Through reverse engineering, an elaborate understanding of the firmware architecture was achieved, unearthing multiple unsafe configurations and exploitable services alive and running on both devices. Results further proved the efficacy of the open-source tools and manual hardware probing in discovering vulnerabilities in embedded systems.

7.2 Mitigation Strategies & Recommendations

This study has uncovered several vulnerabilities, and preventative security measures should be established as a remedy to these vulnerabilities. This section elaborates on well-laid-down practical methods for ensuring the security of embedded IoT devices. These methods are intended to limit even the most possible attack surfaces while achieving better resilience of the whole firmware.

- **Secure Firmware Development Practices**

Manufacturers should implement modular design of firmware to rid themselves of outdated hard-coded credentials and will carry out periodic code reviews. Moreover, the firmware must be signed to prevent unauthorized modification or modification at a later state.

- **Cryptographic Improvements**

Outdated hashing algorithms such as MD5 or NTLM must give way to newly secured algorithms such as bcrypt, scrypt, or Argon2, among others. The crypto goes beyond having salts for passwords. Keeping sensitive data must be done with specific secure storage mechanisms.

- **Hardware-Level Security Enhancements**

The router should implement secure boot using TPM or hardware-based keys and restrict physical access. Furthermore, best practice recommends disabling used debug ports, such as UART/JTAG

- **User-Level Security Recommendations**

Changing default passwords, regularly updating firmware, disabling unused services, and applying WPA2 or WPA3 for encryption. Another layer of security may be proud of network segmentation and actively monitoring router activity and traffic.

7.3 Summary of Findings

Severe vulnerabilities related to security in the firmware of TP-Link TL-WR845N routers were thoroughly discovered during this investigation. It relied on very old password hashing methods which could easily be broken like MD5 (\$1\$) and NTLM. There existed embedded DES cryptographic keys into the firmware which makes this vulnerability horribly catastrophic since DES cannot be said to be safe anymore.

These threats mean very important risks like:

- **Unauthorized access** to the router's admin interface,
- **Credential leakage** through weak hash cracking,
- **Firmware tampering**, where attackers could modify and reflash the device.

Such vulnerabilities highlight the urgent need for secure firmware development practices and regular security reviews in consumer grade hardware.

Chapter 8

CONCLUSION

8.1 Conclusion

Scads of Embedded Systems vulnerability that exist in IoT devices were demonstrated through hands-on learning involving practical initiatives to detect hardware and firmware vulnerabilities. The survey on two consumer routers TP-Link TL-WR845N and D-Link DWR-116 opened a Pandora's box of issues that arose from improper security measures, weak implementation of security protocols, and absence of tamper-resistant design.

The approach integrates interacting with the device hardware through UART (Universal Asynchronous Receiver/Transmitter) in addition to firmware reverse engineering. UART interfaces enable such communication with devices, thus acquiring bootable access login and attendant system logs, which would make firmware dumping and analysis possible. Analyzing the logs gives sensitive information like unprotected files, encrypted files, and tripped validation processes.

Among many, this approach serves well to allow reproducibility of results, thereby addressing a practical avenue of follow-up studies. In addition, the study validated the hypothesis that low-level software systems in cybersecurity pay little attention to interfaces between hardware and firmware. This study also showed glaring omissions in security provisions that should be there in public-use devices.

Embedded systems security is viewed in the larger context of network and consumer security with the combination of theoretical and practical approaches. This case exemplifies the need for the interaction of cybersecurity, software, hardware, and human factors to be considered at different levels. In view of the vulnerabilities presented by IoT systems, therefore, there is an urgent need for a reassessment of these systems along modern lines in view of increasing cyber warfare directed at smart and industrial systems.

8.2 Contributions of the Study

The contributions of this study are both practical and educational. It connects a critical gap between academic research and real-world security analysis by demonstrating how vulnerabilities can be identified without needing expensive tools or proprietary software access.

The study's contributions can be summarized as follows:

- **Exposure of Critical Vulnerabilities**
 - The research uncovered various practical flaws in two widely used routers, including unauthenticated UART access, exposure of system configurations, and improper access controls in firmware.
- **Demonstration of Methodology**
 - A repeatable workflow was established starting from hardware teardown and UART identification to firmware extraction, analysis, and vulnerability documentation. This workflow can be adapted for analyzing other embedded devices.
- **Security Awareness Contribution**
 - By publicly documenting the risks in popular consumer devices, the project contributed to increasing awareness for the importance of embedded device security.
- **Promotion of Secure Design Principles**
 - The study highlights the importance of s secure boot processes, encrypted communication protocols, and hardened firmware designs and deployments in mitigating hardware-level attacks.
- **Support for Research and Education**
 - The project also serves as a valuable teaching resource for students, cybersecurity professionals, highlighting the real-world relevance of concepts like UART exploitation, memory analysis, and firmware reverse engineering.

8.3 Future Research Directions

The project lays the foundation for more comprehensive research in the field of IoT hardware security and provides various promising directions for future exploration:

- **Dynamic Testing in Live Environments:** While this study primarily focused on static analysis and UART-based communication and a little dynamic analysis through emulation, future research can incorporate dynamic testing in live network environments. This would allow for the identification of runtime vulnerabilities, buffer overflows, and insecure network behaviors under different usage conditions.
- **Expanded Device Coverage:** An important next step would be to scale the analysis across a wider spectrum of devices including smart cameras, smart plugs, wearable devices, and industrial control systems. This broader coverage would help validate the scalability of the methodology and uncover common design flaws across manufacturers.
- **AI/ML-Based Automation:** With the rapid growth of IoT devices, manual vulnerability analysis becomes increasingly infeasible. Future research can leverage machine learning models for automated firmware classification, anomaly detection, and threat prediction. Techniques such as taint analysis, natural language processing of logs/config files, and pattern recognition in binaries could significantly enhance the speed and depth of assessments.
- **Development of Secure Firmware Deployment Models:** As vulnerabilities are discovered and patched, another area of research could be the design and deployment of secure firmware update mechanisms. This includes cryptographic validation, rollback prevention, and user-notified OTA (Over-the-Air) updates.
- **User Behavior and Policy Analysis:** A comprehensive security solution must also consider the end-user. Research can be extended to explore how user behavior affects device security—such as failing to update firmware or using default credentials. Coupling technical hardening with user education and behavior modeling can offer a holistic approach to IoT security.

REFERENCES

- [1] Andrei Costin, “Embedded Devices Security and Firmware Reverse Engineering”, ResearchGate, 2013.
- [2] Cui, Ang & Costello, Michael & Stolfo, Salvatore, “When Firmware Modifications Attack: A Case Study of Embedded Exploitation”, 2013.
- [3] Marc Fyrbiak, Sebastian Strauß, “Hardware Reverse Engineering: Overview and Open Challenges”, Proc. of the IEEE 2nd International Verification and Security Workshop (IVSW), Greece, 2017.
- [4] Mohieldin S, “Hardware Hacking 101: Introduction to JTAG”, River Loop Security, May, 2017.
- [5] Sebastian Vasile, David Oswald, and Tom Chothia, “Breaking all the Things - A Systematic Survey of Firmware Extraction Techniques for IoT Devices⁶”, University of Birmingham, 2017.
- [6] Jin-bing Hou, Tong Li, Cheng Chang, “Research for Vulnerability Detection of Embedded System Firmware”, Procedia Computer Science, Volume 107, 2017.
- [7] Y.David, Nimrod Partush and Eran Yahav, “FirmUp: Precise Static Detection of Common Vulnerabilities in Firmware”, Proc. of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating System, United States of America, June 2018.
- [8] Alharbi, R & Aspinall, “An IoT analysis framework: An investigation of IoT smart Cameras’ vulnerabilities”, Proc. of the Living in the Internet of Things: Cybersecurity of the IoT, United Kingdom, 2018.
- [9] A. Adithyan, K. Nagendran, R. Chethana, G. Pandey D. and G. Prashanth K, "Reverse Engineering and Backdooring Router Firmwares", Proc. of the 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2020.

- [10] John A. Chandy, "Hardware Hacking: An Approach to Trustable Computing Systems Security Education", Proc. of The Colloquium for Information System Security Education (CISSE) 19th Annual Conference, Las Vegas, United States of America, June 2021.
- [11] Maroua Ahmid, Okba Kazar," A Comprehensive Review of the Internet of Things Security", Journal of Applied Security Research, Aug 2021.
- [12] Jee Young Lee, Jungwoo Lee, "Current Research Trends in IoT Security: A Systematic Mapping Study", Mobile Information Systems, March 2021.
- [13] Akash C. Koturwar, Dr. V. K. Pachghare, Sharad Hange, " A Practical approach for Firmware Reverse Engineering", International Journal of Advance Research, Ideas and Innovations in Technology, 2021.
- [14] Zahra, Samman & Gong, Wei & Khattak, Hasan Ali & Shah, Munam & Song, Houbing. "Cross-Domain Security and Interoperability in Internet of Things", IEEE Internet of Things Journal, 2021.
- [15] Nadir, Ibrahim & Mahmood, Haroon & Shah, Ghalib, "A taxonomy of IoT firmware security and principal firmware analysis techniques", International Journal of Critical Infrastructure Protection, 2022.
- [16] S. Hemram, G. J. W. Kathrine, G. M. Palmer and S. E. V. Ewards, "Firmware Vulnerability Detection in Embedded Systems and Internet of Things", Proc. of the International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), Trichy, India, 2022.
- [17] L. Cao, J. Chen and J. Li, "Working principle and application analysis of UART", Proc. of the IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), Changchun, China, 2023.
- [18] Pranav Vardhan,"Hardware Hacking: An Approach to Evaluate the Importance of Hardware Hacking and its Associated Risks", IJNRD - INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT, October-2023.


- [19] Endres Puschner, Christof Paar, “Security Analysis of IoT Devices: From the system level to the logic level”, IEEE Solid-State Circuits Magazine, January 2023.
- [20] Kuzminykh, Ievgeniia & Yevdokymenko, Maryna & Sokolov, Volodymyr, “Encryption Algorithms in IoT: Security vs Lifetime”, SSRN Electronic Journal, 2023.
- [21] Junjian Ye, Xincheng Fei, Xavier de Carné de Carnavalet, Lianying Zhao, Lifa Wu, “Detecting command injection vulnerabilities in Linux-based embedded firmware with LLM-based taint analysis of library functions”, Computers & Security, 2024.
- [22] W. Huang and G. Sheng, "Analysis and Research on UART Communication Protocol", Proc. of the 2024 4th Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS), Shenyang, China, 2024.
- [23] Yuhao Qiu "An improved method for reverse engineering ECU firmware", Proc. of the SPIE 13175, International Conference on Computer Network Security and Software Engineering, China, June 2024.
- [24] Mahmud Hossain ,Golam Kayas ,Ragib Hasan ,Anthony Skjellum , “A Holistic Analysis of Internet of Things (IoT) Security: Principles, Practices, and New Perspectives”, Proc. of the Cyber Security in the New "Edge Computing + IoT" World, 2023.
- [25] Zhou, Wei & Shen, Shandian & Liu, Peng, “IoT Firmware Emulation and Its Security Application in Fuzzing: A Critical Revisit”, Future Internet, 2025.

APPENDIX

1. D-Link Security Advisory – DWR-116 Vulnerability Report – SAP10424

Search by product, keyword, model.

HomeSupportForumsSecurity AdvisoriesShopEnglish | French

Security Announcement

Announcement > SAP10424

(non-US) DWR-116 : H/W Rev. A1/A2/A3 : F/W 1.00 : End-of-Life (EOL) / End-of-Service (EOL) : Vulnerability Report

Publication ID: SAP10424

Resolved Status: Yes

Published on: 12 March 2025 12:17 GMT

Last updated on: 12 March 2025 2:39 GMT

Overview

The (non-US product) DWR-116 : All Models, Derivative Models, Hardware Revision A1/A2/A3, and All Firmware reached their End-of-Life (EOL)/ End-of-Service Life (EOS) lifecycle. D-Link Corporation and D-Link North America (D-Link Systems, Inc.) recommend that all current users take one or more of the following actions:

1. Transition to a current-generation product.
2. Perform comprehensive data backup.
3. Contact our local regional office for further recommendations or information ([LINK](#)).

By standard industry practice, products that have reached EOL/EOS status may no longer receive technical support or firmware updates. Please read the detailed information and recommendations provided below.

3rd Party Report information:

Reported: (Non-US) DWR-116 H/W: 03/10/2025:: Affiliation: Cybersecurity Lab, Department of Cybersecurity, SRM Institute of Science and Technology, Ramapuram, Chennai

Author: Aman Sharma – amansharma_dot_amsh0208_at_gmail_dot_com
Sathya Priya S – sathyapriya80_at_gmail_dot_com
Preethi D – mail_dot_dpreethi_at_gmail_dot_com

Details: Firmware D-link_DWR-116-V1.00 - the presence of hardcoded root credentials in the /etc/passwd file and /etc/smbpasswd file within the extracted firmware

Affected Models

| Model | Region | Hardware Revision | End of Support | Legacy Website | Last Updated |
|---------|--------|-----------------------------|----------------|---------------------|--------------|
| DWR-116 | Non-US | All Series A1 H/W Revisions | 09/05/2017 | No (Non-US Product) | 03/11/2025 |
| | | All Series A2 H/W | | No (Non-US | |

| | | | | | |
|---------|--------|-----------------------------|------------|----------------------|------------|
| DWR-116 | Non-US | Revisions | 07/22/2019 | Product) | 03/11/2025 |
| DWR-116 | Non-US | All Series A3 H/W Revisions | 09/05/2017 | No (Non-US Product)) | 03/11/2025 |

Recommendation for EOL /EOS Products

In line with industry practice, D-Link may periodically determine that certain products have reached a stage where further support or development is no longer attainable. This decision may be driven by commonly acknowledged factors such as technology evolution, market requirement, innovation, product efficiency, or the need for product replacement due to superior functionality.

For US Consumers

When a product reaches EOL/EOS status, which we have always announced in advance for an extended period, no further extended support, updates, or development may be available.

We may be unable to address devices or firmware issues for such products, as development and customer support may have been discontinued. If you are outside the US, please contact your regional D-Link office for an inquiry.

We recommend discontinuing such products and caution that continued use may harm other connected devices. If users continue using these devices, please update them to the latest known firmware on the Legacy Website links above. Additionally, users should frequently update a device's unique password to access its web configuration and always have Wi-Fi encryption enabled with a strong and unique password.

Regarding the Security Update for Your Device

Installing firmware updates is critical in addressing security vulnerabilities in your devices. We strongly urge all users to install the relevant updates and regularly check for further updates. After downloading the firmware update, it is essential to ALWAYS validate its success by comparing the firmware version on your product interface to the firmware update version.

Please note that beta software, beta firmware, or hot-fix release is still undergoing rigorous testing before its official release. This ensures it is of the highest quality and meets our stringent standards. Due to such nature, we do not provide express or implied warranties regarding its suitability or usability. It is essential to understand that the user assumes all risk and liability for its use.

NOTE: Our products have different hardware revisions, so please check your device's hardware revision before downloading the corresponding firmware update. The hardware revision can be found on the product label next to the serial number or on the device's web interface.

[GPL Source Code](#)

[End of Product Life-Cycle](#)

[Terms of Use](#)

[Privacy](#)

[Do Not Sell My Info – CA Residents Only](#)

[Contact Us](#)