engineering proceedings

MDPI

Proceeding Paper

Auditable Security Assessment of Proprietary and Open-Source Wi-Fi Router Firmware: A Reproducible Approach for Academic Infrastructures †

Leonardo de Paiva Souza [1,*], Robson de Oliveira Albuquerque [2], Luis Javier García Villalba [3], Fábio Lúcio Lopes Mendonça [2] and Georges Daniel Amvame Nze [2]

[1] Professional Graduate Program in Electrical Engineering (PPEE), Department of Electrical Engineering, Faculty of Technology, University of Brasília (UnB), Brasília 70910-900, Brazil
[2] Faculty of Technology, University of Brasília (UnB), Brasília 70910-900, Brazil; robson.albuquerque@unb.br (R.d.O.A.); fabio.mendonca@unb.br (F.L.L.M.); georges@unb.br (G.D.A.N.)
[3] Group of Analysis, Security and Systems (GASS), Universidad Complutense de Madrid (UCM), 28040 Madrid, Spain; javiergv@fdi.ucm.es
* Correspondence: leonardops@unb.br; Tel.: +55-61-982688857
† Presented at the First Summer School on Artificial Intelligence in Cybersecurity, Cancun, Mexico, 3–7 November 2025.

Abstract

Wi-Fi router security is a real concern for universities and research centers that rely on strong, dependable networks for everything they do. In this study, we took a close look at four popular Wi-Fi router firmwares using open-source tools such as Binwalk, CVE-Bin-Tool, and Semgrep. We carefully examined the file systems, cross-referenced them with the National Vulnerability Database (NVD), and searched for outdated software like BusyBox and OpenSSL. What we found was clear: proprietary firmwares had more Critical and High vulnerabilities, while OpenWrt stood out for being more secure, easier to update, and openly maintained by its community. Our reproducible process automates how we gather evidence and map vulnerabilities, making firmware auditing more practical and trustworthy. These results make a strong case for using open-source firmware as a safer, more manageable choice for institutional networks.

Keywords: busybox; cve; embedded linux; firmware analysis; network hardening; openssl; openwrt; static analysis; wi-fi security; auditability; reproducibility

1. Introduction

Reliable Wi-Fi is the backbone of academic and research environments, especially in engineering departments. Labs, automation setups, and sensor networks all depend on strong wireless connections. But when router firmware is outdated or not properly configured, it puts these networks at real risk, such as opening the door to unauthorized users and making it easier for data to be compromised. At the University of Brasília's Department of Electrical Engineering, for example, these kinds of security flaws are an everyday concern for both labs and administrative offices. In this study, we took a practical, step-by-step approach to analyzing four popular Wi-Fi router firmwares, from TP-Link Archer C7 (US) V5.0, D-Link DIR-842 R1, Intelbras RG1200, to OpenWrt 22.03.5, using open-source tools (Binwalk, Semgrep, and CVE-Bin-Tool) and public databases like the NVD. We looked for old software, misconfigurations, and risky services. Our aim was

Academic Editors: Gabriel Sánchez Pérez and Ana Lucila Sandoval Orozco

Published: 12 February 2026

simple: count the known vulnerabilities, compare the risks between proprietary and open-source firmware, and show how openness and transparency can make networks safer. The method we used is practical and can be adapted for use in both academia and industry.

2. Related Work

Firmware security has gained prominence as the number of IoT devices, such as routers and controllers, has grown rapidly. ENISA's Threat Landscape for IoT [1] emphasizes that most network vulnerabilities stem from outdated or unsupported firmware, highlighting the need for structured auditing approaches. Recent frameworks have advanced large-scale automated analysis. Costin et al. [2] pioneered the static analysis of embedded firmware, revealing recurring flaws such as hardcoded credentials and obsolete libraries. Zhao et al. [3] introduced UVScan for detecting third-party component misuse, while Feng et al. [4] proposed HERMESCAN, enhancing CVE correlation in Linux-based systems. Li et al. [5] developed AutoFirm for large-scale library reuse detection, and Balgavy and Muench [6] presented FIRMLINE, an extensible pipeline for non-Linux firmware. Complementary works such as Karonte [7] and FirmAE [8] explored emulation and multi-binary analysis. Unlike these large-scale frameworks, which depend on complex infrastructures or partial source access, this study adopts a practical, transparent, and fully reproducible methodology tailored to academic Wi-Fi router firmware. By employing open-source and auditable automation (collect_evidence.sh), it bridges the gap between theoretical research and institutional security validation.

3. Methodology

This section describes the methodological framework adopted to identify and classify vulnerabilities in Wi-Fi router firmware, comparing proprietary and open-source images. The analysis was conducted exclusively through static methods, following the OWASP Firmware Security Guidelines and the NIST SP 800-115 and SP 800-160 standards [9,10]. The process was designed to ensure reproducibility, auditability, and alignment with cybersecurity best practices for embedded systems in academic environments.

3.1. Overview

A static analysis approach was employed, combining reverse engineering and CVE correlation using the National Vulnerability Database. The reproducible workflow (Figure 1) was divided into five stages: (1) Environment setup; (2) Firmware acquisition; (3) Static inspection; (4) Vulnerability classification; and (5) Mitigation proposals.

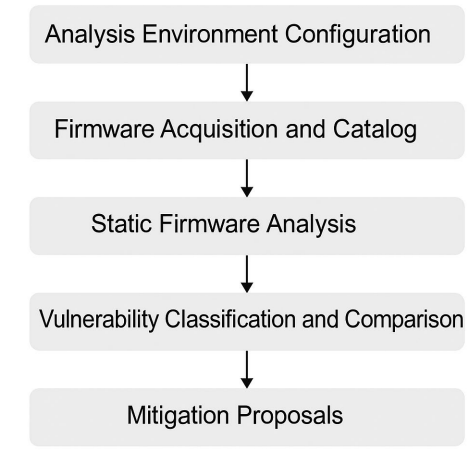

Figure 1. General workflow of the methodology applied in the static analysis of Wi-Fi router firmware.

3.2. Analysis Environment Configuration

All experiments were performed on a Kali Linux 2025.2 virtual machine (4 vCPUs, 8 GB RAM), ensuring a controlled and repeatable environment. The analysis pipeline combined Binwalk for firmware extraction, Semgrep for detection of insecure code patterns, and CVE-Bin-Tool for mapping known vulnerabilities to the NVD. Four firmware images, from TP-Link Archer C7 (US) v5.0, D-Link DIR-842 R1, Intelbras RG1200, to OpenWrt 22.03.5, were obtained from official vendor repositories and verified via MD5 and SHA-256 checksums. File systems were unpacked and inspected to identify misconfigurations, embedded credentials, outdated libraries, and unsafe initialization scripts. The CVE-Bin-Tool classified all findings under the CVSS v3.1 severity levels: Critical, High, Medium, Low, and Unknown. To standardize and automate the process, an auditable automation script (collect_evidence.sh) was developed. This script orchestrates extraction, scanning, and artifact collection, generating logs and reports to ensure full reproducibility of the analysis workflow.

3.3. Vulnerability Classification and Comparison

Vulnerabilities were categorized as Critical, High, Medium, Low, or Unknown according to CVSS v3.1. Proprietary firmwares showed higher concentrations of severe issues, while OpenWrt presented mostly low- or unknown-severity cases, indicating a smaller attack surface (Figures 2 and 3). Mitigation measures include disabling insecure services, enforcing encrypted access (SSH/HTTPS), adopting modular open-source firmware, applying VLAN segmentation, and enabling continuous monitoring.

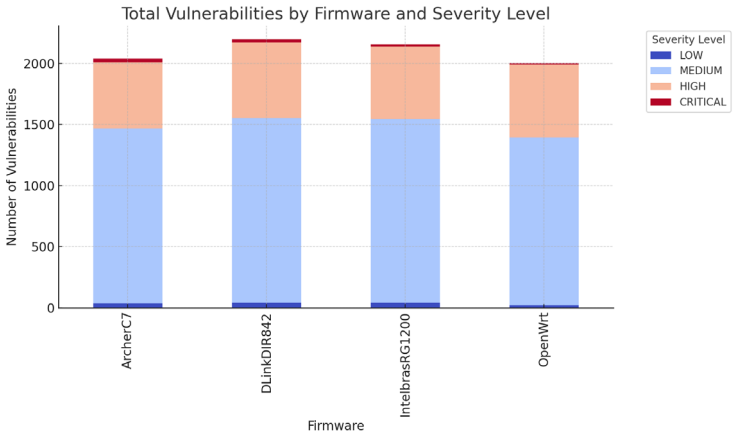

Figure 2. Vulnerabilities by severity category (aggregated counts per firmware) based on CVE-Bin-Tool.

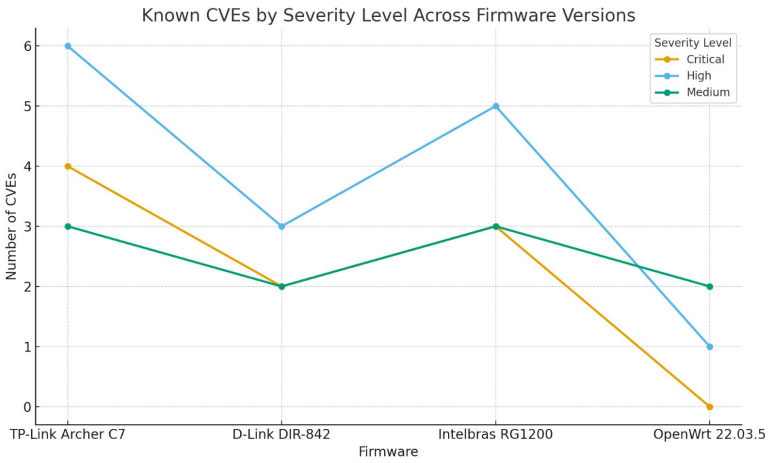

Figure 3. Summary of unique CVEs identified per firmware and main vulnerable components.

4. Results and Discussion

When we analyzed the four Wi-Fi router firmwares, TP-Link Archer C7, D-Link DIR-842, Intelbras RG1200, and OpenWrt 22.03.5, we saw the same problems crop up again and again, in the proprietary options. OpenWrt, on the other hand, proved to be more reliable and easier to maintain. The most common issues we found were hardcoded passwords, insecure network services such as HTTP, Telnet, and FTP, outdated cryptographic libraries, weak logins, and open ports that shouldn't be accessible. These problems put the privacy, reliability, and security of academic networks at risk. In contrast, OpenWrt offers up-to-date components and safer default settings, thanks to its active community.

4.1. Vulnerability Mapping

CVE-Bin-Tool identified 13, 7, 11, and 3 vulnerabilities, respectively, in TP-Link, D-Link, Intelbras, and OpenWrt firmware. Proprietary images concentrated Critical and High CVSS 3.1 scores, while OpenWrt displayed mostly Low or Unknown (Figures 2 and 3).

4.2. Academic Network Impact

Firmware vulnerabilities in academic networks may cause service disruption, data tampering, and unauthorized VLAN access. At ENE/FT-UnB, such flaws could trigger denial-of-service in monitoring systems and compromise experimental data. OpenWrt's modular design and community maintenance reduce residual risk, whereas proprietary firmware, often reliant on outdated components, exposes users to greater risk, as noted by Costin et al. [2], Redini et al. [3], and Zhao et al. [7]. Enhancing resilience requires adopting auditable firmware, disabling legacy services, enforcing secure protocols, and network segmentation. These results confirm that community-maintained firmware benefits from faster updates and improved configurations, validating the proposed workflow as a reliable model for academic network auditing.

5. Conclusions

While our approach focused on just four router models and relied solely on static analysis, it still managed to spot vulnerabilities without running the firmware itself. In the future, testing with live emulation tools like FirmAE or QEMU would be a great next step. Our results make it clear: OpenWrt has fewer and less serious security issues than proprietary firmware, making it a strong candidate for those looking for a safer, more manageable solution. Our open-source process provides a repeatable way to audit firmware, encouraging more institutions to choose open, regularly updated, and easy-to-verify options.

Author Contributions: Conceptualization, L.d.P.S.; methodology, L.d.P.S.; software, L.d.P.S.; validation, R.d.O.A., L.J.G.V., F.L.L.M. and G.D.A.N.; formal analysis, L.d.P.S.; investigation, L.d.P.S.; resources, L.J.G.V. and F.L.L.M.; data curation, L.d.P.S.; writing—original draft preparation, L.d.P.S.; writing—review and editing, R.d.O.A., L.J.G.V., F.L.L.M. and G.D.A.N.; visualization, L.d.P.S.; supervision, G.D.A.N.; project administration, L.d.P.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author(s).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. ENISA. Threat Landscape for IoT 2024. European Union Agency for Cybersecurity. 2024. Available online: https://www.enisa.europa.eu/publications (accessed on 15 January 2025).
2. Costin, A.; Zaddach, J.; Francillon, A.; Balzarotti, D. A Large-Scale Analysis of the Security of Embedded Firmwares. In Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, USA, 20–22 August 2014; Volume 23, pp. 195–210.
3. Zhao, X.; Zhang, X.; Tian, Y.; Wang, Q.; Pu, Y.; Lyu, C.; Beyah, R. UVScan: Detecting Third-Party Component Usage Violations in IoT Firmware. In Proceedings of the USENIX Security Symposium 2023, Anaheim, CA, USA, 9–11 August 2023.
4. Feng, X.; Sun, R.; Zhu, X.; Xue, M.; Wen, S.; Liu, D.; Nepal, S.; Xiang, Y. HermeScan: Faster and Better Detecting Vulnerabilities in Linux-Based IoT Firmware. In Proceedings of the Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 26 February–1 March 2024.
5. Chen, Y.; Ma, F.; Zhang, Y.; He, Y.; Wang, H.; Li, Q. AutoFirm: Automatically Identifying Reused Libraries inside IoT Firmware at Large-Scale. *arXiv* **2024**, arXiv:2406.12947. [CrossRef]
6. Balgavy, A.; Muench, M. FirmLine: A Generic Pipeline for Large-Scale Analysis of Non-Linux Firmware. In Proceedings of the NDSS Workshop on Binary Analysis Research (BAR) 2024, San Diego, CA, USA, 1 March 2024.
7. Redini, N.; Machiry, A.; Wang, R.; Spensky, C.; Continella, A.; Shoshitaishvili, Y.; Kruegel, C.; Vigna, G. Karonte: Detecting Insecure Multi-Binary Interactions in Embedded Firmware. In Proceedings of the 2020 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 18–21 May 2020; pp. 1544–1561. [CrossRef]
8. Kim, M.; Kim, D.; Kim, E.; Kim, S.; Jang, Y.; Kim, Y. FirmAE: Towards Large-Scale Emulation of IoT Firmware for Dynamic Analysis. In Proceedings of the ACSAC 2020, Austin, TX, USA, 7–11 December 2020. [CrossRef]
9. NIST. *SP 800-115: Technical Guide to Information Security Testing and Assessment*; National Institute of Standards and Technology (NIST): Gaithersburg, MD, USA, 2008. [CrossRef]
10. NIST. *SP 800-160 Volume 1 Revision 1: Systems Security Engineering—Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems*; National Institute of Standards and Technology (NIST): Gaithersburg, MD, USA, 2022.