Unveiling firmware weaknesses: An approach for large-scale security analysis

Osmany Barros de Freitas[1], **Lourenço Alves Pereira Júnior**[1]

[1]Divisão de Ciência da Computação - Instituto Tecnológico de Aeronáutica (ITA)
São José dos Campos, SP - Brazil

{osmany,ljr}@ita.br

Abstract. *The COVID-19 pandemic has driven widespread adoption of remote work, altering corporate network perimeters to include Small-Office and Home-Office (SOHO) Routers and exposing infrastructures to IoT threats. We researched and developed a scalable vulnerability analysis method for embedded systems, focusing on vulnerability enumeration and automated source code analysis. In Brazil, we studied 159 router models, obtaining firmware samples from 131. Our analysis revealed more vulnerabilities in official firmware compared to open-source versions, highlighting the security benefits of the latter. Using Binary Code Similarity Analysis (BCSA), we created BinclustRE, a tool for grouping similar binaries to prioritize impactful firmware samples analysis.*

1. Introduction

While Internet of Things (IoT) devices offer numerous benefits across various domains, their inherent vulnerabilities introduce new security concerns, as previous research has highlighted [Alrawi et al. 2019]. Cyberattacks targeting IoT devices can potentially disrupt industries and economies due to their widespread use. Small Office/Home Office (SOHO) routers, integral to the IoT ecosystem, enable connectivity between end devices and the internet service provider, becoming more prominent in networks

In response to these security challenges, manufacturers regularly release firmware updates to patch known vulnerabilities. However, the diverse range of IoT devices complicates the timely delivery of patches for each unique vulnerability. Even when patched firmware is available, users often neglect to update their devices, leaving them exposed to N-day vulnerabilities that malicious actors can exploit. An audit report, [Synopsys 2023], revealed that 53% of IoT applications still contain high-risk vulnerabilities.

1.1. Motivations

In recent years, remote work has become a trend, unlikely to revert to a purely in-person model within organizations [WEFORUM 2022]. The COVID-19 pandemic accelerated the transition to remote work environments and distance education, emphasizing the need to understand and mitigate emerging vulnerabilities in smart cities [Alfonso et al. 2021]. The formation of geographically distributed teams has become easier with improved connectivity services and the ubiquity of Wi-Fi routers [Romana et al. 2020]. These routers connect IoT devices to access networks and Internet service providers.

Nevertheless, the heterogeneity of remotely located devices has led organizations to relax security policies previously based on network assets within a physical perimeter. In this context, program analysis techniques are essential tools for unveiling latent

vulnerabilities in IoT devices, encompassing static and dynamic approaches to analyze file system content, binaries, and kernels [ACI 2018, Weidenbach and vom Dorp 2020]. Dynamic analysis, for example, enables scanning and coverage-guided fuzzing of web interfaces and services within IoT device firmware [Costin et al. 2016, Zhang et al. 2020].

1.2. SCREEN Project

Motivated by the challenges of enhancing the security of IoT devices, the SCREEN project (**S**craper, **C**lustering, **RE**-hosting, and **E**xploitatio**N**) was conceived as a framework for scalable testing of firmware images, specifically designed to handle a vast repository of firmware images. The project leverages an open-source intelligence strategy to obtain freely available firmware online and then execute the hardware image (with some modifications) in an emulated environment. This enables vulnerability assessment and exploitation without the need for the original hardware.

1.3. Objectives

To achieve our objectives, we contributed to the SCREEN project in the *Re-Hosting*, *Clustering*, and *Exploitation* modules, as specific objectives outlined below.

- *Enhance firmware emulation capabilities for kernel mode execution*. Concerning the *Re-Hosting* module, most research primarily focuses on emulating user mode, often utilizing a generic and modified kernel. However, this approach compromises the fidelity of the emulated setup. The most promising works simulate the expected values for each peripheral's kernel or implement these peripherals in C to compile with QEMU, enabling a dedicated virtual machine for emulation. While these proposals provide advancements for our research, scalability issues persist due to the need for manual intervention in the implementation of peripherals.
- *Start implementation of the exploitation module*. We studied static and dynamic vulnerability analysis processes, incorporating the best practices and introducing automated source code analysis. With this approach, we successfully enumerated known vulnerabilities in binaries and the kernel, identified errors in the web source code, and ultimately discovered a zero-day registered at CVE-2022-46552 [1].
- *Apply the methodology in a national context*. Based on this methodology, we conducted a case study in the Brazilian scenario, focusing on SOHO Wi-Fi routers used in homes and small offices to quantify the threats present in these devices. During this study, we compared the most recent versions of the official firmware and the open-source version when available. From this, we found that the solution maintained by the open-source community offers a higher level of security compared to versions officially disclosed by vendors.
- *Cluster most common binaries artifacts to improve vulnerability firmware analysis*. The development of BinclustRE addresses the *Clustering* module in our studies with a modular tool that employs reverse engineering and machine learning techniques to cluster binaries by similarity, allowing the identification of the most common elements in a given repository. Through this classification, it is possible to optimize efforts for vulnerability analyses.

[1] https://nvd.nist.gov/vuln/detail/CVE-2022-46552

2. Related works

The study conducted by [Fiorenza et al. 2020] analyzes the use of HTTPS and vulnerabilities found in the implementations of Brazilian websites. Additionally, the case study by [Ponce et al. 2022] serves as an example, using OSINT techniques to enumerate vulnerabilities in Brazilian Internet devices. These approaches offer awareness to bolster security for systems affecting a large segment of society.

Within the domain of firmware security analysis, [Helmke and Dorp 2022] used static firmware analysis to evaluate the security of routers sold in the European market, focusing on the kernel, but without revealing the manufacturers analyzed. Taking a similar approach, [ACI 2018] provided an analysis focused on equipment in the North American market. Both of them offered valuable insights applicable to the Brazilian context, although they used proprietary tools and an opaque methodology.

To develop our process, we examined related works in vulnerability analysis, focusing on their analytical capabilities and scalability. Table 1 provides a detailed comparison focusing on these works and highlighting the diverse approaches taken by frameworks over the years (for more details, please refer to our full thesis 5.2). Furthermore, our research encompasses all these features, establishing itself as a comprehensive work that advances the integration approach and can potentially enhance vulnerability enumeration and detection in firmware images.

3. Vulnerability Enumeration

3.1. Methodology

To achieve the Vulnerability Enumeration objective, we proposed a generic firmware analysis methodology illustrated in Figure 1, including the following phases: (1) obtaining firmware images from manufacturer websites; (2) extracting the kernel and file system content; (3) creating a repository on GitHub; (4) automated code analysis using the Semgrep tool; (5) static analysis with an emphasis on identifying kernel and binary versions, as well as checking for passwords and private keys; and finally, (6) searching for known vulnerabilities through the public Common Vulnerabilities and Exposures (CVE) database. We applied this methodology to both official firmware and open-source versions.

The analysis framework was built with `Python` and `Shell` scripts, automating the entire process from data collection to firmware image analysis, ensuring scalability.

Table 1. Related Works: Vulnerability Analysis

	OSINT	Static Analysis	Source Code Analysis	Binary Analysis	Kernel Analysis	Vulnerability Enumeration	Scalability
Firmalice (2015)	x	x		x			x
Firmadyne (2016)	x						x
Costin (2017)	x	x	x	x			x
ACI (2018)	x	x			x	x	x
Firm-AFL (2019)	x						x
FirmAE (2020)	x						x
Toso (2021)	x	x			x		x
Dorp (2022)	x	x		x		x	x
Our Work	x	x	x	x	x	x	x

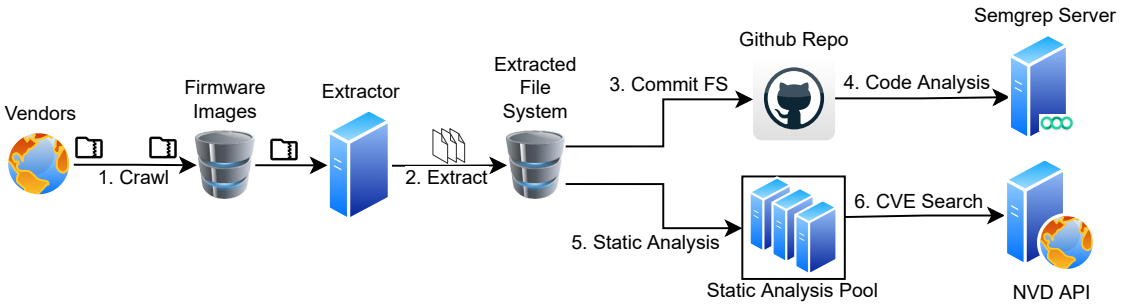

Figure 1. Static methodology for firmware analysis

While it applies to any firmware architecture, our research focused on SOHO routers, as detailed in the motivation. However, due to the impossibility of accurately identifying the models in use in homes and establishments, as well as the unavailability of firmware from devices provided by internet service providers, we employed the same method used in the USA [ACI 2018] and Europe [Helmke and Dorp 2022], which relies on models available for purchase in the market.

Therefore, we employed the Market Share criterion to identify the models considered in this study's selection process. This involved assessing the equipment available for purchase on the most prominent e-commerce platforms in the country, ensuring accessibility for users. From market analysis and data collected between August and October 2022, it was possible to identify 158 distinct models of SOHO routers available in the Brazilian market, distributed by various manufacturers, with a total of 19 brands, where TP-Link, D-Link, and Intelbras hold more than 50% of the models available for sale in the main e-commerces. Compared to the models tested by [ACI 2018], our list features 17 devices from 4 different manufacturers.

3.2. Results

We implemented python scripts for crawling vendors' websites, 133 firmware images were collected from 13 manufacturers in various formats, including raw or compressed formats (e.g., .gz, .rar, and .zip), resulting in a total data size of 1.4 GB. Due to some protective measures implemented by developers, extracting information from 80 samples was possible, achieving a success rate of 60.15%. Among the analyzed models, the MIPS architecture predominates in 83.75% of cases, while ARM represents the remaining 16.25%. We collected 14 distinct user hashes, including `admin`, `root`, `user`, and `support`, from 36 official firmware images. The most common passwords found were `1234`, `admin`, and `sohoadmin`. We also found 88 pre-generated private keys corresponding to models from 6 manufacturers.

We compared official firmware with compatible open-source images. We obtained samples for two models compatible with Tomato, 14 with DD-WRT and 30 with OpenWrt, covering 1, 3, and 7 manufacturers, respectively. The collected samples totaled 297 MB. Regarding architecture, we found only one ARM model, the TP-Link Archer C8. Neither hashes nor private keys were found in the images.

In the official versions, the predominant kernel family is 2.6.x, present in 41.3% of the samples, followed by 3.10.x at 26.7%. In the community-maintained samples,

version 5.10.x appears in 44.4% of the samples, followed by 3.18.x at 15.6%. Analyzing the overall data, we found that only 12% of the official firmware has kernel versions that are less than 4 years old, whereas in the open-source versions, this value rises to 84.4%.

From the binary analysis, 84 vulnerable versions were found, totaling 1474 known vulnerabilities in the official versions. In the open-source versions, 15 samples were found to have a total of 142 vulnerabilities. By applying this methodology to the latest official firmware images, we identified and ranked the top 10 devices with the highest number of vulnerabilities in their binaries, as shown in the Table 2.

Vendor	Model	Firmware	Low	Medium	High	Critical	Total
tp-link	TL-WR941H	TL-WR941HPv2_br-up-2-0-1	20	138	55	13	226
tp-link	Archer C60	ArcherC60v3_eu-up-ver1-2-1	20	129	55	13	217
tp-link	Archer C1200	Archer_C1200_US_V3_211219	13	123	57	11	204
tp-link	Archer C8	ArcherC8-US-V4_170421	14	120	57	11	202
tp-link	Deco M5	Deco_M5_1.6.0_220525	20	93	54	12	179
tp-link	Deco M9 Plus	Deco_M9_Plus_V1_1.5.1_210126	20	93	54	12	179
tp-link	Deco M4	Deco_M4_V2_1.5.4_220113	20	98	47	12	177
tp-link	Deco E4	Deco_E4R_V1_190823	20	98	47	12	177
tp-link	Archer C7	c7v5_us-up-ver1-2-1-P1	17	87	47	10	161
d-link	DIR-846	DIR846enFW100A53DBR-Retail	10	88	50	9	157

Table 2. Top 10 Wi-Fi routers featuring vulnerable binaries.

Semgrep analyzed all the source code foundand identified potential vulnerabilities four times more frequently in official versions than in open-source ones. The tool reported 3921 and 917 possible flaws in analyzing official and open-source firmware images that need validation. MITRE developed a list to enumerate common types of vulnerabilities and categorize these flaws using the Common Weakness Enumeration (CWE).

Table 3. Top 25 MITRE CWEs found in official firmware images. criticality refers to the Likelihood of exploitation.

Official	Criticality	#	Top 25
CWE-79	HIGH	1621	2
CWE-20	HIGH	545	4
CWE-22	HIGH	64	8
CWE-798	*unknown*	168	15
CWE-94	MEDIUM	20	25

Open-source	Criticality	#	Top 25
CWE-79	HIGH	411	2
CWE-20	HIGH	227	4
CWE-798	*unknown*	108	15

Among these findings, especially those classified as `exec-use`, a zero-day vulnerability was identified in the D-Link DIR-846 router, reported as CVE-2022-46552. Validation was conducted in the laboratory by emulating the image with the SCREEN platform and later confirmed using the physical equipment. This vulnerability was then officially reported to D-Link for patch.

4. Clustering Binaries by Similarity

According to [Liu et al. 2020], the IoT devices that develop similar functions usually have the same software, and most use the `Linux` Operating System. The possibility of utilizing open-source in IoT devices can make this software present in machines from different vendors. This practice has become increasingly common because open-source code brings cost reduction, flexibility, and transparency advantages.

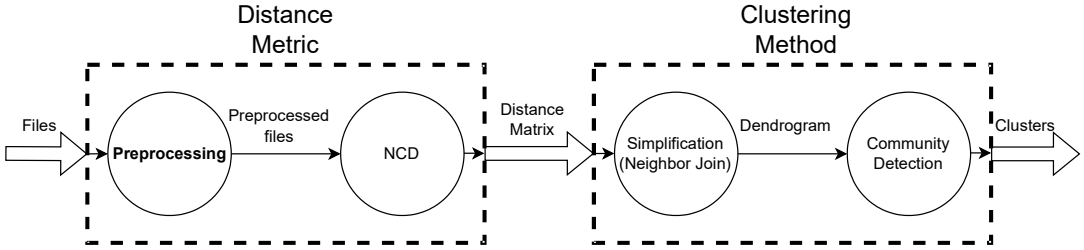

Figure 2. Example of embedding the `Damicore` pipeline to our architecture with the inclusion of our novel preprocessing step.

Binary Code Similarity Analysis (BCSA) is crucial for determining similar programs [Kim et al. 2023]. While similarity analysis serves as an initial step for clustering, to the best of our knowledge, there is no initiative to form clusters specifically for vulnerability detection. This is noteworthy despite the existence of numerous clustering studies for malware analysis, as highlighted in the literature review on machine learning for malware detection conducted by [Gibert et al. 2020]. Some techniques utilize dynamic analysis to group malware based on behavior. In contrast, others employ static analysis, such as Control-Flow Graph (CFG) comparison, to identify groups of malware executables.

4.1. BinclustRE

Therefore, we present `BinclustRE`. This Open-Source tool allows the use of different binary analysis strategies to determine the detection of binary clusters, aiming to help identify software vulnerabilities. Our tool aims to address these challenges by offering a flexible solution for binary code similarity analysis and clustering.

`BinclustRE` adopts the `Damicore` [Sanches et al. 2011] methodology as the core, utilizing the Normalized Compression Distance (NCD) as a similarity metric and creating clusters through neighbor-joining (NJ) and community detection algorithms. This clustering strategy is inspired by phylogenetics and complex networks. Thus, we add a preprocessing step to the `Damicore` original methodology that allows it to work well with compiled binary files instead of source code.

A preprocessing stage is crucial for our pipeline due to its reliance on the NCD, which treats executable files as byte sequences and seeks patterns among them. This makes architecture a critical factor in similarity analysis, as binaries with the same architecture have identical opcodes, while similar instructions in different architectures do not. Consequently, `BinclustRE` implements strategies such as string extraction to capture texts independent of compilation, control-flow graph analysis to map program flow with minimal dependence on compilation options, and binary lifting to revert executables to a state resembling intermediary representation or source code, Figure 2. Therefore, it was possible to apply this approach to various types of binaries found in firmware images and distinguish groups based on their similarity.

5. Conclusions

Our study examined the most popular SOHO routers in Brazil, revealing that their preinstalled firmware is less secure than open-source alternatives. Additionally, we demonstrated how our methodology can identify zero-day vulnerabilities. Finally, we present `BinclustRE`, a versatile tool for dynamic binary clustering that features caching and

multi-threading capabilities. Its architecture is designed for experimenting with various techniques, and the codebase facilitates the seamless integration of new strategies.

5.1. Contributions

1. Integration of best practices to formulate a scalable methodology for analyzing firmware in IoT devices on a large scale, independent of hardware.
2. Application of this methodology in the context of Wi-Fi routers in the Brazilian market, offering an overview of the current security status of these devices.
3. A comparative security analysis between official and open-source firmware confirms that systems maintained by the open-source community are more secure.
4. The `BinclustRE` tool, featuring a modular architecture, introduces a preprocessing phase to `Damicore`'s pipeline, enhancing the capacity to cluster binaries based on similarity and facilitating the analysis of large-scale vulnerabilities.

5.2. Scientific production

- FREITAS, O.; CORRÊA, F.; SANTOS, A.; JUNIOR, L. P. Caracterização das vulnerabilidades dos roteadores wi-fi no mercado brasileiro. In: Anais do XLI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos. Proceedings [...]. Porto Alegre, RS, Brasil: SBC, 2023. p. 183–196. ISSN 2177-9384. Available at: https://sol.sbc.org.br/index.php/sbrc/article/view/24538.
- FREITAS, O.; JUNIOR, L. P. Caracterização de vulnerabilidades em dispositivos IOT e o emprego em ações cibernéticas. In: I Seminário de Pesquisa em Defesa Nacional. Escola de Comando e Estado-Maior do Exército (ECEME), Rio de Janeiro, RJ, Brasil: DECEx,2023.
- TAFFAREL, F.; FREITAS, O.; JUNIOR, L. P. Análise de vulnerabilidades em larga escala nos roteadores wi-fi por meio de web-fuzzing. In: Anais do XXI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais. Porto Alegre, RS, Brasil: SBC, 2023b. Available at: https://sol.sbc.org.br/index.php/sbseg/.
- TAFFAREL, F.; FREITAS, O.; DANTAS, F.; TOSO, G. D.; JUNIOR, L. P. Consciência situacional cibernética no contexto de firmwares de ativos de rede. In: Simpósio de Aplicações Operacionais em Áreas de Defesa 2022 (SIGE2022). Proceedings [...]. [S.l.: s.n.], 2022. Available at: https://www.sige.ita.br/edicoes-anteriores/2022/st/225949%201.pdf
- TAFFAREL, F.; FREITAS, O.; ALMEIDA, F. S.; JUNIOR, L. P. Análise em larga escala de vulnerabilidades em roteadores wi-fi: Ampliando a consciência situacional cibernética. In: Simpósio de Aplicações Operacionais em Áreas de Defesa 2023 (SIGE2023). Available at: https://www.sige.ita.br.

Thesis Avaiability

This thesis is available online at the following URL: http://www.bdita.bibl.ita.br/tesesdigitais/79667.pdf

References

ACI (2018). Securing iot devices: How safe is your wi-fi router? `https://www.theamericanconsumer.org/wp-content/uploads/2018/09/FINAL-Wi-Fi-Router-Vulnerabilities.pdf`. Accessed: 26/12/2022.

Alfonso, I., Garcés, K., Castro, H., and Cabot, J. (2021). Self-adaptive architectures in iot systems: a systematic literature review. *Journal of Internet Services and Applications*, 12(1):1–28.

Alrawi, O., Lever, C., Antonakakis, M., and Monrose, F. (2019). Sok: Security evaluation of home-based iot deployments. In *2019 IEEE symposium on security and privacy (sp)*, pages 1362–1380. IEEE.

Costin, A., Zarras, A., and Francillon, A. (2016). Automated dynamic firmware analysis at scale: A case study on embedded web interfaces. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '16, page 437–448, New York, NY, USA. Association for Computing Machinery.

Fiorenza, M., Kreutz, D., Escarrone, T., and Temp, D. (2020). Uma análise da utilização de https no brasil. In *Anais do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 966–979, Porto Alegre, RS, Brasil. SBC.

Gibert, D., Mateu, C., and Planes, J. (2020). The rise of machine learning for detection and classification of malware: Research developments, trends and challenges. *Journal of Network and Computer Applications*, 153:102526.

Helmke, R. and Dorp, J. v. (2022). Towards reliable and scalable linux kernel cve attribution in automated static firmware analyses. DOI: 10.48550/ARXIV.2209.05217.

Kim, D., Kim, E., Cha, S. K., Son, S., and Kim, Y. (2023). Revisiting binary code similarity analysis using interpretable feature engineering and lessons learned. *IEEE Transactions on Software Engineering*, 49(4):1661–1682.

Liu, K., Yang, M., Ling, Z., Yan, H., Zhang, Y., Fu, X., and Zhao, W. (2020). On manually reverse engineering communication protocols of linux-based iot systems. *IEEE Internet of Things Journal*, 8(8):6815–6827.

Ponce, L., Gimpel, M., Fazzion, E., Ítalo Cunha, Hoepers, C., Steding-Jessen, K., Chaves, M., Guedes, D., and Jr., W. M. (2022). Caracterização escalável de vulnerabilidades de segurança: um estudo de caso na internet brasileira. In *Anais do XL Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 433–446, Porto Alegre, RS, Brasil. SBC.

Romana, S., Grandhi, J., and Eswari, P. R. L. (2020). Security analysis of soho wi-fi routers. In *2020 International Conference on Software Security and Assurance*.

Sanches, A., Cardoso, J. M., and Delbem, A. C. (2011). Identifying merge-beneficial software kernels for hardware implementation. In *2011 International Conference on Reconfigurable Computing and FPGAs*, pages 74–79.

Synopsys (2023). Open source security and risk analysis report.

WEFORUM, W. E. F. (2022). Employers are giving workers the work from home days they want. `https://www.weforum.org/agenda/2022/07/work-from-home-employers-workers-work-life/`. Accessed: 05/01/2023.

Weidenbach, P. and vom Dorp, J. (2020). Home router security report 2020.

Zhang, C., Wang, Y., and Wang, L. (2020). Firmware fuzzing: The state of the art. In *Proceedings of the 12th Asia-Pacific Symposium on Internetware*, pages 110–115.