Segurança Cibernética em Roteadores Wi-Fi: abordagem automatizada para Coleta e Análise de Firmware

Guilherme Bertolino[1], **Françoa Taffarel**[1] e
Lourenço Alves Pereira Junior[1]

[1]Instituto Tecnológico de Aeronáutica (ITA) – São José dos Campos, SP – Brazil

{guilhermegmb,taffarel,ljr}@ita.br

Abstract. *Wireless routers are ubiquitous in contemporary social life, making them a significant concern for cybersecurity and user privacy. Previous studies have presented methods to create firmware image databases to assess router security. However, these methodologies are outdated, as manufacturers make it increasingly difficult to obtain these images. To overcome this challenge, this article presents an automated methodology that includes the firmware's downloading, extraction, and static analysis. The initial results include the acquisition of 262 firmware images from 10 manufacturers and the identification of 7,257 and 3,892 vulnerability indicators from the use of Semgrep and CodeQL, respectively.*

Resumo. *Os roteadores sem fio são onipresentes na vida social contemporânea. Com isso, tornaram-se uma preocupação para segurança cibernética e privacidade dos usuários. Estudos anteriores apresentaram métodos para criar bancos de dados de imagens de firmware a fim de avaliar a segurança dos roteadores. Entretanto, essas metodologias estão desatualizadas, pois os fabricantes dificultam a obtenção dessas imagens. Para superar esse desafio, este artigo apresenta uma metodologia automatizada composta por download, extração e análise estática do firmware. Os resultados iniciais foram a obtenção de 262 firmware de 10 fabricantes, além da identificação de 7.257 e 3.892 indicadores de vulnerabilidade resultado da ação do Semgrep e CodeQL.*

1. Introdução

Os roteadores sem fio são dispositivos cruciais para a interconexão de redes, encaminhando pacotes de dados entre a Internet e redes locais. Essa função confere a eles um papel fundamental na segurança da infraestrutura de comunicação, pois operam como o primeiro ponto de defesa contra ameaças externas [De Keersmaeker et al. 2023]. De acordo com [Ye and et. al. 2024], vulnerabilidades identificadas nesses dispositivos podem ser exploradas por agentes mal-intencionados para interceptar dados, executar ataques de negação de serviço (DoS) e, de maneira geral, comprometer a integridade da rede.

A segurança dos roteadores tornou-se mais urgente por dois fatores principais. Primeiro, o crescimento acelerado de dispositivos IoT, com estimativa de 29,7 bilhões conectados até 2027, muitos dos quais dependem de roteadores sem fio [Analytics 2023]. Segundo, o aumento do trabalho remoto devido à pandemia de COVID-19 expandiu o

perímetro digital das empresas, expondo seus ativos a vulnerabilidades em dispositivos SOHO (*small office home office*).[ICP 2024] [Mudgerikar and Bertino 2021].

Este trabalho insere-se no contexto do projeto SCREEN [Freitas et al. 2023] [Taffarel et al. 2023] [Toso and Pereira 2021] (**S**craper, **C**lustering, **RE**-hosting, and **E**xploitatio**N**), um *framework* desenvolvido para detectar vulnerabilidades na *interface web* de roteadores. No âmbito deste projeto, o artigo apresenta a metodologia para o desenvolvimento de um *web crawler* destinado ao *download* automático de *firmware* de roteadores, além da extração dos sistemas operacionais encontrados e da realização de uma análise estática dos códigos extraídos.

2. Trabalhos Relacionados

De acordo com [Wright et al. 2021], a pesquisa sobre vulnerabilidades em dispositivos IoT tem recebido investimentos consideráveis, refletindo o grande interesse em melhorar a segurança desses dispositivos. Este trabalho insere-se no *framework* SCREEN e apresenta o desenvolvimento de um *web crawler* para criar um *dataset* de *firmware*, com o objetivo de possibilitar a emulação e análise de vulnerabilidades. O escopo do estudo é limitado aos roteadores vendidos no Brasil em sua última versão [Freitas et al. 2023].

Os *frameworks* como `Firmadyne` [Chen et al. 2016] e `FirmAE` [Kim et al. 2020] estão na vanguarda da emulação e análise em larga escala de vulnerabilidades em *firmware* IoT. O `Firmadyne` inclui um módulo *crawler* próprio, independente do restante do *framework*, composto por 42 *spiders* manualmente escritos para buscar e baixar arquivos de *firmware* nos sites de fabricantes de roteadores. Metadados, como o nome do fabricante, modelo e número de versão, são registrados para cada imagem coletada. No entanto, mudanças nos websites dos fabricantes e a implementação de técnicas que dificultam a operação de *crawlers* automáticos frequentemente resultam na desatualização da maioria dos *spiders* do `Firmadyne` na época da escrita deste trabalho.

Na extração de sistemas de arquivos e análise estática, [Costin et al. 2016] apresentou um *framework* para análise automática de vulnerabilidades em *firmware* IoT. Scripts foram criados para extrair e processar sistemas de arquivos UNIX de um *dataset* de *firmware*. A combinação de análises estática e dinâmica identificou vulnerabilidades em 45 dispositivos, mas as ferramentas estáticas focaram apenas em arquivos PHP.

Assim, esse trabalho concentra-se na criação de uma nova interface de aquisição de *firmware* e extração de suas *features*, dada a desatualização das ferramentas anteriores, visando melhorar o *dataset* do projeto SCREEN e outros projetos de pesquisa que desejem realizar análises em roteadores. Também é proposta a utilização de analisadores de código-fonte mais recentes para avaliar os sistemas de arquivos extraídos dos *firmware*.

3. Metodologia

A Figura 1 expõe a metodologia proposta por esse trabalho, apresentando as seguintes fases: (1) obtenção das imagens de *firmware* das páginas dos fabricantes; (2) extração do sistema de arquivos das imagens; (3) criação do repositório no `GitHub`; (4) análise estática automática de vulnerabilidades usando as ferramentas de integração contínua do `GitHub` com o `CodeQL` e o `Semgrep`.

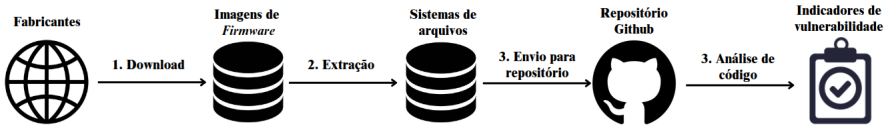

Figura 1. Metodologia

3.1. Aquisição de *firmware*

O *web crawler* foi desenvolvido utilizando a biblioteca `Scrapy` em *python*, com um *spider* escrito para o site de cada um dos 10 fabricantes selecionados, o critério de escolha foi a relevância no mercado brasileiro de roteadores, conforme identificado por [Freitas et al. 2023]. Os *scripts* escritos possuem como ponto de partida a página principal de venda de roteadores no site do fabricante, a qual contém a listagem de todos os modelos.

Como alguns fabricantes fazem uso de elementos gerados dinamicamente em seus sites, nesses casos, utilizou-se adicionalmente a biblioteca `Scrapy-Splash` e, em último caso, a biblioteca `Selenium` com o *webdriver* do `Firefox`. Ambas as bibliotecas operam utilizando um navegador para acessar o endereço solicitado e possuem ferramentas para carregar conteúdo gerado dinamicamente, bem como para clicar em elementos na tela ou ou rolar a página. O `Splash` é um navegador leve escrito em `Lua`, especificamente projetado para uso em *web crawlers* e de fácil integração com o `Scrapy`. Entretanto, em alguns casos, o site requisitado não respondeu bem a esse navegador, sendo necessário utilizar o `Selenium` com um navegador de uso geral, no caso, o `Firefox`.

Por fim, foi escrito um script em `bash` para orquestrar a inicialização dos *spiders* e paralelizar sua operação. Além disso, para manter a portabilidade em qualquer sistema e facilitar a operação do módulo por outros pesquisadores, foi escrito um template *YAML* para rodar o *web crawler* em um container `Docker`, utilizando da ferramenta de volumes para adicionar os *firmware* em um diretório da máquina *host*.

3.2. Extração de sistema de arquivos

Utilizando-se da ferramenta de extração de *firmware* `Binwalk`, foi desenvolvido um *script* em `Python` para recuperar o sistema de arquivos a partir do *firmware* adquirido no site do fabricante. A heurística utilizada para identificar sistemas de arquivos UNIX foi a presença de quatro ou mais diretórios UNIX (/bin/, /usr/, etc.) dentro de um diretório.

A ferramenta de extração desenvolvida baseia-se em uma busca em profundidade (DFS) nos resultados fornecidos pelo `Binwalk`. Um resultado com o nome contendo as palavras-chave *"filesystem"*, *"archive"* ou *"compressed"* ativa uma busca em largura (DFS) no arquivo gerado pela extração do `Binwalk`, procurando por um diretório contendo quatro ou mais diretórios UNIX. Em caso positivo, o sistema de arquivos é extraído para uma pasta e encerra-se a extração nesse *firmware*. Todas as buscas possuem um limite de profundidade, a fim de garantir que o programa termine de executar.

Uma melhoria feita no extrator foi o uso das ferramentas de extração de terceiros `jefferson` e `sasquatch` para sistemas de arquivos `JFFS2` e `SquashFS` respectivamente, melhorando consideravelmente as taxas de sucesso na extração. Essa melhoria ocorre porque as ferramentas de extração desenvolvidas pelos desenvolvedores

Tabela 1. Resultados da aquisição e extração de *firmware*

	TP-link	Intelbras	Netgear	D-link	Multilaser	Ubiquiti	Asus	Tenda	Mercusys	Trendnet	Total
Imagens baixadas	87	48	26	25	18	17	14	13	8	6	262
Extraídas	57	29	15	3	5	16	14	2	0	0	141
Taxa de extração	66%	60%	58%	12%	28%	94%	100%	15%	0%	0%	54%

de sistemas de arquivos são pouco robustas a implementações modificadas dos sistemas de arquivo, frequentemente falhando na extração das imagens adquiridas.

Apesar dessas melhorias, nem sempre é possível recuperar o sistema de arquivos a partir do *firmware*. Isso ocorre porque alguns fabricantes distribuem apenas uma imagem parcial dos seus produtos, impedindo a reconstrução do sistema de arquivos. Também pode ocorrer da imagem conter múltiplos sistemas de arquivos parciais para serem montados, necessitando de lógica adicional para sua reconstrução. Outros problemas incluem imagens com o sistema de arquivos encriptado, imagens com um sistema de arquivos não reconhecido ou imagens para outros sistemas operacionais.

3.3. Análise estática

As ferramentas escolhidas para a análise estática foram `Semgrep` e `CodeQL`, adequadas para analisar código em linguagens interpretadas como `JavaScript` e `PHP`, comuns nas interfaces web de roteadores. Essas ferramentas detectam vulnerabilidades por padrões e consultas, suportam várias linguagens e permitem personalização, melhorando a segurança do código sem executá-lo. Ambas integram-se com o `GitHub`. Foi criado um repositório para armazenar os sistemas de arquivos extraídos, separando-os em diferentes *branches* para evitar sobrecarregar a memória do *container* durante a análise.

4. Resultados Preliminares

O primeiro resultado fornecido por esse trabalho são os *scripts* do *web crawler* e do extrator de sistemas de arquivos, disponíveis para uso por quaisquer outros pesquisadores que necessitem de uma ferramenta para *download* automático de *firmware* e extração de *features* específicas deles.

Nesse sentido, a Tabela 2 apresenta o número de *firmware* adquiridos pelo *crawler*, bem como o número de imagens cuja extração do sistema de arquivos obteve sucesso. Esse projeto focou nas imagens de roteadores vendidos pelos principais fabricantes no Brasil e em suas versões mais recentes. No entanto, essa abrangência pode ser facilmente expandida com a criação de novos *spiders* e a adição de links para páginas de fabricantes em outras regiões no arquivo `'vendor_links.py'`. Por fim, a taxa de sucesso na extração é baixa e varia entre os fabricantes, devido às diferentes formas como divulgam suas imagens, que podem incluir sistemas de arquivos parciais, encriptados ou separados em múltiplos pontos do *firmware*.

Na análise estática de vulnerabilidades, os sistemas de arquivos foram submetidos a uma análise automática utilizando as regras do `Semgrep` e do `CodeQL`. Os resultados, apresentados na Tabela 2, servem como indicadores de possíveis vulnerabilidades, mas não garantem sua presença no código. Para confirmar a existência de código vulnerável, é necessária a validação das vulnerabilidades diretamente no roteador.

Tabela 2. Resultados da análise estática

Ferramenta	Resultados	Resultados de alta severidade
Semgrep	7257	1871
CodeQL	3892	1427

5. Conclusões

Este trabalho desenvolveu uma metodologia automatizada para a aquisição e análise estática de *firmware* de roteadores sem fio, destinada à integração no *framework* SCREEN. Foram adquiridas 262 imagens de *firmware* de 10 fabricantes, com sucesso na extração dos sistemas de arquivos em 141 delas. A análise identificou 7.257 indicadores de vulnerabilidade pelo `Semgrep` e 3.892 pelo `CodeQL`, demonstrando a eficácia da metodologia proposta.

Os trabalhos futuros focarão na expansão do escopo de estudo para incluir novos fabricantes e regiões, melhorar as ferramentas de extração para incluir a extração do *kernel* e do *bootloader* das imagens, além de uma análise mais profunda dos resultados da análise estática, a fim de identificar quais os achados mais promissores para a criação de *exploits*.

Agradecimentos

Este trabalho foi financiado pelo CNPq.

Referências

Analytics, I. (2023). State of iot 2023: Number of connected iot devices growing 16% to 16.7 billion globally. *IoT Analytics*. Acessado em 04/07/2024.

Chen, D. D., Woo, M., Brumley, D., and Egele, M. (2016). Towards automated dynamic analysis for linux-based embedded firmware. In *NDSS*, volume 1, pages 1–1.

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 Asia CCS*.

De Keersmaeker, F., Cao, Y., Ndonda, G. K., and Sadre, R. (2023). A survey of public iot datasets for network security research. *IEEE Communications Surveys Tutorials*, 25(3):1808–1840.

Freitas, O., Corrêa, F., Santos, A., and Junior, L. P. (2023). Caracterização das vulnerabilidades dos roteadores wi-fi no mercado brasileiro. In *Anais do XLI SBRC*, PA, RS, Brasil. SBC.

ICP (2024). Post-pandemic: The evolution of remote working. *ICP*.

Kim, M., Kim, D., Kim, E., Kim, S., Jang, Y., and Kim, Y. (2020). FirmAE: Towards large-scale emulation of iot firmware for dynamic analysis. In *ACSAC*, Online.

Mudgerikar, A. and Bertino, E. (2021). Iot attacks and malware. *Cyber Security Meets Machine Learning*, pages 1–25.

Taffarel, F., de Freitas, O. B., and Junior, L. A. P. (2023). Análise de vulnerabilidades em larga escala nos roteadores wi-fi por meio de web-fuzzing. In *Anais do XXIII SBSeg*. SBC.

Toso, G. and Pereira, L. A. (2021). Enumeração de sistemas operacionais e serviços de firmwares de roteadores sem-fio. In *Anais Estendidos do XXI SBSeg*, PA, RS, Brasil. SBC.

Wright, C., Moeglein, W. A., Bagchi, S., Kulkarni, M., and Clements, A. A. (2021). Challenges in firmware re-hosting, emulation, and analysis. *ACM Comput. Surv.*, 54(1).

Ye, J. and et. al. (2024). Exposed by default: A security analysis of home router default settings. In *Proceedings of the 19th ACM Asia CCS*. ACM.