Análise em Larga Escala de Vulnerabilidades em Roteadores Wi-Fi: Ampliando a Consciência Situacional Cibernética

Françoa Taffarel, Osmany Barros de Freitas, Felipe Silveira de Almeida, Lourenco Alves Pereira Jr

Instituto Tecnológico de Aeronáutica - ITA

Resumo — Os roteadores sem-fio progrediram para garantir a conectividade entre os dispositivos IoT à Internet. Essa evolução também aumentou a importância de análises de segurança, devido aos crescentes ataques cibernéticos direcionados ou em massa por agentes maliciosos. No entanto, uma restrição na realização dessas análises em larga escala é a necessidade de acesso ao dispositivo físico. Neste artigo, apresentamos uma metodologia semiautomatizada que combina a emulação de imagens de *firmware* de roteadores com o *web-fuzzing* da *interface web* utilizando o Nuclei. Os resultados iniciais foram a identificação de 6.293 possíveis falhas, a criação de 27 *templates* do Nuclei e a validação do CVE-2022-46552.

I. INTRODUÇÃO

Com o crescimento de 18% nas conexões de dispositivos *Internet of Things* (IoT) em 2022, alcançando 14,3 bilhões de aparelhos, e a previsão de atingir 16,7 bilhões em 2023 [1], a segurança desses sistemas é uma preocupação constante. Isso é especialmente relevante em casas inteligentes, onde dispositivos IoT auxiliam em tarefas diárias. Os roteadores sem-fio, essenciais para a conectividade deste dispositivos IoT, foram alvos, como *a botnet* Mirai ainda ativa em 2023 [2]. Assim, cada vez mais a análise de vulnerabilidades em roteadores sem-fio desempenha um papel essencial na segurança cibernética, possibilitando a adoção de medidas preventivas a fim de minimizar riscos e impactos aos usuários, além de atender a conformidades regulatórias [3].

Portanto, este artigo descreve uma metodologia semiautomatizada capaz de auxiliar na descoberta de vulnerabilidades na *interface web* por meio da análise dinâmica de roteadores sem-fio em larga escala. A contribuição proposta pelo artigo é a integração da capacidade de emulação de imagens de *firmware* de roteadores em larga escala por meio do *framework* FirmAE com a análise de vulnerabilidades usando a técnica de *web-fuzzing* com *templates* da ferramenta Nuclei. Além disso, outra contribuição consiste na criação de *templates* específicos para o contexto de roteadores sem-fio a partir da análise de código-fonte e de vulnerabilidades conhecidas. Esses modelos, construídos com base em *Yet Another Markup Language* (YAML), têm a finalidade de estabelecer os procedimentos para o envio e processamento das requisições HTTP.

II. Trabalhos Relacionados

A pesquisa em vulnerabilidades de dispositivos IoT aumentou por meio de grandes investimentos para encontrar falhas, especialmente em roteadores sem-fio [4]. Dessa forma, pesquisadores estão usando análise de vulnerabilidades para melhorar a segurança desses dispositivos. Contudo, enfrentam desafios como a complexidade das ameaças cibernéticas e a falta de padrões de segurança rigorosos.

Nesse contexto, [5], [6] e [2] , respectivamente, adotaram a técnica da análise estática das imagens de *firmware* dos roteadores para caracterizar vulnerabilidades de roteadores sem-fio mais comuns no mercado europeu, norte-americano e brasileiro. Além disso, os autores [7-8] utilizam a técnica de análise dinâmica em seus *frameworks* para localizar vulnerabilidades em imagens de *firmware*. No entanto, nenhum desses trabalhos realiza a análise do código-fonte do sistema de arquivos com ênfase na *interface web*. Com a mesma motivação, o trabalho de [9] apresentou o UCRF que realiza de forma mútua a análise estática no binário do *back-end* da *interface web* dos roteadores e posteriormente executa dinamicamente um *fuzzer* para descobrir vulnerabilidades. Porém, a proposta limita-se à análise de 10 roteadores físicos e, consequentemente, possui baixa escalabilidade.

Com o objetivo de reduzir a dependência de dispositivos físicos e melhorar a capacidade de análise de vulnerabilidades, a emulação de imagens de *firmware* tornou-se uma solução valiosa para os pesquisadores. Neste contexto, a vanguarda da emulação em larga escala é representada por *frameworks* como Firmadyne, FirmAE e ALEmu [10-12]. Esses *frameworks* aplicam a técnica de *re-hosting* para executar o *firmware* em um ambiente emulado. No entanto, [13], que utiliza o *framework* FirmAE, não realiza a análise de código-fonte e seu *web-fuzzer* possui baixa escala por possuir poucas regras de verificação para validar as mais recentes vulnerabilidades.

Assim, o principal desafio deste trabalho em andamento não se limita apenas à integração das técnicas de emulação e análise de vulnerabilidades, mas também abrange a busca por escalabilidade e a capacidade de validar vulnerabilidades. Para isso, será utilizada a ferramenta Nuclei que desempenha um papel fundamental neste processo possibilitando a detecção automatizada e em escala de vulnerabilidades em aplicações, infraestruturas e produtos, graças à estrutura modular de seus *templates* de execução de *web-fuzzing* [14]. Esses *templates* serão alimentados com evidências oriundas da análise do código-fonte do sistema de arquivos das imagens de *firmware* possibilitando descobrir novas vulnerabilidades (zero-days). Além disso, será possível também a validar em outros dispositivos vulnerabilidades anteriormente reportadas (1-day *vulnerabilities*) utilizando os seus dados em *templates* específicos para o contexto de roteadores sem-fio.

III. Metodologia para validação de vulnerabilidade em escala

A Figura 1 expõe a metodologia proposta por esta pesquisa, a qual é fundamentada na emulação e análise de vulnerabilidade, em escala, das imagens de *firmware*, presentes em um banco de dados, por meio da técnica da re-hosting. As imagens presentes nesse repositório são obtidas com auxílio de *web crawlers* a partir das páginas dos fabricantes, ou de modo manual diretamente do dispositivo físico através de *interfaces* como JTAG, UART e USB. Nesse contexto, o *framework* FirmAE executa as heurísticas para checagem de emulação a fim de obter informações sobre quais imagens são emuláveis.

Em seguida, são aplicadas as funções do FirmAE para extração de sistema de arquivos das imagens disponíveis, sendo aproveitadas nessa fase as boas práticas implementadas por [2]. O resultado da extração é enviado para um repositório no Github. Posteriormente, esse conteúdo será analisado pela ferramenta Semgrep, que utiliza uma abordagem de pesquisa de padrões para detectar funções vulneráveis no código-fonte encontrado no conteúdo extraído dos sistemas de arquivos.

Os dados reportados pelo Semgrep são considerados indícios de possíveis vulnerabilidades que necessitam de validação. Esses resultados incluem informações como a localização da vulnerabilidade em determinado arquivo, a severidade e os detalhes sobre as regras correspondente a detecção. Após uma análise manual desses achados, são preparados *templates* da ferramenta Nuclei, a fim de validar as possíveis falhas em toda a base de dados, proporcionando uma análise abrangente e precisa das vulnerabilidades.

Para avaliar de forma eficiente numerosas imagens de *firmware*, é realizada a paralelização da emulação, utilizando contêineres com a ferramenta Docker. Cada imagem é emulada de forma independente em um contêiner, que possui internamente os pacotes e dependências utilizados para emulação. Isso garante uma emulação confiável mesmo em situações com múltiplas interfaces de rede. Por fim, com o firmware emulado com sucesso e a interface *web* acessível, o Nuclei executa os *templates* disponíveis na base de dados, utilizando requisições HTTP para realizar a verificação das vulnerabilidades e geração de um relatório final.

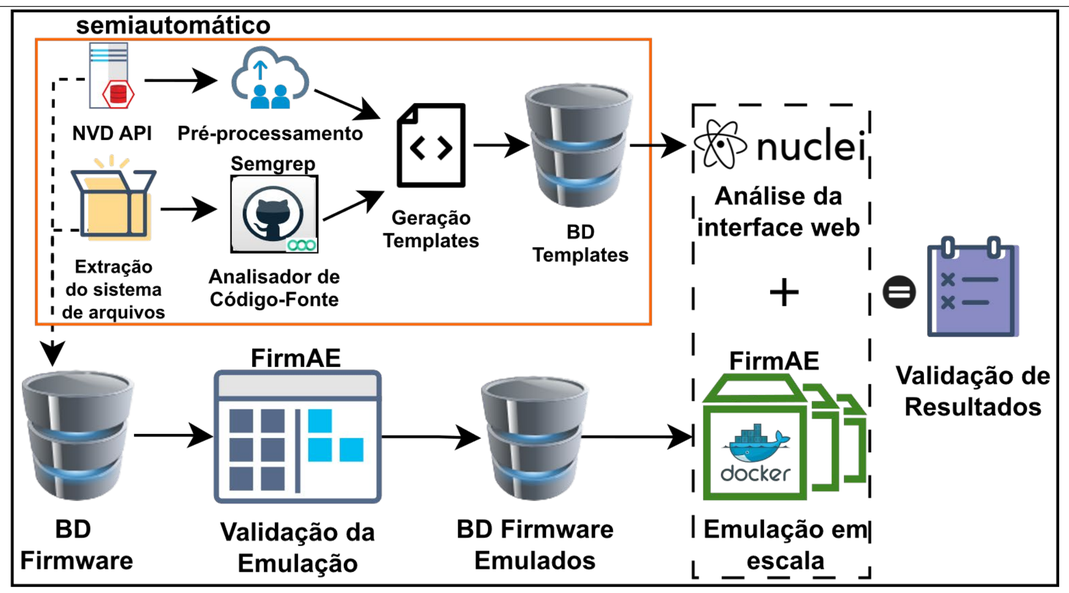

Figura 1. Metodologia

IV. Resultados Preliminares

Todos os experimentos foram conduzidos em um servidor com quatro CPUs Intel® Xeon® E3-1225-v6-3.30GHz, 32 GB de RAM DDR4 e 4 TB de HD. Com sistema operacional Ubuntu 20.04, com o PostgreSQL 12.15 e o Docker v20. A base de dados utilizada para testar a metodologia é composta por imagens de *firmware* dos fabricantes TP-Link e D-Link das quais 1748 são provenientes dos estudos feitos por [15] e outras 65 da base utilizada por [2]. Em seguida desta base, a ferramenta Semgrep analisou o sistema de arquivos extraído pelo *framework* FirmAE e reportou, respectivamente, os valores de 2509 e 3784 indícios de vulnerabilidades para os fabricantes TP-Link e D-Link. A ênfase desta análise foi colocada em 369 possíveis falhas consideradas de alta criticidade, particularmente aquelas relacionadas à execução remota de código.

Durante a emulação e análise de vulnerabilidades, os resultados preliminares indicaram que 219 (12%) da imagens de *firmware* foram emuladas simultaneamente por meio da paralelização por contêineres e a ferramenta Nuclei executando os *templates* gerados obteve êxito em validar a vulnerabilidade de execução remota de comandos com privilégios de super-usuário root no roteador *D-Link DIR-846*. A falha resulta da injeção de código malicioso via requisição *POST* devido à falta de sanitização no arquivo *SetIpMacBindSettings.php*. O fabricante foi notificado por canal oficial e a falha recebeu a numeração CVE-2022-46552. Os dados dispostos na Tabela I expõem os valores relacionados à aplicação da metodologia proposta.

Tabela I. Resultados Preliminares

Base de dados	Fabricante	Imagens Emuláveis	Resultados *Semgrep*	Templates Criados
846	TP-Link	144	2509	23
967	D-Link	75	3784	4

III. CONCLUSÃO

Nesta pesquisa em andamento, foi exposta uma arquitetura semiautomatizada para descoberta de vulnerabilidades em larga escala em roteadores sem-fio por meio da análise dinâmica de vulnerabilidades de suas *interfaces web*. Esta arquitetura tem como objetivo enfrentar esse desafio aproveitando as capacidades de emulação do *framework FirmAE* e o poder de detecção de vulnerabilidades da ferramenta *Nuclei*. Como trabalhos futuros, pretende-se expandir o repositório de *templates* para detecção de vulnerabilidades, utilizando tecnologias de Processamento de Linguagem Natural (NLP).

REFERÊNCIAS

1. Analytics, I. (2023). State of iot 2023: Number of connected iot devices growing 16% to 16.7 billion globally. IoT Analytics. Acessado em 25/05/2023.
2. Freitas, O., et. al. (2023). Caracterização das vulnerabilidades dos roteadores wi-fi no mercado brasileiro. In XLI SBRC, Porto Alegre, RS, Brasil. SBC.
3. ANATEL (2023). Ato n° 2436 - requisitos mínimos de segurança cibernética. Acessado em 19 de junho de 2023.
4. Wright, C., et. al. (2021). Challenges in firmware re-hosting, emulation, and analysis. ACM Comput. Surv., 54(1).
5. Helmke, R. and vom Dorp, J. (2022). Towards reliable and scalable linux kernel cve attribution in automated static firmware analyses.
6. ACI (2018). Securing IoT devices: How safe is your wi-fi router?. Acessado em junho de 2023.
7. Zheng, Y., et al. (2019). FIRM-AFL:High-Throughput Greybox fuzzing of IoT firmware via augmented process emulation. In USENIX Security 19.
8. Redini, N., Machiry, et al. (2020). Karonte: Detecting insecure multi-binary interactions in embedded firmware. In 2020 IEEE SSP.
9. Qin, C., et al. (2023). Ucrf: Static analyzing firmware to generate under-constrained seed for fuzzing soho router. Computers & Security, page 103157.
10. Chen, D., et. al. (2016). Towards automated dynamic analysis for linux-based embedded firmware. In NDSS, volume 1, pages 1–1.
11. Kim, M., et. al. (2020). FirmAE: Towards large-scale emulation of iot firmware for dynamic analysis. In ACSAC, Online.
12. He, H., Xiong, X., and Zhao, Y. (2023). Alemu: A framework for application-layer programs emulation of embedded devices. In 4th (ICCEA).
13. Zhang, H., Lu, K., Zhou, X., et al. (2021). Siotfuzzer: fuzzing web interface in iot firmware via stateful message generation.
14. Solanki, H. V. (2023). Limiting attack surface for infrastructure applications using custom yaml templates in nuclei automation. Master's thesis, Dublin, NCI.
15. Toso, G. and Pereira, L. A. (2021). Enumeração de sistemas operacionais e serviços de firmwares de roteadores sem-fio. In XXI SBSeg, Porto Alegre, RS, Brasil. SBC.