Caracterização das vulnerabilidades dos roteadores Wi-Fi no mercado brasileiro

Osmany Barros de Freitas[1], Françoa Taffarel Rosário Corrêa[1], Aldri Luiz dos Santos[2] e Lourenço Alves Pereira Junior[1]

[1]Divisão de Ciência da Computação – ITA – São Jose dos Campos, SP – Brazil

[2]Departamento de Ciência da Computação – UFMG – Belo Horizonte, MG – Brazil

{osmany,taffarel,ljr}@ita.br, aldri@dcc.ufmg.br

Abstract. *Characterizing the vulnerabilities of Wi-Fi routers is essential to identify and quantify the risks and threats present in the digital ecosystem that permeates the routine of users. This paper analyzes the firmware of Wi-Fi routers in Brazilian e-commerce. The results indicate a predominance of Linux on MIPS and ARM architectures, with an average lag of* 5 *to* 10 *years between the release of kernel and the most recent version of firmware. As a result, we observed* 1344 *and* 72 *vulnerabilities on average in kernel and applications; and* 54 *indicators of compromises that can lead to vulnerabilities in the web interface. On the other hand, replacing an open-source firmware (OpenWrt, DD-WRT, Tomato) reduces the average vulnerabilities to* 291*,* 12*, and* 21 *for kernel, applications, and web interface, respectively. This investigation also allowed the report of a new remote code execution vulnerability (zero-day).*

Resumo. *Caracterizar as vulnerabilidades dos roteadores Wi-Fi é essencial para identificar e quantificar os riscos e ameaças presentes no ecossistema digital que permeia a rotina dos usuários. Este artigo analisa os firmwares de roteadores Wi-Fi presentes no mercado brasileiro. Os resultados indicam uma predominância de Linux nas arquiteturas MIPS e ARM, com defasagem média de* 5 *a* 10 *anos de lançamento do kernel em relação à versão mais recente do firmware. Observaram-se* 1344 *e* 72 *vulnerabilidades em média no kernel e nas aplicações; e* 54 *indicadores de comprometimentos que podem levar a vulnerabilidades na interface web. Por outro lado, a substituição por um firmware open-source (OpenWrt, DD-WRT, Tomato) reduz as médias de vulnerabilidades para* 291*,* 12 *e* 21 *para kernel, aplicações e interface web, respectivamente. Por fim, esta investigação permitiu o relato de uma nova vulnerabilidade (zero-day) de execução remota de código.*

1. Introdução

Nos últimos anos a modalidade de trabalho remoto tem se transformado em uma tendência, e dificilmente o mundo retornará ao modelo puramente presencial de execução de tarefas nas organizações [WEFORUM 2022]. Aumenta-se a flexibilidade de formação de equipes geograficamente distribuídas, pois os serviços de conectividade atuais permitem que a distância deixe de ser um impeditivo. Ainda, os roteadores Wi-Fi apresentam-se como equipamentos pervasivos, pois possibilitam a conectividade entre os dispositivos finais e a rede de acesso com o provedor de serviço de internet.

Assim, ressalta-se sua importância no cotidiano das pessoas em residências, comércios e pequenas indústrias. No entanto, políticas e mecanismos de segurança fundamentados nos ativos de rede localizados no perímetro físico das organizações sofrem um respectivo relaxamento provocado pela heterogeneidade dos dispositivos localizados remotamente.

Juntamente com a popularidade e benefícios proporcionados pela utilização dos dispositivos da Internet das Coisas (IoT), há também os riscos devido às ações maliciosas que se destinam para além dos ataques direcionados, como também em larga escala, como o Mirai, de 2016, que comprometeu milhões de dispositivos para criação de uma *botnet*. Relatório de vulnerabilidades em dispositivos IoT divulgado em 2020 pela Paloalto [Networks 2020], indica que mais de 50% dos equipamentos utilizados no mundo estão vulneráveis a ataques de severidade média a alta. Logo, incidentes desta natureza representam potenciais riscos em aspectos econômicos, geopolíticos e sociais. A detecção de vulnerabilidades em roteadores Wi-Fi fundamenta-se em técnicas para identificação automática de falhas nos *firmwares* dos dispositivos [Feng et al. 2022, Redini et al. 2020], analisando seus sistemas, binários e *kernel*, e empregando metodologias como análises estáticas, dinâmicas, execução simbólica e emulação. Portanto, a análise destas vulnerabilidades torna-se essencial para identificar e quantificar os riscos e ameaças presentes no ecossistema digital que permeia a rotina de seus usuários.

Este artigo apresenta um estudo dos principais roteadores Wi-Fi (categoria *small-office, home-office*—SOHO) disponibilizados no mercado brasileiro. Foi possível enumerar os equipamentos (modelos e fabricantes) com maior oferta; em seguida realizou-se a identificação do sistema operacional e os softwares de sistema, bem como a interface Web de gerenciamento dos dispositivos. Por fim, foi conduzida uma análise estática dos artefatos disponíveis com a finalidade de mapear vulnerabilidades existentes e indicadores de comprometimento que podem levar a descoberta de vulnerabilidades. Portanto, a contribuição deste trabalho consiste de: (*i*) a expansão dos trabalhos relacionados ao proporcionar uma metodologia mais abrangente, (*ii*) a enumeração dos softwares existentes nos roteadores Wi-Fi presentes no mercado brasileiro, (*iii*) a identificação do nível de vulnerabilidade incorporada ao adquirir esses ativos de rede, (*iv*) a quantificação do aumento de segurança ao substituir o *firmware* original por uma versão *open-source* compatível.

O artigo está organizado da seguinte maneira: A Seção 2 discute os trabalhos relacionados. A Seção 3 apresenta a metodologia utilizada nesta pesquisa. Em seguida, a Seção 4 relata uma análise descritiva dos *firmwares* em estudo e na Seção 5 são discutidos os resultados. Por fim, a Seção 6 conclui o trabalho.

2. Trabalhos Relacionados

A transformação digital proporcionada pela IoT, traz consigo desafios inerentes dessa ubiquidade: Privacidade e Segurança. Assim, torna-se primordial aferir o estado atual das tecnologias que compõem este ecossistema, observando, por exemplo, as vulnerabilidades existentes. O estudo conduzido por [Fiorenza et al. 2020] analisa o uso de HTTPS e as vulnerabilidades encontradas nas implementações dos sites brasileiros. Soma-se como exemplo, o estudo de caso realizado por [Ponce et al. 2022] que utiliza técnicas de

OSINT[1] para enumerar vulnerabilidades presentes em dispositivos da Internet brasileira. Assim, cria-se uma percepção da consciência situacional e, por consequência, a melhora da segurança de sistemas que impactam grande parcela da sociedade.

Ainda no contexto de segurança, os roteadores SOHO, que constituem os *gateways* mais usados para conectar dispositivos IoT, tornam-se relevantes para análise de vulnerabilidades e riscos envolvidos. Isto motivou a proposta de [Helmke and Dorp 2022] que usou análise estática de *firmwares* para uma percepção analítica da segurança de roteadores comercializados no mercado europeu, porém com ênfase no *kernel* e sem revelar os fabricantes analisados. Adotando semelhante abordagem, [ACI 2018] proporcionou análise com foco em equipamentos do mercado norte-americano. Apesar de o estudo utilizar ferramenta proprietária e metodologia opaca, foi possível observar um direcionamento para aplicação no contexto brasileiro.

É notório que possuir o dispositivo físico para realizar análises de vulnerabilidades representa elevado custo para um projeto quando deseja-se atuar com vasto repositório. Então, a fim de contornar a dependência dos roteadores para realização dos estudos, [Toso and Pereira 2021] aplicaram a técnica do *re-hosting* que consiste na execução, em um ambiente emulado, dos *firmwares* de roteadores disponibilizados pelos fabricantes. Essa técnica permitiu-lhes a obtenção de informações relacionadas aos sistemas de arquivos e softwares embarcados. No entanto, novamente observa-se a ausência de foco nos equipamentos em uso pelos brasileiros e a necessidade de realizar uma verificação das vulnerabilidades conhecidas, bem como estudar os códigos em *userspace*, essencialmente páginas web utilizadas para administração dos dispositivos.

Dessa forma, com ênfase distinta dos outros trabalhos, neste artigo foram direcionados esforços para enumeração e análise estática de roteadores Wi-Fi SOHO no cenário brasileiro. A partir da identificação dos modelos mais comuns, foi possível quantificar as vulnerabilidades existentes, evidenciando a obsolescência adotada pelos fabricantes quanto aos binários e *kernels* utilizados. Este trabalho utilizou também a inspeção automatizada do código encontrado nos sistemas de arquivo dos *firmwares* analisados, técnica empregada na descoberta de novas vulnerabilidades. Por fim, destacou-se a recomendação direta ao usuário quanto ao uso de *firmwares open-source*, proporcionando melhor nível de segurança aos dispositivos.

3. Metodologia

Esta seção detalha o processo aplicado para a condução da análise. A Figura 1 representa as fases da metodologia: (1) obtenção das imagens dos *firmwares*, das páginas do fabricantes; (2) extração do *kernel* e conteúdo do sistema de arquivos; (3) criação de repositório no *github*; (4) análise automática de código com ferramenta `Semgrep`; (5) análise estática com ênfase na identificação das versões de *kernel* e binários, além da verificação de senhas e chaves privadas; e por fim, (6) consulta por vulnerabilidades conhecidas, através da base pública de *Common Vulnerabilities and Exposures* (CVE).

O escopo deste trabalho foi os roteadores Wi-Fi *SOHO*, no entanto, devido à impossibilidade de se realizar a exata identificação dos modelos em uso nas

[1] *open-source Intelligence*: Técnicas que coletam e analisam informações disponíveis em fontes públicas.

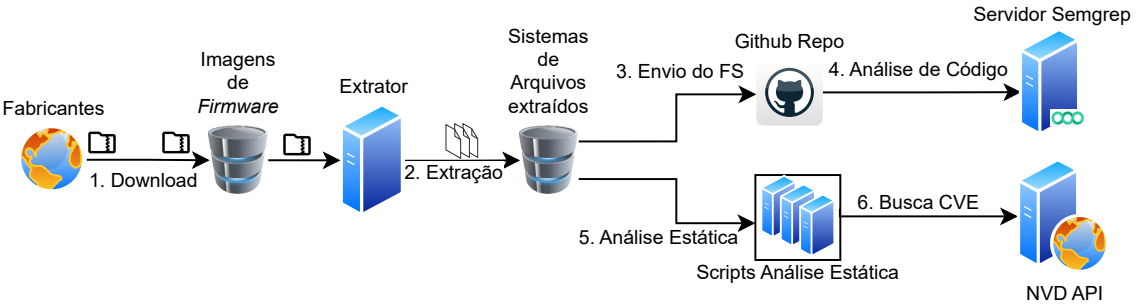


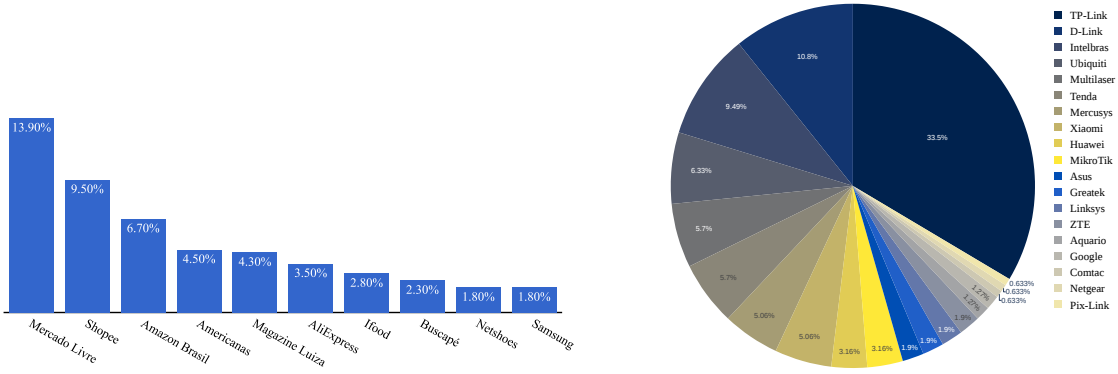
Figura 1. Metodologia

residências e estabelecimentos, como também a indisponibilidade dos *firmwares* dos equipamentos fornecidos pelos provedores de internet, utilizou-se o mesmo método de [ACI 2018] e [Helmke and Dorp 2022], que utilizam os modelos disponíveis à venda no mercado. Assim, empregou-se o critério *Market Share*[2] para a identificação dos modelos considerados na seleção deste trabalho, observando, nos *e-commerces* mais relevantes do país, os equipamentos disponíveis à venda e, portanto, acessíveis aos usuários. Como fonte de pesquisa, foram considerados os principais *e-commerces* de eletrônicos do Brasil (Figura 2(a)), segundo indicativos do *Market Share* considerando o faturamento de vendas, conforme publicado pela *Conversion* [Conversion 2022]. A Figura 2(b) apresenta a participação dos fabricantes no mercado nacional, num total de 19 marcas, sendo TP-Link, D-Link e Intelbras detentoras de mais de 50% dos modelos comercializados.

3.1. Coleta de Evidências

A identificação dos roteadores Wi-Fi SOHO disponíveis no mercado foi realizada com o auxílio de *web crawlers*, utilizando a biblioteca em *python* `Scrapy`, responsáveis por coletar dados como: nome do modelo, fabricante, estatísticas de vendas, avaliações e preços. As coletas realizadas por este estudo foram efetuadas nos meses de Agosto e Outubro de 2022, resultando em 158 modelos de roteadores.

Na sequência tem-se a obtenção do *firmware*, que pode ocorrer através da extração



(a) *Market Share* de *e-commerces* (b) *Market Share* de fabricantes

Figura 2. *Market Share*

[2]Em tradução livre, Parcela de Mercado, é uma porcentagem que corresponde à relevância da sua empresa diante dos competidores da indústria em que ela atua.

fisicamente do aparelho, porém esse procedimento exige a utilização do hardware, o que compromete a escala da pesquisa e inviabiliza a proposta deste estudo. Por esta razão, a busca por fontes digitais para aquisição deste recurso torna-se a melhor estratégia para a proposta desta pesquisa. Assim, o processo de *download* das imagens de *firmware* como fonte exclusivamente os sites oficiais dos fabricantes, evitando assim a aquisição de qualquer imagem modificada por terceiros que pudessem conter artefatos ou configurações estranhas às originalmente encontradas no equipamento. Neste processo, foram consideradas apenas as versões de *firmware* mais recentes para cada modelo, possibilitando a análise no ambiente mais seguro disponibilizado pelos fabricantes.

3.2. Extração das amostras

Após a criação da base local, as imagens foram descompactadas e extraídos o *kernel* e o conteúdo do sistema de arquivos, pois neste estão presentes os arquivos necessários para a condução da análise estática, como binários, arquivos de configuração, *scripts*, além dos arquivos relacionados às páginas web responsáveis pela gerência dos dispositivos. É valido ressaltar que o êxito desta etapa depende muito do projeto de construção do *firmware*, tendo em vista que alguns fabricantes utilizam seus próprios formatos para produção das imagens, em outros casos são utilizados algoritmos de compactação com variações ofuscadas ou criptografadas.

Nesta pesquisa, foi utilizado o extrator do *framework* FirmAE [Kim et al. 2020], que tem como base o *binwalk* [Heffner 2013], capaz de identificar e extrair arquivos de sistemas. O *binwalk* emprega a técnica de identificação de padrões dos cabeçalhos de arquivos para extraí-los da imagem de *firmware*. Durante o processo de extração foram realizadas atualizações dos pacotes internos da imagem *docker* utilizada pelo extrator do *FirmAE*, proporcionando uma melhora de 17% no sucesso da extração dos arquivos.

3.3. Identificação de vulnerabilidades anteriormente reportadas

Este trabalho propõe o mapeamento das vulnerabilidades conhecidas, identificadas pelo CVE ID. As consultas foram realizadas por API[3] disponibilizada pelo *National Vulnerability Database* (NVD), cuja base é sincronizada com a MITRE, responsável por supervisionar o programa CVE. O *Common Vulnerability Scoring System* (CVSS) é um sistema, vinculado ao CVE, capaz de medir a severidade de uma vulnerabilidade.

Como este trabalho focou na análise estática, sem a emulação completa do *firmware*, a detecção da versão dos binários foi processada executando-se o arquivo isoladamente em modo de emulação de usuário - *QEMU User Space Emulation* com a inserção de parâmetros com o intuito de extrair a versão do binário e consultar por CVEs.

Em relação ao *kernel*, a atribuição de CVEs baseada apenas na versão é insuficiente para promover o correto mapeamento das falhas associadas aos dispositivos analisados. A heterogeneidade dos *kernels*, falhas específicas de arquiteturas ou *drivers*, além de *patches* aplicados diretamente pelos fabricantes, são alguns dos desafios desta tarefa. Além disso, há os fatores inerentes ao repositório de CVEs, ocasionando várias ocorrências de falsos positivos, existem situações, por exemplo, em que o CVE foi registrado sem especificar a versão do *kernel* vulnerável. Outro caso recorrente refere-se às falhas que foram reportadas sem considerar a diferença temporal entre diversos

[3]https://nvd.nist.gov/vuln/search

versionamentos do *kernel*, por exemplo, a consulta direta considera que a versão 4.18, de 12/08/2018, é posterior *kernel* 4.4.241, que é de 29/10/2020, fazendo uma equivocada atribuição de falha a uma versão mais recente.

A enumeração de vulnerabilidades para *Kernel* ainda é um problema em aberto e alguns estudos propõem técnicas para obtenção de resultados mais confiáveis. Duas promissoras abordagens apresentadas por [Helmke and Dorp 2022] são *file−based matching* e *commit−based matching*. A primeira considera a identificação dos arquivos comprometidos pela falha e a verificação da ocorrência destes no *kernel* analisado, a segunda avalia os *patches* de correção em cada *release* de *kernel*, uma iniciativa do projeto *Linux Kernel CVEs*[4], porém ambas perspectivas ainda possuem limitações. Esta pesquisa, no entanto, focou em 3 técnicas de mitigação de falsos positivos que consideram: (*i*) a data do CVE em relação ao *release* do *Kernel*; (*ii*) falhas associadas a arquiteturas específicas; e (*iii*) o descarte de resultados em que a falha afeta softwares de terceiros e não o *kernel* especificamente. Portanto, ressalta-se que a proposta desta metodologia visa apresentar estimativas, cujas métricas foram aplicadas igualitariamente em todas as amostras.

3.4. Análise automática de código com ferramenta Semgrep

Com o propósito de melhorar a eficiência da verificação de vulnerabilidades nos *firmwares* coletados, foi utilizada uma abordagem que usa pesquisa de padrões para detectar funções vulneráveis nos arquivos de códigos presentes nos sistemas de arquivos extraídos. Para tal, foi utilizada a ferramenta *open-source* `Semgrep`, responsável por realizar a análise estática de código fonte, compatível com mais de 20 linguagens, que utiliza assinaturas para encontrar possíveis falhas. Segundo [Kluban et al. 2022] a principal vantagem da utilização dessas regras deve-se a transparência e adaptabilidade, permitindo ao `Semgrep` alcançar um índice de 98% de correspondência de funções vulneráveis para o conjunto de dados utilizados em sua pesquisa sobre a medição de funções JavaScript.

4. Análise Descritiva dos *Firmwares*

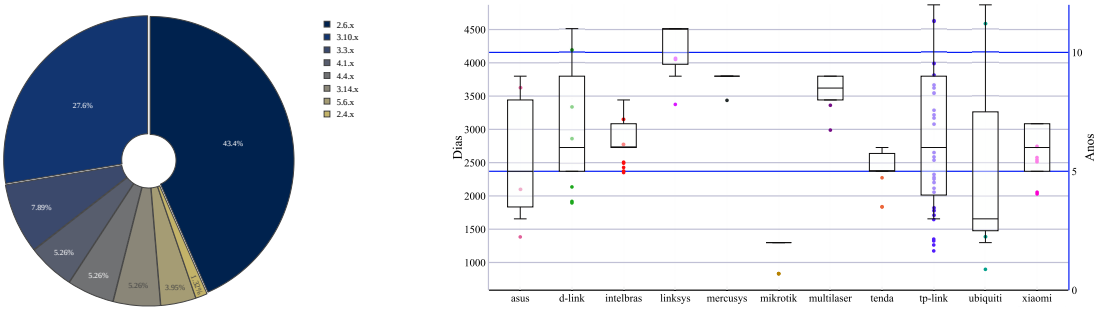
A análise descritiva tem por objetivo descrever e resumir os dados obtidos para o estudo, oferecendo uma percepção mais ampla da base utilizada na pesquisa, quantificando e caracterizando os binários, *kernels* e configurações identificadas nas amostras.

4.1. *Firmwares* oficiais

Foram coletadas 133 imagens de 13 fabricantes, em diversos formatos como *raw* ou compactados (*.gz, .rar e .zip*), somando 1.4 GB de dados. Alguns fabricantes, porém, indisponibilizam os seus *firmwares* para download, em geral, tais empresas adotam a política de atualização *Over-the-air* (OTA), método de distribuição remoto diretamente do desenvolvedor, de maneira automática ou ainda com intervenção do usuário através de algum aplicativo ou interface de gerência do dispositivo.

Devido a algumas medidas de proteção implementadas pelos desenvolvedores, foi possível extrair informações de 80 amostras, uma taxa de 60,15% de sucesso, contemplando 11 dos 13 fabricantes. Dentre os modelos analisados, a arquitetura MIPS predomina em 83,75%, enquanto ARM representa as demais amostras com 16.25%. Todas as imagens processadas utilizam o Linux como sistema operacional embarcado,

[4]https://www.linuxkernelcves.com/



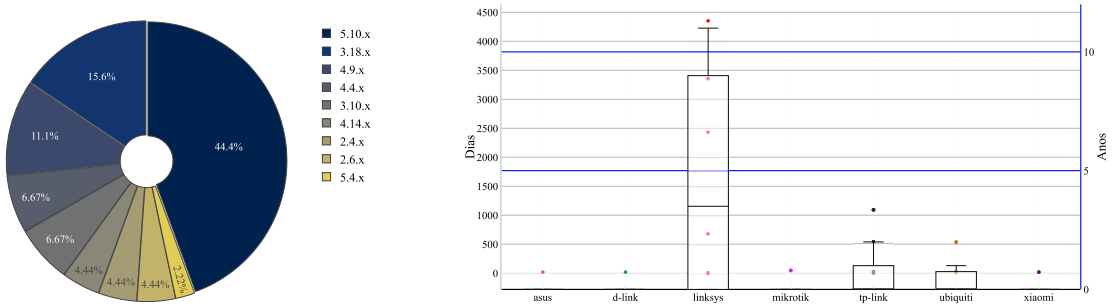
(a) Versões de *Kernel* (b) Diferença temporal entre lançamento do *firmware* e *kernel*

Figura 3. Caracterização dos *Firmwares* Oficiais

com 08 versões distintas de *kernel*. O *kernel* 2.4.20 é a versão mais antiga encontrada e data de 2002, utilizada pelo modelo WRT54G da *Linksys*, enquanto a mais recente é a 5.6.3 de 2020, presente nos modelos da *Mikrotik*. As versões 2.6.x e 3.10.x, cujos suportes encerraram em 2016 e 2017 respectivamente, representam 71% das amostras, Figura 3(a).

Diferentemente do lançamento do *kernel*, a data dos *firmwares* (lançamento do conjunto de *kernel* e todos softwares inclusos na distribuição) permite verificar que a versão mais antiga data de 2010, enquanto que no ano da coleta, 2022, foram lançadas 27 versões. Das imagens processadas, 61,3% foram disponibilizadas entre 2020 e 2022. Por outro lado, a análise da linha do tempo das versões de *kernel* revela que os mais recentes são de 2020. Sendo 2009 o ano com maior representatividade, devido à utilização da versão 2.6.31 em 21 modelos. Há ainda na base, *firmwares* que utilizam *kernel* de 2002 e 2006, por exemplo. A Figura 3(b) projeta um panorama do perfil, por fabricante, entre o lançamento do *firmwares* e a versão do *kernel* utilizada. Em algumas imagens esta diferença atinge 12 anos, já em 48 *firmwares* foi observado que possuem pelo menos 06 anos entre seu lançamento e a versão do *kernel* encontrada. Enquanto o fabricante Mikrotik apresenta média de 02 anos, há 07 outros fabricantes cuja média supera 05 anos, evidenciando a obsolescência do *kernel* utilizado em seus sistemas embarcados. A média atingida pelas amostras foi de 2556 dias entre o *release* do *firmware* e o *kernel* adotado.

Os arquivos `/etc/passwd` e `/etc/shadow` são responsáveis pelo armazenamento das credenciais das contas em sistema Unix. Em uma ação de pós-exploração, estes arquivos são utilizados por atacantes em busca de senhas para utilização em novos acessos remotos. Ao longo da análise foi possível coletar 14 *hashes* distintos de usuários como `admin`, `root`, `user` e `support`, em 36 *firmwares* de 04 fabricantes. Todos os *hashes* utilizam `md5` como algoritmo de criptografia, sendo as senhas mais comuns: `1234`, `admin` e `sohoadmin`. Em geral, os roteadores suportam HTTPS para acesso às páginas *web* de administração, tal funcionalidade é implementada pelo protocolo SSL/TLS. Ocorre que alguns fabricantes geram previamente as chaves privadas no *firmware* do equipamento, logo, para um atacante interceptar esta comunicação, basta obter a imagem do *firmware* e extrair a chave privada, executando o ataque conhecido por *man-in-the-middle*. Em 15 amostras foram encontradas 88 chaves privadas, correspondendo a modelos de 06 fabricantes.



(a) Versões de *Kernel* (b) Diferença temporal entre lançamento do *firmware* e *kernel*

Figura 4. Caracterização dos *firmwares open-source*

4.2. *Firmwares open-source*

A utilização de *firmwares* de código *open-source* emerge como alternativa às versões oficiais, proporcionando ao usuário maior controle dos recursos disponíveis em seus equipamentos e a possibilidade de configuração mais adequada ao perfil de uso. Entre as opções mais populares, estão *DD-WRT*, *OpenWRT* e *Tomato*. O DD-WRT é um projeto com ênfase na estabilidade dos *firmwares* em relação à segurança para diversos roteadores. O OpenWRT, por sua vez, é um projeto mais antigo, sendo o único da lista com binários exclusivamente *non-free*. Como resultado desta estratégia, há diversos roteadores sem o pleno suporte devido à exigência de *drivers non-free* para seu funcionamento. O Tomato apresenta versão mais simplificada e interface mais moderna e amigável, porém conta com menor comunidade de desenvolvimento e lista limitada de modelos suportados, pois é compatível apenas com equipamentos com *chipset* Broadcom.

A fim de comparar os resultados obtidos com os *firmwares* oficiais, foram identificados os modelos suportados pelos projetos *open-source*. Foram obtidos *firmwares* de 02 modelos compatíveis com o Tomato, 14 com o DD-WRT e 30 com o OpenWrt, contemplando 01, 03 e 07 fabricantes, respectivamente. Os modelos E900 e WRT54G, ambos da *Linksys*, foram os únicos suportados pelos 03 projetos. Seguindo a mesma metodologia aplicada às versões oficiais, considerou-se a versão mais recente de cada projeto. As amostras obtidas totalizaram 297 MB, com imagens variando de 1.7 MB a 16.5 MB. Quanto à arquitetura, apenas 01 modelo ARM, o Archer C8 da TP-Link.

As imagens dos *firmwares open-source* apresentaram versões de *kernel* de 09 versões distintas, com destaque para 3.18.x e 5.10.x, que representam 60% do repositório, Figura 4(a). A mais antiga delas é 2.4.37.9, lançada em 01/02/2010, presente no *firmware* do modelo WRT54G, enquanto a mais recente é a 5.10.146 encontrada nos modelos com a versão 22.02.3 do *OpenWrt*, disponibilizada em 28/09/2022. Dentre as amostras coletadas, 93% foram lançadas entre 2020 e 2022. As exceções a esta estatística aplicam-se aos modelos mais antigos, WRT54G e E900 da *Linksys*. A análise do quão atualizado é o *kernel* em relação ao *release* do *firmware*, revelou que OpenWrt apresentou um perfil mais consistente entre os três projetos. A maior diferença foi observada na versão 19.07.10, que utiliza um *kernel* 4.14.275 com 48 dias de lançamento, enquanto que no OpenWrt 21.02.0 o *kernel* 5.4.143 havia sido lançado há apenas 09 dias. O DD-WRT possui resultados que variam de 05 dias a 07 anos de diferença, o Tomato utilizou versões de *kernel* lançadas entre 06 e 09 anos da data de compilação seus *firmwares*. Ao observar o

suporte dado pelo Tomato às versões atuais, esse é o perfil de apenas a 02 modelos. Porém, a média entre o lançamento do *firmware* e o *kernel* utilizado, considerando todas amostras dos 03 projetos, ficou em 411 dias, Figura 4(b), mais de 6 vezes menor que a média dos *firmwares* oficiais. Diferentemente dos softwares oficiais, foram irrecuperáveis *hashes* de senhas e chaves privadas, sinalizando coerência com as boas práticas de segurança da informação.

5. Resultados e Discussões

A execução da análise estática produziu resultados em 03 categorias: binários, *Kernel* e código fonte. A identificação dos binários e do *kernel* visou a enumeração de vulnerabilidades conhecidas, através dos CVEs. A análise de código fonte, que se baseou, essencialmente, no conteúdo das páginas web, foi processada pela ferramenta `Semgrep`, que identificou possíveis vulnerabilidades em comparação com padrões conhecidos.

5.1. Análise Estática nos *Firmwares* Oficiais

O processo automatizado de coleta das versões e consulta por CVEs foi aplicado a 5894 binários dos quais 84 binários são vulneráveis e totalizam 1474 vulnerabilidades conhecidas. Daqueles comprometidos, o lançamento de 88,1% ocorreu há mais de 04 anos. Aqui destaca-se o `openssl` 0.9.8, disponibilizado em 05/07/2005, atualmente com 96 vulnerabilidades, segundo base de dados do NVD, encontrado no *firmware* V1.06B1 do modelo DIR-859 da D-Link e no *firmware* v201214 do Archer C20 V5 da Tp-Link.

Os binários vulneráveis mais comuns foram o `busybox`, `dnsmasq`, `iptables`, `openssl` e `openvpn`. A Figura 5 mostra a média das vulnerabilidades encontradas nos binários. As partes superiores das Figura 5, 6, 7 e 8 apresentam *boxplots* construídos com os valores das distribuições médias das vulnerabilidades agrupados por fabricante. Em números absolutos, 12 modelos apresentaram ao menos 10 falhas críticas, sendo todos da TP-Link. Por sua vez, considerando a média por aparelho, a Asus surge com 07 falhas, seguida pela TP-Link com 06 ocorrências de maior severidade. Nos grupos das vulnerabilidades classificadas com criticidade Alta e Média, a Xiaomi apresentou média de 27 e a TP-link 44 falhas, respectivamente. A marca TP-Link liderou a lista com 09 dos 10 modelos mais vulneráveis: TL-WR941HP, Archer C60 V3, Archer C1200, Archer C8, Deco M5, Deco M4, Deco M9 Plus, Deco E4, Archer C7, e D-Link DIR-846.

Mais que a periodicidade dos fabricantes no lançamento de atualizações de seus *firmwares*, o que tem se mostrado ineficaz, as evidências deste artigo indicam que essa tendência estatística é fruto da utilização de binários desatualizados nos *builds* das imagens. Por exemplo, o modelo Archer C7 V5 da TP-Link possui a versão de *fimware* mais recente do repositório, lançada em 08/11/2022, porém utiliza versões do `proftpd` de 2011, `busybox` de 2012, `openssl` de 2015 e o *kernel* 3.3.8 de 2012, i.e., *firmware* com conteúdo desatualizado e inseguro. Como consequência, vulnerabilidades famosas relacionadas ao `openssl` como *Heartbleed* (`www.heartbleed.com/`), CVE-2014-0160, foram encontrada em 04 modelos, *Poodle* (`www.openssl.org/~bodo/ssl-poodle.pdf`), CVE-2014-3566, em 06 aparelhos e *Freak Attack* (`www.freakattack.com/`), CVE-2015-0204, em 05 dispositivos.

A análise das versões de *kernel* revelou o mesmo padrão verificado nos binários, em que o *firmware* é lançado com versões antigas e até mesmo descontinuadas do *core*

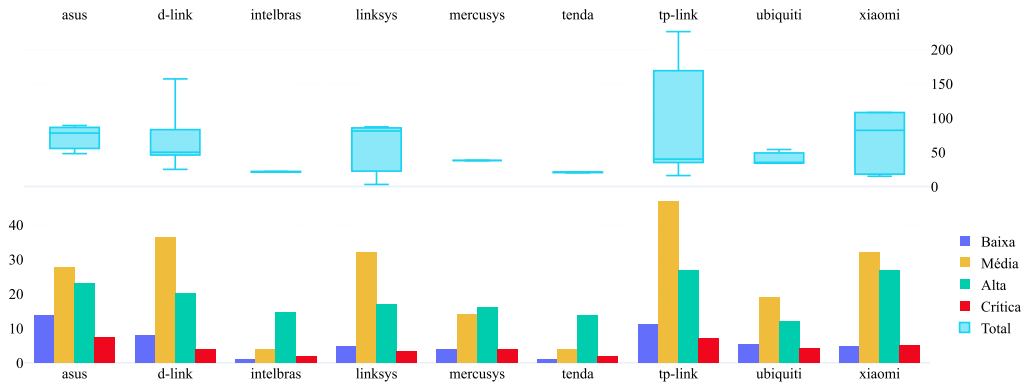


Figura 5. Média das vulnerabilidades do binários por fabricante em *firmwares* oficiais

do sistema. O menor intervalo observado foi de 02 anos, no modelo *UniFi UAP* da Ubiquiti e nos modelos da Mikrotik, mas há casos como os modelos EAP110 e EAP115 da TP-link cuja diferença é de 12 anos. Foi observada uma média de 1344 CVEs de *kernel* por modelo analisado. A Figura 6 apresenta uma estimativa da média das vulnerabilidades existentes nos modelos de cada fabricante. Algumas medidas foram adotadas para minimizar a ocorrência de falsos positivos, mas servem como indicadores dos riscos relativos à implementação de *kernel* adotada.

5.2. Análise Estática nos *Firmwares open-source*

Dentre as 46 imagens obtidas, foram processados 2168 binários distintos, sendo apenas 15 amostras com falhas conhecidas, compreendidas por versões dos binários `busybox`, `dnsmasq`, `iptables` e `openvpn`, totalizando 142 vulnerabilidades. O binário com maior ocorrência de falhas é o `dnsmasq 2.55`, encontrado na versão 10.03.1 do OpenWrt, última compatível com o Linksys WRT54G, com 22 CVEs associados, sendo 03 críticos. Este caso, no entanto, difere-se do ao perfil predominante no projeto, que inclui em suas imagens versões atualizadas dos binários. Pode-se mencionar a sua versão mais recente 22.03.1, na qual foi encontrado apenas 01 binário vulnerável, que é o `busybox 1.35.0`, em sua última versão disponível, ainda sem *patch*.

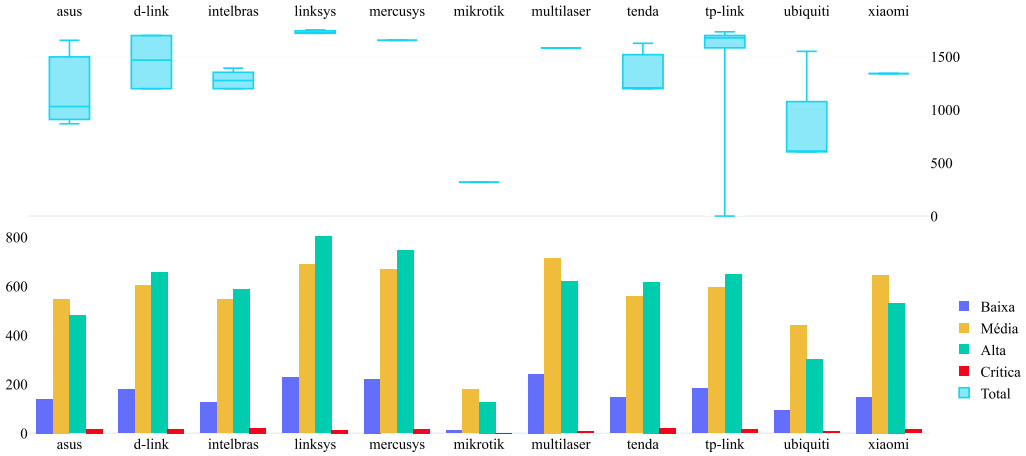


Figura 6. Média das vulnerabilidades de *kernel* em *firmware* Oficiais

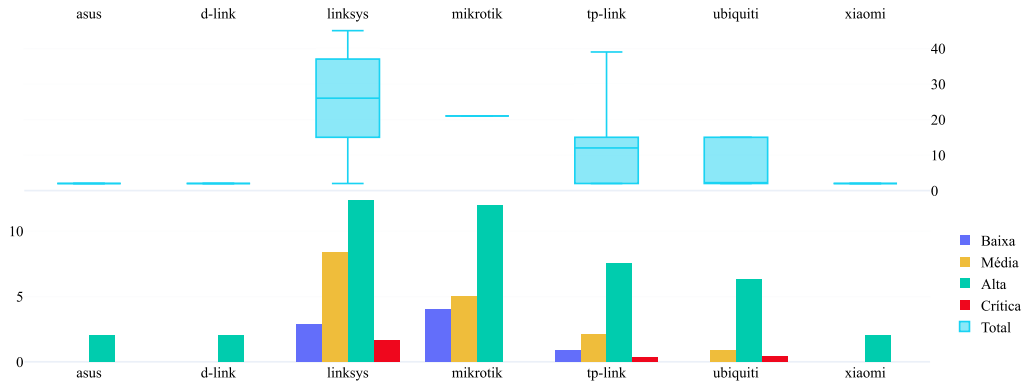


Figura 7. Vulnerabilidade média nos binários dos *firmwares open-source*

Os *firmwares open-source*, por sua vez, apresentam resultados mais favoráveis à segurança da informação em relação aos oficiais. A média total de vulnerabilidades nos modelos são inferiores, nos modelos da Xiaomi este valor reduziu 97%, a Asus 96,8%, a Ubiquiti 81,5%, a TP-Link em 66,6% e a Linksys, com os modelos mais antigos da análise, apresentou redução de 55%, tornando a superfície de ataque mais restrita.

A Figura 7 apresenta as médias das vulnerabilidades dos binários, por fabricante considerando os 03 projetos. Apesar da existência de alguns modelos antigos que elevam os índices, ainda assim os resultados são bem inferiores aos existentes nas versões oficiais. Apenas nos modelos da *Linksys, Tp-Link e Ubiquiti* foram encontradas vulnerabilidades críticas. Uma análise destes indicadores mostrou que tais falhas estão relacionadas a vulnerabilidades de *heap* ou *buffer overflow*, geralmente associadas à possibilidade de execução remota de comando. Seguindo a tendência observada na análise dos binários, a utilização de versões de *kernel* atualizadas fundamenta a constituição de um ambiente mais seguro, Figura 8. A redução de vulnerabilidades ocorreu em todos os modelos analisados e foi ainda mais expressiva do que a comparação dos binários. Os modelos da Asus diminuíram 98,42% das vulnerabilidades totais, D-Link 98,22%, Xiaomi 98,05%, TP-Link 96,23%, Ubiquiti 75,45%, Mikrotik 68,5% e Linksys 46,5%.

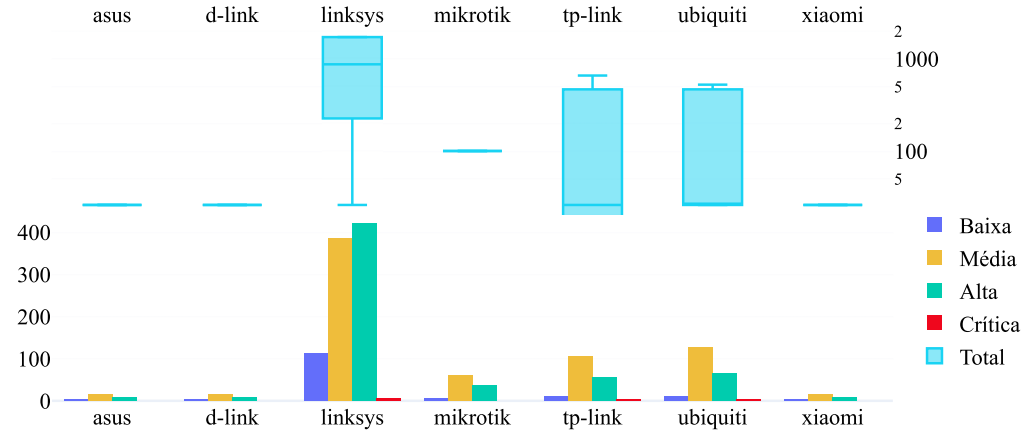


Figura 8. Vulnerabilidade média nos *kernels* dos *firmwares open-source*

5.3. Relatório da ferramenta Semgrep

A ferramenta reportou na análise dos *firmwares* oficiais e *open-source*, respectivamente, os valores de 3921 e 917 possíveis falhas que precisam passar por validação as evidências. Essas falhas são categorizadas conforme *Common Weakness Enumeration* (CWE) que é uma lista desenvolvida pela MITRE e visa enumerar os tipos comuns de vulnerabilidades.

Nesse contexto, a Tabela 1 apresenta os CWE presentes no Top 25 MITRE que foram resultados encontrados na análise. A maior ocorrência foi o CWE-79, relacionado a *Cross-site Scripting* e ocupa o segundo lugar no *Ranking* MITRE, correspondendo a cerca de 41,3% dos achados relacionados pelo `Semgrep`. Em relação aos *firmwares open-source* o mesmo CWE-79 também é o que lidera com mais achados representando cerca de 44,8%. Os CWEs possuem um sistema de pontuação que utiliza em seu cálculo a criticidade de uma vulnerabilidade a fim de identificar qual a probabilidade de um invasor explorá-la com sucesso. Assim, este trabalho utilizou a probabilidade de exploração de um CWE para classificar os riscos associados aos *firmwares* da base de dados.

A Figura 9(a) expõe a distribuição da média da criticidade dos CWE encontrados pelo `Semgrep` por fabricante. Sua análise permite observar que as evidências nos códigos do fabricante Asus possuem maior nível de criticidade nos *firmwares* oficiais. A Figura 9(b) mostra os resultados da análise nas amostras *open-source*, que possuem médias mais uniformes entre os fabricantes. A parte superior da Figura 9 apresenta boxplots construídos com os valores das distribuições médias das probabilidades de exploração dos CWE por fabricante. Em comparação com os *firmwares* oficiais, verifica-se considerável redução nas evidências encontradas, indicando códigos potencialmente mais seguros nas versões *open-source*, com exceção ao fabricante Ubiquiti, cuja média subiu de 8 evidências em seus *firmwares* oficiais para 19 na versão de código aberto.

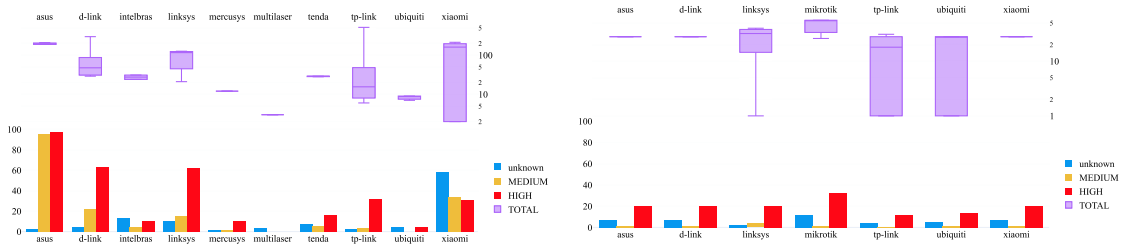
Durante a análise manual das evidências reportadas pelo `Semgrep` foi encontrada uma nova vulnerabilidade *(zero-day)*[5], que possibilita a execução remota de comandos com privilégios de super-usuário `root` no roteador DIR-846 do fabricante D-LINK.A validação desta vulnerabilidade foi verificada em um roteador físico. Neste caso, a partir de um usuário autenticado na página web de administração do roteador é possível realizar a injeção de código, por falta de sanitização na aplicação. A falha em questão, está presente no arquivo `SetIpMacBindSettings.php` que contém a função `exec`, a qual recebe uma variável que é manipulada pelo usuário. Assim, um invasor pode executar comandos arbitrários enviando uma carga maliciosa por meio de uma requisição `POST`. Por fim, como boas práticas as informações relacionadas foram transmitidas ao fabricante para atualização, bem como foi registrado o CVE-2022-46552 [Mitre 2023].

Tabela 1. CWE inclusos no Top 25 MITRE encontrados nos *firmwares* oficiais. Criticidade refere-se a *Likelihood of Exploit* (Probabilidade de Exploração).

Oficiais	Criticidade	#	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	Criticidade	#	Top 25
CWE-79	HIGH	411	2
CWE-20	HIGH	227	4
CWE-798	*unknown*	108	15

[5]É uma nova vulnerabilidade, recém-descoberta, ainda desconhecida do fabricante ou desenvolvedor.



(a) *Firmwares* Oficiais (b) *Firmwares open-source*

Figura 9. Distribuição da média da probabilidade de exploração dos CWE

5.4. Discussões

Uma prática fortemente recomendada aos usuários, visando reduzir a superfície de ataque em suas redes é a atualização do *firmware*, embora seja parcialmente eficaz para garantir a plena segurança dos roteadores, visto que os fabricantes ainda utilizam binários e *kernel* obsoletos no lançamento novas versões. É importante que seja consultada regularmente a disponibilidade de novas versões para o dispositivo e assim aplicada a devida substituição do software. Ainda é notória a utilização de credenciais padrão, e a respectiva reconfiguração na página de administração do roteador é mandatória para reduzir as chances de um invasor ter acesso imediato à gerência da rede local. Observou-se ainda, a disponibilidade de dispositivos fora de linha *End-of-Service-Life* (EoSL). Portanto, a observância de marca, modelo e ciclo de vida do produto deve ser considerada. Dispositivos EoSL pararam de ser comercializados oficialmente por seus fabricantes, a consequência disso é a falta de atualização oficial dos *firmwares* para estes modelos, sentenciando os usuários à permanência de um sistema potencialmente mais vulnerável.

Uma alternativa estratégica para minimizar a necessidade de custos de substituição de novos equipamentos ou a utilização de *firmwares* oficiais extremamente vulneráveis é a adoção de *firmwares open-source*. Ainda que os projetos faltem com o suporte a todos os modelos disponíveis, a lista de equipamentos compatíveis é vasta suficiente para que os usuários possam encontrar aqueles que atendam às suas demandas. Este trabalho revelou que os índices de obsolescência e vulnerabilidades encontrados nestes substitutos são bem inferiores às versões oficiais. Outra vantagem é a possibilidade de atualizar os binários à medida que a comunidade disponibiliza novas versões dos pacotes nos repositórios, minimizando ainda mais a ocorrência de vulnerabilidades no dispositivo.

6. Conclusões

Este trabalho identificou o conjunto de roteadores *SOHO* mais comercializados no Brasil através da análise do *market share* dos principais sites *e-commerces* de eletrônicos. Nesse contexto, analisando esse conjunto de dados com foco em segurança cibernética foi possível confirmar que os *firmwares* disponibilizados pelos fabricantes possuem nível segurança menor do que as versões *open-source*. Soma-se ainda como resultado dessa pesquisa que sistema de gerenciamento web possui falhas e precisa ser melhor avaliado, com pouco esforço foi possível achar um *zero-day*. Como trabalhos futuros, pretende-se ampliar a análise de código fonte, com ênfase na web, aperfeiçoar heurísticas para detecção de versão dos binários, utilizar diferentes abordagens para extração de *firmware* e expandir a análise para escala global.

Agradecimentos

Este trabalho tem apoio financeiro do Programa de Pós-graduação em Aplicações Operacionais—PPGAO/ITA, da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) processo #2020/09850-0, e da CAPES. Apoio CNPq proc. #313641/2020-0.

Referências

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. acessado em 26/12/2022.

Conversion (2022). E-commerce no brasil: conheça os principais dados, o market share, o crescimento e as principais estatísticas, com atualização mensal! https://www.conversion.com.br/blog/relatorio-ecommerce-mensal/. acessado em 10/11/2022.

Feng, X., Zhu, X., Han, Q.-L., Zhou, W., Wen, S., and Xiang, Y. (2022). Detecting vulnerability on iot device firmware: A survey. *IEEE/CAA Journal of Automatica Sinica*, pages 1–17.

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.

Heffner, C. (2013). Github repository: Binwalk. https://github.com/ReFirmLabs/binwalk. Publicado em 11/11/2013; acessado em 11/09/2022.

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, 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 *Annual Computer Security Applications Conference (ACSAC)*, Online.

Kluban, M., Mannan, M., and Youssef, A. (2022). On measuring vulnerable javascript functions in the wild. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '22, page 917–930, New York, NY, USA. ACM.

Mitre (2023). CVE-2022-46552. Available from MITRE, CVE-ID CVE-2022-46552. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-46552.

Networks, P. A. (2020). 2020 unit 42 iot threat report. acessado em 30/12/2022.

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.

Redini, N., Machiry, A., Wang, R., Spensky, C., Continella, A., Shoshitaishvili, Y., Kruegel, C., and Vigna, G. (2020). Karonte: Detecting insecure multi-binary interactions in embedded firmware. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1544–1561.

Toso, G. and Pereira, L. A. (2021). Enumeração de sistemas operacionais e serviços de firmwares de roteadores sem-fio. In *Anais Estendidos do XXI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, pages 178–191, Porto Alegre, RS, Brasil. SBC.

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/. acessado em 05/01/2023.