Uma abordagem para mitigação de *phishing* utilizando eBPF/XDP

Pedro Martins dos Santos[1], **Jéferson Campos Nobre**[1]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{jcnobre,pmsantos}@inf.ufrgs.br

Abstract. *Although widely studied as a cyberattack vector, phishing remains a prevalent threat, driven by its operational simplicity and high effectiveness in spreading malware. This scenario makes it crucial to use robust solutions to block communications through malicious domains. This paper presents an efficient approach to mitigate phishing-based attacks and command and control (C2) communications using eBPF technology and the XDP framework. The study presents a comparative analysis between the proposed solution and the RPZ method. The results demonstrate a reduction in average CPU consumption of 70% with an average increase in latency of 0.0126 ms, presenting a promising alternative to improve the security and performance of DNS services..*

Resumo. *Embora amplamente estudado como vetor de ataque cibernético, o phishing continua sendo uma ameaça predominante, impulsionado por sua simplicidade operacional e alta eficácia na disseminação de malware. Este cenário torna crucial a utilização de soluções robustas para bloquear comunicações através de domínios maliciosos. Este artigo apresenta uma abordagem eficiente para mitigar ataques baseados em phishing e comunicações de comando e controle (C2) utilizando a tecnologia eBPF e o framework XDP. O estudo apresenta uma análise comparativa entre a solução proposta e o método RPZ. Os resultados demonstram uma redução média do consumo de CPU de 70% com incremento médio na latência de 0,0126 ms, apresentando uma alternativa promissora para aprimorar a segurança e o desempenho dos serviços DNS.*

1. Introdução

A Internet consolidou-se como infraestrutura vital da sociedade contemporânea, proporcionando benefícios transformadores que vão desde a facilidade de comunicação até o acesso instantâneo à informação. Neste cenário, o DNS (*Domain Name Server*) desempenha um papel fundamental, permitindo que os usuários possam se conectar a sites e aplicativos ao traduzir domínios em linguagem humana para os endereços IP numéricos que fundamentam toda a comunicação em rede. Paralelamente à expansão e integração da Internet em praticamente todos os aspectos da vida moderna, o panorama de ameaças cibernéticas evoluiu em complexidade e sofisticação, criando desafios sem precedentes para organizações de todos os portes. Neste cenário de vulnerabilidade crescente, agentes maliciosos aprimoraram suas ações, desenvolvendo técnicas refinadas como o *phishing*, através do qual exploram domínios fraudulentos elaborados para capturar credenciais e informações sensíveis. Além disso, esses domínios maliciosos frequentemente servem

como infraestrutura crítica para operações de *botnets*, estabelecendo canais de comando e controle que permitem aos atacantes orquestrar remotamente o vazamento de dados confidenciais e executar ações maliciosas. Esta simbiose entre domínios maliciosos e infraestruturas de ataque representa um dos maiores desafios contemporâneos à segurança cibernética.

Esta crescente sofisticação dos vetores de ataque se manifesta de forma particularmente preocupante nas estatísticas recentes sobre incidentes de segurança. O *phishing* persiste como um dos mecanismos de comprometimento mais prevalentes no ecossistema de segurança cibernética contemporâneo, demonstrando a eficácia desta técnica. Dados estatísticos recentes evidenciam a magnitude desta problemática: aproximadamente 86% das organizações registraram ao menos uma tentativa de acesso a domínios de *phishing* por parte de seus colaboradores, enquanto 70% das entidades corporativas detectaram usuários expostos a anúncios contendo conteúdo malicioso. Adicionalmente, pesquisas indicam que 48% das empresas identificaram atividades de *malware* especificamente orquestradas para a subtração de dados confidenciais, conforme levantamento realizado pela plataforma de segurança Cisco Umbrella, cuja base analítica compreende mais de 24.000 organizações distribuídas em 190 países [Umbrella 2022]. Corroborando esta tendência preocupante, análises de tráfego DNS conduzidas pela Akamai revelaram que aproximadamente 13% dos dispositivos monitorados tentaram estabelecer comunicação com domínios associados a *malware*. O mesmo estudo identificou que 26% dos dispositivos comprometidos interagiram com domínios vinculados a Infraestruturas de Ataque Inicial (IABs), enquanto 9% dos sistemas infectados estabeleceram conexões com domínios relacionados a operadores de *Ransomware-as-a-Service* (RaaS), evidenciando a sofisticação e a estruturação dos ecossistemas criminosos contemporâneos [Akamai 2023].

Estas estatísticas alarmantes sobre incidentes relacionados a domínios maliciosos evidenciam uma vulnerabilidade sistêmica que merece atenção: a exploração do protocolo DNS como vetor deste tipo de ataque. Esta vulnerabilidade ocorre, principalmente, devido às fragilidades inerentes ao próprio protocolo DNS, que não foi originalmente concebido com robustos mecanismos de segurança, somado ao fato de que o tráfego DNS frequentemente escapa à inspeção das ferramentas de segurança convencionais, como firewalls de rede. Diante deste cenário de ameaças, o serviço de resolução de nomes emerge como componente estratégico na arquitetura de segurança corporativa, potencializando a mitigação de riscos cibernéticos através da implementação de DNS Firewalls. Esta abordagem proativa permite não apenas a detecção precoce de ameaças, mas também impede efetivamente que ativos já comprometidos estabeleçam comunicação com infraestruturas de comando e controle externas, interrompendo assim o ciclo de vida do ataque antes do vazamento de dados ou da propagação lateral. Adicionalmente, a implementação de controles no nível DNS proporciona uma visibilidade ampliada sobre os padrões de comunicação dos elementos da rede, complementando significativamente as capacidades das soluções de segurança tradicionais e fortalecendo a postura de segurança organizacional contra as sofisticadas técnicas de ataque evidenciadas nas estatísticas apresentadas.

A evolução das pesquisas na área de prevenção a ataques baseados em domínios maliciosos tem seguido predominantemente duas vertentes metodológicas: análise passiva e análise ativa do tráfego DNS [Khormali et al. 2021]. Embora estas abordagens apresentem contribuições relevantes para a detecção precoce de ameaças, o advento do

plano de dados programável nos oferece novas fronteiras tecnológicas a serem exploradas, permitindo o descarregamento estratégico de parte da computação diretamente para a camada de rede. Diversos estudos têm investigado o potencial da tecnologia eBPF (*extended Berkeley Packet Filter*) para o desenvolvimento de soluções avançadas voltadas à segurança cibernética. Entre as implementações existentes, destacam-se iniciativas focadas na proteção contra ataques volumétricos que visam o esgotamento de recursos do serviço DNS. Por exemplo, [Kostopoulos et al. 2020] apresenta uma solução baseada em XDP (*eXpress Data Path*) para a proteção de servidores DNS autoritativos, enquanto a [Bertin 2017] implementa esta mesma tecnologia para realizar mitigação e filtragem de ataques DDoS (*Distributed Denial of Service*) em sua infraestrutura de CDN (*Content Delivery Network*). Contudo, apesar destes avanços, existem poucos estudos que exploram o potencial da tecnologia eBPF especificamente para implementação de DNS Firewalls em servidores resolvedores, com o objetivo explícito de controlar a comunicação de *botnets*, mitigar ataques de *phishing* e interromper canais de comando e controle de malwares que utilizam domínios maliciosos como infraestrutura operacional. No cenário atual, os serviços de resolução DNS mais amplamente implementados, tanto em software livre como BIND, PowerDNS e Unbound, quanto em soluções proprietárias como Infoblox e Bluecat, fundamentam suas funcionalidades de DNS Firewall no mecanismo RPZ (*Response Policy Zones*) [Vixie and Schryver 2017].

Para abordar diretamente esta lacuna identificada na literatura científica e superar as limitações das tecnologias atuais, este trabalho propõe um mecanismo de DNS Firewall para bloqueio eficiente de resoluções DNS direcionadas a domínios maliciosos. A solução desenvolvida integra as tecnologias eBPF e XDP, aproveitando a capacidade de programabilidade do plano de dados. O XDP atua especificamente na camada mais baixa da pilha de rede, permitindo que os pacotes DNS sejam interceptados e analisados imediatamente após serem recebidos pela interface de rede, antes de qualquer processamento adicional pelo kernel. É importante ressaltar que o mecanismo proposto opera com base em uma lista de domínios maliciosos previamente carregada no sistema, não realizando análise dinâmica ou heurística para detecção de novas ameaças em tempo real. Esta estratégia de implementação elimina a necessidade de consultas adicionais para verificação de legitimidade, concentrando-se na eficiência do bloqueio de domínios já conhecidos como maliciosos. Embora o escopo atual seja focado em uma abordagem determinística baseada em listas, o *framework* desenvolvido estabelece as fundações para evoluções futuras promissoras, incluindo a incorporação de mecanismos de análise dinâmica e algoritmos heurísticos para detecção em tempo real de domínios maliciosos emergentes, capacidades que poderão ser exploradas em trabalhos subsequentes. Esta capacidade de processamento precoce, que ocorre antes mesmo que os pacotes alcancem as camadas superiores do kernel, representa uma vantagem significativa sobre metodologias convencionais. Diferentemente das abordagens tradicionais baseadas em RPZ, que demandam consultas que competem por recursos computacionais, a solução proposta opera diretamente no fluxo de processamento de pacotes, otimizando drasticamente o consumo de recursos do sistema e proporcionando um mecanismo de proteção substancialmente mais eficiente contra ameaças baseadas em domínios maliciosos.

Este artigo está organizado da seguinte maneira. Na Seção 2 é apresentada a fundamentação teórica. Na Seção 3 são discutidos os trabalhos relacionados. Na Seção 4 são apresentados os detalhes da implementação da arquitetura proposta neste trabalho,

enquanto na Seção 5 são descritos os experimentos realizados, além de seus resultados. Por fim, a conclusão é apresentada na Seção 6.

2. Fundamentação Teórica

Esta seção apresenta os conceitos fundamentais necessários para a compreensão do trabalho desenvolvido. Inicialmente, aborda-se a infraestrutura de resolução de nomes, explorando o funcionamento do DNS, suas extensões de segurança (DNSSEC) e mecanismos de controle de resposta (RPZ), elementos essenciais para o entendimento da arquitetura proposta. Em seguida, são discutidos os fundamentos de programabilidade no kernel, com foco no *Extended Berkeley Packet Filter* (eBPF) e no *eXpress Data Path* (XDP), tecnologias que proporcionam execução eficiente de código no nível do kernel e processamento de alta performance para pacotes de rede. Estes componentes teóricos estabelecem a base conceitual necessária para as contribuições apresentadas nas seções subsequentes.

2.1. Infraestrutura de Resolução de Nomes: DNS, Extensões de Segurança e Mecanismos de Controle

O *Domain Name Server* (DNS) é essencial para a operação da Internet, atuando como um mecanismo que traduz nomes de domínio legíveis em endereços IP, facilitando a identificação e localização de dispositivos em uma rede [Kurose and Ross 2022]. Sua arquitetura é formada por uma base de dados distribuída e resolvedores que consultam e fornecem informações sobre domínios através de servidores de nomes, responsáveis pelo armazenamento de dados específicos em arquivos de zona contendo Registros de Recursos (RR). Para garantir resiliência e balanceamento de carga, cada domínio deve possuir pelo menos um servidor de nomes primário e um secundário [Liu and Albitz 2006].

Entretanto, a infraestrutura DNS apresenta vulnerabilidades significativas que motivaram o desenvolvimento de extensões de segurança. O DNSSEC (*DNS Security Extensions*), introduzido em 1997 [Arends et al. 2005], estabelece especificações que fortalecem a segurança do DNS através de criptografia assimétrica, permitindo a autenticação inequívoca da origem dos dados e garantindo sua integridade durante a transmissão. Essas medidas mitigam vulnerabilidades críticas amplamente exploradas, como envenenamento de cache e técnicas de *spoofing*. É importante ressaltar que o DNSSEC não provê confidencialidade nas comunicações ou proteção contra ataques de negação de serviço, concentrando-se especificamente na autenticação e integridade dos dados.

Complementando as extensões de segurança, o RPZ (*Response Policy Zone*) constitui um mecanismo que permite aos administradores de servidores DNS capacidade de controlar as respostas a consultas DNS conforme necessidades específicas [Vixie and Schryver 2017]. O RPZ foi desenvolvido pelo *Internet System Consortium* (ISC) e inicialmente integrado como componente do serviço BIND. Atualmente, sua configuração é viável em qualquer DNS de código aberto baseado no BIND a partir da versão 9.8, além de ser a base para soluções comerciais proprietárias como Infoblox e Bluecat. A operacionalização do RPZ fundamenta-se na criação de uma zona DNS especializada que armazena registros definidores de políticas de resposta específicas para domínios identificados como vetores de atividades maliciosas, incluindo *phishing*, distribuição de *malware* e infraestruturas de comando e controle. Estas políticas baseiam-se em listas curadas de domínios comprometidos, fornecidas por serviços especializados em inteligência de ameaças cibernéticas.

Os sistemas de resolução de nomes de domínio que utilizam RPZ implementam diferentes tipos ações para gerenciar respostas em consultas DNS e garantir a segurança da rede. A resposta NXDOMAIN (*Non-Existent Domain*) indica diretamente ao solicitante que o domínio consultado não existe. Por sua vez, a resposta CNAME (*Canonical Name*) implementa um redirecionamento do usuário para uma notificação específica, funcionando como um importante mecanismo de comunicação com os usuários finais sobre políticas de acesso ou questões de segurança. A ação DROP representa uma abordagem mais radical, caracterizada pela supressão completa da resposta à consulta DNS. Esta medida estratégica integra frequentemente políticas de segurança avançadas para bloquear comunicações potencialmente maliciosas, interrompendo completamente o canal de comunicação antes mesmo que qualquer resposta seja gerada. Independentemente da estratégia adotada, todas estas respostas demandam processamento por parte do serviço DNS, consumindo recursos computacionais que poderiam ser alocados para o atendimento de consultas legítimas.

2.2. Programabilidade no Kernel: Extended Berkeley Packet Filter e eXpress Data Path

O eBPF representa uma evolução significativa do clássico *Berkeley Packet Filter* (BPF) [McCanne and Jacobson 1993], originalmente concebido como um simples filtro de pacotes com instruções bem definidas. O eBPF expandiu este conceito, introduzindo novas instruções que permitem a execução segura de código diretamente no kernel do sistema operacional, eliminando a necessidade de modificações no código-fonte do kernel ou carregamento de módulos externos.

Uma característica fundamental do eBPF é sua orientação a eventos, possibilitando uma visão ampla e detalhada do sistema através de diversos pontos onde os programas podem ser anexados e acionados. Esta característica torna o eBPF particularmente valioso para aplicações de observabilidade, segurança e funções avançadas de rede, como balanceamento de carga[Community 2024].

Os programas eBPF podem ser desenvolvidos em linguagens de alto nível como C, Go e P4. Quando escritos em C, são compilados utilizando a biblioteca libbpf, que insere o programa no kernel via chamada de sistema (*syscall*). Antes da execução, o código passa pelo Verificador eBPF, garantindo sua segurança, e posteriormente o compilador JIT (*Just in Time*) traduz o bytecode genérico para instruções de máquina específicas. A integração dos programas eBPF com o kernel ocorre em pontos específicos denominados *Hooks*, permitindo interceptar e processar eventos do sistema. Dependendo do tipo de *hook* utilizado, é possível interceptar pacotes de rede em diferentes estágios do processamento, monitorar chamadas de sistema, observar funções específicas do kernel e coletar métricas de desempenho.

Um componente essencial da arquitetura eBPF são os mapas BPF, estruturas que compartilham informações entre o espaço do kernel e o espaço do usuário. Estes mapas armazenam dados persistentemente no modelo *key/value* e são acessíveis por diferentes programas eBPF em ambos os contextos[Community 2024]. Entre os principais tipos disponíveis de mapas, destacam-se o BPF_MAP_TYPE_HASH para armazenamento de propósito geral com suporte a chaves e valores compostos, o BPF_MAP_TYPE_LRU_HASH que incorpora funcionalidade de exclusão automática das

entradas menos utilizadas, e o BPF_MAP_TYPE_ARRAY que oferece estrutura vetorial para armazenamento genérico. A manipulação destes mapas é realizada através de funções auxiliares específicas como bpf_map_update_elem, bpf_map_delete_elem e bpf_map_lookup_elem.

Complementando o ecossistema de programabilidade no kernel, o eXpress Data Path (XDP) emerge como um framework que opera em conjunto com o eBPF, executando programas diretamente no contexto do kernel antes da manipulação dos pacotes de dados [Høiland-Jørgensen et al. 2018]. Esta abordagem possibilita o processamento de pacotes no nível mais próximo de sua recepção pelo *hardware*, antes que o kernel realize qualquer tratamento, resultando em alto desempenho.

O XDP implementa três modos distintos de operação, cada um oferecendo diferentes compromissos entre desempenho e compatibilidade. No modo nativo (*Native Mode*), os programas são executados diretamente no *driver* de rede, interceptando os pacotes antes de sua entrada no kernel, o que minimiza a latência e maximiza a vazão. O modo genérico (*Generic Mode*) processa os pacotes através do kernel mantendo a lógica dentro do *framework* eBPF, servindo como alternativa para dispositivos sem suporte ao modo nativo, embora com desempenho inferior. Por fim, o modo *offload* permite a execução dos programas diretamente em *hardware* de rede especializado como SmartNICs, proporcionando desempenho superior aos demais modos graças à descarga do processamento para hardware dedicado.

3. Trabalhos relacionados

A pesquisa em segurança DNS, especialmente focada em mitigação de ameaças como *phishing* e *botnets*, tem evoluído significativamente nos últimos anos. Esta seção analisa trabalhos relevantes que fundamentam a abordagem proposta neste estudo.

[Khormali et al. 2021] realizaram uma revisão abrangente sobre ataques de *phishing* e *botnets* que exploram vulnerabilidades do DNS, revelando que essas ameaças persistem apesar dos numerosos esforços de pesquisa para sua detecção e mitigação. Os autores destacam a contínua evolução das táticas utilizadas pelos atacantes, especialmente em ambientes móveis e de aplicativos de mensagens, o que apresenta desafios significativos para os métodos tradicionais de detecção.

[Bilge et al. 2011] propuseram o EXPOSURE, um sistema pioneiro que emprega análise passiva de DNS para detectar automaticamente domínios maliciosos. O sistema utiliza 15 características extraídas do tráfego DNS, categorizadas em quatro grupos: baseadas em tempo, em respostas DNS, em valores TTL e no próprio nome de domínio. Durante um período de análise de dois meses e meio, processando mais de 100 bilhões de consultas DNS, o EXPOSURE alcançou uma taxa de detecção de aproximadamente 98% com apenas 1% de falsos positivos. Embora tenha demonstrado resultados promissores, o sistema focou principalmente na detecção de ameaças, sem enfatizar a otimização de recursos computacionais ou a redução de latência, aspectos prioritários na proposta atual.

Avançando nesta linha, [Marques et al. 2021] desenvolveram uma solução de firewall DNS baseada em aprendizado de máquina para detecção de domínios maliciosos em tempo real. Utilizando um *dataset* com 34 características e 90.000 registros de DNS enriquecidos com fontes OSINT, os autores avaliaram seis algoritmos de aprendizado supervisionado, com destaque para o CART combinado ao método de seleção de

características RFE, que alcançou 96% de precisão e tempo de classificação de apenas 0,013 segundos. Contudo, o estudo não apresenta análises específicas sobre o impacto na latência de resolução DNS em ambientes de produção com alto volume de requisições, nem avalia o consumo de recursos computacionais, aspectos fundamentais que a presente proposta busca aprimorar.

A eficiência no processamento de pacotes é crucial para soluções de segurança DNS. [Bertin 2017] apresenta uma arquitetura inovadora implementada na Cloudflare para mitigação de ataques DDoS, que migra de soluções baseadas em kernel *bypass* e BPF clássico para uma implementação utilizando XDP e eBPF. Esta abordagem demonstra como o processamento de pacotes pode ser otimizado ao nível do kernel, permitindo a inspeção direta na camada mais baixa possível sem necessidade de *bypass* e com custo computacional mínimo para o descarte de pacotes maliciosos. Os resultados indicaram um desempenho significativamente superior em termos de pacotes filtrados por segundo quando comparado ao Iptables tradicional, sem comprometer a latência da rede. [Sommese et al. 2022] investigaram o impacto de ataques DDoS na infraestrutura DNS, combinando dados de atividade DoS inferidos de uma *darknet* com medições DNS contemporâneas durante um período de 17 meses. Seus resultados revelaram que até 5% do *namespace* DNS sofreu ataques, com alguns casos apresentando aumentos de 100 vezes no tempo de resolução DNS ou completa inacessibilidade.

[Kostopoulos et al. 2020] desenvolveram abordagens significativas para mitigação de ataques DNS no plano de dados utilizando o framework XDP. Em seu trabalho inicial, implementaram Filtros de Bloom para mapear zonas DNS em servidores autoritativos, enquanto [Kostopoulos et al. 2021] aplicaram classificadores Naive Bayes em servidores recursivos. Ambas as soluções demonstraram eficiência excepcional, com a abordagem baseada em Naive Bayes mantendo quase 99% do processamento de requisições legítimas durante ataques. Entretanto, essas implementações apresentam limitações importantes: a primeira depende do conhecimento completo das zonas DNS, raramente disponíveis para servidores resolvedores, enquanto a segunda enfrenta restrições do verificador eBPF, incluindo o limite de 200 caracteres para nomes de domínio e a representação imprecisa de probabilidades no kernel. A presente proposta se diferencia fundamentalmente ao utilizar uma abordagem que mantém listas de domínios, porém eliminando a restrição de comprimento de nomes do eBPF, aspecto crucial para a detecção eficaz de domínios de *phishing* que frequentemente excedem esse limite.

No contexto de otimização de desempenho para aplicações de rede baseadas em eBPF/XDP, [Capeletti 2022] conduziram uma análise abrangente das capacidades e limitações dessas tecnologias em planos de dados programáveis. Através de experimentos sistemáticos utilizando SmartNIC, os autores avaliaram três modos XDP (*Generic*, *Native* e *Offload*) processando pacotes de diferentes tamanhos e com algoritmos variando a quantidade de acessos à memória. Os resultados demonstraram que os modos XDP *Generic* e *Native* conseguem manter taxa de transferência máxima (10 Gbit/s). Embora forneçam contribuições valiosas sobre os limites operacionais dessas tecnologias, o trabalho não avaliou especificamente aplicações DNS nem testou o modo XDP *Offload* devido a limitações técnicas, aspectos que complementariam a compreensão do desempenho em cenários de segurança DNS.

A análise dos trabalhos relacionados evidencia a complexidade e vulnerabilidades inerentes ao sistema DNS, sublinhando a necessidade crítica de abordagens para salvaguardar esta infraestrutura fundamental da Internet. Contudo, observa-se uma lacuna significativa na literatura atual: a ausência de avaliações robustas sobre o impacto das soluções de análise em tempo real no desempenho dos serviços DNS.

A proposta apresentada se diferencia ao utilizar as vantagens da eficiência computacional proporcionada pelo eBPF/XDP para detecção de domínios maliciosos. A arquitetura desenvolvida, estruturada em plano de controle e plano de dados, permite o gerenciamento efetivo de ameaças, enquanto preserva o desempenho operacional mesmo sob cargas elevadas de tráfego. A implementação de uma estratégia de *hashing* otimizada viabiliza a identificação quase instantânea de domínios maliciosos, minimizando o impacto na latência das resoluções DNS.

Desta forma, este trabalho apresenta uma solução equilibrada que alia segurança robusta e desempenho superior, demonstrando que a proteção contra ameaças cibernéticas pode ser implementada sem comprometer a eficiência operacional dos serviços DNS, mesmo em ambientes de alta demanda.

4. Arquitetura

A presente seção descreve a estrutura e o funcionamento da solução do DNS Firewall XDP, detalhando os aspectos fundamentais de sua implementação. Na arquitetura dos programas eBPF/XDP, as responsabilidades são divididas entre o espaço do kernel e o espaço de usuário. Conforme ilustrado na Figura 1 a arquitetura da solução é composta por dois componentes principais escritos em linguagem C com eBPF: um programa em espaço de kernel (dnsfw_xdp.kern.c) e uma aplicação em espaço de usuário (arquivo dnsfw_xdp.c) responsável pela carga, configuração e monitoramento do programa eBPF. Esta organização arquitetural permite estabelecer uma clara separação de responsabilidades, otimizando tanto o gerenciamento quanto o desempenho da solução.

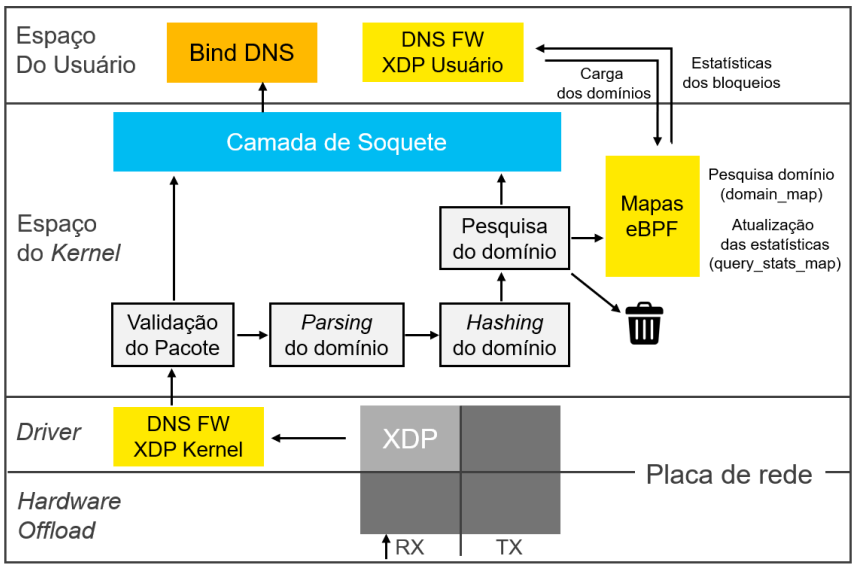

Figura 1. Arquitetura DNS Firewall XDP

O programa em espaço do usuário tem como principais responsabilidades o carregamento da lista de domínios maliciosos no mapa eBPF xdp_domains_map, a apresentação em tempo real das estatísticas de bloqueios e a geração de informações de controle por meio de arquivos de log. A partir de um arquivo local contendo a lista de domínios, os nomes são lidos, tratados e convertidos em chaves de 32 bits por meio de uma função de *hash* baseada na multiplicação por 33. Essas chaves são inseridas no mapa eBPF com as respectivas strings de domínio, utilizando a chamada bpf_map_update_elem. Além disso, o módulo de controle realiza a fixação (*pinning*) dos mapas no sistema de arquivos virtual /sys/fs/bpf, garantindo sua persistência mesmo após o término da aplicação. Também é de sua responsabilidade a carga do programa eBPF compilado, sua verificação e anexação ao *hook* XDP da interface de rede selecionada.

O programa em espaço do kernel é implementado como uma função eBPF XDP, cujo objetivo é inspecionar e filtrar pacotes DNS com base em uma lista de domínios bloqueados. São definidos dois mapas do tipo BPF_MAP_TYPE_HASH: o xdp_domains_map, utilizado para armazenar os domínios maliciosos, e o xdp_query_stats, empregado para manter estatísticas de quantas vezes cada domínio foi consultado. A cada pacote recebido, o programa realiza inicialmente a validação dos cabeçalhos Ethernet, IP e UDP, verificando se o protocolo utilizado é UDP com destino à porta 53 (DNS) e se o pacote corresponde especificamente a uma requisição do tipo *query*. Apenas os pacotes que atendem a esses critérios são submetidos à análise detalhada; os demais, incluindo pacotes TCP, mensagens que não sejam do tipo *query* ou destinadas a outras portas, são imediatamente liberados, sem processamento adicional, sendo encaminhados para as camadas superiores.

Uma vez validado o pacote como uma requisição DNS, o domínio requisitado é extraído da mensagem DNS. Durante esse processo, é determinado o comprimento da string, respeitando os limites de tamanho impostos pelo verificador do eBPF. A implementação foi construída para suportar nomes canônicos completos com até 253 caracteres, em conformidade com a [Mockapetris 1987], que estabelece os padrões do sistema de nomes de domínio (DNS). Essa característica amplia a compatibilidade do sistema com domínios válidos utilizados na internet, garantindo a correta inspeção de requisições DNS independentemente do comprimento do nome consultado. O nome do domínio é então processado pela função de *hash* de 32 bits ***djb2***, função proposta por Bernstein [Bernstein 1991], que aplica a multiplicação do valor acumulado por 33 e soma do valor ASCII de cada caractere da entrada. O resultado é utilizado como chave para consulta no mapa de domínios maliciosos.

O uso dessa função de *hash* é justificado por sua simplicidade e boa distribuição para entradas textuais, como nomes de domínios. Apesar do espaço de endereçamento teórico de 2^{32} (4.294.967.296) valores distintos, colisões tornam-se prováveis antes de atingir esse limite, em função do paradoxo do aniversário. Segundo a teoria de *hashing*, a chance de colisão atinge 50% com aproximadamente $\sqrt{2^{32}}$ = 65.536 elementos distintos [Knuth 1998]. Para os experimentos realizados, estima-se o uso de aproximadamente 6 mil domínios, valor abaixo do limiar crítico. No entanto, caso seja necessária a ampliação da base de domínios, uma alternativa seria o uso de uma função de *hash* mais robusta ou o aumento do espaço da chave para 64 bits, o que implicaria maior consumo de memória e possível impacto no desempenho.

Após o cálculo do *hash*, o programa realiza uma consulta ao mapa xdp_domains_map. Se o domínio for encontrado, o pacote é descartado (XDP_DROP) e o contador correspondente no mapa xdp_query_stats é atualizado, incrementando a quantidade de vezes que o domínio foi requisitado. Caso o domínio não esteja presente no mapa, o pacote é liberado (XDP_PASS) e encaminhado normalmente ao serviço de resolução de nomes.

Os códigos-fonte dos programas eBPF/XDP e os scripts para realização dos experimentos desenvolvidos podem ser acessados no repositório [1].

5. Avaliação de desempenho

Nesta seção, apresenta-se a avaliação de desempenho da solução proposta de DNS Firewall, implementada utilizando eBPF/XDP, em comparação com o DNS Firewall baseado em RPZ. O sistema avaliado é um resolvedor DNS, que será executado utilizando o BIND 9.16.23, configurado exclusivamente como resolvedor de nomes recursivo. O foco da avaliação é o componente de filtragem de domínios, ou seja, o módulo de DNS Firewall, conforme ilustrado na Figura 2.

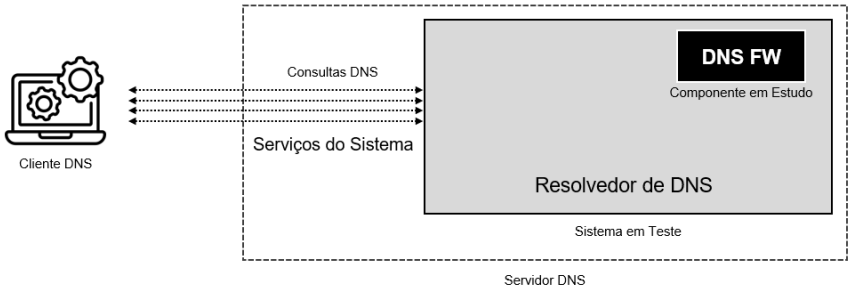

Figura 2. Avaliação de desempenho

Os experimentos foram conduzidos em um ambiente de teste controlado, composto por dois elementos principais: um servidor e um cliente gerador de carga. O servidor é equipado com um processador Intel Core i5-7500 de 3,4 GHz, 16 GB de RAM e executa o sistema operacional Red Hat Enterprise Linux 9.5 com kernel versão 5.14.0. O cliente, responsável pela geração das consultas DNS, consiste em uma estação com processador Intel Core i5 de 3,0 GHz, 16 GB de RAM, rodando Red Hat Enterprise Linux 9.4, também com kernel 5.14.0. Ambos os dispositivos estão interligados por meio de um switch Cisco Gigabit Ethernet e possuem conectividade com a Internet. Devido às limitações de *hardware* disponíveis, o carregamento do programa eBPF no kernel foi realizado utilizando o modo genérico.

Os experimentos utilizaram um subconjunto do dataset de classificação de domínios desenvolvido pelo *Canadian Institute for Cybersecurity* em parceria com a *Bell Canada* [Razavi et al. 2021], composto por domínios legítimos e maliciosos. Para garantir consistência nos testes, tanto a solução RPZ quanto a baseada em XDP foram configuradas, em todos os experimentos, com uma lista contendo 6.654 domínios maliciosos.

[1] https://github.com/psantos-it/dnsfw

A avaliação de desempenho do sistema foi conduzida considerando três métricas principais:

- **Vazão**: Utilizada para quantificar a capacidade máxima de processamento de consultas DNS. Foram conduzidos testes de estresse na operação de resolução de nomes, com o objetivo de identificar os limites superiores de desempenho e avaliar o impacto da adoção das soluções de DNS Firewall na vazão do sistema.
- **Consumo de CPU**: Esta métrica é particularmente relevante dado que um dos principais benefícios propostos pela programabilidade do plano de dados é a otimização no uso de recursos de processamento. A análise foi realizada a partir de cargas compostas por domínios legítimos e diferentes proporções de domínios maliciosos, permitindo observar a variação no uso da CPU conforme o cenário de bloqueio.
- **Tempo de resposta**: Considerando que o processamento adicional de pacotes no plano de dados pode introduzir latência em comparação ao encaminhamento convencional, foram conduzidas medições sistemáticas para quantificar o impacto da solução proposta no tempo de resolução de domínios

5.1. Vazão

Para fins de comparação da vazão máxima, foi utilizado o resperf [Nominum 2012] versão 2.12.0, ferramenta desenvolvida pela Nominum/Akamai e *Domain Name System Operations Analysis and Research Center* (DNS-OARC). Ela envia consultas DNS a uma taxa controlada e constantemente crescente. Usamos como base o sistema sem a utilização de DNS Firewall, isso permite determinar um valor de referência para comparação entre a solução RPZ e a solução proposta utilizando eBPF/XDP.

Os experimentos foram executados com duração de 60 segundos, simulando consultas de forma crescente até 100.000 consultas por segundo utilizando uma base fornecida pela Nominum com 1 milhão de domínios. Durante os ensaios, o resperf monitora as respostas do servidor, bem como as taxas de respostas e taxas de falhas. Além disso, o resperf continua esperando respostas por 40 segundos depois de parar de enviar tráfego. Foi considerada como taxa de divergência quando a diferença entre consultas enviadas e respostas recebidas supera 2% do total de queries enviadas, sinalizando que o sistema entrou em esgotamento e consultas começam a ser descartadas. O mapa de domínios maliciosos do sistema XDP e o arquivo de zona com domínios a serem bloqueados do RPZ foram carregados para os experimentos com um subconjunto de 6.654 entradas do nosso dataset.

Na Figura 3, é possível observar que a vazão do sistema sem proteção de DNS Firewall atinge um máximo de 51698 *queries* por segundo e, utilizando a proteção de DNS XDP, atinge uma taxa máxima de 47904 queries por segundo. Uma redução de 3794 queries por segundo na vazão máxima do sistema.

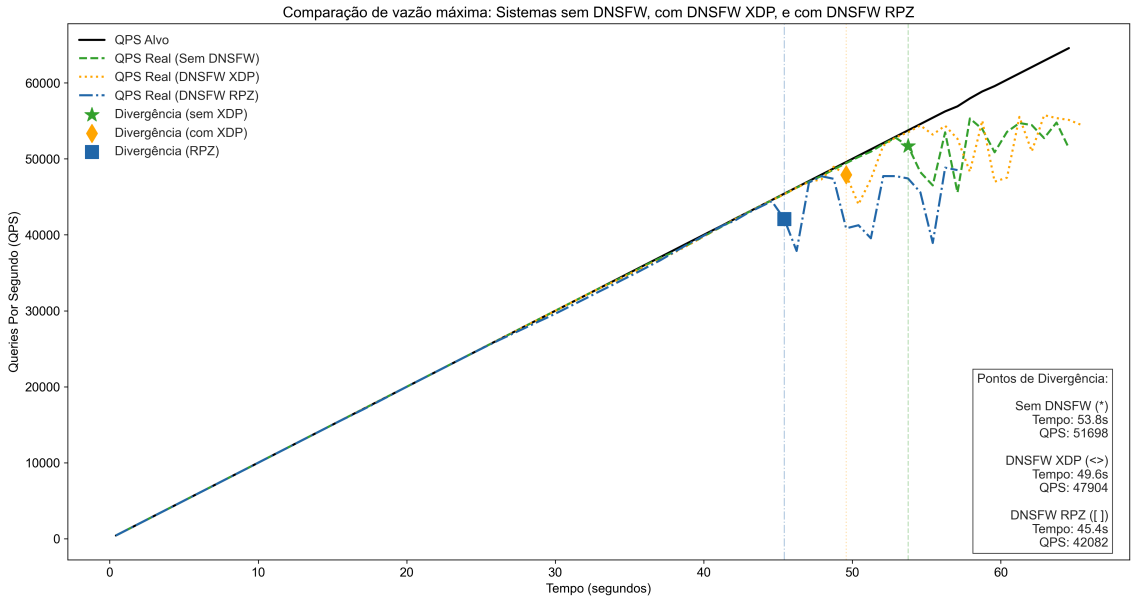

Figura 3. Comparativo da vazão máxima entre os 3 sistemas

Também é possível observar que, comparando a vazão do sistema utilizando DNS Firewall RPZ com a solução utilizando o processamento de pacotes no plano de dados, o XDP consegue entregar uma taxa máxima superior, sendo 5822 queries por segundo a mais.

5.2. Consumo de CPU

Para avaliação do consumo de recursos de CPU pelos sistemas, inicialmente foi considerado um cenário composto exclusivamente por domínios legítimos, de forma a evitar bloqueios durante os testes. Nesse contexto, foram reproduzidos os mesmos experimentos de vazão máxima, e as métricas de consumo de CPU foram coletadas com o auxílio da ferramenta *System Activity Report* (SAR), um utilitário do sistema operacional Linux utilizado para o monitoramento de recursos. Na Figura 4, observa-se que o pico de consumo de CPU do DNS Firewall XDP foi de 31,23%, ligeiramente inferior ao pico da solução com RPZ, que atingiu 31,91%, e levemente superior ao ambiente sem firewall, cujo pico foi de 30,92%. A média de consumo de CPU não foi considerada neste caso, pois os sistemas atingem o ponto de exaustão em momentos distintos, o que resulta em durações variáveis nos experimentos, como evidenciado na Figura 3. Como a coleta realizada pelo SAR foi feita em intervalos fixos de 60 segundos, a média poderia induzir a interpretações imprecisas. Também foram realizados experimentos com um subconjunto de domínios maliciosos para avaliar o impacto no consumo de CPU em cenários de bloqueios sendo executados. Foram utilizados 9 subconjuntos: o primeiro com 10% dos domínios maliciosos, aumentando 10% a cada nível até 90%. O experimento gerou consultas DNS durante 60 segundos, usando o conjunto de forma proporcional. Também foram realizados testes com dois fatores distintos: com cache DNS ativado e desativado.

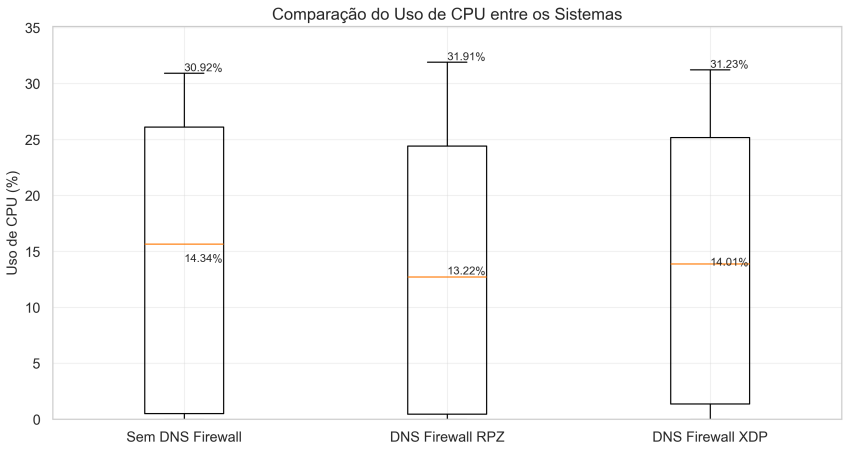

Figura 4. Comparativo do consumo de CPU

Na Figura 5, podemos observar que a solução de DNS Firewall utilizando XDP resultou em consumo de CPU significativamente menor que suas contrapartes com RPZ. Outra constatação é que a vantagem do XDP sobre o RPZ aumenta conforme cresce o percentual de domínios maliciosos. A redução média de CPU quando se compara o RPZ com Cache e o XDP com Cache é de aproximadamente 70%, chegando no cenário de 90% de domínios maliciosos a 93% de redução (de 16,3% para 1,2% de uso de CPU). Já no cenário sem Cache, a redução média de CPU é de aproximadamente 42%, chegando no cenário de 90% de domínios maliciosos a 91% de redução (de 10,6% para 1,0% de uso de CPU).

É importante destacar que a redução do consumo de CPU será ainda maior caso seja executado o *offload* do processamento para o *hardware* dedicado (*SmartNICs*).

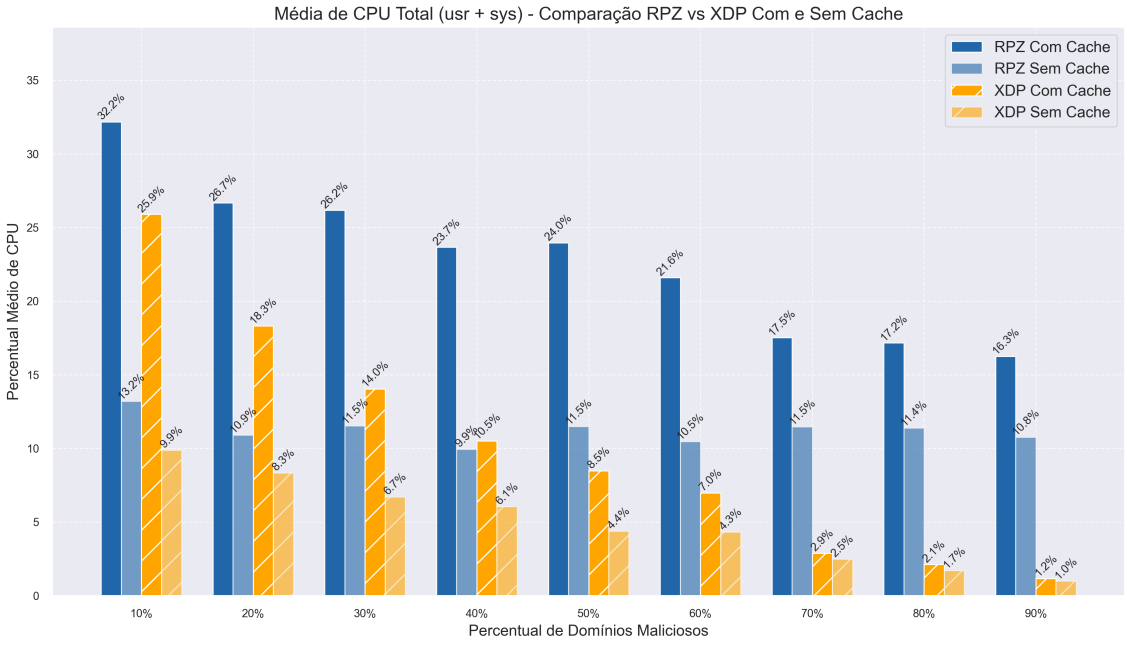

Figura 5. Comparativo do Consumo de CPU com níveis de bloqueios

5.3. Tempo de Resposta

Considerando que o servidor avaliado atua como DNS recursivo, ou seja, realiza consultas adicionais a servidores de nível superior quando não possui a resposta em cache, optou-se por habilitar o cache local e conduzir os experimentos com um único domínio e previamente presente no cache. Essa escolha teve como objetivo eliminar interferências externas, como latência de conexão com a internet, variabilidade na carga dos servidores DNS externos e efeitos de cache nesses servidores, assegurando a reprodutibilidade dos resultados.

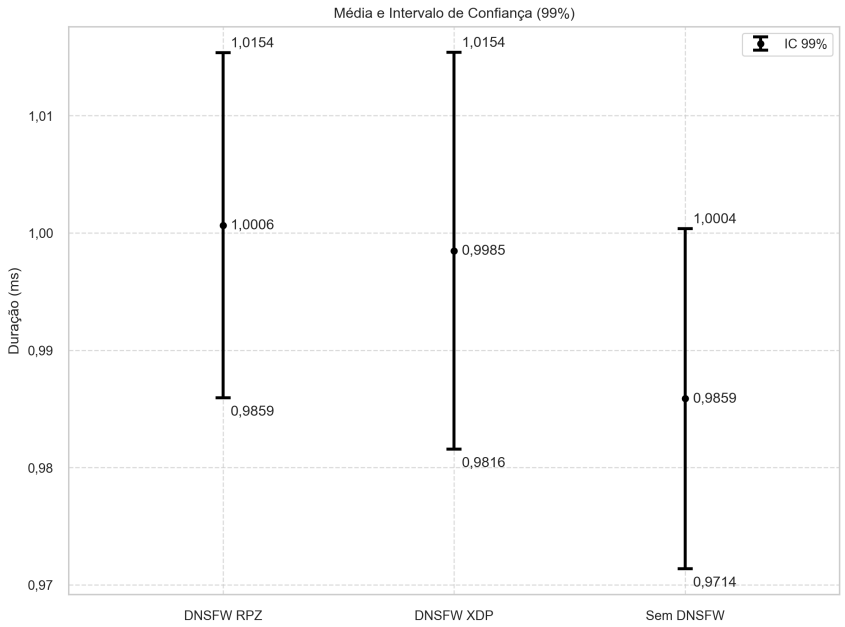

Figura 6. Comparativo do tempo de resposta

Para a geração das consultas DNS, utilizou-se o Dnspyre [Tantalor93 2024] que permite definir um número de solicitações simultâneas a serem enviadas, o intervalo das consultas, bem como o tipo de consulta e nome de domínio. Os experimentos foram executados com registros A, amplamente utilizados para resolução de nomes de domínio para endereços IPv4.

O domínio escolhido para o experimento foi *google.com*, por figurar como o mais popular na Tranco List [Le Pochat et al. 2019]. Ao todo, foram executadas 1000 consultas, com uma taxa de 5 solicitações por segundo. O tempo médio de resposta no ambiente sem o DNS Firewall foi de 0,9858 ms, enquanto no ambiente com a solução baseada em XDP foi de 0,9984 ms. Como ilustrado pela Figura 6, a introdução do DNS Firewall XDP resultou em um acréscimo médio de apenas 0,0126 ms no tempo de resposta. A análise estatística, baseada em intervalos de confiança com nível de 99%, mostra que há sobreposição entre os intervalos das duas configurações, indicando que não há diferença estatisticamente significativa entre os tempos de resposta.

Comparando com a solução de DNS Firewall RPZ é possível verificar que novamente os intervalos se sobrepõem e as médias estão contidas entre si, porém o tempo médio de resposta durante o experimento foi 0,0022 ms inferior. Apesar dessa diferença,

os dados não sustentam uma conclusão estatisticamente significativa sobre a superioridade de uma abordagem em relação à outra no que diz respeito ao tempo de resposta. Ainda assim, os resultados são promissores, pois indicam que a implementação baseada em XDP oferece desempenho comparável às soluções consolidadas, com impacto mínimo na latência.

6. Conclusão

Este artigo apresentou uma abordagem para a implementação de DNS Firewall, utilizando eBPF, demonstrando sua eficácia como mecanismo de mitigação de ameaças cibernéticas. A solução proposta foi desenvolvida com base no framework XDP, permitindo a interceptação e análise de pacotes DNS diretamente no plano de dados do kernel Linux, o que viabiliza respostas em tempo real com mínimo impacto sobre o desempenho do sistema.

A avaliação experimental realizada concentrou-se em métricas críticas de desempenho, como tempo de resposta de consultas, vazão e consumo de recursos, evidenciando a eficácia da proposta. Os resultados indicaram que a proposta supera abordagens tradicionais, como o uso de RPZ, alcançando uma redução de até 93% no consumo de recursos computacionais em situações de alto volume de bloqueio de domínios. Em cenários sem bloqueios, a proposta oferece uma maior vazão máxima do sistema; em contrapartida, gera um incremento de latência praticamente insignificante no tempo de resposta.

Os resultados indicam que a utilização de eBPF para o bloqueio de domínios maliciosos representa uma alternativa promissora, especialmente para serviços DNS com grande escala de tráfego, como os resolvers públicos. Além de contribuir para o aumento da segurança, a solução proposta também promove maior eficiência na operação desses serviços. Em trabalhos futuros, é possível realizar o *offload* da solução para uma *SmartNIC* o que certamente irá gerar melhores resultados, evidenciando ainda mais os benefícios da adoção da solução.

Referências

Akamai (2023). Attack Superhighway A Deep Dive on Malicious DNS Traffic. `https://www.akamai.com/resources/state-of-the-internet/attack-superhighway-a-deep-dive-on-malicious-dns-traffic`.

Arends, R., Austein, R., Larson, M., Massey, D., and Rose, S. (2005). Rfc 4033: Dns security introduction and requirements.

Bernstein, D. J. (1991). The djb2 hash function. `http://www.cse.yorku.ca/~oz/hash.html`.

Bertin, G. (2017). Xdp in practice: integrating xdp into our ddos mitigation pipeline. In *Netdev 2.1*, volume 2.

Bilge, L., Kirda, E., Krügel, C., and Balduzzi, M. (2011). Exposure: Finding malicious domains using passive dns analysis. In *Network and Distributed System Security Symposium*.

Capeletti, I. F. (2022). Análise de desempenho de aplicações ebpf/xdp em planos de dados programáveis. Monografia de graduação, Universidade Federal do Pampa, Alegrete, RS, Brasil. Trabalho de Conclusão de Curso – Curso de Ciência da Computação.

Community (2024). eBPF Docs. `https://ebpf-docs.dylanreimerink.nl/`.

Høiland-Jørgensen, T., Brouer, J. D., Borkmann, D., Fastabend, J., Herbert, T., Ahern, D., and Miller, D. (2018). The express data path: fast programmable packet processing in the operating system kernel. In *Proceedings of the 14th International Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '18, page 54–66, New York, NY, USA. Association for Computing Machinery.

Khormali, A., Park, J., Alasmary, H., Anwar, A., Saad, M., and Mohaisen, D. (2021). Domain name system security and privacy: A contemporary survey. *Computer Networks*, 185:107699.

Knuth, D. E. (1998). *The Art of Computer Programming: Sorting and Searching*, volume 3. Addison-Wesley, Boston, 2 edition.

Kostopoulos, N., Kalogeras, D., and Maglaris, V. (2020). Leveraging on the xdp framework for the efficient mitigation of water torture attacks within authoritative dns servers. In *2020 6th IEEE Conference on Network Softwarization (NetSoft)*, pages 287–291.

Kostopoulos, N., Korentis, S., Kalogeras, D., and Maglaris, V. (2021). Mitigation of dns water torture attacks within the data plane via xdp-based naive bayes classifiers. In *2021 IEEE 10th International Conference on Cloud Networking (CloudNet)*, pages 133–139.

Kurose, J. F. and Ross, K. W. (2022). *Computer networking: a top-down approach.* Pearson Education Limited.

Le Pochat, V., Van Goethem, T., Tajalizadehkhoob, S., Korczyński, M., and Joosen, W. (2019). Tranco: A research-oriented top sites ranking hardened against manipulation. In *Proceedings of the 26th Annual Network and Distributed System Security Symposium*, NDSS 2019.

Liu, C. and Albitz, P. (2006). *DNS and BIND (5th Edition)*. O'Reilly Media, Inc.

Marques, C., Malta, S., and Magalhães, J. (2021). Dns firewall based on machine learning. *Future Internet*, 13(12):309.

McCanne, S. and Jacobson, V. (1993). The bsd packet filter: a new architecture for user-level packet capture. In *Proceedings of the USENIX Winter 1993 Conference Proceedings on USENIX Winter 1993 Conference Proceedings*, USENIX'93, page 2, USA. USENIX Association.

Mockapetris, P. V. (1987). Rfc1035: Domain names - implementation and specification.

Nominum (2012). resperf Performance Tool Manual. `https://www.dns-oarc.net/files/dnsperf/2.0.0.0/resperf.pdf`.

Razavi, A., Mahdavifar, S., Maleki, N., Habibi Lashkari, A., and Broda, M. (2021). Classifying malicious domains using dns traffic analysis. In *Book*.

Sommese, R., Claffy, K., van Rijswijk-Deij, R., Chattopadhyay, A., Dainotti, A., Sperotto, A., and Jonker, M. (2022). Investigating the impact of ddos attacks on dns infrastructure. In *Proceedings of the 22nd ACM Internet Measurement Conference*, IMC '22, page 51–64, New York, NY, USA. Association for Computing Machinery.

Tantalor93 (2024). dnspyre. https://github.com/Tantalor93/dnspyre.

Umbrella, C. (2022). 2022 DNS Discoveries. https://learn-cloudsecurity.cisco.com/umbrella-library/2022-dns-discoveries.

Vixie, P. A. and Schryver, V. (2017). DNS Response Policy Zones (RPZ). Internet-Draft draft-ietf-dnsop-dns-rpz-00, Internet Engineering Task Force. Work in Progress.