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

Ataque de Negação de Serviço por Reflexão Amplificada: Explorando o Protocolo Domain Name System Para a Detecção e Identificação de Refletores

Yan Victor dos Santos

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

Orientador
Prof. Dr. João José Costa Gondim

Brasília
2019

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

Ataque de Negação de Serviço por Reflexão Amplificada: Explorando o Protocolo Domain Name System Para a Detecção e Identificação de Refletores

Yan Victor dos Santos

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

Prof. Dr. João José Costa Gondim (Orientador)
CIC/UnB

Prof. Dr. Eduardo Adilo Pelinson Alchieri
CIC/UnB

Prof. Dr. Marcos Fagundes Caetano
CIC/UnB

Prof. Dr. Edison Ishikawa
Coordenador do Bacharelado em Ciência da Computação

Brasília, 11 de Julho de 2019

Dedicatória

Dedico este trabalho a minha mãe Francisca das Chagas Chaves Santos, ao meu pai José Maria Santos Silva, a toda a minha família e amigos, a professora Alessandra Lisboa, ao professor Marcos Paulo e a todos que me apoiaram durante o meu período de graduação.

Agradecimentos

Agradeço ao departamento de Ciência da Computação (UnB) pelo fornecimento dos recursos que permitiram a realização deste trabalho. Agradeço também ao Dr. João José Costa Gondim por orientar, incentivar e por também fornecer recursos, além de ter visto potencial em minha capacidade de pesquisa na área de segurança da informação. Agradeço ainda à minha família e amigos, pois sempre me apoiaram fornecendo uma base sólida necessária para a minha dedicação aos meus objetivos. Por fim, agradeço aos diversos professores de graduação que me apoiaram durante toda esta jornada; em especial à professora Genaína Nunes Rodrigues, pioneira no despertamento da minha paixão pela pesquisa.

Resumo

O avanço da tecnologia permitiu o aumento de serviços automatizados na Internet, gerando a necessidade de confiabilidade em sistemas computacionais. O grande número de servidores tornou possível o surgimento de novas vulnerabilidades, e consequentemente, novos formatos de ataque surgiram, comprometendo a segurança de servidores e de consumidores. Um dos ataques mais recorrentes é conhecido como Ataque de Negação de Serviço, ou Distributed Denial of Service (DDoS), que possui como principal objetivo dificultar, ou até mesmo impedir a comunicação entre clientes e servidores. Existe também uma outra abordagem que abusa do protocolo User Datagram Protocol (UDP) e do processamento de servidores legítimos, que são utilizados como intermediários para a realização do ataque, atuando como refletores. Dentro da comunidade científica, há uma série de problemas a serem considerados para a construção de novas mitigações do ataque. Porém, é indiscutível, para tanto, a necessidade de estudá-los, devido aos prejuízos que a indisponibilidade de serviço gera em grandes corporações. Da mesma forma que os ataques acompanham a evolução das soluções, a comunidade de segurança acompanha a evolução desses ataques. Este trabalho propõe uma ferramenta que realize a detecção e a identificação de refletores, principais agentes em ataques de negação de serviço por reflexão amplificada. Para a realização da detecção, os métodos utilizados até o momento consideram apenas a análise de pacotes referentes ao protocolo Domain Name System (DNS). Uma série de testes foi realizada em conjunto à pesquisa de [1] realizada na Universidade de Brasília, que permitiram analisar a eficiência da detecção do refletor em diversos níveis de dificuldade, com métrica de quantidade de pacotes por segundo. Para tanto, a ferramenta, que inserida em determinada estrutura, permitiu detectar e identificar o refletor, bloqueando-o de maneira estratégica; fornecendo caminhos alternativos para reestabelecer uma comunicação entre o servidor e clientes.

Palavras-chave: DDoS, Ataque de negação de serviço, Refletores, Segurança da informação

Abstract

The advancement of technology allowed the increase of automated services on the Internet, generating the need for reliability in computational systems. The large number of servers made it possible for new vulnerabilities to appear, and consequently, new attack formats emerged, compromising the security of servers and consumers. One of the most recurring attacks is known as Denial of Service Attack (DDoS), whose main purpose is to hamper or even prevent communication between clients and servers. There is also another approach that abuses the protocol User Datagram Protocol (UDP) and the processing of legitimate servers, which are used as intermediates for the realization of the attack, acting as reflectors. Within the scientific community, there are a number of problems to be considered for building new mitigations of the attack. However, it is unquestionable, therefore, the need to study them, due to the damages that the unavailability of service generates in large corporations. In the same way that the attacks accompany the evolution of the solutions, the security community follows the evolution of these attacks. This work proposes a tool that performs the detection and identification of reflectors, the main agents in denial of service attacks by amplified reflection. In order to perform the detection, the methods used so far consider only the analysis of packages referring to the Domain Name System (DNS) protocol. A series of tests was carried out in conjunction with the research [1], carried out at the University of Brasilia, which allowed the analysis of the reflector detection efficiency in several levels of difficulty, with a number of packages per second metric. To do so, the tool, which inserted in a certain structure, allowed to detect and identify the reflector, blocking it in a strategic way; providing alternative ways to reestablish communication between the server and clients.

Keywords: DDoS, Denial of Service, Reflectors, Information Security

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

ARP Address Resolution Protocol.

DDoS Distributed Denial of Service.

DNS Domain Name System.

DoS Denial of Service.

LAN Local Area Network.

MAC Media Access Control.

NTP Network Time Protocol.

SSDP Simple Service Discovery Protocol.

TCP Transmission Control Protocol.

UDP User Datagram Protocol.

WAN Wide Area Network.

Capítulo 1

Introdução

Com o aumento do número de dispositivos capazes de se conectar à Internet, o acesso a serviços que antes eram realizados presencialmente passou a ser realizado por meio plataformas on-line, que incentivou empresas, universidades e grandes corporações a disponibilizarem seus serviços na web. A rapidez de negociações via Internet é determinante para o sucesso de atendimentos em massa, o que influencia na maneira da sociedade produzir e consumir. Esta evolução levou ao surgimento de sistemas de informação capazes de atender demandas de alto risco, que precisavam garantir a confiabilidade, integridade e, mais importante, a disponibilidade do serviço prestado. Em contrapartida, a indisponibilidade dos serviços passou a interessar usuários mal-intencionados.

Concomitante ao avanço da tecnologia computacional, surgiram diversas modalidades de ciberataques, dentre eles: o ataque de negação de serviço. Diante de um tráfego malicioso cujo objetivo é saturar a rede interna, a proteção de um servidor torna-se uma difícil tarefa. Afinal, como impedir a negação de serviço causada por requisições que são, em um primeiro momento, consideradas legítimas? Como separar um tráfego de usuários comuns de um ataque em execução? A resposta para a segunda pergunta é relativamente simples. Verifica-se a quantidade de pacotes que chegam em um determinado espaço de tempo, considerando os endereços IP's de origem. Mesmo após a detecção do cenário, a resposta para a primeira questão exige uma avaliação mais aprofundada, diante da complexidade da estrutura da Internet.

1.1 Motivação

Milhares de acessos a sites na rede reforçam a existência de uma grande diversidade de usuários. Por este motivo, existem servidores dispostos a prover serviços que podem movimentar milhões de dólares anualmente, gerando a necessidade de ampliar a segurança contra possíveis ataques, pois uma única ameaça poderia comprometer completamente o

funcionamento de uma empresa e até mesmo levá-la à falência. É necessário considerar que a indisponibilidade de um serviço poderia gerar situações catastróficas, como por exemplo, Bancos internacionais que movimentam bilhares de transações financeiras, terem seus serviços indisponibilizados gerando transtornos incalculáveis aos seus clientes. Toda esta complexidade requer uma estrutura de segurança capaz de conter quaisquer tipos de ataques, prevendo e defendendo as ações de invasores para manter o serviço funcionando perfeitamente.

Para que uma plataforma se mantenha segura ao nível a que é exposta, é necessário investimento sério em segurança computacional que seja de fato robusta. Porém, nem toda corporação é capaz de manter uma estrutura neste patamar, ou até mesmo acreditam não ser necessário implantá-la, afinal, nem todos acreditam que poderão ser alvejados até que um ataque de fato ocorra. Portanto, é válido reforçar que todos estão sujeitos a este cenário, e que por vários motivos muitas vítimas não possuem um nível de segurança compatível com o que a comunidade de usuários mal intencionados tem para oferecer de pior, tornando reais os desastres cibernéticos.

Um estudo aprofundado pode contribuir para suportar eficientemente um ataque, se detectarmos exatamente o funcionamento da origem em tempo real. Como citado anteriormente, a primeira pergunta sobre a mitigação do ataque pode ser previamente respondida com uma sequência de etapas que realizam a defesa da vítima, começando por um dos passos mais importantes: categorização do ataque. A detecção de uma variante do ataque sugere levar a uma solução precisamente direcionada, evitando perdas consideráveis de serviços úteis e importantes para as corporações e seus consumidores. Afinal, para se defender de forma eficiente, é interessante ter ciência de quem está atacando e quais são as suas ferramentas. Por meio da necessidade, diversos trabalhos sobre detecção dos tipos de ataque surgiram.

A realização deste trabalho foi motivada pela metodologia de detecçao de Arp Spoofing, descrita no trabalho situado por [6], que tornou possível detectar outros tipos de ataques utilizando a mesma idéia de implementação, incluindo a reflexão amplificada. Além do mais, a necessidade de mitigação de DDoS torna essencial o passo de detecção da ameaça e suas variações. Revelar a existência de refletores e expor seus endereços físicos em uma lista negra reduz fortemente o esforço de bloqueio do atacante, pois o agente refletor normalmente não possui más intenções, podendo ser a chave para conter o fluxo, se ele tiver ciência do ataque. O sucesso desta abordagem fortalecerá o entendimento e a importância de dar continuidade a pesquisas das variações do DDoS.

1.2 Justificativa

Segundo os dados divulgados pela DDoS Mon [2], é possível visualizar o quão problématica é a segurança de diversas empresas, inclusive nos dias atuais, pois como podemos ver, países como China e Estados Unidos lideram o ranking de países que mais sofrem ataques de negação de serviço por reflexão amplificada, com um pouco mais de 250 mil ocorrências de ataque desde abril de 2019. O Brasil fica em décimo lugar, com um pouco mais de 10 mil ocorrências apenas nos últimos 3 meses.

Ainda segundo a DDoS Mon, ataques por negação de serviço têm liderado dentre os diversos tipos de ataques do gênero, sendo responsável por 70% das ocorrências segundo a observação.

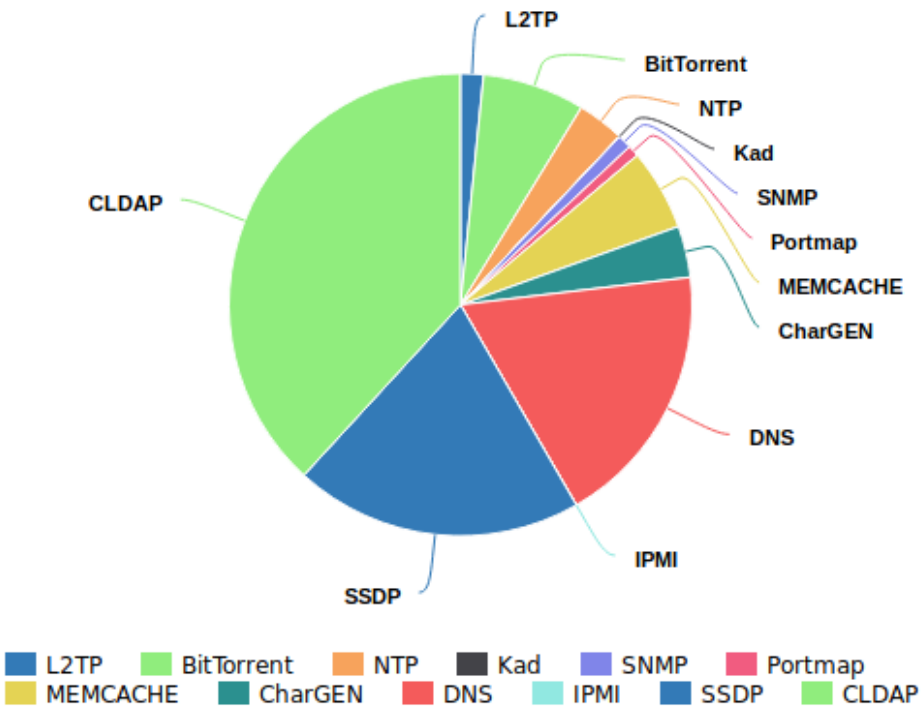

Figura 1.1: Frequência de Protocolos Usados para Reflexão [2]

Conforme observado na Figura 1.1, estão entre os protocolos mais utilizados para a execução do ataque no primeiro semestre de 2019: Domain Name System (DNS) e Simple Service Discovery Protocol (SSDP). O protocolo Network Time Protocol (NTP) também possui uma alta taxa de uso. As estatísticas revelam um cenário preocupante, sendo imprescindível reação constante e evolutiva contra ameaças. A necessidade da evolução de sistemas de segurança contra este tipo de ataque é uma resposta para a tendência de

crescimento nos próximos anos, devido a complexidade da estrutura da rede mundial de computadores (local onde todos estão sujeitos aos diversos tipos de inseguranças).

1.3 Objetivo

Para este trabalho, vamos considerar métodos já consolidados para detecção de ataques DDoS volumétricos. Uma vez que a detecção já tenha ocorrido, verificamos uma nova configuração de elementos que abrem portas para a detecção e a identificação de refletores, agentes responsáveis por amplificar o tráfego. Enquanto processam pacotes vindos de uma ou mais máquinas com endereço camuflado (técnica de IP *spoofing*), os refletores inundam a vítima de pacotes legítimos, causando sua indisponibilidade, o que caracteriza este formato de ataque como negação de serviço.

O referido trabalho tem como principal objetivo o desenvolvimento de uma ferramenta de detecção e identificação de refletores em ataques de negação de serviço por reflexão amplificada, considerando apenas o escopo de pacotes referentes ao protocolo DNS. Ao final da discussão, será proposto um cenário exemplificativo acerca da atuação do detector em uma determinada rede, com o intuito de contribuir com a mitigação do ataque. Os resultados serão apresentados após a discussão sobre o teste realizado, em conjunto com o trabalho que desenvolveu a ferramenta de ataque, descrito em [1].

1.4 Organização do Texto

Este trabalho está organizado da seguinte forma:

- capítulo 2: abordagem sobre os principais conceitos necessários para o entendimento do trabalho;
- capítulo 3: abordagem sobre a metodologia de trabalho desenvolvida, juntamente com a ferramenta de detecção de refletores;
- capítulo 4: apresentação sobre os testes realizados em laboratório e seus resultados. Discussão e análise dos dados;
- capítulo 5: apresentação sobre a conclusão e trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Neste capítulo será apresentada uma fundamentação teórica a respeito do funcionamento do ataque de negação de serviço, ataque de negação de serviço por reflexão amplificada, protocolo DNS e servidores DNS.

2.1 Ataque de Negação de Serviço - DoS

Ataques de negação de serviço geralmente acontecem quando um ou mais atacantes agem com a intenção de dificultar ou até mesmo impedir a disponibilidade de serviço de um servidor. Normalmente, a negação é resultado de um grande fluxo de pacotes enviados pelo atacante com o objetivo de inundar a rede do servidor, desta maneira qualquer cliente que tentar se comunicar com ele não obterá sucesso.

Embora a inundação seja uma das formas mais comuns de realizar o ataque Denial of Service (DoS), existem outras maneiras de realizá-lo, como por exemplo, causar *buffer overflow* ao consumir todos os recursos do servidor, tais como memória, CPU, disco, etc [7]. Desta maneira o servidor poderia ter dificuldades de processar todos os pacotes, podendo ter problemas de gerenciamento do sistema operacional, ficando indisponível para a rede.

Ataques DoS podem ser confundidos com problemas de conectividade, dentre outros típicos problemas que tornam a conexão lenta. Da perspectiva do cliente, é possível suspeitar do ataque quando há perda completa de conexão ou lentidão no carregamento de recursos do site que não costumam ser recorrentes.

Com o avanço dos sistemas de defesa, o DoS precisou evoluir concomitantemente, gerando variações do ataque. Com o tempo, o poder de processamento das máquinas servidores aumentou, e pesquisas foram realizadas sobre a mitigação de DoS, o que dificultou o sucesso de agentes mal intencionados. Portanto, para gerar pacotes suficientes para causar DoS, os atacantes passaram a aumentar também a intensidade de pacotes

no tráfego por meio de amplificação, característica do ataque distribuído de negação de serviço (DDoS).

2.2 Ataque Distribuído de Negação de Serviço - DDoS

Ataques DDoS são famosos pelos seus grandes feitos, demandando uma resposta ágil por parte da vítima para conseguir mitigá-lo. Este formato de ataque tem o mesmo objetivo do DoS, com uma estrutura semelhante. A diferença se encontra na maneira como o atacante divide a tarefa de uma máquina para diversas máquinas na rede, por meio de um *malware*, criando um exército de *bots* dispostos a amplificar o tráfego a ser enviado à rede da vítima [8]. Quando enfim o endereço de cada bot é direcionado ao servidor, o tráfego enviado ultrapassa o limite suportado pela vítima, causando a negação de serviço. Este tipo de ataque mais elaborado também possui outras maneiras de ser feito.

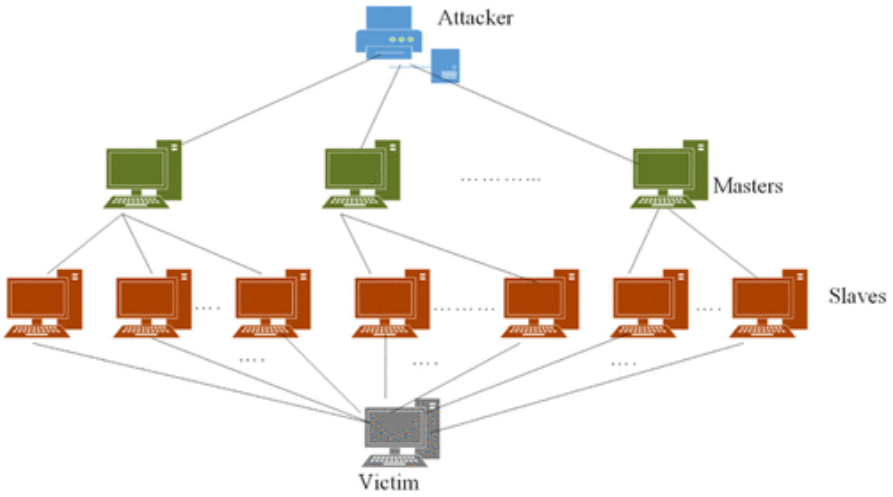

Figura 2.1: Estrutura básica do ataque distribuído de negação de serviço [3]

Segundo a Cloudflare [9], o maior ataque DDoS ocorreu em 2018. Este ataque teve como alvo o GitHub [10], chegando a ter um tráfego de entrada de 1.3 terabytes por segundo (Tbps), com uma taxa de envio de 126.9 milhões por segundo. A característica mais atenuante neste ataque foi a utilização de servidores Memcached como refletores, camuflando o endereço de origem e ampliando o ataque por volta de 50 mil vezes. O GitHub conseguiu mitigar com sucesso, detectando o ataque aproximadamente 10 minutos depois de seu início. Embora este ataque tenha ficado famoso pelo seu feito, existiu uma outra vítima 5 dias após com feito de 1.7 Tbps, mas a identidade desta não foi divulgada.

O sucesso dos ataques é o principal motivo da necessidade de uma boa estratégia de mitigação. Além da estrutura exemplificada a respeito do DDoS, também há outra forma de fazê-lo: amplificação por meio de refletores.

2.3 Protocolo Domain Network System - DNS

Para compreendermos os ataques de negação de serviço por reflexão amplificada, é necessário entender como servidores que utilizam protocolos favoráveis a eles funcionam.

Servidores que utilizam protocolo User Datagram Protocol (UDP) [11] tem como objetivo oferecer serviços rápidos e atender inúmeras requisições, dispensando conectividade. O principal intuito deste protocolo é prover datagramas de demandas com alto volume de dados, sendo estes dados pouco sensíveis. Normalmente sua aplicação é útil para vídeos, áudio, *streamings*, etc. Existem diversos protocolos que fazem uso de datagramas UDP, e dentre eles está o protocolo DNS.

O Protocolo DNS é responsável por facilitar a busca por *websites* de uma forma legível para os usuários. Se não fosse por este protocolo, todos os usuários precisariam decorar o endereço numérico de cada servidor, dificultanto os acessos por limitações de memorização. Por este motivo, o protocolo DNS facilita a maneira de encontrarmos um servidor na web, apelidando-o com um nome mais amigável e criando o mapeamendo de apelido por um endereço real. Portanto, o usuário não precisa mais decorar o endereço IP, mas apenas o apelido atribuído ao *website*. A implementação deste protocolo é realizada por servidores DNS, que recebem milhares de requisições não orientadas à conexão e, por meio de uma hierarquia de domínios, traduzem nomes atribuídos a um servidor retornando o seu endereço IP real, permitindo o acesso ao serviço.

Existem diversos tipos de *queries* a um servidor DNS, variando de acordo com o parâmetro *type* passado no pacote. Como por exemplo, *type* A retorna um endereço de 32 bits, enquanto AAAA retorna um enderço de 128 bits, e assim por diante. Vale destacar que é possível obter a resposta de todos os *types*, bastando passar o argumento ANY, onde o servidor DNS retornará uma resposta muito maior que a requisição realizada [12]. A fórmula a seguir define o fator de amplificação [13]:

Fator de Amplificação = tamanho(resposta) / tamanho(requisição)

Como podemos observar na figura 2.2, a resposta para a requisição do tipo "ddos.dns.com: type ANY, class IN"gerou uma resposta muito maior, retornando todos os *types* definidos pelo servidor.

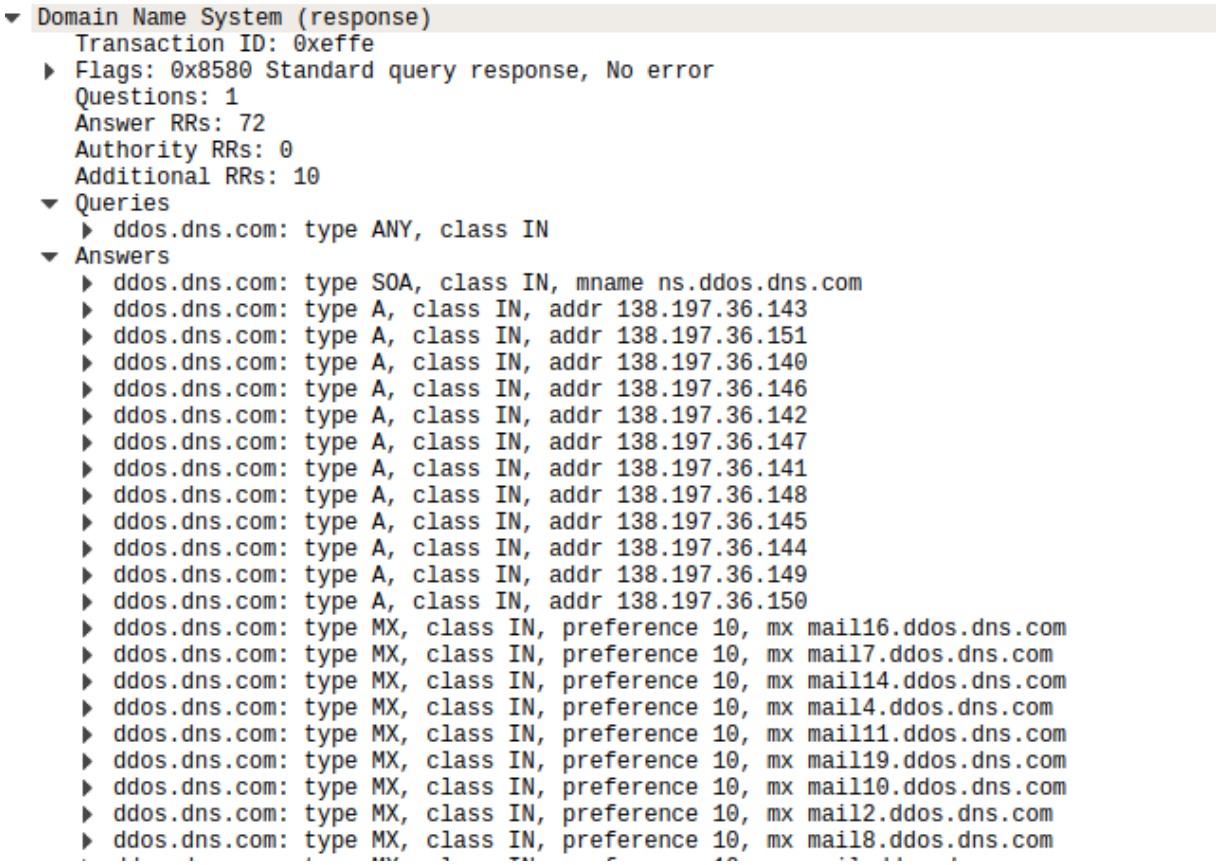

Figura 2.2: Exemplo de pacote de resposta para *query* do tipo ANY - Captura Wireshark

2.4 Ataque de Negação de Serviço por Reflexão Amplificada

Para que seja possível validar a metodologia deste trabalho, é necessário compreender como o ataque por reflexão amplificada é estruturado. Primeiro, é necessário que servidores que utilizam protolos farováveis estejam vulneráveis para a realização da reflexão. Como este trabalho tem como foco o estudo do protocolo DNS, vamos nos a ter tão somente aos servidores DNS. O fluxo pode ser resumido pelos seguintes passos: processamento de requisições DNS válidas por um cliente e retorno de um endereço IP válido (endereço do servidor desejado) para o domínio passado como entrada. Desta maneira, o cliente pode se conectar com o servidor.

Com o passar dos anos, a quantidade de serviços e *websites* cresceram exponencialmente, aumentando ainda mais a necessidade dos serviços de servidores DNS, o que quer dizer: aumento de requisições DNS na rede. Pela alta demanda, DNS trabalha sem necessidade de conexão, característica do protocolo UDP. Esta falta de conexão permite aos atacantes a realização de ataques camuflados, pois não há necessidade de *handshaking* das duas pontas.

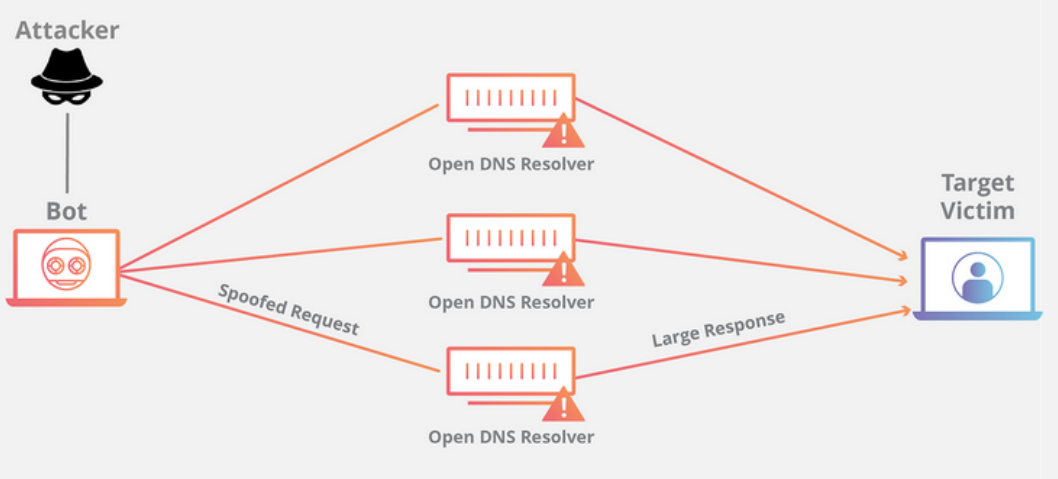

Figura 2.3: Estrutura básica do ataque de negação de serviço por reflexão [4]

O ataque funciona da seguinte forma:

- Após a montagem da estrutura de máquinas conforme a explicação sobre DDoS, cada máquina realiza uma técnica conhecida como *IP Spoofing* [14], que adultera o endereço de origem do pacote para o endereço da vítima. Desta maneira, o servidor que recebe a requisição entende que o pacote partiu da vítima, e não do atacante;
- A rede de ataque então envia requisições DNS válidas com adulteração da origem, diretamente para um servidor DNS ou um conjunto de servidores DNS. Normalmente atribui-se aos pacotes o parâmetro ANY, para que o retorno do servidor seja o maior possível, aumentando ainda mais o fluxo de saída;
- O servidor DNS então recebe as requisições, faz processamentos ANY e responde milhares de pacotes com todos os *types* diretamente para a vítima, sem sequer perceber que está sendo utilizado para refletir o tráfego. Esta característica o torna um refletor;
- A rede da vítima é então inundada de pacotes DNS de respostas válidas. A saturação da rede por inundação leva então o servidor a ter dificuldades de atender clientes, levando-o eventualmente à completa indisponibilidade. O tráfego resultante pelo refletor é caracterizado como amplificação. No fim, o esforço inicial mínimo do atacante gera um grande impacto negativo sobre o servidor vítima.

Como esquematizado na figura 2.3, o atacante tem uma grande facilidade de agir de forma anônima, uma vez que seu endereço de origem real não será identificado após aplicar o *spoofing* nos pacotes de saída. Neste cenário, os servidores DNS só conseguem responder as requisições para a vítima, que por sua vez não realizou requisições equivalentes que justificassem o tráfego. Para entendermos melhor a escolha recorrente do ataque de

negação de serviço por reflexão amplificada, vamos listar os principais pontos que levam à escolha do ataque:

- Poucos recursos podem gerar grandes impactos;
- A origem não ataca diretamente a vítima. Isto significa que, para a vítima, a origem do pacote de resposta não é o atacante, e sim o refletor;
- Para o refletor, a origem do pacote de requisição é a vítima, e não o atacante;
- O ataque é camuflado;
- A resposta pode ser bem maior que a requisição;
- A resposta é rápida;
- O tráfego é considerado legítimo pelo lado do refletor;
- Enquanto o ataque é trivial, a detecção e a resposta ao ataque ainda é lenta [15].

A mesma idéia do ataque por reflexão vale para outros servidores disponíveis na *web*, tais como SSDP, NTP e memcached, apresentados nos trabalhos [16, 17, 18]. A facilidade de execução em comparação aos poucos recursos gastos incentiva essa variante de DDoS. Os dados explicados na seção 1.2 exemplifica claramente como este tipo de ataque é alvejado por agentes mal intencionados da rede, afinal, 70% dos ataques de negação de serviço ainda são executados por meio da amplificação.

Em uma análise sobre a utilização de *memcache* para ataques de negação de serviço por reflexão, pesquisadores da Cloudflare destacaram diversos problemas que permitem a utilização de refletores para este tipo de ataque. "A especificação do UDP mostra ser um dos melhores para amplificação! Há absolutamente zero verificações, e os dados são entregues ao cliente com velocidade máxima! Além disso, o pedido pode ser minúsculo e a resposta enorme", como descrito pela Kaspersky [19]. Embora servidores DNS não sejam os principais candidatos a refletores, ainda assim sua porcentagem é preocupante, mostrando que há muito a ser compreendedido e modificado para obtermos um bom resultado de mitigação a respeito do ataque discutido. É exatamente por este motivo que este trabalho tem o objetivo de contribuir para a comunidade de segurança com a detecção (e identificação) de possíveis refletores em ataques DDoS, facilitando a tomada de decições em relação à mitigação do problema. Esta detecção será discutida de maneira aprofundada nas próximas seções.

Capítulo 3

Ferramenta de Detecção e Identificação de Refletores

Neste capítulo será explicada a ferramenta utilizada para a realização da detecção de refletores em ataques de negação de serviço por reflexão amplificada.

3.1 Estrutura Baseada em Detecção de Arp Spoofing

O trabalho realizado sobre detecção de Arp Spoofing [6] foi uma grande inspiração para o algoritmo implementado por este trabalho. A detecção de Arp Spoofing é resultante de um estudo aprofundado sobre o formato do protocolo Address Resolution Protocol (ARP), que basicamente se refere à dois tipos de mensagens: Request (*who has*) e Reply (*is At*). Embora este protocolo seja simples, é passivo de uma vulnerabilidade que permite um usuário interceptar o tráfego da rede sem ser percebido, apenas redirecionando a rota para seu *host*, repassando o tráfego para o destino logo em seguida.

Assim como descrito no trabalho [6], a técnica de Arp Spoofing consiste em envenenar a cache dos *hosts* que se deseja interceptar o tráfego. O enevenenamento é causado por meio de envios de *replies* sem *requests* antecessores. A mensagem Reply deve conter o seguinte formato: **arp reply *victim_host_ip* is at *my_MAC_address***, mapeando o endereço físico da vítima para o Media Access Control (MAC) do atacante, fazendo com que pacotes destinados à vítima sejam enviados diretamente para o interceptor.

A detecção de ARP Spoofing se baseia na contradição de fluxo contabilizado que existe entre o número de pedidos em relação ao número de requisições, isto é, se a quantidade de *replies* for maior que a quantidade de *requests*, há uma grande probabilidade de estarmos diante de uma tentativa de envenenamento.

Para facilitar a visualização do ataque, vamos considerar um interceptor R escutando o tráfego entre dois *hosts* A e B:

1. R - ARP reply para A: B *is at* MAC_R
2. A - tabela ARP de A: "B está em MAC de R"
3. R - ARP reply para B: A *is at* MAC_R
4. B - tabela ARP de B: "A está em MAC de R"

O esquema acima descreve como o atacante R realiza o envenenamento nas caches de A e B, interceptando a comunicação entre eles. No processo acima, é possível detectar um tráfego de pacotes do tipo *reply* sem *requests* pendentes. Após a detecção, o administrador é rapidamente alertado. Segundo o trabalho [6], ainda é possível que haja falsos positivos durante a detecção, casos especiais que não deveriam ser considerados como tentativa de Arp Spoofing, acabam sendo identificados desta forma.

O algoritmo de detecção de refletores em negação de serviço segue os mesmos princípio de Arp Spoofing. Para mostrar isso, é preciso explorar melhor a estrutura dos pacotes DNS [5].

3.2 Viabilidade da Detecção de Refletores

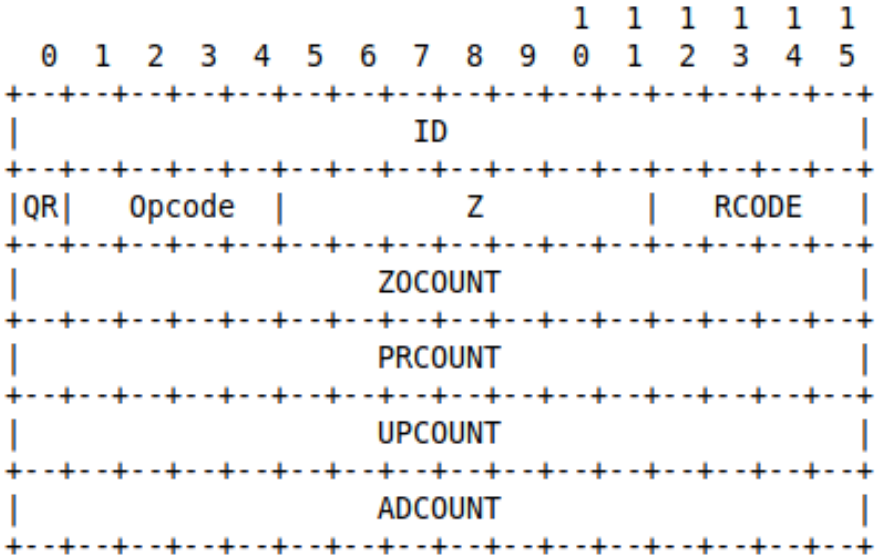

Figura 3.1: Estrutura do pacote DNS [5]

Um pacote DNS possui os campos mostrados na figura 3.1, tais como:

- ID: identificador do pacote;
- QR: identifica que o pacote é Request (0) ou Reply (1);

- Opcode: define o tipo de requisição;
- Z: reservado para uso futuro;
- RCODE: código de resposta
- ZOCOUNT: número de RR's na seção Zona;
- PRCOUNT: número de RR's na seção de pré-requisito;
- UPCOUNT: número de RR's na seção de atualização;
- ADCOUNT: número de RR's na seção de dados adicionais.

Assim como no protocolo ARP, os pacotes do protocolo DNS possuem um campo que descritivo relacionado à pedidos ou respostas: **QR**. Este campo nos permite validar as ações de quaisquer refletores em ataques.

Para facilitar a visualização do ataque, vamos considerar um refletor R respondendo à um enorme volume de requisições de um host atacante A com endereço forjado da vítima V:

1. A - envia pacote DNS: (Request, origem V, destino R, type ANY)
2. R - envia pacote DNS: (Reply, origem R, destino V)
3. V - recebe Reply sem Request pendente

O esquema acima descreve, de forma genérica, o caminho seguido pelo atacante A para atingir a vítima V, por meio do refletor R. Assim como no Arp Spoofing, ataques por reflexão geram um enorme volume de pacotes de resposta sem requisições pendentes. Este tipo de tráfego é característico de refletores, permitindo então detectá-lo por meio da quantidade de pacotes enviados, vindos da mesma origem, cujo campo QR sempre está setado para 1, indicando repostas de requisições válidas. Portanto, a estrutura do pacote DNS nos permite automatizar a detecção em tempo real dos refletores, como podemos ver no fluxo acima, no passo 3.

3.3 A Ferramenta e a Detecção de Refletores

A implementação da ferramenta foi realizada em linguagem C, recuperando todo o tráfego de entrada do servidor. Após receber os pacotes do servidor, a ferramenta de deteção valida os protocolos utilizados. Caso seja pacote do protocolo UDP com DNS, então o pacote é considerado passivo de ser analisado pelo detector.

A figura 3.2 detalha a estrutura da API e do detector de refletores por meio de um diagrama de classes. Segundo a decomposição do diagrama, a classe:

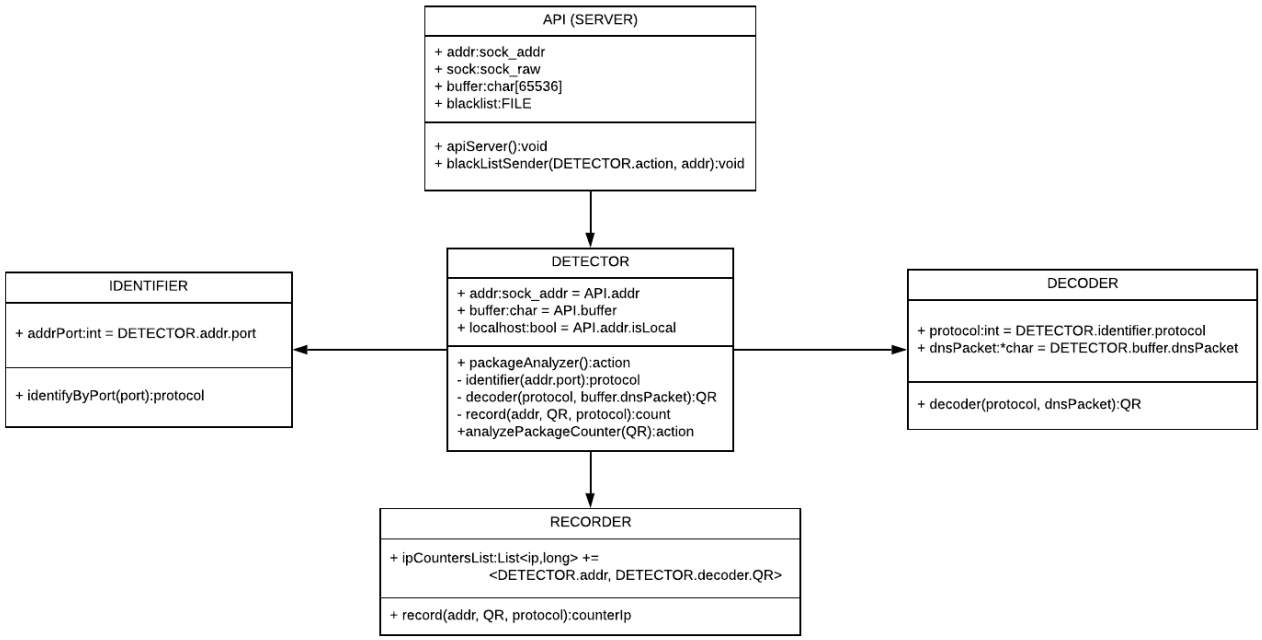

Figura 3.2: Diagrama de Classes da API e Detector de refletores.

- API Server: módulo auxiliar de mitigação. É responsável por recuperar pacotes em qualquer interface da máquina. Sua área de dados guarda informações de endereço do pacote, conteúdo do pacote e uma lista de endereços IP's em uma *blacklist*. Quando o pacote é DNS, chama o serviço do módulo Detector para analisar o pacote;
- Detector: responsável por analisar o pacote, este módulo recebe o endereço do *host* externo à rede e o pacote DNS. Esta classe chama os serviços de outros módulos, tais como Identifier, Decoder e Recorder. Também detecta contradições nas operações, indicando ao módulo API a necessidade de bloquear o *host* da mensagem;
- Identifier: responsável por detectar o protocolo do pacote através da porta de origem [20];
- Decoder: responsável por recuperar a informação sobre o tipo de *query*, ou seja, se é REPLY ou RESPONSE;
- Recorder: responsável por guardar o estado de todos os *hosts* DNS que em algum momento prestaram serviços ao servidor interno da rede. O estado é definido por meio de uma lista de contadores, indicando a quantidade de REPLIES e RESPONSES para cada possível refletor DNS. Cada contador é mapeado para um endereço IP, viabilizando a identificação do agente.

Para cada requisição, o valor do contador correspondente ao *host* é incrementado em 1. Ao receber uma resposta de um servidor DNS, o contador referente a ele é decrementado também em 1. Vale destacar que um tráfego normalizado gera a tendência de que os valores dos contadores devem zerar eventualmente, uma vez que a quantidade de requisições terão a mesma quantidade de respostas de servidores DNS. O algoritmo está descrito a seguir:

se pacote DNS:
  se dns_query **e** origem local:
    contador[host] = contador[host] + 1
  se dns_response **e** origem não local:
    contador[host] = contador[host] - 1

se contador[host] $<$ limite minimo permitido:
  host é um possível refletor!

Durante a análise, constatamos que não há necessidade de controlar com rigidez os contadores de valor positivo, uma vez que requisições válidas com origem interna não são indicadores de ameaça. Após um limite definido, o valor positivo do contador é zerado para não precisar manter seu estado desnecessariamente, reduzindo o tamanho da lista de contadores. Em contra partida, como descrito no algoritmo acima, valores negativos estão sob observação, e se atingir o limite negativo (muitos *replies* sem *requests*), o servidor DNS responsável pela contradição será considerado suspeito.

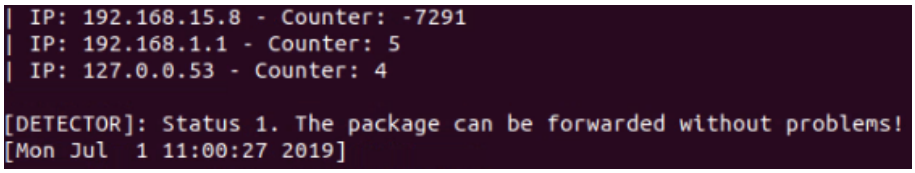

Figura 3.3: Detecção de requisições suspeitas. Contador negativo referente a um *host*.

A estratégia definida acima não acessa dados em bancos de dados, diferentemente do trabalho descrito por [13], que utiliza a mesma estratégia de análise de pacotes de resposta. O acesso ao banco reduz a performance da ferramenta. Ao invés de executar buscas em tabelas, os valores referentes aos refletores são salvos em memória, cujo acesso é extremamente rápido e de fácil acesso. É possível captar e registrar dados por meio de outras ferramentas auxiliares, tal como o Wireshark.

3.4 A Ferramenta e a Identificação de Refletores

Como pode ser visto nos módulos da Figura 3.2, o Recorder guarda uma lista de contadores que mantém informações sobre o par *<IP, Counter>*. Sempre que um contador negativo ultrapassa o limite de tolerância definido pelo administrador, retorna-se as informações referentes ao *host* responsável por refletir o ataque. O módulo Detector recupera essas informaçãos e produz como saída uma lista de endereços que devem ser bloqueados pelo servidor *proxy*, que passa a conter o ataque com o objetivo de proteger a vítima.

Para melhorar ainda mais o desempenho da detecção e identificação, utilizamos o conceito de temporalidade. Este conceito sugere que, se um objeto foi referenciado recentemente, há uma grande probabilidade de que ele seja referenciado novamente. Da mesma forma, se um refletor enviar milhares de pacotes, então este endereço deve ser observado mais intensamente do que outros possíveis refletores que não foram referenciados recentemente. Para cada *reply* sem *request*, o *host* responsável passa a ser o primeiro da lista. Desta maneira ele sempre será acessado primeiro, como mostrado na Figura 3.3, onde o *host* **192.168.15.8** é listado em primeiro na lista, com contador negativo. Sempre que um refletor é detectado, o mesmo é identificado, retirado da lista de contadores e inserido na lista negra.

```
================== DETECTOR MODULE ==================
|Analyzing packet...
[DETECTOR]: Status 0. [BLACKLIST] The host 192.168.15.8 was found in the bla
cklist.
@Action: All packets from this host must be BLOCKED!
[Mon Jul  1 11:00:29 2019]
=============== END - DETECTOR MODULE ===============
```

Figura 3.4: Momento da Detecção de um Refletor 2 segundos após o registro da Figura 3.3.

Se, em algum momento, o módulo de detecção entender que há presença de tráfego malicioso iniciado por um refletor, este módulo irá inserir o IP na *blacklist*, como mostrado na Figura 3.4. Em seguida, o módulo API Server irá avisar ao *Firewall* mais próximo do refletor identificado a necessidade de rejeitar todos os pacotes vindos deste endereço. Quanto mais perto o Proxy estiver do refletor, melhor será para a rede interna, pois assim é possível encontrar rotas alternativas para outros clientes, mesmo que a rede esteja sob ataque de negação de serviço. Assim a disponibilidade do serviço será garantida.

As informações relativas ao limite máximo de pacotes de resposta para um mesmo *host* deverá ser definido pelo administrador da rede, seguindo as suas próprias políticas de segurança.

O fluxo do ataque e a detecção dos refletores pode ser visualizada no diagrama de sequência da Figura 3.5, em que atacante envia uma grande quantidade de requisições com endereço de origem adulterado para o endereço da rede da vítima. O servidor DNS,

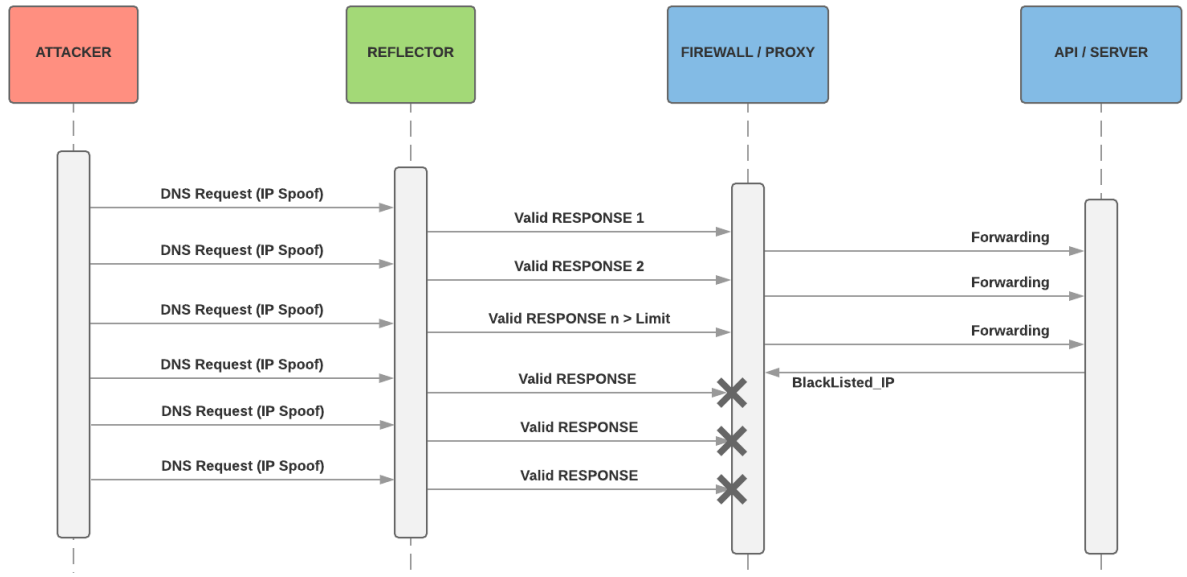

Figura 3.5: Diagrama de Sequência de detecção de um refletor.

por sua vez, processa a requisição do atacante e envia uma resposta válida para o endereço marcado como origem que, no caso, é o endereço da vítima.

Para testarmos a eficiência da detecção, foi montada uma estrutura de testes baseando-se no diagrama de sequência e o esquema da Figura 3.6. Portanto, além do atacante, refletor e vítima, se encontram também o Firewall (ou *proxy*) e o detector, que está localizado na mesma máquina do servidor vítima. Desta forma, o refletor responde diretamente para o *proxy*, que logo encaminha os pacotes de resposta para a vítima. Após o contador negativo do *host* refletor atingir o limite de respostas sem requisições, o detector envia uma mensagem para o proxy via conexão Transmission Control Protocol (TCP), pedindo para que os pacotes vindos do endereço identificado possam ser rejeitados. A partir deste momento, como pode ser visto no diagrama, nenhum patoce será encaminhado para a vítima.

3.5 Estrutura de Detecção e Identificação de Refletores em Tempo Real

Uma bateria de testes foi realizada em laboratório juntamente do trabalho [1], que implementou uma ferramenta de ataque DDoS por reflexão amplificada. Esta ferramenta contribuiu para o refinamento do detector. Para a realização dos testes, utilizamos quatro máquinas e dois roteadores, criando duas subredes. Sobre as máquinas, uma foi utilizada para realizar o ataque, uma para o servidor DNS, uma para manter o *proxy* e a última foi reservada para o serviço (vítima), juntamente com a API Server de detecção de refletores.

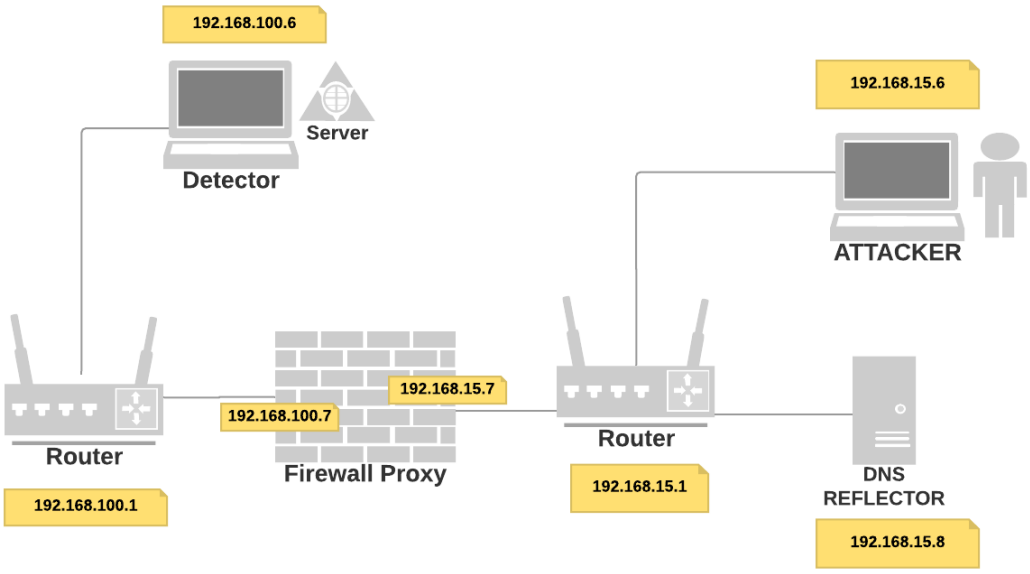

Figura 3.6: Esquema de Teste de Ataque de Negação de Serviço por Reflexão Amplificada e Detecção/Identificação de Refletores.

Como pode ser visto na Figura 3.6, o esquema foi montado seguindo o fluxo do diagrama de sequência da Figura 3.5. As máquinas da sub-rede **192.168.100.255** são o refletor, servidor e Firewall. Na outra subrede, com endereço **192.168.15.255**, se encontram o atacante, o servidor DNS e a segunda interface do Firewall.

É importante destacar que, para reduzir a perda de pacotes durante a detecção do ataque, optamos por utilizar duas interfaces distintas no *proxy*, uma reservada apenas para pacotes vindos da Wide Area Network (WAN), e outra interface para a comunicação na Local Area Network (LAN).

A estrutura de teste foi criada com o objetivo de, não só testar a capacidade de detecção da ferramenta, mas também de exemplificar um cenário de detecção juntamente com a mitigação do ataque. No caso, é possível implementar uma rota alternativa ligada ao roteador **192.168.100.1**, enquanto o *proxy* sofre a inundação do ataque. Portanto, neste exemplo, existe uma forma de mitigar o ataque por meio da contribuição da ferramenta implementada por este trabalho.

Durante a realização dos testes, obtivemos uma série de resultados que serão descritos na próxima seção.

Capítulo 4

Testes e Resultados

Após a construção da ferramenta, foi realizada uma série de testes de detecção de refletores. A estrutura da rede, apresentada na Figura 3.6, nos ajudou a compreender o funcionamento de um ataque DDoS por reflexão amplificada. A decisão da arquitetura foi incentivada pela demonstração de uma possível mitigação, e não apenas do processo de detecção do refletor em si.

4.1 Rede de Dispositivos

Foram utilizados quatro máquinas e roteadores para a realização do ataque e detecção dos refletores. A especificação está definida na tabela 4.1:

Tabela 4.1: Captura de Pacotes - Especificação dos Dispositivos

Máquina	**Hardware**	**OS**	**Application**	**Link Type**
Detector	Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz (with SSE4.2)	Linux 4.15.0-52-generic	Dumpcap (Wireshark) 2.6.8 (Git v2.6.8 packaged as 2.6.8-1∼ubuntu18.04.0)	Ethernet
Proxy	Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz (with SSE4.2)	4.15.0-51-generic	Dumpcap (Wireshark) 2.6.8 (Git v2.6.8 packaged as 2.6.8-1∼ubuntu18.04.0)	Ethernet
DNS Reflector	Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz (with SSE4.2)	Linux 5.0.0-17-generic	Dumpcap (Wireshark) 2.6.8 (Git v2.6.8 packaged as 2.6.8-1)	Ethernet

A Figura 4.1 mostra a estrutura da rede utilizada na realização dos testes, baseando-se no esquema apresentado na Figura 3.6.

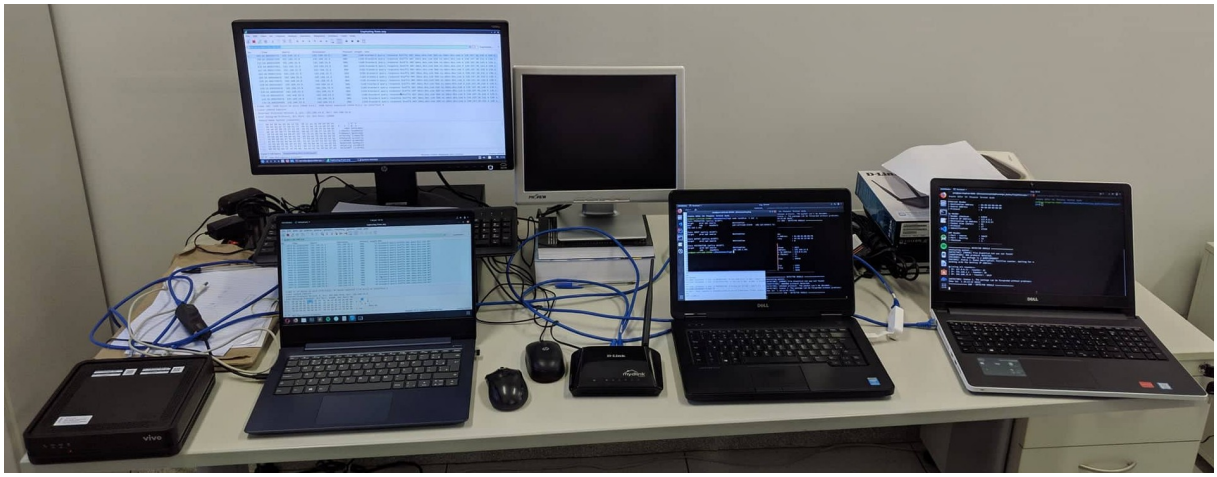

Figura 4.1: Dispositivos Utilizados para Testes.

4.2 Detectando e Identificando Refletores - Análise de Dados

Para a realização do ataque, utilizamos uma ferramenta [1] capaz de gerar pacotes DNS com endereços IP's de origem forjada, que foram direcionados ao servidor DNS **192.168.15.8**. As requisições ANY então são respondidas e enviadas para a vítima, cuja estrutura interna executa uma filtragem do tráfego através do *proxy*. Portanto, o endereço de origem forjado pelo atacante é apontado para a interface **192.168.15.7** do *firewall*. Em seguida, o pacote é encaminhado ao servidor pela interface **192.168.100.7**, enquanto o detector escuta os pacotes de entrada e realiza a detecção do refletor DNS.

A coleta de dados foi realizada por meio de níveis de ataques, ou seja, aumentamos a quantidade de pacotes que a ferramenta de ataque foi capaz de enviar ao refletor a cada teste realizado. O objetivo da execução em níveis diferentes foi a verificação do desempenho da ferramenta de detecção sob o estresse da rede em diferentes massas de ataque. Os dados referentes aos recursos utilizados pelo detector também foram coletados. É necessário observar que os níveis de ataque testados iniciaram a partir do 2, pois o nível 1 realizava o envio de apenas 1 pacote por segundo, sem estresse so servidor. As tabelas a seguir apresentam os dados coletados nos testes em cada nível de ataque:

Tabela 4.2: Captura de Pacotes - Ataque Nível 2

Máquina	**Qtd. Pacotes**	**Tempo de Execução (s)**
Limite Detector	100	10
Proxy	147	14.0025
Refletor DNS	147	14.0030

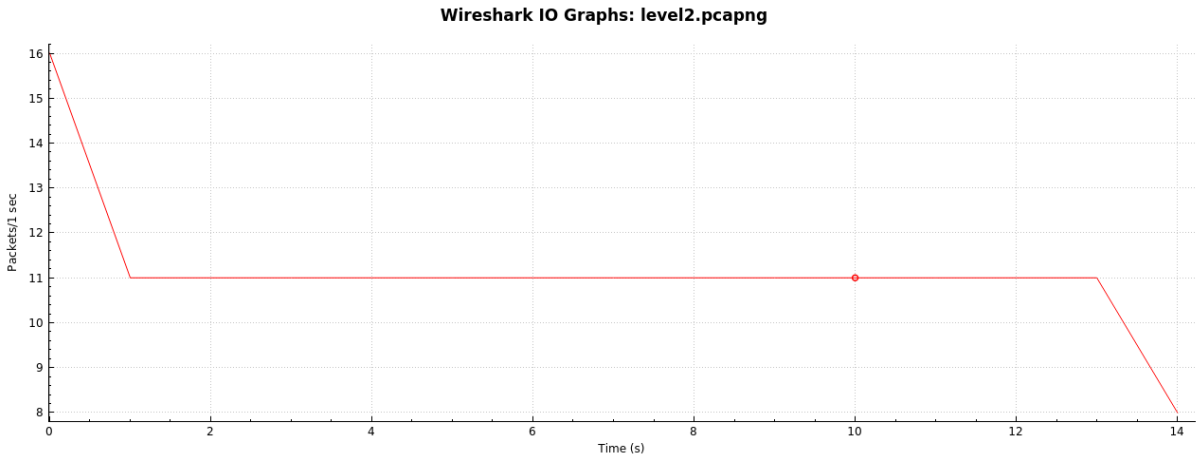

Figura 4.2: Ataque em Nível 2: Quantidade de pacotes por segundo.

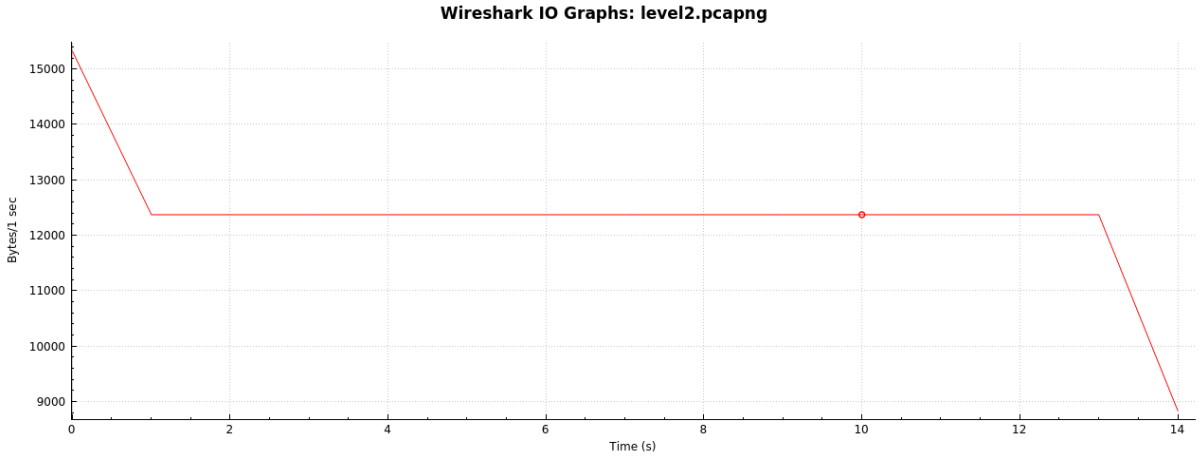

Figura 4.3: Ataque em Nível 2: Quantidade de bytes por segundo.

Os dados apresentados na Tabela 4.2, Figura 4.2 e 4.3 mostram a maneira como o ataque foi realizado. Neste nível, foram enviados 147 pacotes (10 pacotes por segundo) pelo refletor até o proxy, que por sua vez encaminhou os pacotes para a vítima. Com um limite de tolerância de 100 pacotes, a ferramenta foi capaz de detectar e identificar o refletor aproximadamente 10 segundos após o início do ataque, que logo em seguida foi inserido em uma *blacklist* e enviado até o *proxy* para impedir o encaminhamento de pacotes, bloqueando o tráfego. Este teste mostrou-se eficiente, pois a quantidade de segundos para completar o ataque condiz com a quantidade de segundos necessária para analisar os pacotes enviados neste intervalo de tempo. O teste teve duração de aproximadamente 14.0030 segundos. É possível observar no gráfico da figura 4.2 a quantidade de pacotes por segundo enviados pelo refletor.

Assim como no nível 2, os dados apresentados na Tabela 4.4 e Figura 4.5 mostram a maneira de como o ataque foi realizado no nível 3. Foram enviados um total de 1.082

Tabela 4.3: Captura de Pacotes - Ataque Nível 3

Máquina	**Qtd. pacotes**	**Tempo de Execução (s)**
Limite Detector	500	6
Proxy	1.082	9.0119
Refletor DNS	1.082	9.0123

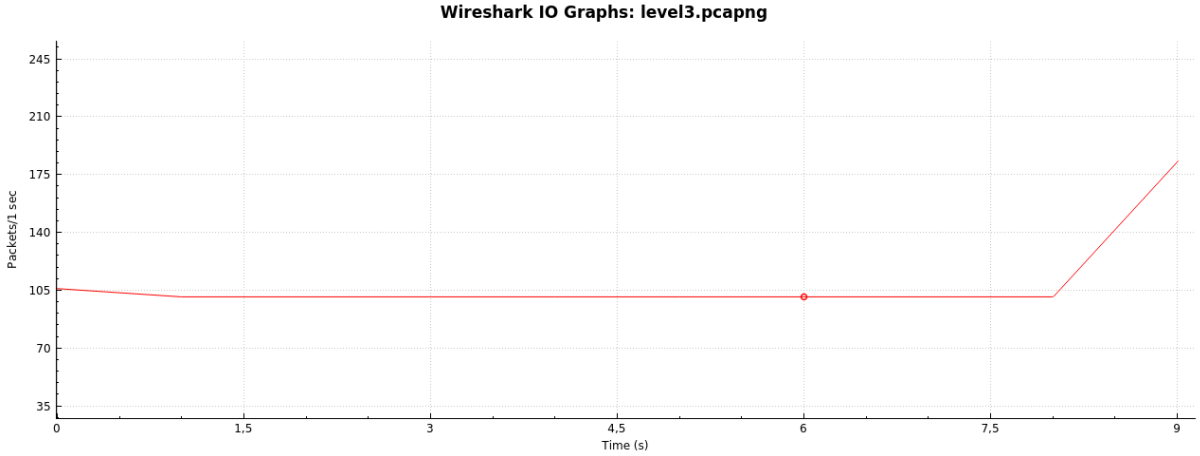

Figura 4.4: Ataque em Nível 3: Quantidade de pacotes por segundo.

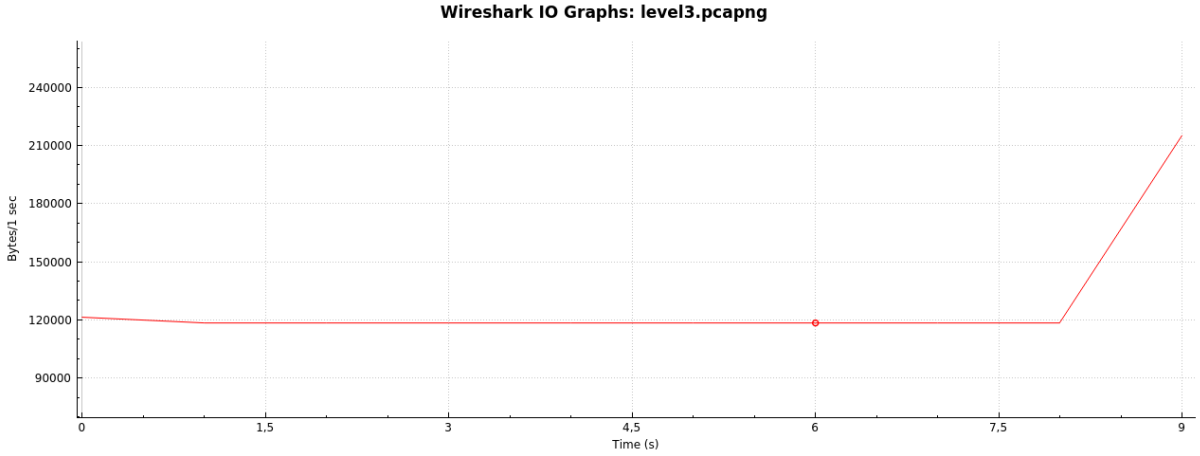

Figura 4.5: Ataque em Nível 3: Quantidade de bytes por segundo.

pacotes (100 pacotes por segundo) pelo refletor até o proxy, que por sua vez encaminhou os pacotes para a vítima. Com um limite de tolerância de 500 pacotes, a ferramenta foi capaz de detectar e identificar o refletor aproximadamente 6 segundos (quase a metade do tempo do ataque total) após o início do ataque, que logo em seguida foi inserido em uma *blacklist* e enviado até o *proxy* para impedir o encaminhamento de pacotes, bloqueando o tráfego. Este segundo teste também mostrou-se eficiente, pois a quantidade de segundos para completar o ataque condiz com a quantidade de segundos necessária para a captura de pacotes enviados neste intervalo de tempo. O teste teve duração de aproximadamente

9.0123 segundos. É possível observar no gráfico da Figura 4.4 a quantidade de pacotes por segundo enviados pelo refletor.

Tabela 4.4: Captura de Pacotes - Ataque Nível 4

Máquina	**Qtd. pacotes**	**Tempo de Execução (s)**
Limite Detector	2000	6
Proxy	3.549	9.0133
Refletor DNS	3.577	9.0140

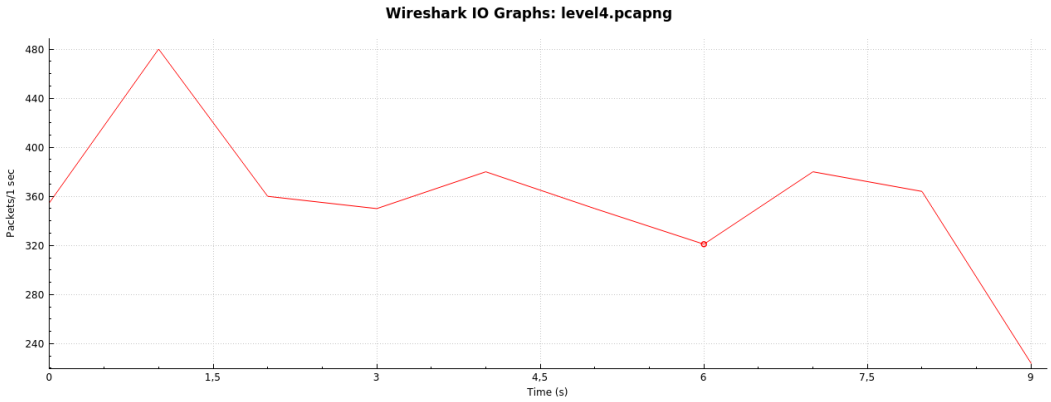

Figura 4.6: Ataque em Nível 4: Quantidade de pacotes por segundo.

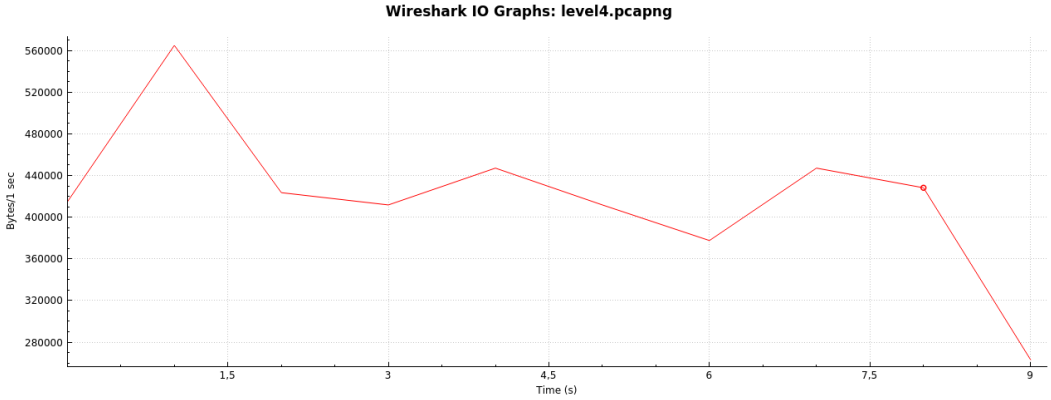

Figura 4.7: Ataque em Nível 4: Quantidade de bytes por segundo.

No nível 4, foram enviados um total de 3.577 pacotes (1000 pacotes por segundo) pelo refletor até o proxy, havendo indícios de saturação durante o encaminhamento de pacotes, havendo perda de 28 pacotes no processo. Com um limite de tolerância de 2000 pacotes, a ferramenta foi capaz de detectar e identificar o refletor aproximadamente 6 segundos após o início do ataque. Este teste também mostrou-se eficiente, pois a quantidade necessária de segundos para completar o ataque condiz com a quantidade necessária de segundos de captura de pacotes enviados neste intervalo de tempo. O teste

teve duração de aproximadamente 9.0140 segundos. É possível observar no gráfico da Figura 4.6 a quantidade de pacotes por segundo enviados pelo refletor.

Tabela 4.5: Captura de Pacotes - Ataque Nível 5

Máquina	**Qtd. pacotes**	**Tempo de Execução (s)**
Limite Detector	5000	9
Proxy	6.802	9.0961
Refletor DNS	7.873	9.0169

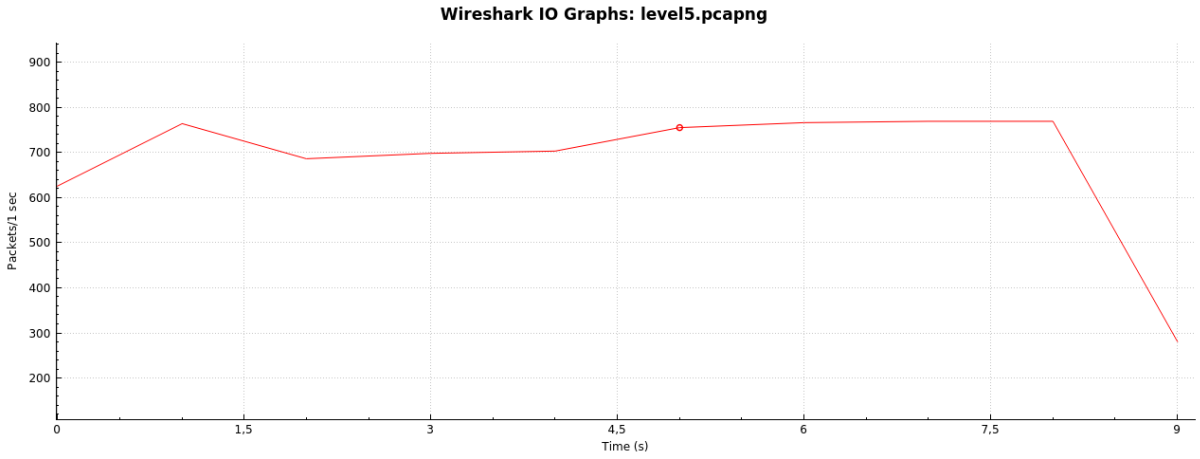

Figura 4.8: Ataque em Nível 5: Quantidade de pacotes por segundo.

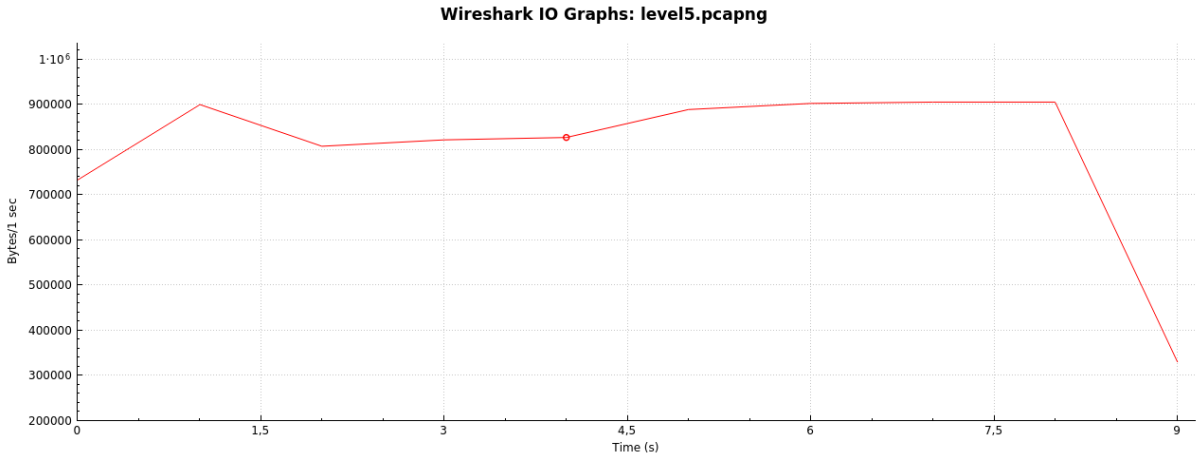

Figura 4.9: Ataque em Nível 5: Quantidade de bytes por segundo.

No nível 4, pôde-se observar um pequeno indício de saturação na rede ao tentar encaminhar pacotes para a vítima. Da mesma forma, o nível 5 também sofre com a saturação devido a grande quantidade de pacotes enviados pelo refletor. Neste nível, o *proxy* chegou a encaminhar 6.802 dos 7.873 pacotes enviados pelo refletor. Com um limite de tolerância

de 5000 pacotes, a ferramenta foi capaz de detectar e identificar o refletor aproximadamente 9 segundos após o início do ataque. Este teste mais uma vez mostra a coerência da detecção, pois a quantidade necessária de segundos para completar o ataque condiz com a quantidade de segundos necessária para completar a detecção. O teste teve duração de aproximadamente 9.0169 segundos. É possível observar no gráfico da Figura 4.8 a quantidade de pacotes por segundo enviados pelo refletor, e no gráfico da Figura 4.9 a quantidade de bytes por segundo.

Tabela 4.6: Captura de Pacotes - Ataque Nível 6

Máquina	**Qtd. pacotes**	**Tempo de Execução (s)**
Limite Detector	10.000	6
Proxy	24.657	9.0980
Refletor DNS	25.547	9.0177

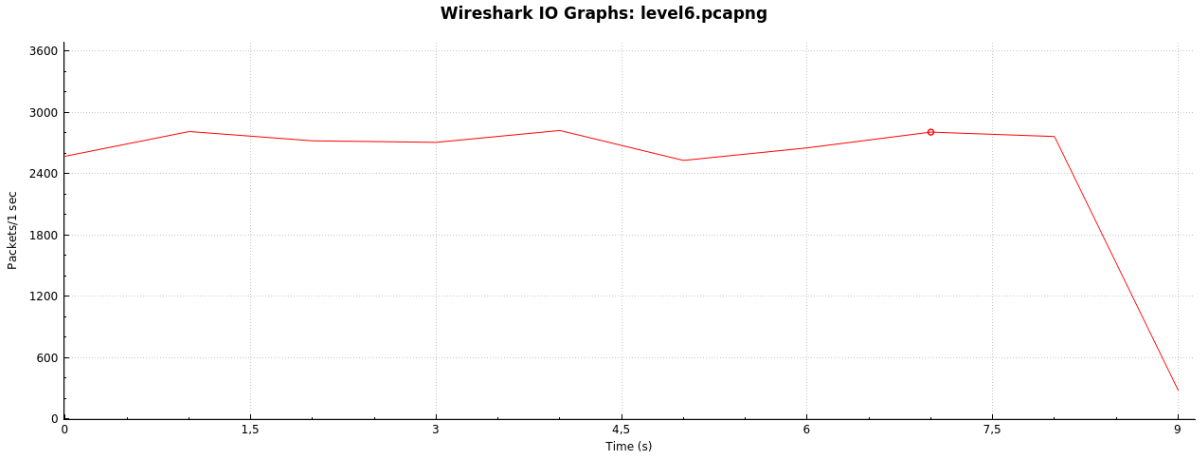

Figura 4.10: Ataque em Nível 6: Quantidade de pacotes por segundo.

No nível 6, a quantidade de pacotes enviados pelo refletor teve um aumento considerável. Este nível também sofreu saturação, assim como o nível 4 e 5. O *proxy* encaminhou 24.657 dos 25.547 pacotes enviados pelo refletor. Com um limite de tolerância de 10.000 pacotes, a ferramenta foi capaz de detectar e identificar o refletor aproximadamente 6 segundos após o início do ataque, 3 segundos a menos que no nível 5. Também nota-se coerência nos dados coletados sobre a detecção. O teste teve duração de aproximadamente 9.0177 segundos. É possível observar no gráfico da Figura 4.10 a quantidade de pacotes por segundo enviados pelo refletor.

O nível 7 obteve um grande sucesso para a ferramenta de ataque. Obtivemos o mais alto valor da quantidade de pacotes enviados até a vítima. A saturação também atingiu um grande pico, indicando a limitação dos equipamentos utilizados. O *proxy* encaminhou apenas 25.028 dos 121.746 pacotes enviados pelo refletor. Com um limite de tolerância de

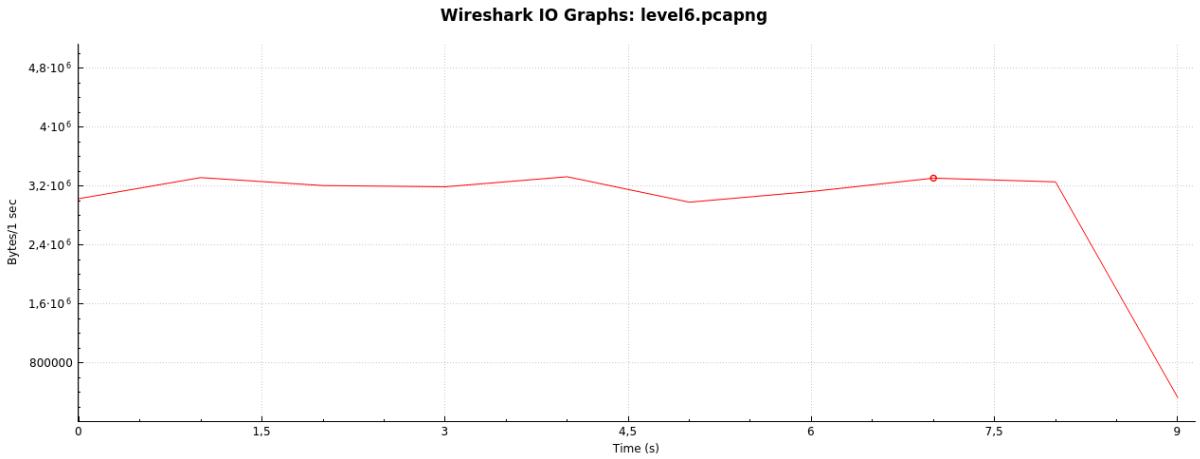

Figura 4.11: Ataque em Nível 6: Quantidade de bytes por segundo.

Tabela 4.7: Captura de Pacotes - Ataque Nível 7

Máquina	**Qtd. pacotes**	**Tempo de Execução (s)**
Limite Detector	20000	9
Proxy	25.028	9.9099
Refletor DNS	121.746	9.0116

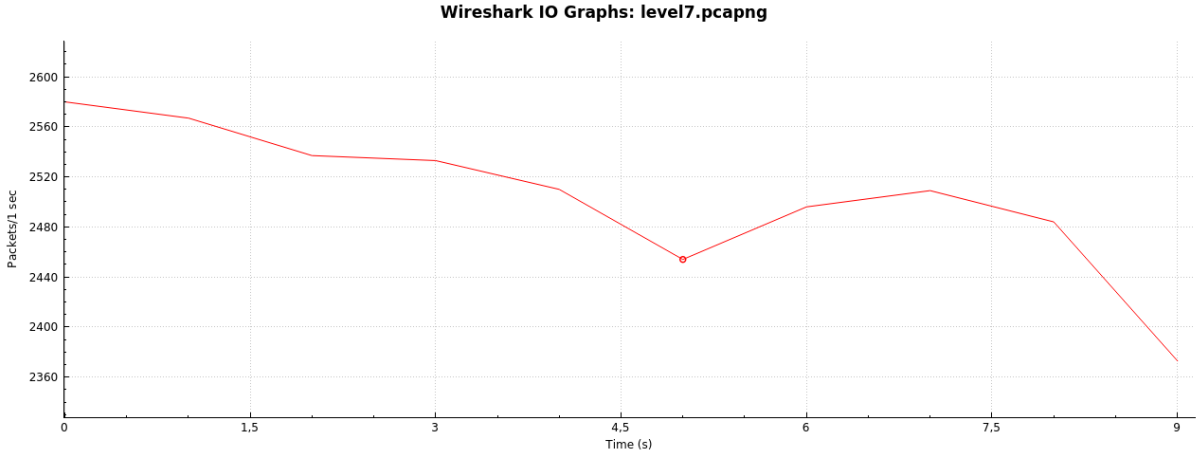

Figura 4.12: Ataque em Nível 7: Quantidade de pacotes por segundo.

20.000 pacotes, a ferramenta foi capaz de detectar e identificar o refletor aproximadamente 9 segundos após o início do ataque. O resultado também indica coerência nos dados coletados sobre a detecção pelo mesmo motivo das fases anteriores. O teste teve duração de aproximadamente 9.0116 segundos. É possível observar no gráfico da Figura 4.12 a quantidade de pacotes por segundo enviados pelo refletor, e na Figura 4.13 a quantidade de bytes por segundo.

A partir do nível 8, há grande saturação por parte do refletor. O atacante é capaz de enviar mais pacotes do que o refletor consegue suportar. Da mesma forma, há grande

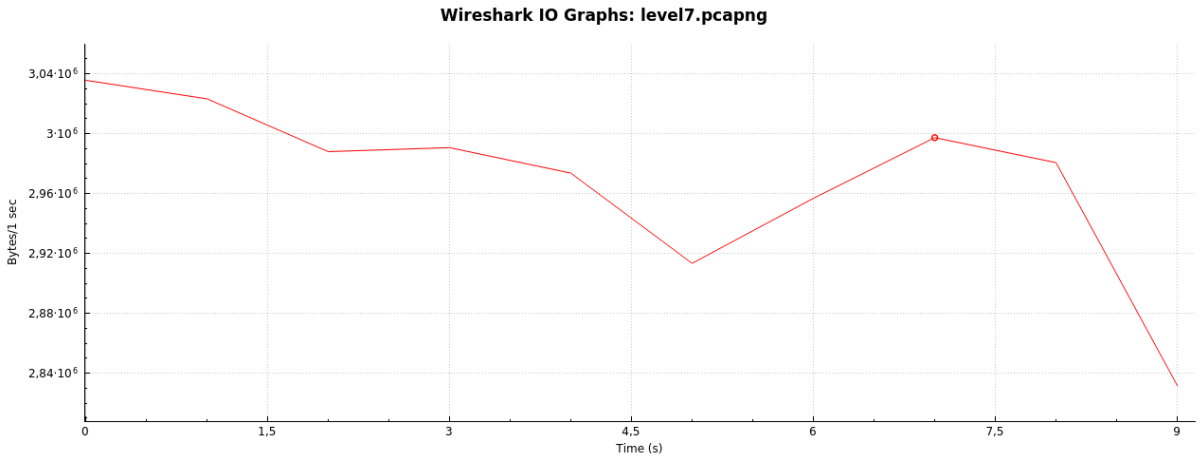

Figura 4.13: Ataque em Nível 7: Quantidade de bytes por segundo.

Tabela 4.8: Captura de Pacotes - Ataque Nível 8

Máquina	**Qtd. pacotes**	**Tempo de Execução (s)**
Limite Detector	10.000	5
Proxy	12.799	5.1899
Refletor DNS	98.648	4.0362

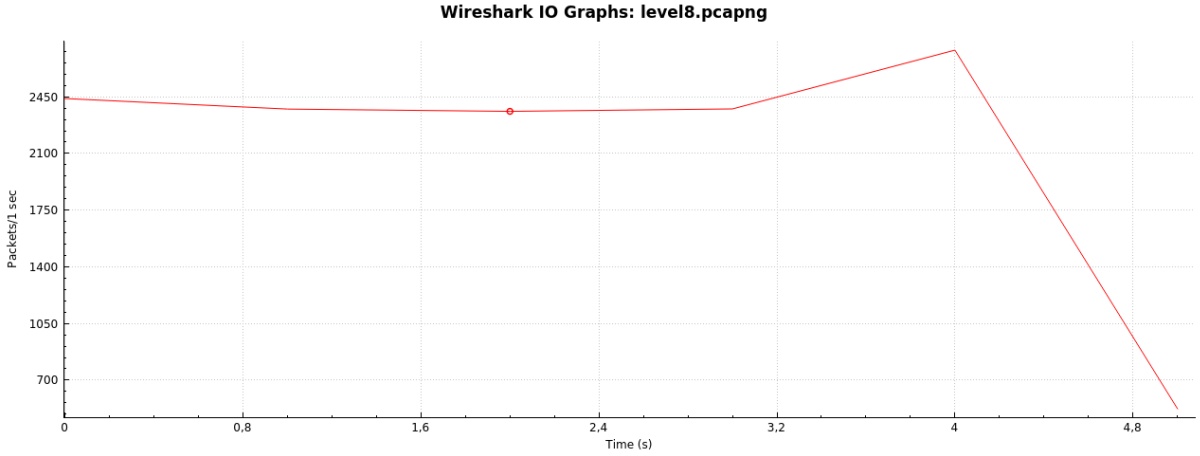

Figura 4.14: Ataque em Nível 8: Quantidade de pacotes por segundo.

saturação por parte do proxy durante o encaminhamento de pacotes. É possível que boa parte se perca também no roteador entre os dois *hosts*. O *proxy* encaminhou apenas 12.799 (metade do nível 7) dos 98.648 pacotes enviados pelo refletor. Com um limite de tolerância de 10.000 pacotes, a ferramenta foi capaz de detectar e identificar o refletor aproximadamente 5 segundos após o início do ataque, que também indica coerência nos dados coletados sobre a detecção, pelo mesmo motivo das fases anteriores. O teste teve duração de aproximadamente 4.0362 segundos. É possível observar no gráfico da Figura 4.14 quantidade de pacotes por segundo enviados pelo refletor.

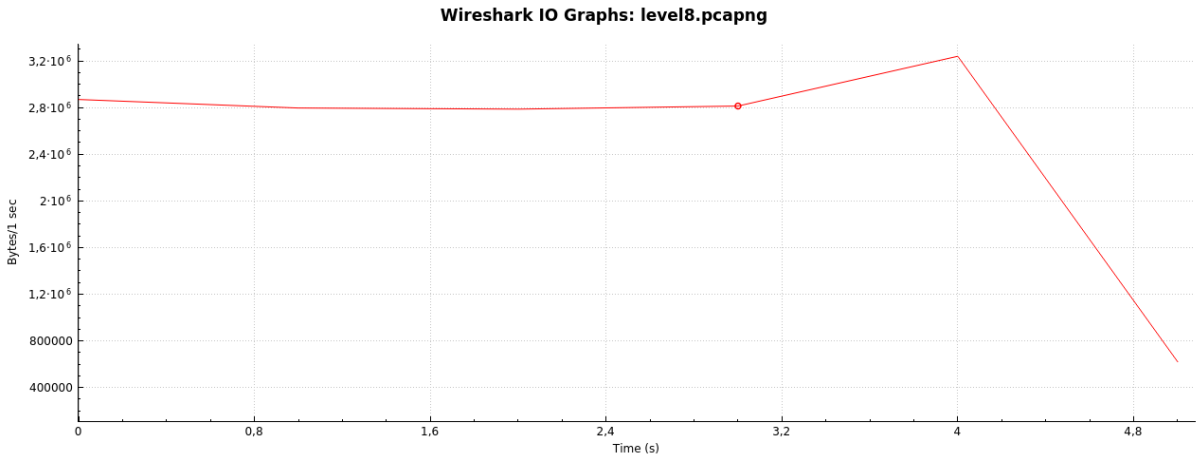

Figura 4.15: Ataque em Nível 8: Quantidade de bytes por segundo.

Tabela 4.9: Captura de Pacotes - Ataque Nível 9

Máquina	**Qtd. pacotes**	**Tempo de Execução (s)**
Limite Detector	10.000	5
Proxy	12.909	5.1886
Refletor DNS	75.995	4.0410

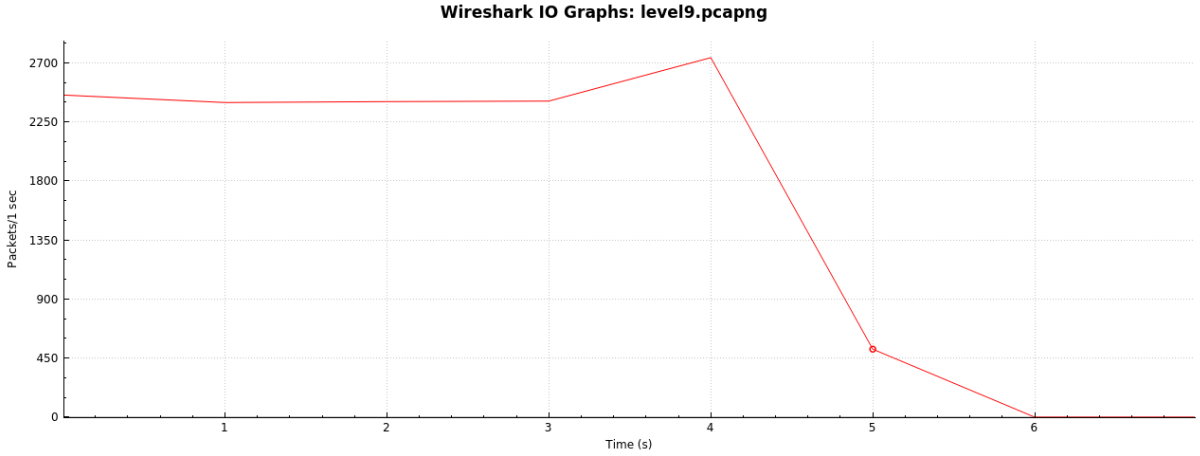

Figura 4.16: Ataque em Nível 9: Quantidade de pacotes por segundo.

O nível 9 obteve resultados parecidos com os resultados do nível 8. Da mesma forma, há grande saturação no envio dos pacotes, desde o atacante até a vítima. O *proxy* encaminhou apenas 12.909 dos 75.995 pacotes enviados pelo refletor. Com um limite de tolerância de 10.000 pacotes, a ferramenta foi capaz de detectar e identificar o refletor aproximadamente 5 segundos após o início do ataque, o que indica mais uma vez coerência nos dados coletados sobre a detecção, pelo mesmo motivo das fases anteriores. O teste teve duração de aproximadamente 4.0410 segundos. É possível observar no gráfico da Figura 4.16 a quantidade de pacotes por segundo enviados pelo refletor.

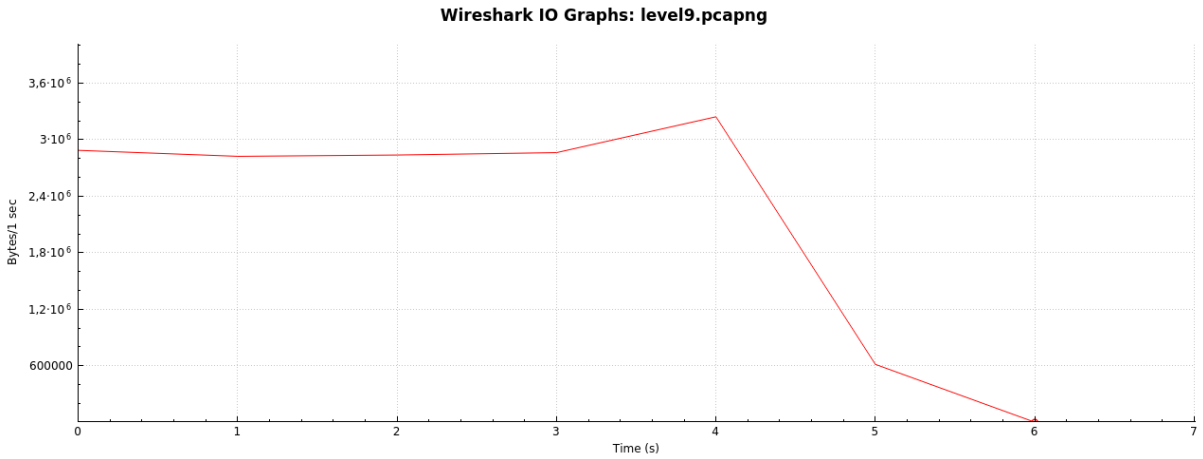

Figura 4.17: Ataque em Nível 9: Quantidade de bytes por segundo.

Tabela 4.10: Captura de Pacotes - Ataque Nível 10

Máquina	**Qtd. pacotes**	**Tempo de Execução (s)**
Limite Detector	10.000	5
Proxy	12.845	5.1869
Refletor DNS	90.863	4.0406

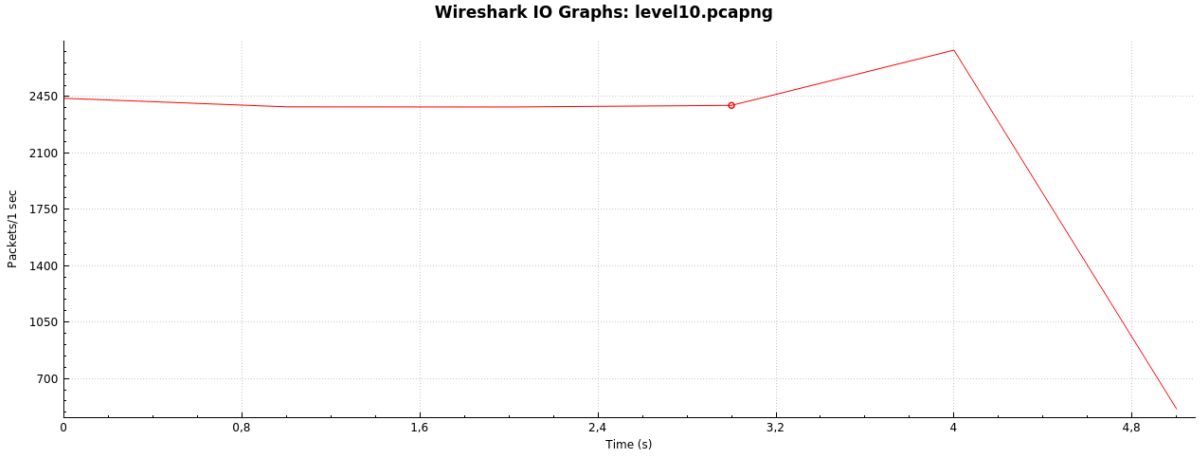

Figura 4.18: Ataque em Nível 10: Quantidade de pacotes por segundo.

Por fim, o nível 10 também não obteve resultados diferentes do nível 8 e 9, pois também há grande saturação no envio dos pacotes. O *proxy* encaminhou apenas 12.845 dos 90.863 pacotes enviados pelo refletor. Com um limite de tolerância de 10.000 pacotes, a ferramenta foi capaz de detectar e identificar o refletor aproximadamente 5 segundos após o início do ataque, indicando pela última vez a coerência dos dados coletados sobre a detecção. O teste teve duração de aproximadamente 4.0406 segundos. É possível observar no gráfico da Figura 4.18 a quantidade de pacotes por segundo enviados pelo refletor.

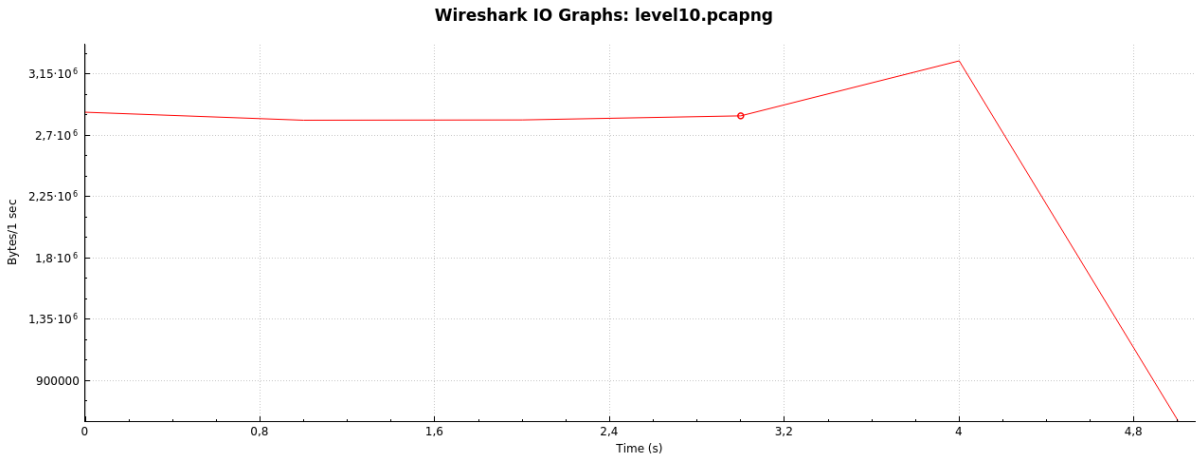

Figura 4.19: Ataque em Nível 10: Quantidade de bytes por segundo.

4.3 Considerações Sobre a Análise

Como observado nos dados, grande parte da saturação se deve às limitações da rede. Foi possível provar que grande parte dos pacotes encaminhados pelo refletor foram de fato processados pelo detector, mostrando a eficiência de captura e de processamento. Esta precisão é essencial para a segurança da rede, pois enquanto a rede está sob ataque, existe a necessidade de bloquear o refletor o mais rápido possível. Todos os testes foram realizados com um limite de tolerância alto, para analisar o tempo de detecção sob estresse da rede após um longo período. Pode-se considerar que este limite seja ainda menor em um cenário real, diminuindo drasticamente o tempo que o detector levará para inserir o endereço do agente de reflexão na *blacklist*, fazendo com que o *proxy* mais próximo a ele inicie ainda mais cedo o processo de rejeição dos pacotes. A ferramenta de detecção foi capaz de identificar o refletor em todos os testes realizados, conforme mostram os resultados.

Capítulo 5

Conclusão e Trabalhos Futuros

5.1 Conclusão

Esta monografia apresentou a viabilidade de detecção de refletores em ataques de negação de serviço por reflexão amplificada, que abusam da vulnerabilidade de servidores DNS para a realização do ataque.

Primeiramente, discutiu-se o crescimento de servidores na internet nos últimos anos, juntamente com o aumento de vulnerabilidades em diversos níveis. Vimos que o avanço tecnológico pode contribuir para a evolução das formas de ataques, e dentre eles citou-se o ataque de negação de serviço por reflexão amplificada. Em seguida, apresentamos a estrutura do ataque e propomos uma forma alternativa aos outros trabalhos discutidos sobre a detecção de um agente refletor.

Discutiu-se a implementação da ferramenta de detecção de refletores, que baseou-se na análise de comportamento do fluxo da rede gerada por meio de um ataque por reflexão de servidores DNS, um tráfego volumoso contendo apenas pacotes de respostas definidos pelo parâmetro QR igual a 1. Este desbalanceamento de pedidos e respostas nos levou a uma estratégia capaz de detectar esta característica. Em seguida, após a implementação, realizamos uma bateria de testes em diversos níveis, em conjunto com o trabalho de [1] que implementou a ferramenta de ataque. Estes testes nos revelaram a capacidade de detecção em todos os casos testados, considerando as limitações de *hardware* e relevando a consistência da quantidade de pacotes enviados pelo refletor até a máquina, e a quantidade de pacotes detectados pela máquina servidor.

Os resultados mostraram coerência na correlação entre a quantidade de pacotes encaminhados pelo *proxy* até a vítima, tempo de envio do atacante e o tempo de detecção e identificação do refletor. É necessário destacar que a decisão sobre qual deve ser o limite de tolerância do detector não foi considerada como sendo uma variável do estudo, mas uma decisão dos administradores da rede. Este limite foi aumentado em seu máximo,

conforme a proposta do nível do ataque, para que pudéssemos ter um maior espectro sobre a coleta de informações. É possível observar que a saturação impediu a rapidez de detecção, uma vez que a quantidade de pacotes analisados foi menor que a quantidade enviada pelo refletor em níveis mais elevados.

Sobre a metodologia utilizada, devemos considerar uma possível ocorrência de falso positivo, no caso onde o refletor adultera o IP de origem de um servidor DNS e então envia um grande volume de pacotes diretamente para a vítima. Desta maneira, o atacante a ser bloqueado não deveria ser considerado um refletor, mas apenas um agente de ataque direto.

5.2 Trabalhos Futuros

Existe necessidade de refinamento da ferramenta, pois, como foi dito na subseção anterior, existem algumas vulnerabilidades na versão atual que podem comprometer a precisão de detecção exclusiva de refletores, sem gerar falsos positivos. Por este motivo, temos a necessidade das seguintes colaborações:

- Tratamento de falsos positivos: Fazer uma segunda verificação durante a detecção de refletores. Esta verificação deve ser responsável por detectar cooperação do servidor DNS durante um tráfego inicialmente considerado malicioso, que o coloca como posição de agente refletor. Esta segunda verificação pode ser feita de várias maneiras, mas sua importância trará precisão na hora de detectar.
- Expansão da Ferramenta: o próximo passo é, devido aos resultados dos testes, viabilizar a detecção também para outros protocolos, tais como SSDP, NTP e SNMP.
- Execução de testes com outro conjunto de equipamentos: a diversificação do cenário de testes nos ajuda a coletar dados mais precisos sobre a integridade dos resultados da detecção. Além do mais, a criação de uma rede mais fiel a um cenário real nos ajudará a prever situações não verificadas nesta versão do trabalho.

Referências

[1] Sousa Saldanha, Rodrigo de: *Ataque distribuído de negação de serviço por reflexão amplificada explorando o protocolo domain name system.* Monografia apresentada como requisito parcial para conclusão do Curso de Engenharia de Computação, 2019. v, vi, 4, 17, 20, 31

[2] Mon, DDoS: *Insight into global ddos threat landscape.* https://ddosmon.net/insight/, 2019. viii, 3

[3] Mahjabin, Tasnuva, Yang Xiao Guang Sun e Wangdong Jiang: *A survey of distributed denial-of-service attack, prevention, and mitigation techniques.* International Journal of Distributed Sensor Networks, página 4, 2017. viii, 6

[4] Cloudflare: *Dns amplification attack.* https://www.cloudflare.com/learning/ddos/dns-amplification-ddos-attack/, 2019. viii, 9

[5] Bound, Jim: *Dynamic updates in the domain name system (dns update).* https://tools.ietf.org/html/rfc2136, 1997. viii, 12

[6] Marco Antônio Carnut, João J. C. Gondim: *Arp spoofing detection on switched ethernet networks : A feasibility study.* página 2, dezembro 2010. 2, 11, 12

[7] Behal, Sunny e Krishan Saluja: *Measuring the impact of ddos attacks on web services -a realtime experimentation.* International Journal of Computer Science and Information Security, 14:323–330, outubro 2016. 5

[8] Gondim, João José Costa, Robson de Oliveira Albuquerque, Anderson Clayton Alves Nascimento, L. Javier García-Villalba e Taewan Kim: *A methodological approach for assessing amplified reflection distributed denial of service on the internet of things.* Em *Sensors*, 2016. 6

[9] Cloudflare: *Famous ddos attacks | the largest ddos attacks of all time.* https://www.cloudflare.com/learning/ddos/famous-ddos-attacks/, 2019. 6

[10] Newman, Lily Hay: *Github survived the biggest ddos attack ever recorded.* https://www.wired.com/story/github-ddos-memcached/, 2018. 6

[11] Postel, J.: *User datagram protocol.* https://tools.ietf.org/html/rfc768, 19. 7

[12] Roolvink, Stephan: *Detecting attacks involving dns servers - a netflow data based approach.* Computer Communication Review, página 36, 2008. 7

[13] Kambourakis, Georgios, Tassos Moschos, Dimitris Geneiatakis e Stefanos Gritzalis: *Detecting dns amplification attacks*. páginas 185–196, outubro 2007. 7, 15

[14] Mukhopadhyay, Debajyoti, Byung Jun Oh, Sang Heon Shim e Young Chon Kim: *A study on recent approaches in handling ddos attacks*. 5º Simpósio Segurança em Informática, novembro 2003. 9

[15] Alexandru G. Bardas, Loai Zomlot, Sathya Chandran Sundaramurthy e HP Labs; Marc R. Eisenbarth HP TippingPoint Xinming Ou, Kansas State University; S. Raj Rajagopalan: *Classication of udp trafc for ddos detection*. 5th USENIX Workshop on Large-Scale Exploits and Emergent Threats, página 2, 2012. 10

[16] Miranda, Igor F.: *Ataque de negação de serviço por reflexão amplificada explorando memcached*. Monografia apresentada como requisito parcial para conclusão do Curso de Engenharia de Computação, 2018. 10

[17] Duarte, Eduardo S.: *Ataque distribuído de negação de serviço por reflexão amplificada usando o protocolo simple service discovery protocol*. Monografia apresentada como requisito parcial para conclusão do Curso de Engenharia de Computação, 2018. 10

[18] Souza Vieira, Alexander André de: *Ataque distribuído de negação de serviço por reflexão amplificada usando network time protocol*. Monografia apresentada como requisito parcial para conclusão do Curso de Engenharia de Computação, 2019. 10

[19] Kaspersky: *Novo método amplifica ataques ddos*. https://www.kaspersky.com.br/blog/novo-metodo-amplifica-ataques-ddos/10120/, 2018. 10

[20] Paxson, Vern: *An analysis of using reflectors for distributed denial-of-service attacks*. Computer Communication Review, 31:7, 2001. 14