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

Eris: uma ferramenta para avaliação de resiliência em servidores DNS

Thais Fernanda de Castro Garcia

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

Orientador
Prof. Dr. João José Costa Gondim

Brasília
2025

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

Eris: uma ferramenta para avaliação de resiliência em servidores DNS

Thais Fernanda de Castro Garcia

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

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

Prof. Dr.
Marcos Fagundes Caetano

Prof. Dr.
Robson de Oliveira Albuquerque

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

Brasília, 27 de janeiro de 2025

Agradecimentos

Agradeço à minha mãe por ser minha melhor amiga e por seu apoio inestimável, que supre a necessidade de qualquer outro. Também agradeço ao meu amor, pelo apoio incondicional e por ter aceitado que inúmeros dates tenham sido resumidos a passar a madrugada olhando terminais Linux. Agradeço aos meus amigos que cruzaram meu caminho no jardim de infância e até hoje se fazem presentes.

Sou imensamente grata à Universidade de Brasília por toda a jornada aqui vivida, que me trouxe até este ponto. Igualmente, agradeço ao meu orientador pelo apoio nessa jornada, pelo conhecimento compartilhado e por sempre conseguir um espaço em sua agenda lotada para me ajudar.

Por fim, minha gratidão também se estende a todos que, mesmo que não citados nominalmente, me apoiaram ao longo deste processo.

Resumo

Ataques de negação de serviço têm se tornado cada vez mais sofisticados e especializados com o passar do tempo, com uso de métodos mais elaborados e potentes para causar indisponibilidade até em sistemas protegidos por medidas de segurança. Para possibilitar o estudo da resiliência em servidores DNS, por meio de testes precisos, foi desenvolvida a Eris, uma ferramenta capaz de recriar um ataque DNS Water Torture configurável, simplificando os testes de resiliência em sistemas desse gênero. A Eris se propõe a ser de fácil utilização e multiplataforma. Sua arquitetura modular é bem estruturada e bem documentada. O desempenho da ferramenta foi avaliado em testes que simularam usuários legítimos, medindo o tempo até que os servidores DNS sob ataque deixassem de responder. Sua arquitetura avançada permite não apenas maior eficiência na indução de indisponibilidade, mas também efeitos em cascata mais pronunciados em servidores dependentes. Enquanto as soluções com o mesmo propósito DNSWaterTorture e dns-flood-ng encontradas se limitam a parâmetros básicos como intervalos fixos entre pacotes e, quando disponível, spoofing de um único IP, a Eris introduz um paradigma de personalização granular através de: controle dinâmico da vazão de pacotes por iteração via sistema de níveis (1-10) ou via arquivo de taxa customizável, modo incremental que adapta progressivamente a intensidade do ataque, capacidade de simulação de múltiplas fontes através de multithreading utilizando uma lista de IPs spoofados, e modo RAID para maximização automática de consultas/segundo. Esta combinação única de features táticas aliada a resultados empíricos comprovam sua singularidade e superioridade quando comparada a ferramentas com o mesmo objetivo, executadas sob o mesmo hardware. Desenvolvimentos futuros podem aprimorar a Eris, aumentar sua eficiência e incorporar novos tipos de ataques focados em DNS, além de expandir as opções de configuração.

Palavras-chave: Negação de serviço, Water Torture, Eris, DNS

Abstract

Denial-of-service attacks have become increasingly sophisticated and specialized over time, employing more elaborate and powerful methods to cause service unavailability, even in systems protected by security measures. To facilitate the study of resilience in DNS servers through precise testing, Eris was developed, a tool capable of recreating a configurable DNS Water Torture attack, simplifying resilience testing for systems of this nature. Eris is designed to be user-friendly and cross-platform, with a well-structured and well-documented modular architecture.

The tool's performance was evaluated in tests that simulated legitimate users, measuring the time until the attacked DNS servers stopped responding. Its advanced architecture allows not only for greater efficiency in inducing unavailability but also for more pronounced cascading effects on dependent servers. While existing solutions such as DNSWaterTorture and dns-flood-ng are limited to basic parameters such as fixed intervals between packets and, when available, spoofing a single IP address, Eris introduces a new paradigm of granular customization through dynamic packet flow control per iteration via a level-based system (1-10) or a customizable rate file, an incremental mode that progressively adapts attack intensity, the ability to simulate multiple sources through multithreading using a list of spoofed IPs, and a RAID mode for automatic query-per-second maximization. This unique combination of tactical features, combined with empirical results, demonstrates its singularity and superiority when compared to other tools with the same objective, executed on the same hardware. Future developments may enhance Eris, increasing its efficiency and incorporating new DNS-focused attack techniques, as well as expanding configuration options.

Keywords: Denial of service, Water torture, Eris, DNS

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

AR DDoS Amplified Reflection DDoS.

DDoS Distributed Denial of Service.

DNS Domain Name System.

DoS Denial of service.

PPS Packets per second.

TTL time to live.

Capítulo 1

Introdução

Com o crescente volume de dados e a expansão de serviços baseados na Internet, a segurança da informação vem se tornando um tópico cada vez mais relevante. Essa segurança é sustentada por quatro pilares principais: Confidencialidade, Disponibilidade, Integridade e Autenticidade, cada um desses pilares desempenha um papel essencial para garantir que informações e sistemas sejam acessados e manipulados de maneira segura e confiável.

Entre esses pilares, a Disponibilidade se destaca como um dos mais desafiadores de se proteger, devido à constante ameaça de ataques de negação de serviço Denial of service (DoS) e ataques distribuídos de negação de serviço Distributed Denial of Service (DDoS). Tais ataques têm como objetivo principal interromper o funcionamento de sistemas, aplicativos, ou serviços, ao sobrecarregar servidores, bem como recursos de rede com um volume massivo de tráfego malicioso.

Nos últimos anos, os ataques DDoS têm se tornado mais frequentes e sofisticados, acompanhando o avanço das tecnologias e o aumento da dependência de serviços digitais. No primeiro semestre de 2024, o número de ataques registrados mais que dobrou em comparação com o segundo semestre de 2023 [3]. Além disso, esses ataques vêm atingindo níveis inéditos de volumetria. Em 2024, foi registrado o maior ataque DDoS da história, alcançando uma impressionante taxa de 5,6 terabits por segundo (Tbps) [4], o que reforça a urgência de soluções robustas e inovadoras para mitigar esses ataques.

Entre as diferentes variações de ataques DDoS, o método conhecido como DNS Water Torture se destaca graças ao seu foco específico e sua capacidade de causar indisponibilidade em larga escala. Ataque esse que tem como foco explorar vulnerabilidades em servidores DNS, inundando-os com consultas para subdomínios aleatórios de domínios existentes. Isso não só sobrecarrega os servidores DNS recursivos e autoritativos, mas também impacta a experiência de inúmeros usuários finais, ao impedir que domínios legítimos sejam resolvidos [5]. Por se tratar de uma variação de DDoS, o DNS Water Torture reproduz todas as características fundamentais desse tipo de ofensiva distribuída,

concentrando-se especificamente na resolução de nomes.

Devido às suas peculiaridades, ferramentas que simulam esse tipo de ataque não são facilmente encontradas para avaliar soluções de mitigação. Durante o processo de desenvolvimento da Eris, foram encontrados apenas dois programas não oficiais que simulam o ataque [6] [7], nenhum dos dois possuindo elementos essenciais para um teste de resiliência detalhado, como por exemplo o controle de injeção de pacotes configuráveis, uso intuitivo e acompanhamento do andamento do ataque, além de apresentarem um volume de tráfego gerado inferior à Eris, quando testadas no mesmo ambiente. A ferramenta proposta neste trabalho, denominada Eris, replica o comportamento do DNS Water Torture, possibilitando que pesquisadores reproduzam a dinâmica do ataque em ambientes controlados e avaliem contramedidas sob diferentes graus de intensidade. Entre suas principais contribuições destacam-se o controle programável da taxa de geração de consultas, a apresentação de métricas em tempo real e a disponibilização do código-fonte, fomentando a evolução colaborativa da solução. Não obstante, a Eris apresenta limitações inerentes à sua arquitetura, notadamente a previsibilidade das consultas DNS produzidas, uma vez que os subdomínios são gerados por um algoritmo pseudo-aleatório determinístico, o que pode facilitar a criação de assinaturas específicas de detecção.

1.1 Objetivos

Dessa forma, o objetivo principal desse trabalho é o desenvolvimento da ferramenta Eris. A ferramenta tem como requisito de projeto reproduzir o comportamento de ataques do tipo DNS Water Torture, porém, de maneira controlada e altamente configurável. Por meio de uma plataforma unificada, a ferramenta vai possibilitar a execução de testes de resiliência em servidores DNS, reproduzindo cenários reais nos quais uma grande quantidade de queries — muitas vezes com subdomínios aleatórios — congestiona os serviços de resolução de nomes. Além disso, é um objetivo fornecer tráfego realista e parametrizável, para auxiliar na avaliação de vulnerabilidades e no desenvolvimento de contra-medidas específicas para esse tipo de ameaça. Dessa forma, infraestruturas críticas — como grandes provedores de internet e serviços essenciais — podem se antecipar a potenciais incidentes, aprimorando a postura de defesa e construindo estratégias de mitigação mais eficazes.

Adicionalmente, a capacidade da Eris em gerar tráfego DNS realista também facilita pesquisas relacionadas ao Domain Generation Algorithm (DGA). Esses algoritmos são frequentemente empregados em malwares com a finalidade de criar domínios de forma dinâmica, dificultando o bloqueio ou a detecção por mecanismos de segurança convencionais. Ao alterar continuamente o nome de domínio de controle, o malware obtém um

canal de comunicação resiliente a blacklisting, pois, quando um domínio é bloqueado, outro é gerado previamente pelo algoritmo e pode ser utilizado.

Logo, por ser capaz de gerar consultas DNS com subdomínios pseudoaleatórios, a ferramenta se torna útil para simular condições semelhantes às que surgem em ataques que utilizam DGA, permitindo avaliar como soluções de detecção e prevenção reagem quando confrontadas com um padrão dinâmico de domínios, além de contribuir com a geração de datasets para treinamento de algoritmos. Esse tipo de teste, feito em um ambiente controlado, é essencial para entender os limites e eficiência das estratégias de segurança adotadas, bem como para desenvolver novos métodos de identificação de algoritmos maliciosos baseados em DGA.

1.2 Organização do documento

Este documento descreve o processo de desenvolvimento da Eris e os testes realizados para atingir os objetivos definidos. O Capítulo 2 aborda os conceitos fundamentais que contextualizam a importância da ferramenta, incluindo uma análise detalhada do ataque *DNS Water Torture* e de suas implicações. No Capítulo 3, é apresentada a ferramenta, onde são descritas as decisões arquiteturais acerca da implementação da ferramenta, suas funcionalidades, seu fluxo de execução e a usabilidade. O Capítulo 4 detalha a metodologia de avaliação da ferramenta e os procedimento seguidos para tal. Em sequência, o Capítulo 5 detalha e apresenta os resultados obtidos nos testes e as conclusões acerca deles. Por fim, o Capítulo 6 apresenta as conclusões obtidas no processo aqui citado.

Capítulo 2

Conceitos Básicos

Neste capítulo, são apresentados os principais conceitos relacionados ao *DNS Water Torture*, incluindo o funcionamento do protocolo Domain Name System (DNS) em si, a modalidade dos ataques DoS e DDoS, e como essa técnica explora a infraestrutura do sistema de nomes de domínio para sobrecarregar servidores e interromper serviços.

2.1 Ataques DoS/DDoS

Um ataque de negação de serviço, ou DoS, tem como objetivo comprometer o pilar da disponibilidade na segurança da informação, tornando deliberadamente um sistema ou infraestrutura inacessível para usuários legítimos, por meio do esgotamento de recursos [8]. Esse objetivo pode ser alcançado, principalmente, por meio de duas estratégias, ataques volumétricos ou abuso de protocolos de rede vulneráveis.

No primeiro caso, o intuito do ataque é enviar um volume de tráfego excessivo de forma que seja superior à capacidade de processamento do alvo, essa estratégia pode ser potencializada usando a distribuição (do inglês, Distributed denial of service (DDoS)), em que vários nós enviam tráfego de forma coordenada, resultando em maior eficiência do ataque e mitigação mais complexa devido à multiplicidade de origens. Adicionalmente, os ataques volumétricos podem ser divididos em ataques diretos e ataques de reflexão. Em ataques diretos o tráfego é enviado diretamente à vítima, por um único atacante ou por uma rede distribuída. Já os ataques com reflexão utilizam de nós intermediários, conhecidos como refletores, para inundar a vítima. Um refletor é qualquer dispositivo que responde ao tráfego recebido, podendo ou não amplificá-lo. Ataques que empregam refletores para aumentar sua volumetria são chamados de DDoS por reflexão amplificada (do inglês, Amplified Reflection DDoS ou Amplified Reflection DDoS (AR DDoS)), multiplicando significativamente o impacto do ataque. Uma técnica comum a ambas essas as categorias de DoS é a falsificação de IP (spoofing), usada para ocultar a origem real do

ataque. Em AR-DDoS, por exemplo, os atacantes enviam solicitações aos refletores utilizando o endereço IP da vítima como origem, fazendo com que as respostas amplificadas sejam direcionadas a ela [9].

No segundo caso, a abordagem se difere ao enviar um volume de tráfego menor a uma taxa mais lenta que a técnica anterior, visando explorar características específicas dos protocolos de rede, detalhes de implementação ou funcionalidades, levando ao esgotamento gradual dos recursos da vítima. Essa categoria de ataque não depende de grandes quantidades de banda para saturar o alvo. Em vez disso, utiliza requisições cuidadosamente arquitetadas para manter o sistema ocupado, bloqueando ou retardando o atendimento de usuários legítimos [10]. Dessa forma, mesmo com uma quantidade aparentemente pequena de tráfego, a vítima gradualmente fica indisponível por falta de memória, threads ou conexões livres.

No decorrer deste trabalho, o foco foi na modalidade de ataques DoS diretos, de modo que a ferramenta desenvolvida explora o protocolo de rede DNS e suas potenciais vulnerabilidades, enquanto ainda sendo capaz de gerar uma volumetria alta de requisições por segundo.

2.2 DNS - Domain Name System

O Sistema de Nomes de Domínio (DNS) é uma infraestrutura essencial para o funcionamento da Internet, responsável por traduzir nomes de domínio legíveis, como www.sales.com, em endereços IP, como 192.168.0.1, que são utilizados por dispositivos para estabelecer comunicação em redes [11] [12]. Essa funcionalidade permite que usuários acessem serviços online sem a necessidade de memorizar endereços IP. O DNS é baseado em uma arquitetura hierárquica e distribuída, composta por diferentes tipos de servidores que desempenham papéis fundamentais no processo de resolução de nomes.

No topo da hierarquia estão os servidores raiz (do inglês, root), responsáveis por direcionar consultas aos servidores de domínio de nível superior (do inglês, Top level domain - TLD), que, por sua vez, indicam os servidores autoritativos, os quais armazenam os registros definitivos de cada domínio. Responsáveis por intermediar esse processo, os servidores recursivos recebem a consulta do cliente e realizam a busca completa da resposta, consultando os servidores necessários e armazenando as informações temporariamente em cache para otimizar consultas futuras.[13].

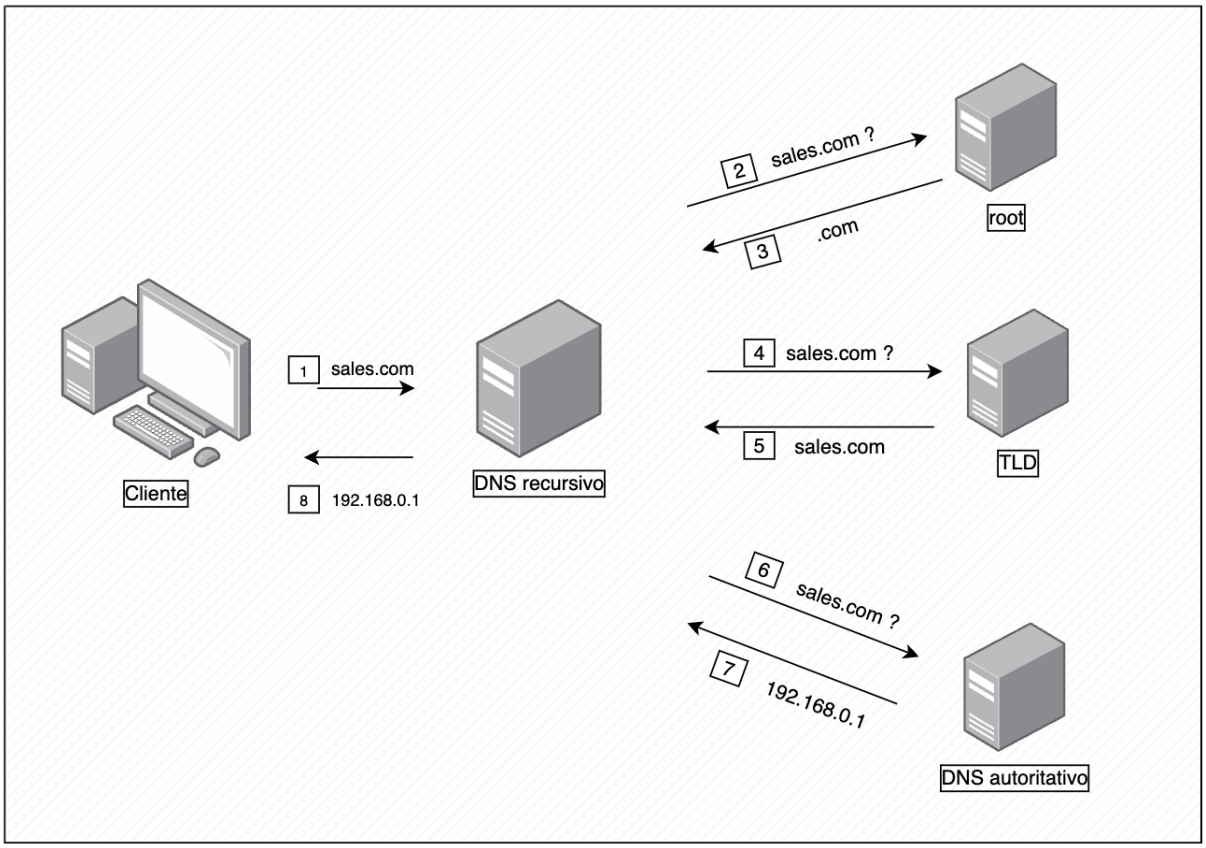

Figura 2.1: Fluxo de resolução DNS [1])

Ou seja, cabe aos servidores recursivos atuarem como intermediários entre os clientes que realizam consultas, como navegadores ou dispositivos, e os servidores autoritativos, que possuem os registros definitivos do domínio buscado. Quando um cliente solicita a resolução de um domínio, o servidor recursivo verifica inicialmente seu cache local em busca de uma resposta armazenada. Caso a informação não esteja no cache, ele inicia uma busca iterativa, enviando a consulta a outros servidores, começando pelos servidores raiz, passando pelos servidores responsáveis pelo domínio de nível superior (TLD), e finalmente alcançando os servidores autoritativos que gerenciam os registros do domínio requisitado, assim como representado na imagem 2.1.

Durante esse processo, o servidor recursivo armazena temporariamente as respostas em seu cache, seguindo o time to live (TTL) definido nos registros DNS. Isso reduz a latência em consultas futuras e diminui a carga sobre servidores autoritativos [1]. Por exemplo, se outro cliente solicitar o mesmo domínio dentro do período de TTL, o resolver retornará a resposta diretamente do cache, sem consultar novamente a hierarquia DNS.

Já uma consulta autoritativa é respondida diretamente por um servidor que detém as informações definitivas para um domínio. Servidores autoritativos são configurados com

zonas DNS, que são segmentos da hierarquia do DNS administrados por um servidor e contêm registros que definem a estrutura e os endereços associados a um domínio [14]. Dentro dessas zonas, encontram-se registros essenciais como A, que associa um nome de domínio a um endereço IPv4; AAAA, que faz o mesmo para IPv6; MX, que define os servidores responsáveis pelo recebimento de e-mails do domínio; e NS, que especifica quais servidores são autoritativos para aquele domínio. Quando um servidor autoritativo recebe uma consulta, ele responde com base em seus próprios dados, sem realizar buscas adicionais. A imagem 2.2 apresenta uma representação simplificada de como funciona a estrutura de zonas [13] para o domínio sales.com, por exemplo.

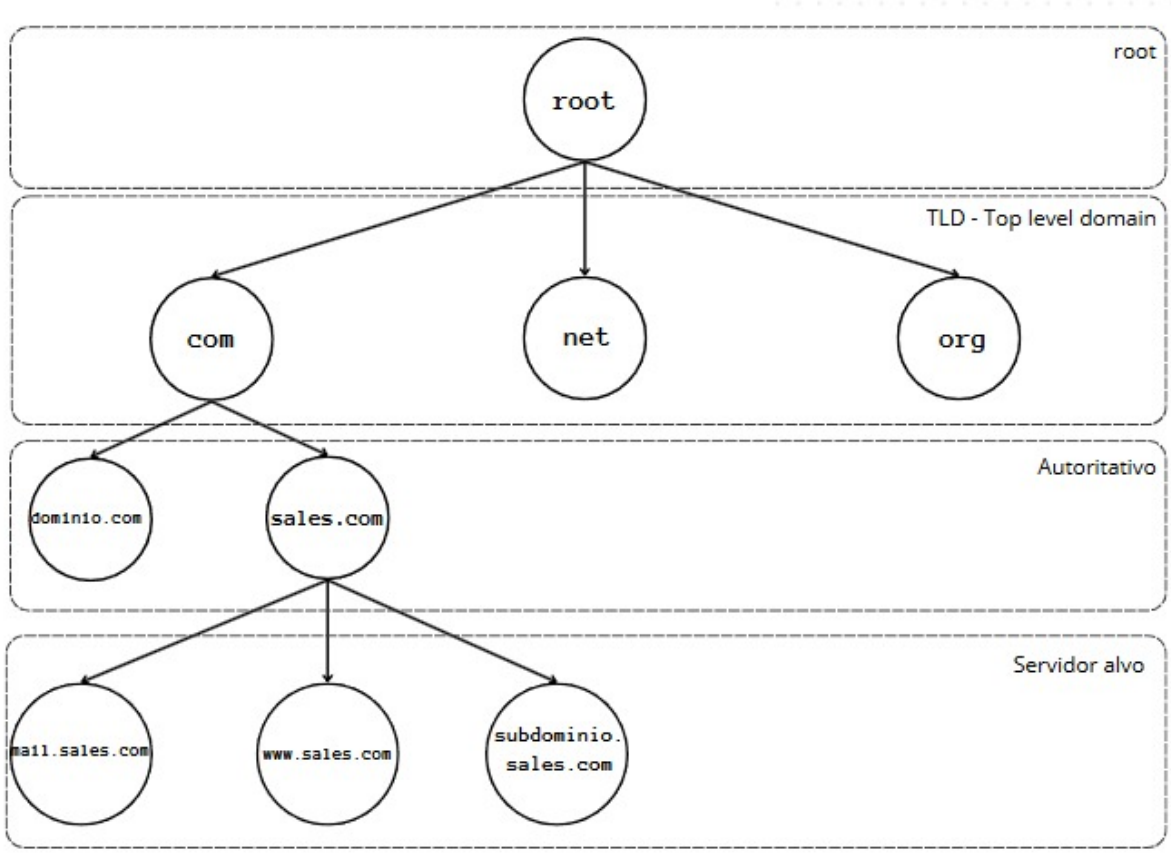

Figura 2.2: Zonas DNS

Ou seja, o DNS é um sistema essencial que, por meio de sua estrutura hierárquica de servidores, garante a resolução de domínio de forma que torna possível a navegação na Internet sem a necessidade de memorizar e acessar endereços de IP diretamente.

2.3 Water Torture

A técnica de ataque DDoS conhecida como *DNS Water Torture* foi popularizada pelo malware Mirai e suas variantes [5] [15]. Tal técnica consiste no envio de um grande volume de solicitações contendo subdomínios pseudoaleatórios a servidores DNS, com o objetivo de exaurir seus recursos computacionais e de rede. O ataque explora de maneira particular o funcionamento do sistema DNS, que gerencia a resolução de nomes de domínio de forma hierárquica. Nesse contexto, os servidores recursivos são forçados não só a verificar cada consulta recebida pelo atacante em seu cache local, como também são obrigados a encaminhar as requisições aos servidores autoritativos correspondentes ao domínio base do ataque. Dessa forma, os servidores recursivos e autoritativos, responsáveis por encaminhar e validar a existência dos subdomínios, sofrem uma sobrecarga significativa, pois precisam processar cada requisição como se fosse legítima, o que aumenta exponencialmente o consumo de recursos computacionais e de rede.

O ataque se baseia no envio de queries DNS com subdomínios aleatórios concatenados ao domínio alvo. Essas queries seguem o formato padrão do protocolo DNS, estruturadas em um cabeçalho e seções subsequentes. O cabeçalho contém campos essenciais, como o identificador único da query (Transaction ID), flags de operação e resposta, contador de questões e registros. Em um ataque DNS Water Torture, as flags são manipuladas de forma estratégica para intensificar o impacto. Por exemplo, a flag `RD` (Recursion Desired) é ativada para forçar os servidores recursivos a buscar a resolução completa da consulta. Como os subdomínios não existem, os servidores não conseguem fornecer uma resposta armazenada e são obrigados a encaminhar a requisição ao servidor autoritativo do domínio base. Esse comportamento gera um efeito cascata que sobrecarrega os servidores ao longo da hierarquia DNS. Em ataques distribuídos, assim como o realizado pelo Mirai, a técnica torna-se exponencialmente mais devastadora.

A estrutura das consultas DNS geradas pela Eris podem ser observados na imagem 2.3

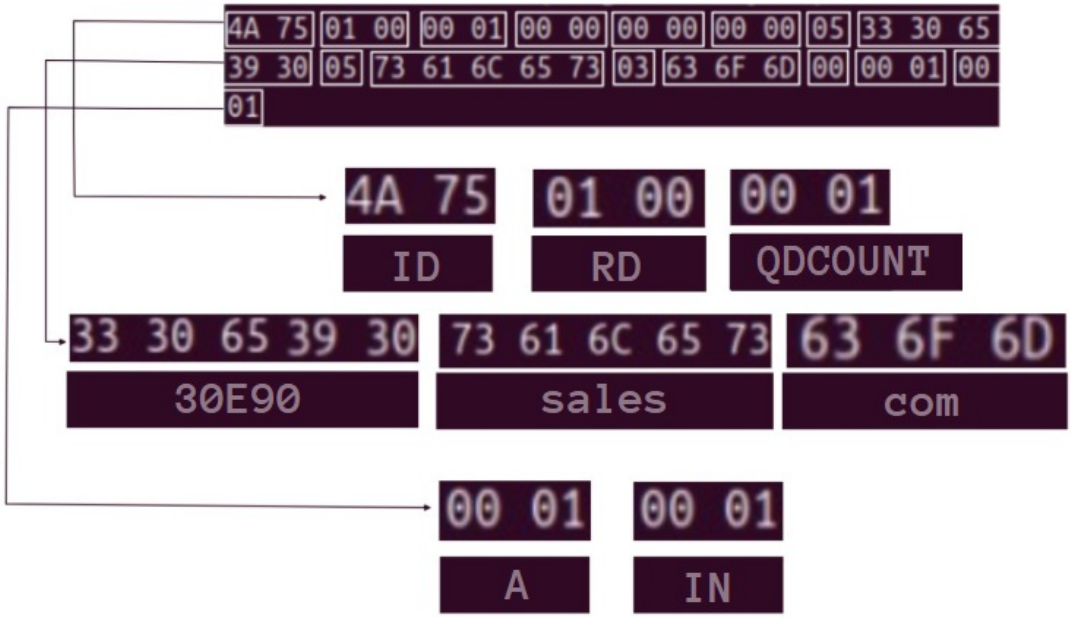

Figura 2.3: Bytes de uma consulta gerada pela Eris

- **ID**: Identificador da transação DNS, usado para correlacionar as respostas às consultas.
- **Flags**: RD (Recursion Desired): indica que o cliente deseja que o servidor faça uma consulta recursiva caso não saiba a resposta.
- **QDCOUNT**: Número de consultas contidas na mensagem DNS. No caso, apenas 1 pergunta.
- **QNAME (`30E90.sales.com`)**: Nome de domínio que está sendo consultado. Nesta representação, cada parte (`30E90`, `sales`, `com`) aparece precedida de um byte que indica o comprimento do rótulo.
- **QTYPE (A)**: Tipo de registro solicitado. `A` significa que se deseja o endereço IPv4 do nome de domínio.
- **QCLASS (IN)**: Classe do registro. `IN` indica um registro no contexto de Internet.

A Figura 2.4 apresenta um exemplo de query realizada por meio do Mirai enquanto a Figura 2.5 apresenta um exemplo de query realizada por meio da Eris. Pode-se notar que ambas seguem a mesma lógica explicada acima.

```
DNS Query Flood(Aka Mirai DNS Water Torture Attack) - Target Domain Names Removed
11:48:43.171738 IP x.x.x.x.47645 > x.x.x.x.53: 59218 [1au] A?
02uqhuovfi1f.<redacted>.com. (xx)

11:48:43.171749 IP x.x.x.x.47371 > x.x.x.x.53: 62949 [1au] A?
qo5etoh5foab.<redacted>.com. (xx)
```

Figura 2.4: Ilustração das consultas realizadas pelo Mirai (Fonte: [2])

```
02-Feb-2025 04:08:06.982 client @0x796f51471e08 42.145.79.186#53 (371b5.sales.com): query: 371b5.sales.com IN A + (192.168.0.4)
02-Feb-2025 04:08:06.982 client @0x796f514776a8 42.145.79.186#53 (d2c80.sales.com): query: d2c80.sales.com IN A + (192.168.0.4)
02-Feb-2025 04:08:06.982 client @0x796f521d0248 42.145.79.186#53 (a6434.sales.com): query: a6434.sales.com IN A + (192.168.0.4)
02-Feb-2025 04:08:06.982 client @0x796f521d5da8 42.145.79.186#53 (12bf6.sales.com): query: 12bf6.sales.com IN A + (192.168.0.4)
02-Feb-2025 04:08:06.982 client @0x796f51d6ffb8 42.145.79.186#53 (a968d.sales.com): query: a968d.sales.com IN A + (192.168.0.4)
02-Feb-2025 04:08:06.982 client @0x796f51d75c68 42.145.79.186#53 (53ed1.sales.com): query: 53ed1.sales.com IN A + (192.168.0.4)
02-Feb-2025 04:08:06.982 client @0x796f51d7b7c8 42.145.79.186#53 (e41fa.sales.com): query: e41fa.sales.com IN A + (192.168.0.4)
02-Feb-2025 04:08:06.982 client @0x796f51d814c8 42.145.79.186#53 (ae442.sales.com): query: ae442.sales.com IN A + (192.168.0.4)
02-Feb-2025 04:08:06.987 client @0x796f51d871c8 42.145.79.186#53 (89ea8.sales.com): query: 89ea8.sales.com IN A + (192.168.0.4)
02-Feb-2025 04:08:06.987 client @0x796f5262cba8 42.145.79.186#53 (17a37.sales.com): query: 17a37.sales.com IN A + (192.168.0.4)
02-Feb-2025 04:08:06.987 client @0x796f526328a8 42.145.79.186#53 (d6265.sales.com): query: d6265.sales.com IN A + (192.168.0.4)
02-Feb-2025 04:08:06.987 client @0x796f51c57118 42.145.79.186#53 (ea79b.sales.com): query: ea79b.sales.com IN A + (192.168.0.4)
02-Feb-2025 04:08:06.987 client @0x796f513c20c8 42.145.79.186#53 (b9f43.sales.com): query: b9f43.sales.com IN A + (192.168.0.4)
02-Feb-2025 04:08:06.987 client @0x796f513bc568 42.145.79.186#53 (663a2.sales.com): query: 663a2.sales.com IN A + (192.168.0.4)
```

Figura 2.5: Ilustração das consultas realizadas pela Eris

Como consequência, a indisponibilidade de um servidor DNS pode resultar na inacessibilidade de inúmeras aplicações e serviços, uma vez que seus registros DNS, essenciais para traduzir nomes de domínio em endereços IP, tornam-se inalcançáveis. Ou seja, fazendo com que o usuário final apenas consiga acessar uma aplicação web caso conheça o IP de seu servidor. Esse impacto não se limita a websites, mas também afeta sistemas corporativos e infraestruturas essenciais, amplificando de maneira significativa o alcance e a gravidade do ataque.

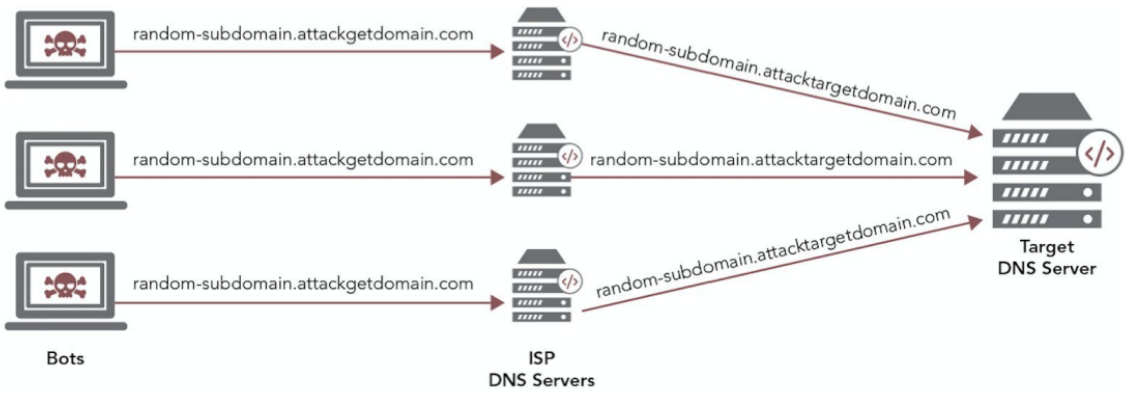

Figura 2.6: Ilustração da técnica de DNS Water Torture (Fonte: [2])

2.4 Mirai

O malware *Mirai* ganhou notoriedade ao explorar vulnerabilidades presentes em dispositivos de Internet das Coisas (IoT), convertendo esses dispositivos em bots para a formação de uma vasta botnet. Originalmente detectado em 2016, o Mirai demonstrou como dispositivos com configurações de segurança inadequadas podem ser alvos fáceis para ataques em larga escala [5]. Essa botnet, composta principalmente por câmeras de segurança, roteadores e outros dispositivos de baixa capacidade, foi utilizada para lançar ataques DDoS massivos, que sobrecarregaram alvos importantes na internet.

Além de sua capacidade de recrutar dispositivos vulneráveis, o Mirai também popularizou a utilização de técnicas sofisticadas, entre elas a estratégia de ataques baseada no envio de queries DNS com subdomínios pseudoaleatórios. Algumas variantes do Mirai incorporaram o que posteriormente ficou conhecido como *DNS Water Torture*, onde o malware realiza consultas para subdomínios inexistentes, forçando servidores DNS recursivos e autoritativos a processarem cada requisição de forma completa, exacerbando o consumo de recursos e dificultando a mitigação do ataque.

O Mirai ilustra a evolução dos ataques DDoS, mostrando que a eficácia não depende somente do volume de tráfego, mas também da exploração de fragilidades específicas na infraestrutura da rede. Sua abordagem de combinar dispositivos IoT comprometidos com técnicas de amplificação e evasão exemplifica os desafios enfrentados por administradores de rede e pesquisadores de segurança atualmente, evidenciando a necessidade de soluções robustas e atualizadas para proteção contra ataques distribuídos.

2.5 Considerações finais

Neste capítulo foram apresentados os principais conceitos envolvidos no desenvolvimento dessa ferramenta, como ataques DoS/DDoS, protocolo DNS, o ataque específico de DNS Water Torture e o contexto de sua utilização passada, no Mirai.

As discussões aqui apresentadas são fundamentais para embasar a análise dos desafios que motivaram o desenvolvimento da ferramenta Eris, que será abordada nos próximos capítulos. Neste sentido, o entendimento dos mecanismos que governam o funcionamento do DNS, a dinâmica dos ataques DoS/DDoS e a utilização de técnicas como o DNS Water Torture e as variações do Mirai, colaboram para um panorama dos riscos associados à infraestrutura da Internet. No próximo capítulo, a ferramenta será apresentada em detalhes, expondo sua arquitetura, funcionamento, fluxo de execução, utilização e disponibilização, contribuindo para a formulação de estratégias eficazes de mitigação e aprimoramento da resiliência dos sistemas.

Capítulo 3

Eris

A Eris é uma ferramenta desenvolvida para a simulação e análise de ataques volumétricos de negação de serviço (DoS), especificamente do tipo DNS Water Torture. Seu nome, inspirado na deusa grega da discórdia [16], reflete a capacidade desse tipo de ataque, e consequentemente da ferramenta, de desestabilizar redes e serviços essenciais em escala amplificada. Projetada para avaliar a resiliência de servidores DNS e testar suas proteções, a ferramenta permite a simulação controlada e altamente configurável de ataques, possibilitando a medição precisa do momento em que o servidor sob análise se torna incapaz de responder adequadamente às requisições maliciosas.

A ferramenta foi desenvolvida utilizando a engine de formação de pacotes e controle de vazão das ferramentas de ataques DoS DamBuster [17] e Linderhof [18] . Entretanto, seu funcionamento foi adaptado para incorporar o comportamento característico do Mirai [5], [15], resultando em uma ferramenta essencialmente nova, capaz de realizar ataques *DNS Water Torture* de forma independente de dispositivos secundários. Tal flexibilidade possibilita sua aplicação em diversos cenários de teste, contribuindo significativamente para o aprimoramento de estratégias de mitigação e proteção contra esse tipo de ameaça.

Nas seções seguintes, serão apresentadas a arquitetura da ferramenta e suas principais funcionalidades.

3.1 Arquitetura

Apesar de aproveitar a base funcional da DamBuster, a Eris integra um sistema específico para a geração de consultas DNS válidas [19]. Cada pacote gerado corresponde a uma consulta DNS única, direcionada ao servidor especificado pelo usuário, com a adição de um subdomínio pseudoaleatório anexado ao domínio configurado no ataque. Esse mecanismo garante a sobrecarga progressiva nos servidores DNS recursivos e autoritativos ao forçá-los a processar resoluções repetidas para domínios inexistentes.

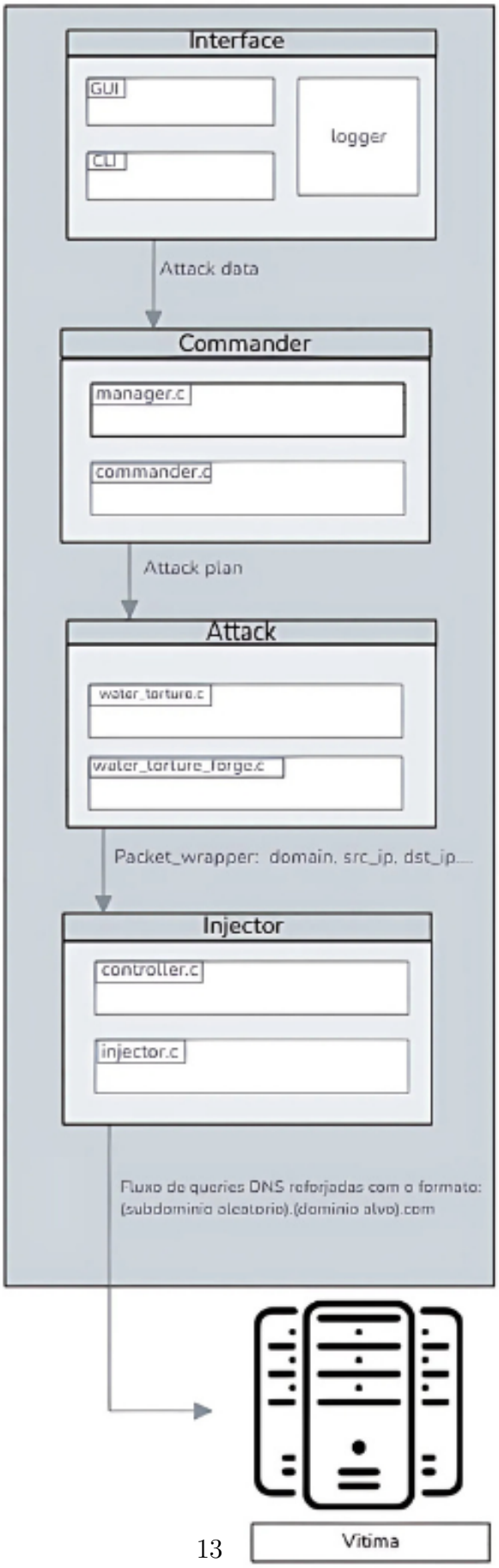

Figura 3.1: Diagrama de arquitetura do software.

A Eris mantém, inicialmente, a estrutura modular da DamBuster, composta por quatro módulos principais, mas realiza alterações significativas em seu funcionamento para se adequar ao novo modelo de ataque. Os detalhes sobre cada módulo são apresentados a seguir:

3.1.1 Interface

Módulo em que as interfaces gráfica (GUI) e de linha de comando (CLI) são implementadas. Esse módulo é responsável por receber os argumentos que serão utilizados no ataque.

3.1.2 Commander

- commander.c: Configura o ambiente inicial (por exemplo, definindo ações para erros fatais), inicia o fluxo principal do programa, chamando a lógica de ataque, com base nos parâmetros recebidos pela interface. Também é responsável por criar e sincronizar as *threads* de ataque, garantindo o encerramento ordenado quando todas finalizam. A ideia é que o módulo se assemelhe a um servidor de comando e controle;
- manager.c: Atua como intermediário entre o commander e o módulo Attack.

3.1.3 Attack

Módulo que contém a lógica principal do ataque *DNS Water torture* e também é responsável pela criação dos pacotes que serão enviados durante o ataque. Sua estrutura é dividida em dois submódulos:

- dnsflood.c: Gerencia o fluxo de alto nível do ataque *DNS flood*, desde a criação do primeiro pacote até o início do processo de injeção.
- dnsfloodforge.c: Cuida dos detalhes de baixo nível da construção e preparação de pacotes DNS com subdomínios aleatórios, tornando cada pacote único para garantir a eficácia do ataque.

3.1.4 Injector

Nesta parte do código, o injector e o controller estão implementados:

- injector.c: Gerencia a mecânica detalhada da injeção de pacotes, desde a criação de pacotes DNS até o envio através de sockets de rede.
- controller.c: Coordena o processo geral de injeção, sincronizando os injectors, controlando o tempo e realizando ajustes dinâmicos durante o ataque.

3.2 Funcionamento

A Eris implementa um ataque de *DNS query flood* com opção de *spoofing* do endereço de origem, permitindo o controle do rate dos pacotes enviados e funcionamento *multithread*. A ferramenta necessita dos parâmetros de alvo e domínio e aceita parâmetros de configuração para o rate (duration, level, increment ou RAID mode), IP de origem para *spoofing*, porta (53 por padrão). Por padrão, a ferramenta envia um pacote por segundo indefinitivamente.

A cada pacote enviado, um subdomínio aleatório é gerado, utilizando uma string randômica hexadecimal concatenada com o domínio base, assim como mostrado na imagem 3.2.

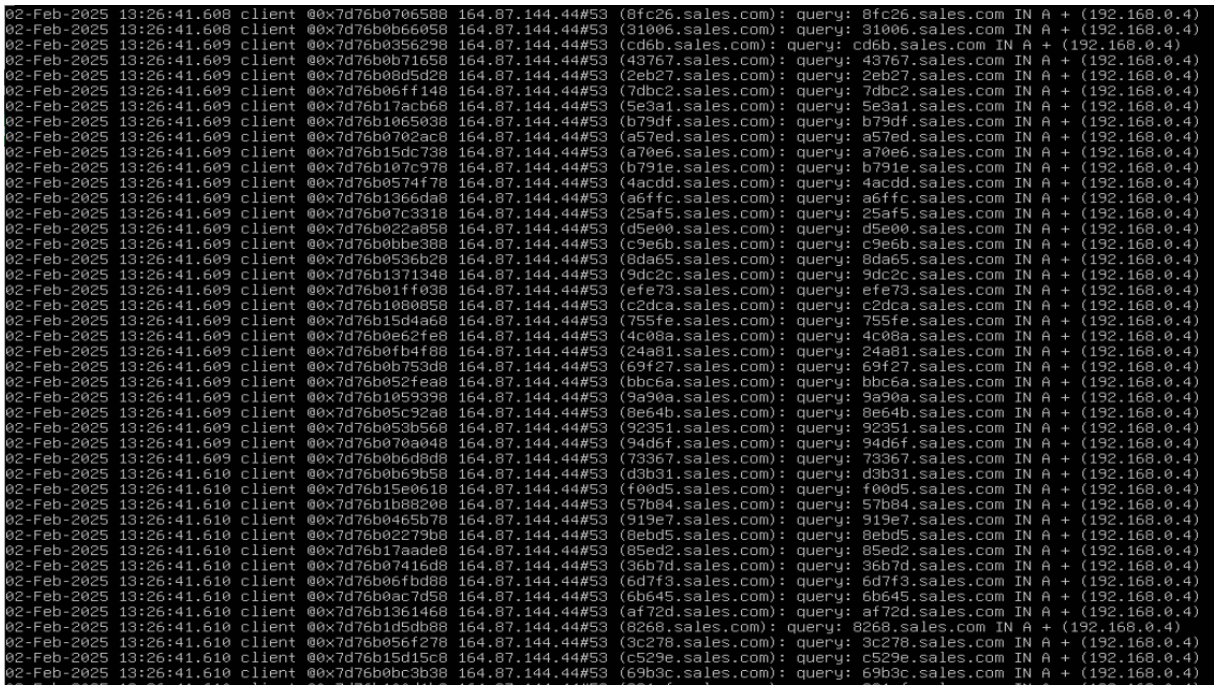

Figura 3.2: Logs do servidor DNS recursivo sob flood do domínio sales.com

Optou-se por gerar nomes de subdomínio hexadecimais para prover deliberadamente uma assinatura detectável para a assinatura gerada pela Eris. Dessa forma, qualquer uso indevido ou indesejável da ferramenta e facilmente identificado, e bloqueado com a aplicação da assinatura.

Figura 3.3: Campos presentes na query DNS.

A estrutura do log do servidor DNS recursivo apresentada nas imagens 3.2 e 3.3 possui os campos descritos abaixo:

(1) **client @0x796f513bc568**: Identificador interno do BIND (serviço hospedado no servidor DNS) que representa a estrutura que armazena detalhes do cliente no servidor DNS.

(2) **116.24.140.227#53**: O endereço IP e a porta (#53) do cliente que fez a consulta DNS.

(3) **(da6be.sales.com)**: O nome de domínio que foi solicitado.

(4) **da6be.sales.com IN A +**: Traduz-se para uma consulta DNS do tipo `A` (endereço IPv4), na classe `IN` (Internet) e com a flag RD ativa (+), para o domínio da6be.sales.com.

(5) **(192.168.0.4)**: O endereço IP onde a requisição foi direcionada.

O *spoofing*, o qual pode ser observado por exemplo no item 2 da imagem 3.3, como objetivo mascarar a identidade do host e tem como consequência, a potencial sobrecarga do host real dono do ip, à medida que ele receberá um grande volume de tráfego, contendo as respostas do servidor DNS destinadas ao atacante. A ferramenta por *default* randomiza um ip de origem, caso nenhum seja passado como argumento de execução e aceita um ip, ou uma lista de ips como entrada. Caso a ferramenta recebe uma lista como argumento, para cada ip recebido, uma thread de execução do ataque será criada com a intenção de simular um tráfego distribuído.

3.3 Estratégias de Uso da Ferramenta Eris para Teste de Resiliência de Servidores DNS

A Eris foi desenvolvida para simular ataques do tipo DNS Water Torture de maneira altamente configurável, permitindo que pesquisadores e administradores de rede avaliem a resiliência de servidores DNS sob diferentes condições de estresse. A seguir, são apresentadas três estratégias que podem ser adotadas, utilizando os recursos de configuração e as funcionalidades detalhadas no documento.

3.3.1 Teste Incremental de Carga

Nesta estratégia, o ataque é iniciado com uma taxa de envio baixa e, progressivamente, a intensidade do tráfego é aumentada. Utilizando os parâmetros de nível e incremento (conforme apresentado na Tabelas 4.2), o operador pode monitorar a degradação do desempenho do servidor DNS à medida que a taxa de consultas aumenta. Esse teste incremental é ideal para determinar o ponto de saturação do servidor e identificar limites operacionais, permitindo a análise detalhada da resposta do sistema a cargas crescentes de requisições.

3.3.2 Teste em Modo RAID para Máxima Vazão

Outra estratégia consiste em configurar a Eris no *modo RAID*, o qual ignora os parâmetros de incremento e outras configurações de taxa, fazendo com que a ferramenta envie o maior número de consultas por segundo possível. Essa abordagem é particularmente útil para avaliar a capacidade do servidor de suportar um ataque de intensidade máxima. Os testes realizados em ambiente bare metal, conforme descritos no Capítulo 4, demonstraram que essa estratégia permite observar rapidamente o ponto de colapso do servidor, bem como o efeito cascata em servidores vizinhos.

3.3.3 Simulação de Ataque Distribuído com Spoofing Múltiplo

Uma terceira estratégia explora a capacidade de injeção multithread da Eris, combinada com a funcionalidade de spoofing de múltiplos IPs. Ao fornecer uma lista de endereços IP de origem – ou seja, simulando várias fontes de ataque – a ferramenta cria diversas threads de execução, cada uma enviando queries com IPs de origem distintos. Essa simulação busca aproximar-se do comportamento real de uma botnet, em que os ataques são distribuídos em várias origens.

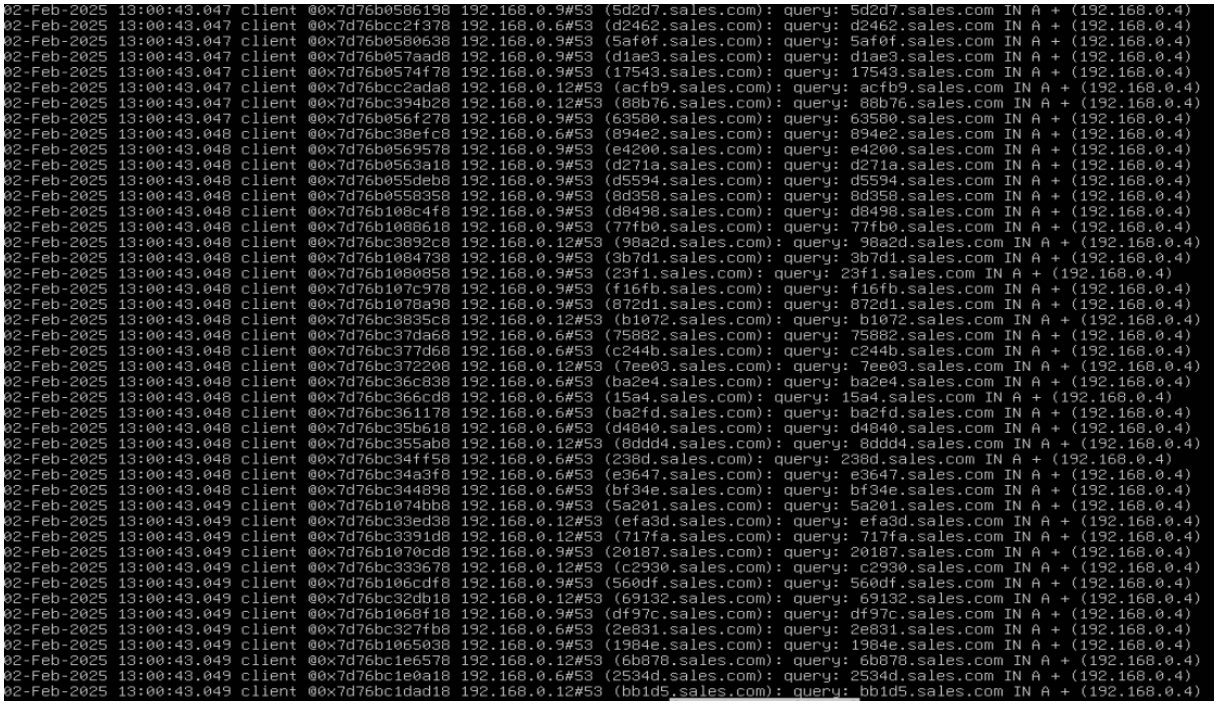

Figura 3.4: Logs do servidor DNS recursivo recebendo pacotes de 3 ips distintos

O comportamento *multithread* segue o que está ilustrado no diagrama da imagem 3.5, onde uma lista

$$\text{n IPs} \quad [IP_n, IP_{n+1}, IP_{n+2}]$$

leva à criação de n *threads* de execução do ataque. O comportamento pode ser observado na prática nos logs do servidor na imagem 3.4.

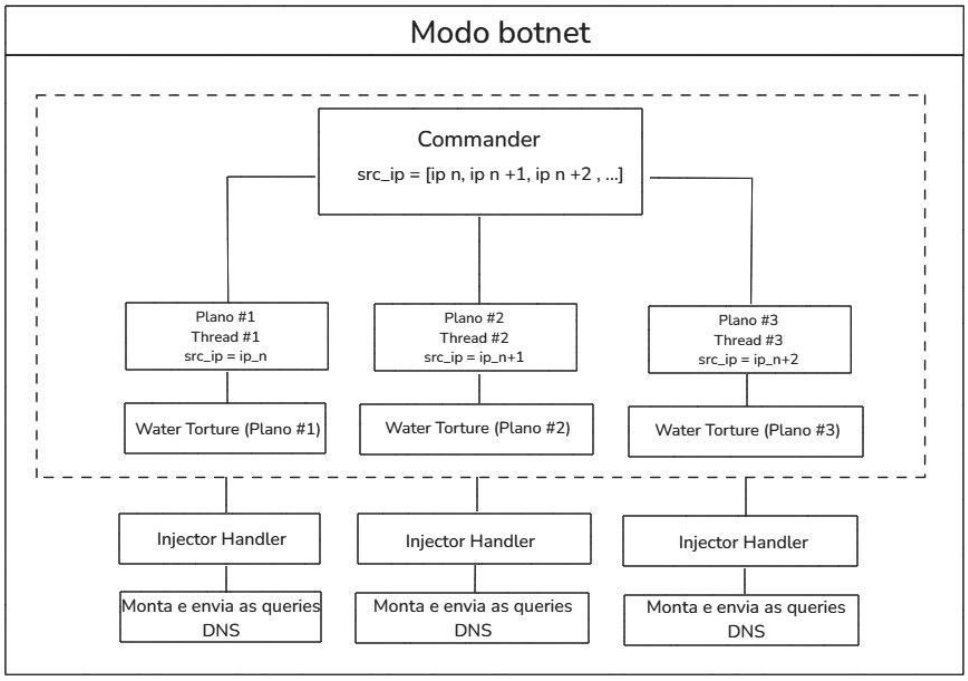

Figura 3.5: Funcionamento Multithread

3.4 Parâmetros

Para seu funcionamento, a ferramenta exige os seguintes parâmetros:

- **Endereço de Destino**: endereço IPv4 do alvo.
- **Domínio**: domínio base para qual os subdomínios serão gerados.

Além dos parâmetros obrigatórios, a ferramenta aceita os seguintes parâmetros de configuração:

- **Endereço de Origem**: Argumento não obrigatório responsável por definir o endereço de IP que sofrerá spoofing, aceita um endereço, ou uma lista passada por meio de arquivo.
- **Porta de Origem**: A porta de origem por padrão é 53, mas pode ser alterada por meio desse parâmetro.
- **Porta de Destino**: A porta de destino por padrão é 53, mas pode ser alterada por meio desse parâmetro.
- **Duração do ataque**: Define a duração total do ataque em milissegundos, caso nenhum valor seja passado, o ataque será executado até que seja interrompido via CTRL+C.

- **Nível de Ataque**: O nível do ataque varia entre 1 e 10 e permite definir quantas consultas serão feitas a cada iteração, de acordo com a fórmula $10^{(\text{level}-1)}$.
- **Modo Incremental**: Permite definir a frequência em que o nível do ataque aumenta, de modo que suba de nível a cada i iterações.
- **Modo RAID**: A flag de modo RAID ignora os outros parâmetros relacionados à taxa de envio de pacotes e faz com que a ferramenta envie a quantidade máxima de consultas por segundo.
- **Arquivo de taxa de Ataque** : Permite o uso de um arquivo para personalizar as taxas de ataque a serem seguidas para cada nível.
- **Intervalo Base de Ataque**: O intervalo base do ataque para cada iteração é 1 segundo, mas pode ser alterado por meio desse parâmetro.

3.5 Execução

A execução da ferramenta é dividida em três etapas principais: configuração, construção de pacotes e injeção de pacotes.

3.5.1 Configuração

Na etapa de configuração, os parâmetros fornecidos pelo usuário são processados para definir o plano de ataque. A interação do usuário pode ser feita via linha de comando ou interface gráfica. Todas as informações são consolidadas em uma estrutura chamada draft, que encapsula os detalhes necessários para as etapas seguintes.

3.5.2 Construção de Pacotes

Na construção de pacotes, o cabeçalho DNS é gerado com um identificador de transação (Transaction ID) aleatório e configurações padrão, como o tipo de consulta (Query Type) definido como A (IPv4) e a classe como IN (Internet). Além disso, o domínio alvo é concatenado com um subdomínio aleatório, formado por uma string hexadecimal, garantindo que cada consulta seja única.

Esse subdomínio é transformado em etiquetas DNS, em que cada segmento do domínio é precedido por um byte que indica seu comprimento. A montagem do pacote segue o protocolo DNS [20], incluindo campos de consulta e classe. Após a criação do pacote DNS, são configurados os cabeçalhos IP e UDP, contendo os endereços de origem e destino, além das portas.

O endereço IP de origem é modificado para simular uma ou múltiplas fontes. Durante o ataque, o subdomínio é remodelado continuamente, assegurando que cada consulta seja única. Essa característica aumenta a aleatoriedade e a efetividade do ataque, dificultando sua mitigação. O pacote modificado é encapsulado em uma estrutura packet wrapper, que armazena o conteúdo e metadados do pacote, garantindo sua consistência antes do envio.

3.5.3 Injeção de Pacotes

A injeção de pacotes começa com a criação do plano de ataque pelo módulo manager, com base nos parâmetros do draft. O módulo commander chama esse plano e executa a função correspondente, como o DNS Flood. A coordenação da injeção é feita pelo módulo controller, que gerencia os injetores responsáveis pelo envio dos pacotes.

No módulo injector, os injetores são configurados por meio da função StartInjector, que cria um socket RAW para cada injetor e ajusta os parâmetros do pacote, como endereços IP e portas. Durante a execução, a função ReForgeDnsPacket remodela subdomínios e ajusta os pacotes dinamicamente, garantindo que cada consulta DNS seja única.

O envio dos pacotes é realizado em loops contínuos. Cada injetor respeita os parâmetros configurados, como taxa de envio (rate), ou modo RAID, para maximizar o tráfego. Quando múltiplos IPs de origem são configurados, o sistema cria cópias profundas do pacote original, ajustando cada cópia para refletir o IP de origem correspondente. O processo de cópia profunda em C é realizado por meio da alocação dinâmica de memória, utilizando a função malloc, que retorna um ponteiro para a nova região de memória. O conteúdo original é copiado para esse novo endereço, garantindo que alterações em uma cópia não afetem as demais, preservando assim o pacote original.

3.6 Usabilidade

A interação com a ferramenta pode acontecer via interface gráfica, ou por linha de comando, facilitando assim sua manipulação.

3.6.1 Interface por linha de comando

Quando executada via linha de comando, o decorrer do ataque pode ser acompanhado pelo output no terminal, assim como o ilustrado na figura A.1. A saída da ferramenta traz informações do nível, pacotes enviados, drop rate, endereço de origem e destino a cada iteração.

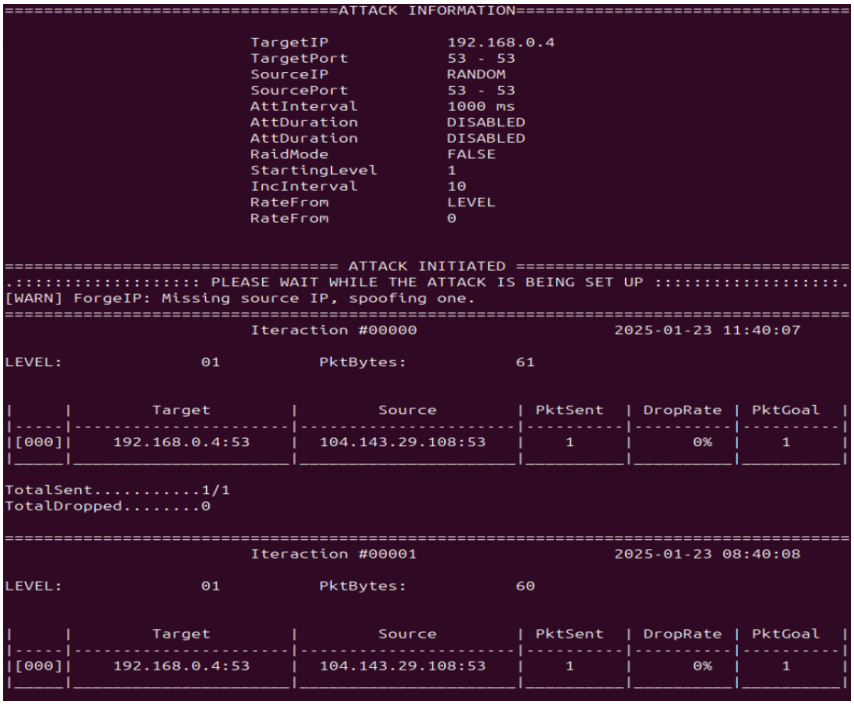

Figura 3.6: Interface via linha de comando

3.6.2 Interface gráfica

A interface gráfica apresentada na imagem 3.7 torna o uso da ferramenta ainda mais intuitivo, à medida que traz visibilidade dos possíveis parâmetros de ataque.

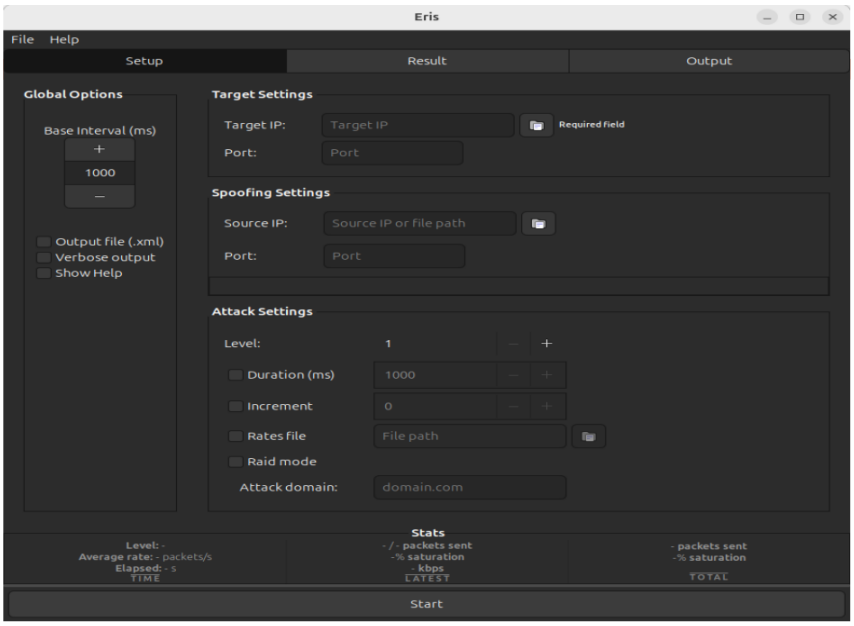

Figura 3.7: Interface gráfica

Com a interface gráfica, é possível visualizar o andamento do ataque tanto pela aba Results quanto pela aba Output.

Figura 3.8: Acompanhamento do ataque via aba Output

A seção Results traz as informações acerca do ataque por meio de um gráfico, assim como ilustrado na imagem 3.9.

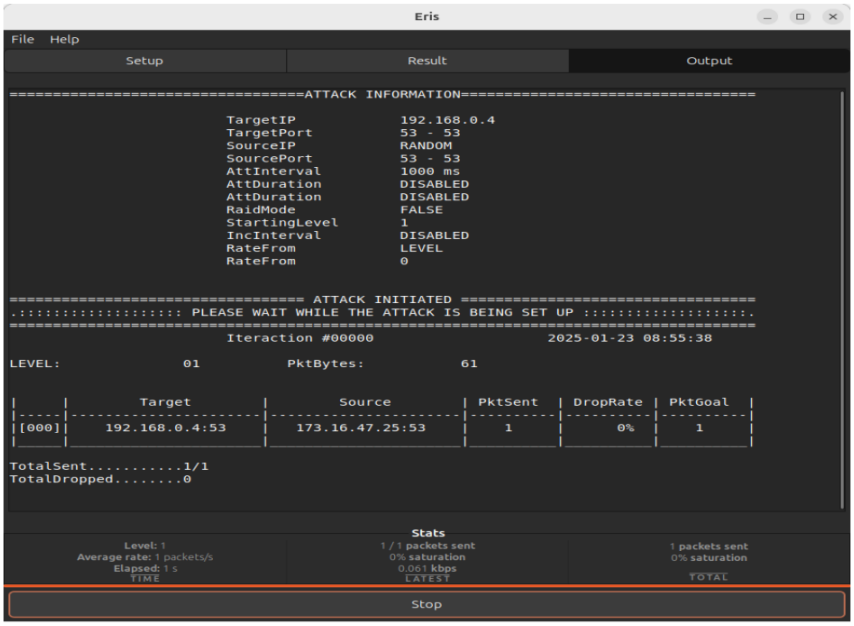

Figura 3.9: Acompanhamento do ataque via aba Results

A área Output traz a mesma saída da ferramenta, quando executada via interface de linha de comando, assim como na figura 3.8.

3.7 Disponibilização

A disponibilização da ferramenta foi feita via dois meios: o código fonte e uma imagem Docker, permitindo assim a liberdade para interagir com o código fonte, caso desejado e a praticidade de utilizar o Docker para sua execução, sem necessidade de configurações adicionais. O código fonte da Eris está disponível no repositório do GitLab [1] e sua imagem Docker pode ser encontrada no Docker hub [2].

3.8 Considerações finais

Este capítulo apresentou as decisões de arquitetura, o funcionamento, o fluxo de execução a utilização e disponibilização da Eris. O capítulo seguinte descreve a metodologia e os procedimentos dos testes aos quais a ferramenta foi submetida.

[1] `https://gitlab.com/thais.f.c.garcia/eris`
[2] `https://hub.docker.com/r/thaisfcg/eris`

Capítulo 4

Metodologia e procedimentos

Este capítulo apresenta a metodologia adotada para testar o desempenho do ataque performado pela Eris e os procedimentos seguidos para tal.

4.1 Metodologia de testes e Procedimentos

4.1.1 Metodologia de testes

A Eris foi projetada para causar indisponibilidade em servidores DNS via um número excessivo de consultas a subdomínios inexistentes de um domínio específico. Dessa forma, para determinarmos ou não seu sucesso, avaliamos a disponibilidade de servidores durante sua execução e a volumetria máxima de pacotes enviados pela ferramenta. Adicionalmente, foram realizados testes para garantir que o tráfego gerado por ela era válido, assim sendo capaz de realizar um verdadeiro ataque de Water Torture e, tal como o Mirai, causar indisponibilidade além do servidor o qual recebe as requisições diretamente.

Logo, com o intuito de validar os aspectos citados acima, foram realizados 4 testes, utilizando o modo incremental e o modo RAID da ferramenta. O modo incremental foi utilizado de acordo com as tabelas 4.1 e 4.2.

Parâmetro	**Valor**
`level (-l)`	1
`increment (-i)`	10

Tabela 4.1: Parâmetros Utilizados

Nível	Pacotes/iteração
1	1
2	10
3	100
4	1000
5	10000
6	100000
7	1000000
8	10000000
9	100000000
10	1000000000

Tabela 4.2: Níveis de ataque

As métricas utilizadas para descrever os testes utilizados são:

- Segundo em que o servidor DNS deixa de responder;
- Taxa máxima de envio;

O segundo em que o servidor deixa de responder é medido por meio da simulação de um usuário legítimo, realizando uma requisição por segundo para um domínio o qual está no cache do servidor. É considerado que o servidor deixou de responder como esperado quando observamos dois erros seguidos. A taxa máxima de envio é a maior quantidade de pacotes por segundo que o atacante foi capaz de produzir e enviar durante o período do teste.

Adicionalmente, o desempenho da ferramenta foi avaliado sob as perspectivas de:

- Hardware do atacante: A ferramenta foi executada em dois diferentes hardwares, seu desempenho foi medido em cada um deles.
- Comparação com ferramentas com a mesma proposta: O desempenho da Eris foi comparado com duas outras ferramentas as quais possuem o mesmo propósito.

4.2 Procedimentos

Seguindo a metodologia descrita, foram montados dois ambientes de teste, um virtual e um com máquinas reais, adicionalmente, foram definidos quatro cenários de teste para validar todos os aspectos relevantes da ferramenta. O tráfego de rede foi coletado em todas as máquinas envolvidas nos testes, utilizando o software Wireshark [21], para a

análise dos resultados obtidos, medição dos dados e geração de gráficos foram utilizadas as bibliotecas Numpy [22] e Matplotlib [23] do Python.

4.2.1 Ambientes de teste virtual

Para a simulação em ambiente virtual, foi montado um sistema com o software VirtualBox, utilizando uma rede NAT com range 192.168.0.0/24, assim como demonstrado no diagrama da imagem 4.1.

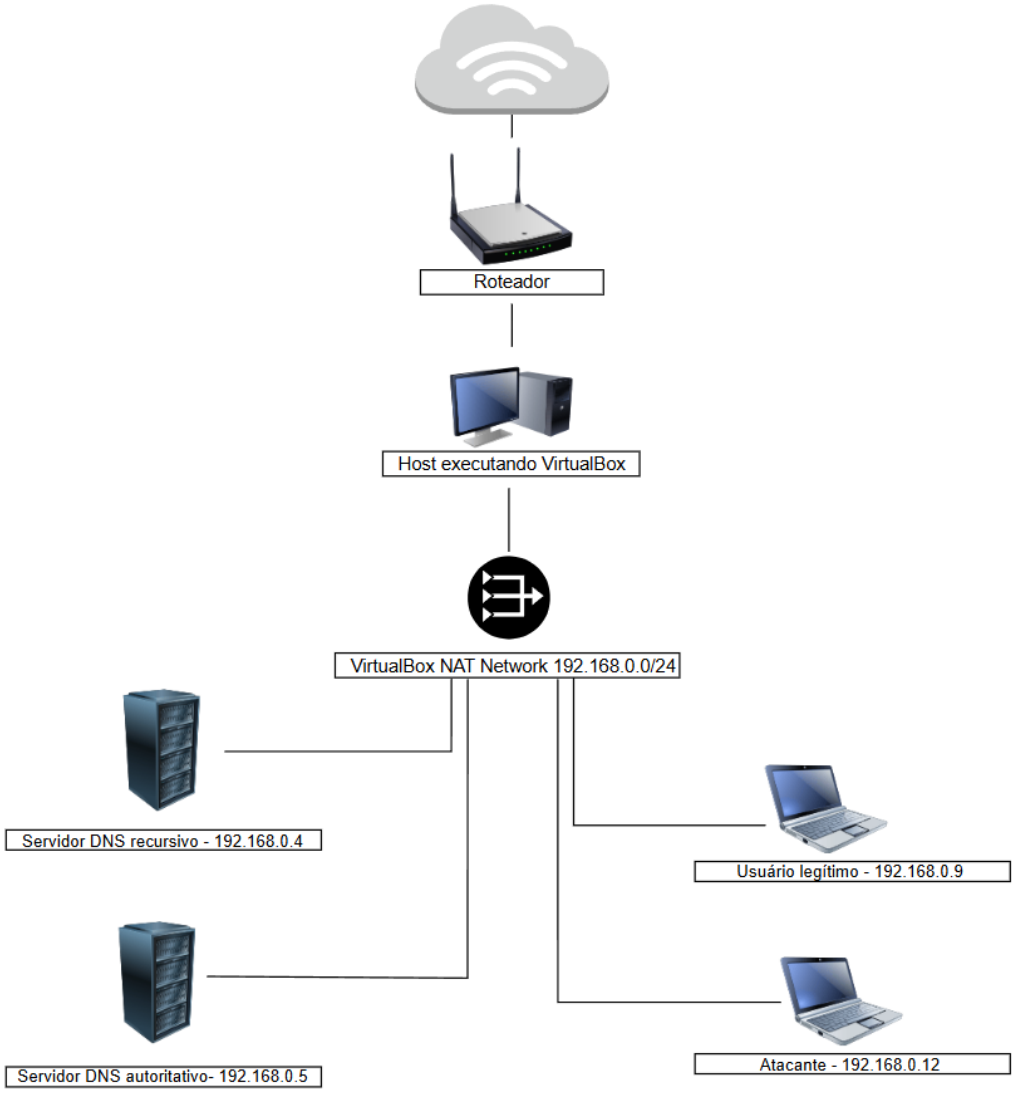

Figura 4.1: Diagrama do ambiente virtual

O ambiente foi configurado conforme a tabela 4.3, composto pelo atacante, o usuário legítimo e os servidores DNS. O modelo DNS utilizado foi simplificado, de modo que o servidor recursivo consulta diretamente o servidor autoritativo responsável pelos domínios testados, eliminando os servidores raiz e TLD.

Nome	**Processador**	**Memória**
Atacante	4 CPU	10GB
Servidor DNS recursivo	4 CPU	10GB
Servidor DNS autoritativo	4 CPU	10GB
Usuário 1	2 CPU	4GB

Tabela 4.3: Configuração dos sistemas virtuais.

O serviço de DNS, em ambos os servidores, foi configurado com o software BIND9 [24], de forma que, o servidor DNS recursivo foi caracterizado por possuir a configuração de recursão ativada e consultar o servidor em 192.168.0.5 em busca dos registros definitivos do domínio requisitado.

O usuário legítimo foi utilizado para medir os tempos de reposta do servidor DNS sob ataque, realizando uma *query* DNS por segundo para o DNS recursivo.

Em ambiente virtual, foram realizados os testes dos cenários 1 e 3 descritos ainda neste capítulo.

4.2.2 Ambiente para teste bare metal

O ambiente para teste foi montado utilizando 5 máquinas físicas, todas conectadas de forma wireless ao mesmo roteador TP-Link TL-WR829N conforme como indicado no diagrama 4.2.Para isolar a rede o roteador foi configurado como *Access Point* isolado (sem uplink para Internet).

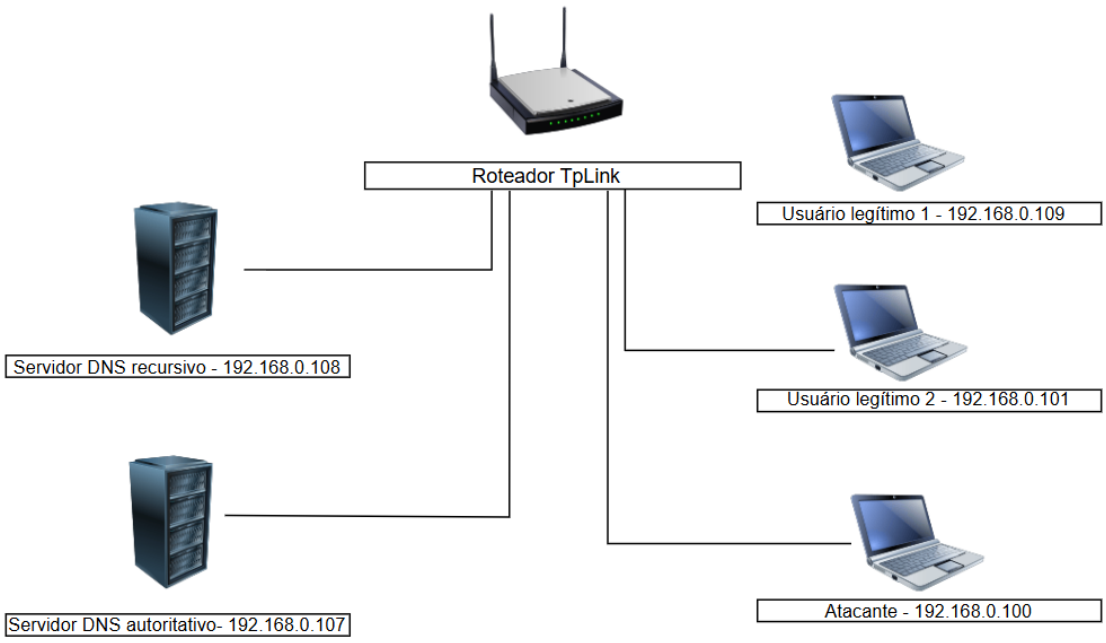

Figura 4.2: Diagrama do bare metal

Assim como no ambiente virtual, o serviço de DNS, em ambos os servidores, foi configurado usando o software BIND9 [24], de forma que, o servidor DNS recursivo possui a configuração de recursão ativada e consulta o servidor autoritativo em busca dos registros definitivos do domínio requisitado.

Os usuários legítimos foram utilizados para medir os tempos de reposta dos servidores DNS sob ataque, realizando uma *query* DNS por segundo cada.

Nome	Processador	Memória
Atacante	Intel Core i7-14700K @ 3,40 GHz	32 GB DDR4 @ 3600 MHz
Servidor DNS recursivo	Intel Core i5-4200U @ 1,60 GHz	6 GB DDR3 @ 1600 MHz
Servidor DNS autoritativo	Intel Core i3-3217U @ 1,80 GHz	4 GB DDR3 @ 1600 MHz
Usuário 1	Apple M2 chip (2022)	8 GB
Usuário 2	Intel Core i5 @ 1,40 GHz	8 GB DDR3 @ 2400 MHz
Roteador TP-Link TL-WR829N	Single-Core CPU	32 MB DDR2

Tabela 4.4: Inventário de dispositivos e roteador utilizados nos testes bare-metal.

O ambiente bare metal foi utilizado para a execução do cenário 2 de teste ainda descrito nesse capítulo.

4.2.3 Padrões de ataque

Para os testes, definiram-se dois padrões de ataque com configurações de taxa de envio distintas. O primeiro padrão adota um esquema incremental, com duração total de 200 segundos, em que a intensidade do ataque aumenta progressivamente, como descrito na tabela 4.5. Nesse padrão, o tempo é dividido em intervalos de 10 segundos, denominados "degraus". A cada degrau, a taxa de envio de pacotes é multiplicada por 10, sendo que cada multiplicação representa um "nível" de ataque mais intenso. Assim, o nível inicial se mantém durante os primeiros 10 segundos e, a cada novo degrau, o nível é incrementado, refletindo um aumento sistemático da vazão.

Parâmetro	Valor
Duração	Intervalo: 1 s Nível: 10 s Total: 200 s
Nível	Inicial: 1 Final: 10

Tabela 4.5: Parâmetros padrão de ataque 1.

Já o segundo padrão de ataque definido utiliza o modo RAID da ferramenta, que envia o maior número de pacotes por segundo possível, surante 200 segundos.

4.2.4 Cenários de teste

Primeiro cenário

Para esse primeiro cenário, o ataque é executado em ambiente virtual, de acordo com as especificações de hardware da figura 4.1 e da tabela 4.3. A Eris nesse cenário foi executada utilizando os parâmetros condizentes com o primeiro padrão de ataque, assim como descrito anteriormente na tabela 4.5.

As capturas de tráfego foram realizadas na máquina do atacante, no servidor DNS recursivo e na máquina do usuário legítimo, usando o Wireshark [21] conforme a tabela 4.6

Tráfego	**Ponto de Captura**	**Destino**	**Objetivo da Medição**
Tráfego do Atacante	Pacotes saindo da máquina do atacante	Servidor DNS	Quantificar o volume de requisições maliciosas
Tráfego do Alvo	Pacotes saindo da máquina do usuário	Servidor DNS	Medir a taxa de requisições legítimas (1 req/s)
Resposta do Servidor	Pacotes saindo do servidor DNS	Máquina do usuário	Avaliar até quando o servidor ainda responde ao usuário

Tabela 4.6: Medições de tráfego para avaliação de resiliência do servidor DNS sob ataque.

O primeiro cenário foi concebido para demonstrar o funcionamento da Eris, evidenciando suas funcionalidades de controle de vazão de tráfego. Utilizando os parâmetros do primeiro padrão de ataque (conforme indicado na Tabela 4.5), o teste objetiva verificar a capacidade da ferramenta em gerar tráfego de consultas DNS com subdomínios pseudoaleatórios de forma controlada e incremental. Dessa forma, busca-se identificar a taxa máxima de envio atingida antes da saturação do servidor DNS recursivo e comprovar a eficácia do controle incremental de vazão implementado na ferramenta.

O resultado esperado é observar um gráfico que toma forma de escada, onde cada degrau tem duração de 10 segundos e onde entre cada degrau a taxa de envio de pacotes por

segundo é multiplicada por 10. Adicionalmente, o resultado esperado é que a ferramenta seja capaz de causar a indisponibilidade desejada no servidor alvo.

Segundo cenário

Para o segundo cenário, foi utilizado o modo RAID da ferramenta, sob o ambiente bare metal. Para medir o desempenho, foram utilizados dois usuários, ambos fazendo uma requisição por segundo. O usuário 1 para o servidor recursivo e o 2 para o servidor autoritativo.

Este teste tem como objetivo avaliar a rapidez com que a ferramenta atinge seu throughput máximo de consultas por segundo e confirmar que o padrão de tráfego gerado se comporta como requisições legítimas, fazendo com que o servidor recursivo encaminhe todas as consultas ao servidor autoritativo. É justamente essa aparente legitimidade das requisições que provoca o efeito dominó: ao sobrecarregar o recursivo, as mesmas consultas são repassadas em massa ao autoritativo, que também acaba ficando indisponível, demonstrando a propagação do impacto por toda a infraestrutura. Resumindo:

Objetivos do teste

1. **Rampa de subida ao throughput máximo:** medir em quanto tempo a Eris atinge sua capacidade plena de envio de pacotes por segundo.
2. **Realismo do tráfego:** verificar se o padrão de requisições da Eris é suficientemente parecido com requisições legítimas, de modo que o servidor recursivo encaminhe todas ao autoritativo causando um efeito domino.
3. **Capacidade de resiliência do servidor:** até qual volume de requisições simultâneas os servidores mantêm serviço aos usuários legítimos.

Coleta e métricas de tráfego As medições de tráfego foram realizadas com o `tshark`, capturando:

Tráfego	**Ponto de Captura**	**Destino**	**Objetivo da Medição**
Tráfego do Atacante	Pacotes saindo da máquina do atacante	Servidor DNS recursivo	Quantificar o volume de requisições maliciosas geradas pela Eris em modo RAID
Resposta do Servidor Recursivo	Pacotes saindo do servidor DNS recursivo	Máquina do usuário 1	Avaliar até quando o recursivo atende as requisições legítimas
Resposta do Servidor Autoritativo	Pacotes saindo do servidor DNS autoritativo	Máquina do usuário 2	Avaliar até quando o autoritativo responde às requisições legítimas

Tabela 4.7: Medições de tráfego detalhadas para avaliação de resiliência dos servidores DNS sob ataque em modo RAID.

Cada fluxo foi registrado em arquivo PCAP separado, permitindo análise detalhada posterior (número de pacotes, picos de PPS e intervalo sem resposta).

Critério de falha Considera-se falha quando o servidor DNS não responde com timeout às duas últimas requisições consecutivas enviadas pelo cliente legítimo, que faz uma requisição por segundo continuamente. O tempo para timeout foi configurado como 15 segundos.

Resultados esperados

- A Eris em modo RAID deverá atingir o pico de PPS *quase que imediatamente* após o início, sem necessidade de parâmetros adicionais de tuning.
- O pico de tráfego malicioso será superior ao observado no primeiro cenário (ambiente virtualizado), devido ao hardware bare-metal mais potente.

- À medida que o servidor recursivo encaminha cada requisição recebida ao servidor autoritativo, este também é progressivamente sobrecarregado, evidenciando o efeito dominó.
- A partir da captura de resposta (Tabela 4.6), será possível quantificar o tempo até falha do serviço de resolução de domínios.

Terceiro cenário

O terceiro cenário consiste na execução de ferramentas que possuem a mesma proposta da Eris, com o objetivo de comparar os resultados e funcionalidades, assim provando a eficiência da ferramenta descrita neste trabalho. Dessa forma, a Eris foi comparada com as ferramentas DNSWaterTorture [6] e dns-flood-ng[7], as quais possuem a mesma proposta.

Ambiente de teste O cenário foi executado no mesmo ambiente virtual descrito anteriormente em 4.1. Cada ferramenta rodou isoladamente, sem concorrência, para evitar interferências entre experimentos.

Ferramentas comparadas Foram avaliadas três soluções de geração de tráfego DNS com proposta semelhante:

- **Eris** (ferramenta proposta neste trabalho).
- **DNSWaterTorture** [6].
- **dns-flood-ng** [7].

Procedimento experimental Cada ferramenta foi configurada para enviar a maior quantidade de pacotes por segundo possível. O experimento seguiu este fluxo:

1. Inicialização da ferramenta com taxa de envio máxima.
2. Captura de todo o tráfego DNS com tshark durante 100 segundos consecutivos.
3. Geração de arquivo PCAP individual para cada ferramenta, garantindo análise isolada.

Métricas de comparação Os seguintes indicadores foram extraídos de cada captura para comparação:

- **Packets per second (PPS) máximo atingido**: pico de pacotes por segundo registrado.

- **Consistência de vazão**: capacidade de manter-se próximo ao PPS máximo ao longo de todo o teste.
- **Validação de consultas**: existência de requisições DNS formatadas incorretamente e, consequentemente, não processadas pelo servidor recursivo.
- **Flexibilidade de configuração**: facilidade para ajustar parâmetros de taxa de vazão do ataque.

Objetivo do teste comparativo Demonstrar que, sob as mesmas condições de hardware e rede:

1. A Eris alcança um PPS máximo igual ou superior ao das demais ferramentas.
2. A Eris mantém sua vazão mais estável (menor desvio de PPS ao longo dos 100 s).
3. A Eris gera consultas DNS válidas em percentual maior, garantindo o encaminhamento completo pelo servidor recursivo.
4. A Eris oferece opções de personalização de taxa e payload superiores, facilitando experimentos futuros mais detalhados.

Expectativa de resultados Espera-se que a Eris seja a única capaz de combinar alto throughput, consistência e geração de consultas válidas de forma configurável, comprovando sua eficiência em comparação às alternativas DNSWaterTorture e dns-flood-ng.

4.3 Considerações finais

Neste capitulo foi apresentada a metodologia de testes seguida nesse trabalho, além dos procedimentos realizados para conclui-los.

No capítulo a seguir, apresentamos os resultados quantitativos obtidos para cada cenário de teste e a as conclusões acerca a eficácia da Eris.

Capítulo 5

Resultados obtidos

5.1 Resultados obtidos por cenário

Este capítulo apresenta os dados coletados durante os testes e os apresenta por meio de gráficos e tabelas. Os resultados são analisados conforme os resultados esperados.

5.1.1 Primeiro cenário

No gráfico da figura 5.1, o tráfego que deixa a máquina do atacante é representado pelo gráfico vermelho e o que é recebido pelo servidor alvo é representado pelo gráfico azul.

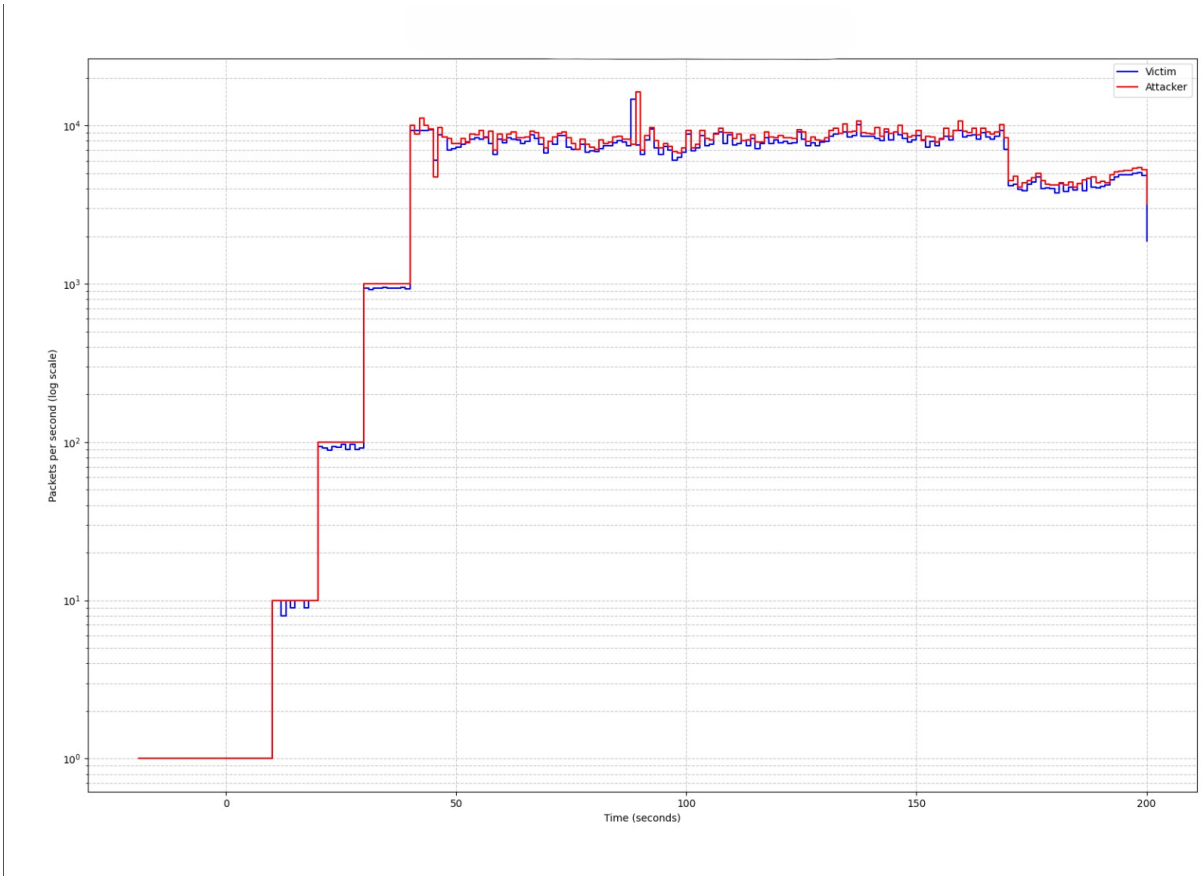

Figura 5.1: Tráfego capturado durante cenário 1.

Dessa forma, podemos confirmar que o ataque segue o definido pelos parâmetros de execução, até que alcance a saturação da taxa de envio. A saturação foi atingida a partir do segundo 59 do ataque, fazendo com que a ferramenta não ultrapassasse o nível 5. A maior taxa de envio observada nessa janela foi de 16.37 kpackets/s no segundo 89, e a media durante o período de saturação foi de 8.63 kpackets/s.

Na imagem 5.2 observa-se a projeção do envio de pacotes, de acordo com o planejado representada pela linha tracejada vermelha, e a taxa de envio real representada pela linha contínua. Podemos observar que a ferramenta atinge o limite do hardware de produção e envio de pacotes antes da metade do tempo total de ataque. De acordo com os parâmetros configurados, deveríamos ver 10 diferentes níveis de ataque, no entanto, a máquina apresenta uma estagnação a partir do nível 5, assim como citado anteriormente.

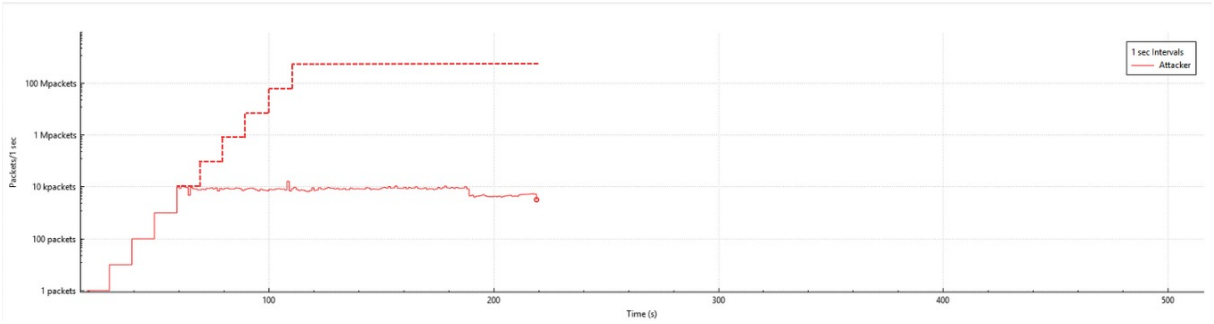

Figura 5.2: Projeção do ataque.

Nível	Pacotes Projetados (por segundo)	Media pacotes Enviados (por segundo)
1	1	1
2	10	10
3	100	100
4	1 kpackets	1 kpackets
5	10 kpackets	8.84 kpackets
6	100 kpackets	8.39 kpackets
7	1 Mpackets	8.51 kpackets
8	10 Mpacktes	8.07 kpackets
9	100 Mpackets	8.82 kpackets
10	1 Gpackets	9.33 kpackets

Tabela 5.1: Comparação entre Pacotes Projetados e Enviados por Nível

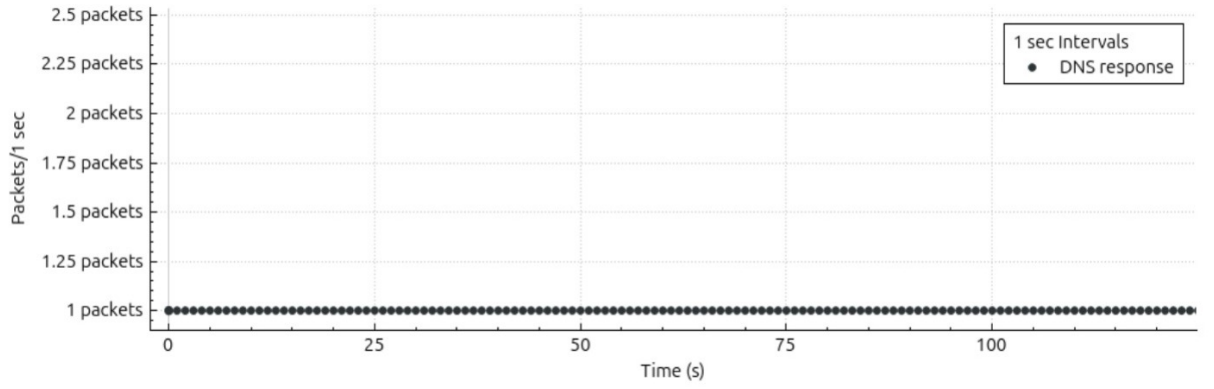

Figura 5.3: Resposta esperada do servidor recursivo DNS sob condições normais.

Na figura 5.4 é possível visualizar a frequência com que o servidor sob ataque responde ao usuário legítimo, o qual faz uma consulta DNS por segundo. A linha observada deveria apresentar os pontos de maneira contínua, assim como representado na imagem 5.3, onde cada ponto representa uma resposta do servidor DNS ao usuário legítimo, totalizando uma resposta por segundo. No entanto, o servidor apresenta o primeiro erro no segundo 15, e apresenta erros consecutivos a partir do segundo 25 sob ataque. Ou seja, o primeiro erro consecutivo é evidenciado com 1 kpackets/s.

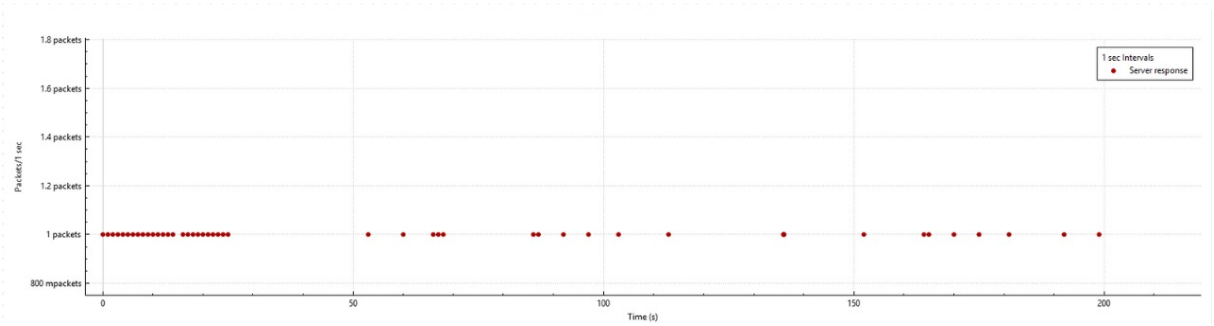

Figura 5.4: Resposta do servidor autoritativo durante cenário 1.

Dessa forma, os resultados esperados de observar o funcionamento do controle de vazão e da eficácia do ataque se confirmam. Além de ficar evidente que, devido à lógica e particularidade do alerta, não é necessária uma taxa de envio de tráfego muito alta para causar indisponibilidade no servidor.

5.1.2 Segundo cenário

O resultado da execução da ferramenta em modo RAID no ambiente bare metal pode ser observado na imagem 5.5, onde o gráfico vermelho representa o volume de pacotes que deixa a máquina do atacante e o azul representa o tráfego recebido pelo servidor DNS recursivo.

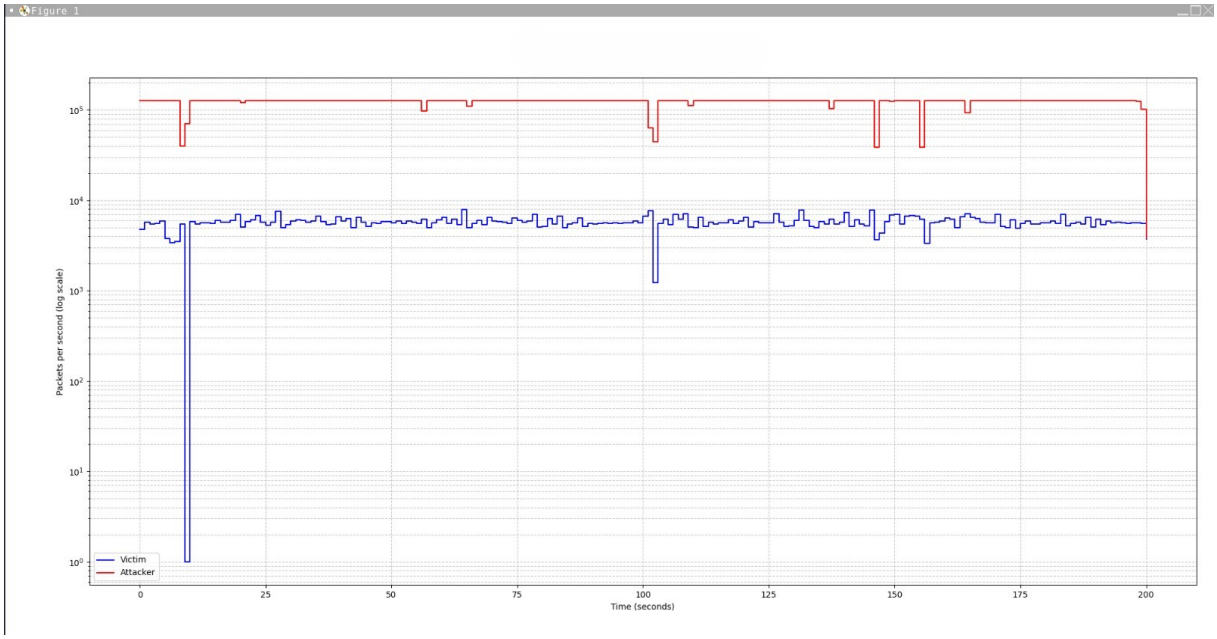

Figura 5.5: Captura do tráfego cenário 2.

Pode-se observar que a ferramenta atingiu a saturação da taxa de envio de pacotes entre o segundo 0 e 1 de execução. A maior taxa observada foi de 126,791 packets/s no primeiro segundo do ataque, confirmando os resultados esperados para esse cenário de observar uma taxa de saturação mais alta que a do anterior.

No entanto, também foi observada uma grande discrepância entre a quantidade de pacotes enviados e a quantidade recebida pelo alvo, assim como observado na tabela. 5.2.

Pacotes	**Ambiente Bare metal**
Maior taxa enviada pelo Atacante	126.791 packets/s
Maior taxa recebida pelo Alvo	7.958 packets/s

Tabela 5.2: Comparação de pacotes enviados e recebidos

Nesse cenário, o DNS recursivo deixa de responder o usuário 1 logo no primeiros segundos do ataque, assim como registrado no gráfico da imagem 5.6.

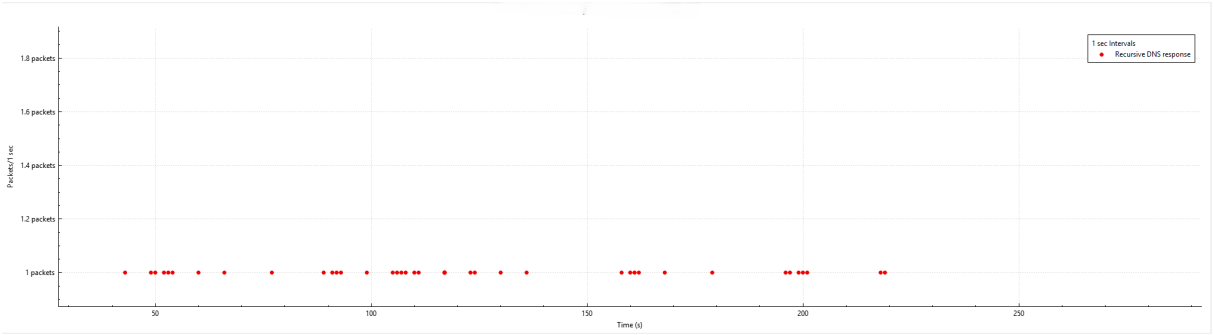

Figura 5.6: Resposta do servidor recursivo sob ataque.

O servidor autoritativo deixa de responder o usuário 2 um segundo após o recursivo, assim como mostrado no gráfico na imagem. 5.7.

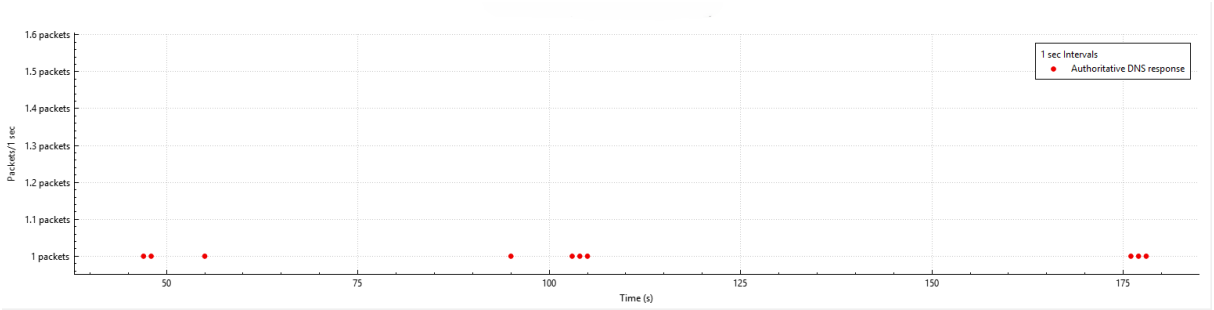

Figura 5.7: Resposta do servidor autoritativo sob ataque.

A resposta dos servidores confirma que o tráfego malicioso está sendo propagado entre os servidores alvo.

5.1.3 Terceiro cenário

No terceiro cenário, buscou-se comparar a eficiência e a eficácia da Eris com as ferramentas DNSWaterTorture e dns-flood-ng[7], ambas com o mesmo propósito de simular ataques DNS Water Torture. Para tanto, os testes foram realizados em ambiente virtual, mantendo as mesmas condições de hardware e configuração para as três ferramentas, com ataques de duração de 100 segundos executados em taxa máxima.

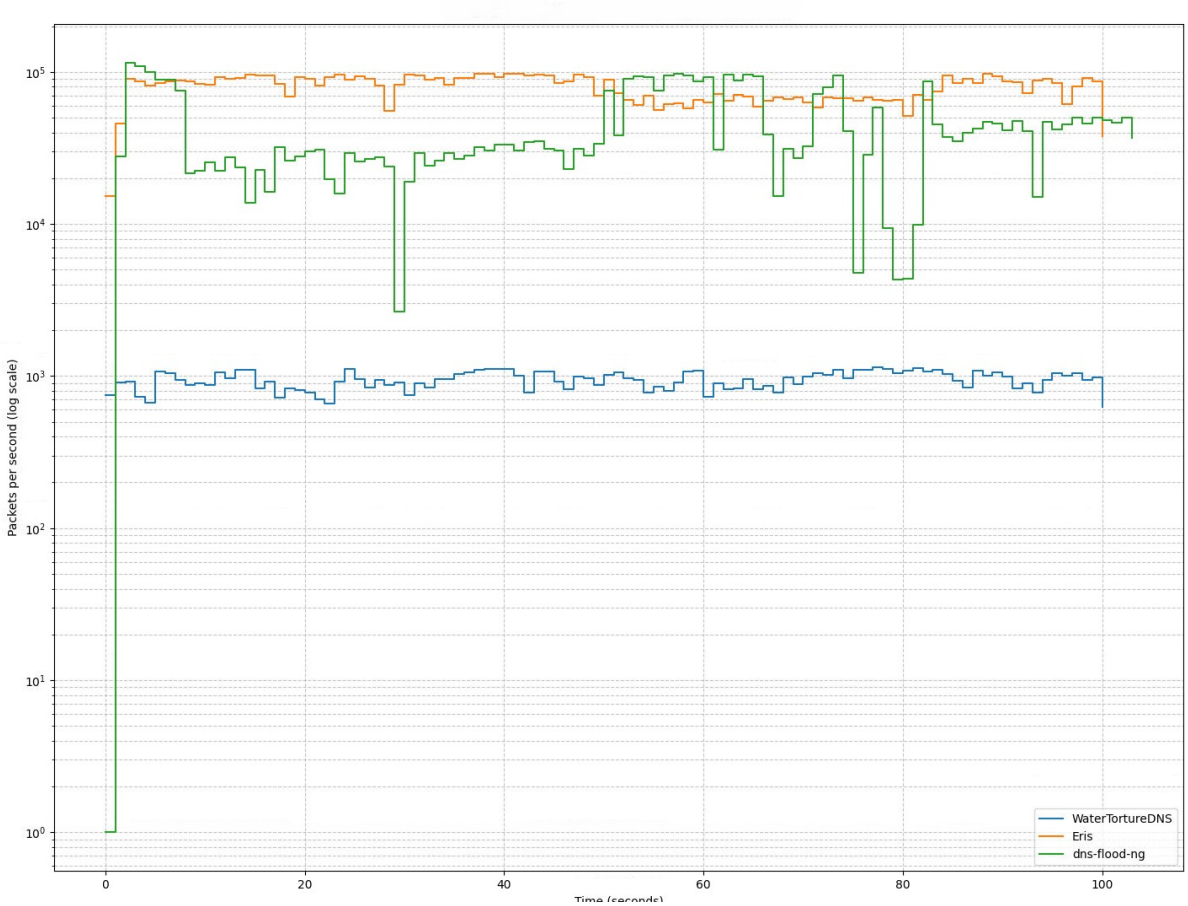

Figura 5.8: Comparativo entre as ferramentas testadas

A dns-flood-ng foi a ferramenta que mais se aproximou da vazão de pacotes por segundo da Eris, conseguindo alcançar 125 mil pacotes por segundo, assim como apresenta a imagem 5.8. No entanto, uma parte significativa do tráfego gerado é composta por pacotes mal formados, assim como mostrado na imagem 5.9.

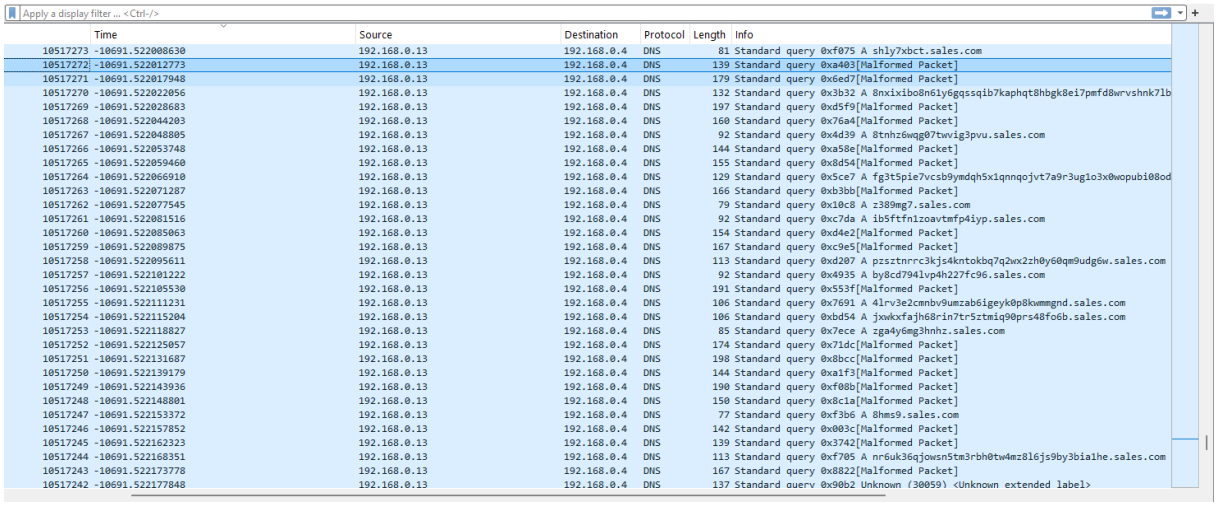

Figura 5.9: Pacotes formados pela dns-flood-ng.

Os resultados advindos da DNSWaterTorture demonstraram que, apesar de apresentar consultas bem formadas, não foi capaz de produzir um volume de tráfego significativo, assim como ilustrado na imagem 5.8.

Em contraste, a Eris se destacou ao proporcionar uma configuração granular dos parâmetros de envio – permitindo desde o controle incremental da taxa até a utilização do modo RAID – o que possibilitou não só um ajuste preciso da intensidade do ataque, mas também a manutenção de uma vazão média mais elevada e consistente. Dessa forma, a ferramenta foi capaz de gerar tráfego realista e eficaz, causando indisponibilidade tanto no servidor recursivo quanto no servidor autoritativo, validando seu desempenho superior frente às alternativas testadas.

5.2 Conclusão acerca dos testes

5.2.1 Primeiro cenário

Diante do apresentado anteriormente, o teste foi eficaz em demonstrar como o controle da taxa de envio de pacotes da Eris funciona, confirmando sua granularidade, assim como é possível observar na tabela 5.1.

5.2.2 Segundo cenário

Observou-se que, ao atingir o servidor recursivo, o servidor autoritativo consultado por ele também perdeu a capacidade de responder às consultas recebidas, indicando que outros

servidores os quais dependem das suas respostas também seriam impactados, mesmo não sendo diretamente atacados. Esse efeito se propaga para usuários finais que dependem desses servidores para resolver domínios, impossibilitando o acesso ao conteúdo hospedado nos servidores desejados. Em resumo, ao direcionar tráfego para apenas um servidor, a ferramenta mostrou-se capaz de causar indisponibilidade em outros servidores e também para os usuários finais.

5.2.3 Terceiro cenário

Adicionalmente, foi constatado que a Eris superou o desempenho de ambas as ferramentas comparadas por uma série de fatores. A ferramenta melhor avaliada, mesmo em sua capacidade máxima, não supera a media de envio de pacotes da Eris, ademais, a ferramenta comparada também não foi capaz de produzir tráfego completamente válido, entregando pacotes mal formados. Por fim, nenhuma das duas ferramentas comparadas possui a opção de personalização da vazão das consultas DNS, de forma que possibilite testes de resiliência mais detalhados.

Ferramenta	**Vazão media**	**Vazão máxima (segundo em que ocorreu)**
WaterTortureDNS	942	1138 (77 s)
Eris	78,841	97,175 (41 s)
dns-flood-ng	43,825	114,903 (2 s)

Tabela 5.3: Comparativo entre as ferramentas.

5.3 Síntese dos resultados

Os testes realizados em todos os cenários demonstraram que a ferramenta foi eficaz em causar indisponibilidade no servidor alvo, em ambas as formas de ataque utilizadas e em ambos os ambientes. Além disso, constatou-se a capacidade da ferramenta de propagar indisponibilidade, ao sobrecarregar tanto os servidores DNS recursivos quanto os autoritativos, ao gerar tráfego pseudoaleatório contínuo que não deixa de se assemelhar a consultas legítimas.

Pode-se notar, no entanto, que nos testes realizados em ambiente bare-metal existe uma discrepância entre a quantidade de pacotes enviados pelo atacante e a quantidade recebida pela vítima. O ponto mais notável sendo logo no início, no segundo 0, onde foi evidenciada uma diferença de 122,013 packets/s, sendo dessa forma, 25 vezes maior do que o recebido pela vítima. Assim como demonstrado na imagem 5.10, portanto, sendo necessário levar em conta para a avaliação da resiliência do servidor o valor de packets/s recebido por

ele. O cenário 1 também apresenta um comportamento de saturação, no entanto, ele se difere devido a sua natureza, no cenário 1, em rede NAT virtual do VirtualBox, o ataque estagnou em cerca de 8,6 kpps, indicando uma limitação pela capacidade de geração de pacotes do host , sem que a recepção pelo alvo apresentasse queda proporcional, o que indica que o gargalo residia no próprio atacante e não no enlace, indicando que isso é um efeito da virtualização a qual limita o hardware simulado, restringindo a taxa máxima de pacotes gerados devido à abstração e compartilhamento de recursos físicos, impedindo que a performance do ataque ultrapasse os limites impostos pela camada de virtualização. Já no cenário 2, com todas as máquinas conectadas via Wi-Fi à mesma interface, o atacante atingiu um pico de 122 kpps enquanto o alvo recebeu apenas 4,8 kpps (diferença de 25×, conforme a figura 5.10), evidenciando perdas massivas e achatamento do throughput devido à saturação física do enlace sem fio. Esse comportamento alinha-se ao reportado em estudos de ataques reflexivos amplificados, nos quais servidores refletores deixam de converter o aumento das requisições de entrada em tráfego de saída ao atingirem o teto de vazão de suas interfaces de rede [25]

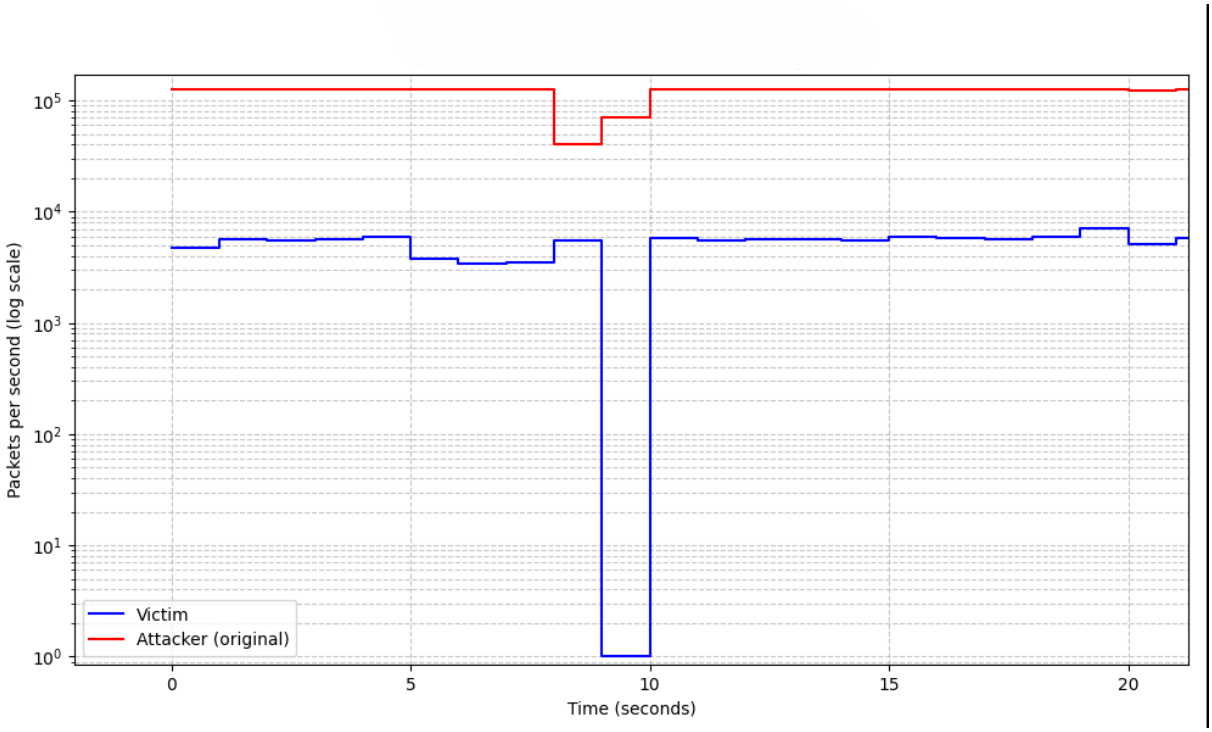

Figura 5.10: Dispersão de pacotes observada no cenário 2.

Nota-se também que o trafego gerado pela Eris torna-se previsível a certo nível por possui um padrão de tamanho na string que compõe o subdomínio randômico, sendo assim um ponto para melhorias futuras.

5.4 Considerações finais

Este capítulo apresentou os resultados obtidos a partir dos testes realizados nesse trabalho, afirmando o desempenho notável da Eris, a qual se sucedeu em causar indisponibilidade em todos os cenários testados, além de se mostrar mais eficaz frente às ferramentas a ela comparadas, de acordo com os critérios eleitos. Por fim, no capitulo a seguir são apresentadas as conclusões acerca desse trabalho.

Capítulo 6

Conclusão

Neste trabalho, exploramos o desenvolvimento e avaliação da Eris, uma ferramenta projetada para medir a resiliência de servidores DNS por meio da simulação controlada de ataques volumétricos do tipo DNS Water Torture. Inicialmente, foram apresentados os fundamentos teóricos sobre ataques de negação de serviço, o funcionamento do DNS e os detalhes do ataque Water Torture. Com base nesses conceitos e nas ferramentas de ataques DDoS DamBuster e Linderhof, foi construída a Eris, que combina facilidade de uso e alto desempenho. A arquitetura da ferramenta foi estruturada em 4 módulos principais, os quais, juntos, são responsáveis por garantir a interação com o usuário, gerenciar o ataque, criar os pacotes e enviá-los. Seu funcionamento foi pensado para atender a cenários de teste variados, permitindo ajustes configuráveis, como taxa de envio de pacotes e geração de subdomínios pseudoaleatórios, garantindo que os ataques simulados sejam realistas e eficazes.

Os testes realizados avaliaram a eficácia da ferramenta em diferentes cenários. No primeiro cenário, foi demonstrada a capacidade da Eris de controlar a vazão de pacotes e gerar indisponibilidade em servidores DNS recursivos. No segundo, foi validada sua habilidade de atingir a taxa máxima de envio de pacotes rapidamente, além de confirmar a legitimidade do tráfego gerado ao ser capaz de reproduzir o efeito cascata do ataque, causando indisponibilidade tanto no servidor DNS recursivo quanto no autoritativo. Por fim, no terceiro cenário, comparações com as ferramentas DNSWaterTorture e dns-flooding comprovaram a superioridade da Eris em termos de vazão de pacotes, tráfego válido e personalização. No entanto, os cenários de teste evidenciaram também os futuros pontos de trabalho da ferramenta, como tornar o padrão do ataque menos previsível, dificultando assim a sua detecção por ferramentas de segurança.

Em conclusão, a Eris se estabeleceu como uma ferramenta eficaz e versátil para o estudo de resiliência em servidores DNS, podendo vir a contribuir significativamente para o aprimoramento de estratégias de mitigação contra ataques desse gênero. Desenvolvimen-

tos futuros podem expandir suas funcionalidades, incorporando novos tipos de ataques DNS e opções adicionais de configuração como ataque simultâneo a domínios diferentes, por exemplo, consolidando-a ainda mais como uma ferramenta aliada para teste de resiliência de servidores DNS.

Referências

[1] Khormali, Aminollah, Jeman Park, Hisham Alasmary, Afsah Anwar e David Mohaisen: *Domain name system security and privacy: A contemporary survey*. arXiv preprint arXiv:2006.15277, June 2020. https://arxiv.org/abs/2006.15277, Preprint. viii, 6

[2] Akamai: *DNS Reflection vs. DNS Mirai: Technical Publication*. Technical report, Akamai Technologies, s.d. https://www.akamai.com/site/en/documents/research-paper/dns-reflection-vs-dns-mirai-technical-publication.pdf, [Online; accessed 12-January-2025]. viii, 10

[3] CISO Advisor: *Alta de 106volume de ataques DDoS*. Technical report, Ciso Advisor, s.d. https://www.cisoadvisor.com.br/alta-de-106-no-volume-de-ataques-ddos/, [Online; accessed 12-January-2025]. 1

[4] Cloudfare: *Ataques DDoS famosos: os maiores ataques DDoS de todos os tempos*. Technical report, Cloudfare, s.d. https://www.cloudflare.com/pt-br/learning/ddos/famous-ddos-attacks/, [Online; accessed 12-January-2025]. 1

[5] Camargo, Camila Imbuzeiro: *Mirai: um estudo sobre botnets de dispositivos iot*. Trabalho de conclusão de curso (bacharelado em ciência da computação), Universidade de Brasília, Brasília, Brasil, 2018. http://bdm.unb.br/handle/10483/21936. 1, 8, 11, 12

[6] Vell3, Nicola: *Dnswatertorture: A dns amplification attack tool*, 2023. https://github.com/nicovell3/DNSWaterTorture, Accessed: 2025-01-23. 2, 34

[7] Kowalewski, Mikolaj: *dns-flood-ng: Dns flood testing tool*, 2023. https://github.com/mikolaj-kow/dns-flood-ng, Accessed: 2025-01-23. 2, 34, 41

[8] Brooks, R., Ilker Ozcelik, Lu Yu, Jon Oakley e Nathan Tusing: *Distributed denial of service (ddos): A history*. IEEE Annals of the History of Computing, PP:1–1, abril 2021. 4

[9] Gondim, Joao e Robson Albuquerque: *Mirror saturation in amplified reflection ddos*. Em *Proceedings of the National Computer Network Security Symposium (JNIC)*, June 2019. https://www.researchgate.net/publication/335261132_Mirror_Saturation_in_Amplified_Reflection_DDoS, Conference Paper. 5

[10] Zhijun, Wu, Li Wenjing, Liu Liang e Yue Meng: *Low-rate dos attacks, detection, defense, and challenges: A survey.* IEEE Access, PP:1–1, fevereiro 2020. 5

[11] Akamai: *O que é DNS? | Como funciona o DNS*, s.d. `https://www.akamai.com/pt/glossary/what-is-dns`, [Online; accessed 22-January-2025]. 5

[12] Cloudflare: *O que é DNS? | Como funciona o DNS*, s.d. `https://www.cloudflare.com/pt-br/learning/dns/what-is-dns/`, [Online; accessed 22-January-2025]. 5

[13] Liu, Cricket e Paul Albitz: *How does dns work?* Em *DNS and BIND*, capítulo 2, páginas 11–36. O'Reilly Media, Sebastopol, CA, USA, 5th edição, 2006, ISBN 978-0-596-10057-5. 5, 7

[14] Mockapetris, P.: *Domain Names - Implementation and Specification.* Request for Comments (RFC) 1035, 1987. `https://www.rfc-editor.org/rfc/rfc1035.html`, Accessed: 2025-02-02. 7

[15] Gamblin, J.: *Mirai source code.* `https://github.com/jgamblin/Mirai-Source-Code`, 2016. Accessed: 2025-01-11. 8, 12

[16] Hesiod: *Theogony, Works and Days.* Oxford World's Classics. Oxford University Press, Oxford, 1988. Translated and edited by M. L. West. 12

[17] Silva, Eduardo Sousa da, Paulo Mauricio Costa Lopes e Joao Gondim: *Dambuster: Uma ferramenta de avaliação de soluções de mitigação de dos volumétrico direto.* Em *Anais Estendidos do Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg)*, September 2023. 12

[18] Dantas, Amanda, Matheus Vieira, Alan Vasques e João Gondim: *Linderhof: uma ferramenta para avaliação de sistemas de mitigação de ataques reflexivos volumétricos (ddos).* Em *Anais Estendidos do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, páginas 25–32, Porto Alegre, RS, Brasil, 2020. SBC. `https://sol.sbc.org.br/index.php/sbrc_estendido/article/view/12398`. 12

[19] GeeksforGeeks: *Dns message format*, 2023. `https://www.geeksforgeeks.org/dns-message-format/`, Accessed: 2025-01-11. 12

[20] Huawei Technologies Co., Ltd: *DNS*, 2023. `https://support.huawei.com/enterprise/en/doc/EDOC1100174721/f917b5d7/dns`, Accessed: 2025-01-23. 20

[21] Team, Wireshark: *Wireshark User's Guide.* Wireshark Foundation, 2023. `https://www.wireshark.org/docs/wsug_html_chunked/`, Acesso em: 10 out. 2023. 26, 31

[22] NumPy Developers: *NumPy Documentation*, 2022. `https://numpy.org/doc/2.2/user/index.html`, Accessed: 2025-02-03. 27

[23] Matplotlib Developers: *Matplotlib Documentation*, 2022. `https://matplotlib.org/stable/users/index.html`, Accessed: 2025-02-03. 27

[24] Internet Systems Consortium (ISC): *BIND 9 Documentation*, 2025. `https://bind9.readthedocs.io/`, Accessed: 2025-01-23. 28, 29

[25] Gondim, João José Costa e Robson de Oliveira Albuquerque: *Reflector saturation in amplified reflection denial of service attack abusing cldap and memcache protocols.* Em Guarda, Teresa, Filipe Portela e Jose Maria Diaz-Nafria (editores): *Advanced Research in Technologies, Information, Innovation and Sustainability*, páginas 248–263, Cham, 2024. Springer Nature Switzerland, ISBN 978-3-031-48855-9. 45

Apêndice A

Script executado pelo usuário para realizar consultas DNS

```python
import sys
import time
import subprocess

def main():
    if len(sys.argv) < 4:
        print('missing args')
        sys.exit(1)

    input_server = sys.argv[1]
    domain = sys.argv[2]

    try:
        interval = float(sys.argv[3])
    except:
        print('cannot parse interval')
        sys.exit(1)

    command = f"dig @{input_server} {domain}"

    while True:
        # subprocess.run(command, shell=True)
        try:
            subprocess.Popen(command.split())
        except Exception as e:
            print(e)
            print('failed to start command')

        time.sleep(interval)

if __name__ == "__main__":
```