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

Ataque de negação de serviço por reflexão amplificada sobre CLDAP utilizando a ferramenta Linderhof

Matheus de Oliveira Vieira

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Orientador
Prof. Dr. João José Costa Gondim

Brasília
2021

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

Ataque de negação de serviço por reflexão amplificada sobre CLDAP utilizando a ferramenta Linderhof

Matheus de Oliveira Vieira

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

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

Prof. Dr. Ricardo Jacobi
CIC/UnB

Prof. Dr. Robson de Oliveira Albuquerque
ENE/UnB

Prof. Dr. João José Costa Gondim
Coordenador do Curso de Engenharia da Computação

Brasília, 28 de maio de 2021

Dedicatória

Dedico este trabalho à minha família que sempre esteve ao meu lado fornecendo todo o apoio de que eu precisava.

Agradecimentos

Agradeço a todos que contribuiíram de alguma forma para o desenvolvimento desta monografia, em especial:

Agradeço ao professor Dr. Gondim pela orientação e todo conhecimento transmitido.

Agradeço também ao Alan Tamer Vasques pelas sugestões e testes realizados neste trabalho.

Agradeço à Amanda Lopes Dantas por toda a ajuda fornecida não somente durante o desenvolvimento desta monografia, mas também durante toda a minha graduação.

Agradeço a todos os meus professores e colegas de curso da Universidade de Brasília pelos ensinamentos e valiosas lições.

Agradeço à Alda de Oliveira Vieira, Álvaro Martins Vieira e Érica de Oliveira Vieira por todo o carinho, apoio e incentivo que me deram durante toda a minha jornada.

Agradeço ainda à GigaCandanga por hospedar o Laboratório Virtual de Segurança Cibernética, onde este trabalho se desenvolveu, e à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo Acesso ao Portal de Periódicos.

Resumo

Ataques de Negação de Serviço (DoS), em particular os Ataques Distribuídos de Negação de Serviço por Reflexão Amplificada (AR-DDoS), vêm se mostrando uma ameaça constante a segurança dos negócios disponíveis na internet. Impulsionados pelo aumento de dispositivos de Internet das Coisas (IoT), que possuem baixo poder computacional e pouca segurança, ataques AR-DDoS têm quebrando recordes de tráfego gerado na rede. Devido a isso, criou-se a ferramenta Linderhof, capaz de gerar ataques de negação de serviço por reflexão amplificada de forma controlada, com o objetivo de permitir o estudo desses ataques e mensurar sua ameaça. Este trabalho se propõe a implementar melhorias na ferramenta permitindo que diferentes características dos ataque AR-DDoS sejam estudados, tais como adicionar uma forma de alterar parâmetros dos protocolos em tempo de execução, permitir a customização da taxa do ataque, adicionar a possibilidade de definir a vítima como sendo um bloco de endereços, salvar as configurações em um arquivo, adicionar uma interface gráfica, implementar a funcionalidade de encontrar refletores e implementar o ataque nos protocolos SNMP e CLDAP. Assim como os trabalhos anteriores que vêm realizando estudos com o auxílio dessa ferramenta, este trabalho apresenta um estudo específico da utilização do protocolo CLDAP para a realização de tais ataques. Os resultados obtidos estão de acordo com os estudos anteriores, demonstrando a saturação dos dispositivos que agem como refletores em baixas taxas de injeção de pacotes.

Palavras-chave: negação de serviço, reflexão, amplificação, cldap

Abstract

Denial of Service Attacks (DoS), in particular the Amplified Reflection Distributed Denial of Service Attacks (AR-DDoS), have been a constant threat to the security of businesses available on the Internet. Driven by the rise of Internet of Things (IoT) devices, which have low computing power and little security, AR-DDoS attacks have been breaking records of traffic generated on the network. Because of this, the tool named Linderhof was created, capable of generating denial of service attacks by amplified reflection in a controlled manner, with the objective of allowing the study of these attacks and measuring their threat level. This work proposes to implement improvements in the tool allowing different characteristics of AR-DDoS attacks to be studied, such as adding a way to change protocol parameters at runtime, allowing customization of the attack rate, adding a possibility to define the victim as an address block, save the settings in a file, add a graphical interface, implement the functionality of finding reflectors and implement the attack in the SNMP and CLDAP protocols. Like previous works that have been carrying out studies with the aid of this tool, this work presents a specific study of the use of the CLDAP protocol to carry out such attacks. The results obtained are compatible with previous studies, demonstrating the saturation of devices that act as reflectors at low packet injection rates.

Keywords: denial of service, reflection, amplificatipn, cldap

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

AR-DDoS Amplified Reflection Distributed Denial of Service.

CIDR Classless Inter-Domain Routing.

CLDAP Connection-less Lightweight Directory Access Protocol.

CoAP Constrained Application Protocol.

DAP Directory Access Protocol.

DDoS Distributed Denial of Service.

DIT Directory Information Tree.

DNS Domain Name System.

DoS Denial of service.

GUI Graphical User Interface.

HOM Hall of Mirrors.

IoT Internet of Things.

LDAP Lightweight Directory Access Protocol.

NTP Network Time Protocol.

SNMP Simple Network Management Protocol.

SSDP Simple Service Discovery Protocol.

Capítulo 1

Introdução

Ataques de negação de serviço (DoS, Denial of service) estão em constante evolução, seja com o aumento em seu volume quanto com o abuso de novos protocolos e estratégias. Em 2020 a AWS sofreu um dos maiores ataques já observados, com um pico de 2.3 Tbps de tráfego gerado, 44% maior que qualquer outro ataque sofrido pela plataforma [3]. Nesse mesmo ano o Google notificou ter sofrido um ataque ainda maior já em 2017, com um pico de 2.5 Tbps [4].

Com o aumento de dispositivos conectados à internet, como dispositivos de Internet das Coisas (IoT, Internet of Things), tornou-se possível a criação de grandes redes de dispositivos infectados, as chamadas *botnets*, que podem ser utilizadas para gerar ataques com grande volume de tráfego. Esse problema tornou-se evidente em 2016 com o surgimento do Mirai [5] [6] [7], um *malware* que infecta dispositivos IoT, como roteadores e câmeras de segurança que estão expostos na internet com pouca ou nenhum proteção, e os coordena para que contribuam em ataques DDoS [8]. Utilizando um pequeno dicionário de combinações de logins e senhas, estima-se que o Mirai infectou mais de 600 mil dispositivos em 2016, sendo 65 mil infectados somente nas primeiras 20 horas de busca [9].

Ataques como estes visam a indisponibilização dos serviços oferecidos pelas vítimas, e com isso são capazes de trazer diversas consequências para elas como prejuízo financeiro, dano à reputação da empresa e perda de seus clientes [10].

Com o intuito de estudar este tema, foi desenvolvida uma ferramenta chamada Linderhof, capaz de realizar ataques de negação de serviço por reflexão amplificada (AR-DDoS, Amplified Reflection Distributed Denial of Service), que se utilizam de diversos protocolos que permitem a ampliação do ataque. A finalidade da ferramenta é a avaliação de soluções de mitigação de ataques AR-DDoS. Com a ajuda desta ferramenta testou-se a utilização de alguns protocolos para a realização de ataques, como o DNS [11], NTP [12], Memcached [13] e CoAP [14]. Nesse ponto do desenvolvimento, havia para cada um

dos protocolos citados uma versão executável, com todas elas compartilhando um núcleo comum.

Houve um esforço de unificação dessas versões em uma única. Assim, uma versão da ferramenta foi desenvolvida incrementalmente sobre uma arquitetura modular totalmente revista e provendo todos os protocolos já disponibilizados. Esse esforço foi consolidado em uma primeira versão da ferramenta publicada em [15] e registrada. Até então, o Linderhof possuía as funcionalidades de escolher o protocolo, os endereços da vítima e refletor, assim como suas portas, o nível e duração do ataque e permitia utilizar um modo incremental onde o nível é elevado constantemente.

Conforme exposto, a referida versão o Linderhof oferecia diversas oportunidades de melhoria. A ferramenta possui somente dois modos de ataque, um com taxa contínua e o outro incremental, além de não permitir alterações nos pacotes criados, diminuindo a quantidade de testes que podem sem realizados, as chances de se obter melhores amplificações e aumentando a facilidade de mitigação. O software pode ser utilizando somente através de linha de comando, dificultando sua usabilidade e exigindo a inserção das configurações desejadas a cada execução. Além disso, apesar de possuir vários protocolos implementados, ainda não possui alguns muito utilizados na atualidade, como o SNMP e o CLDAP.

1.1 Objetivos

O objetivo geral deste trabalho é implementar melhorias na ferramenta Linderhof, tanto de funcionalidades quanto de usabilidade, e assim consolidar sua versão 2.0.0.

Especificamente, as seguintes funcionalidades serão acrescentadas:

- possibilidade de alterar parâmetros dos protocolos em tempo de execução;
- customização da taxa do ataque;
- definição do alvo como sendo um bloco de endereços;
- salvamento das configurações em um arquivo para que possam ser reutilizadas;
- implementação de uma interface gráfica para facilitar o uso da aplicação;
- busca na rede por máquinas que poderiam ser utilizadas como refletores;
- implementação dos ataques com os protocolos SNMP e CLDAP.

1.2 Contribuições

As contribuições deste trabalho se deram em duas fases. Na primeira, foi consolidada a versão 1.0.0 da ferramenta Linderhof que gerou um registro de software, um artigo [15] e foi utilizada em [16], [17] e [2]. Na segunda fase, foram implementadas as melhorias na ferramenta segundo a seção 1.1 gerando a versão 2.0.0.

Apesar do autor ter participado das duas fases, este documento descreve apenas as contribuições da segunda fase, estando os da primeira incluídas no Apêndice A, onde está o artigo [15]. O certificado de registro de software está disponível no Apêndice B.

O Linderhof está sob guarda do autor e do orientador deste trabalho, podendo ser disponibilizada mediante autorização para fins de pesquisa.

1.3 Organização deste documento

Conforme dito anteriormente, este documento descreve as contribuições da segunda fase e se inicia com a conceituação dos ataques de negação de serviço, como suas características, efeitos e algumas das técnicas desenvolvidas para burlar as formas de mitigação. E então será apresentado as características do CLDAP, protocolo que se pretende adicionar à ferramenta.

No terceiro capítulo apresenta-se a ferramenta Linderhof, sua arquitetura, funcionalidades, bem como a descrição e funcionamento das melhorias incluídas na versão 2.0.0. Em seguida, descreve-se as características do protocolo CLDAP que permite que ele seja utilizado como refletor em ataques DoS.

No quarto capítulo são demonstradas as funcionalidades inseridas à ferramenta. Além disso, testa-se o protocolo implementado em ataques realizados em ambiente controlado para se obter suas eficiência em ataques reais.

Ao final conclui-se que o objetivo de amadurecer a ferramenta Linderhof foi alcançado dado o correto funcionamento das melhorias implementadas.

Capítulo 2

Referencial teórico

Nesse capítulo apresentamos os principais conceitos utilizados para o desenvolvimento deste trabalho.

2.1 Ataque de negação de serviço

Um Ataque de Negação de Serviço (DoS, Denial of service) busca esgotar os recursos computacionais de um serviço e com isso impedir o acesso de usuários legítimos a ele [18]. Isso acontece ao enviar um grande volume de pacotes para a vítima, mais do que ela consegue suportar, até que ela não consiga responder as requisições como esperado, impedindo o correto funcionamento do serviço.

A evolução desse ataque chama-se Ataque Distribuído de Negação de Serviço (DDoS, Distributed Denial of Service), e tem como novidade a participação de várias máquinas no ataque, gerando um tráfego ainda maior e dificultando sua rastreabilidade [19]. Essa evolução tornou-se necessária no momento em que o poder de processamento e a infraestrutura das vítimas foram evoluindo, exigindo que os atacantes acompanhem essa evolução.

As máquinas que passaram a ser utilizadas nos ataques DDoS são chamadas de *bots*, e quando fazem parte de um grande grupo de outros *bots* que exercem a mesma função é dito que este dispositivo integra uma *botnet* [20]. São dispositivos infectados por um *malware* que faz com que sejam controlados por uma outra máquina que as coordena, apontando quais devem ser as vítimas e quando o ataque deve acontecer.

Segundo [1], os ataques DDoS podem ser classificados em diversos grupos, como mostrados na Figura 2.1.

A primeira das duas principais classificações são os ataques de **baixo volume**. Eles utilizam-se de falhas nos protocolos ou implementações para confundir e prejudicar as vítimas. Os ataques podem abusar tanto do funcionamento dos protocolos, como por

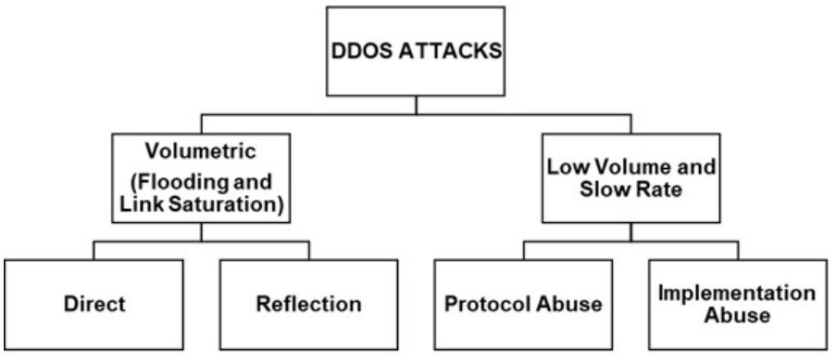

Figura 2.1: Taxonomia dos ataques DDoS.

exemplo o *handshake* do TCP, ou de implementações de protocolos ou serviços que não previram, por exemplo, a utilização de pacotes mal formados, que ao serem recebidos podem causar danos ao funcionamento dos serviços.

A outra classificação dos ataques são os **volumétricos**. Eles baseiam-se em enviar uma grande quantidade de tráfego para a vítima até que sua largura de banda se esgote, dificultando que usuários legítimos acessem os recursos desejados.

Dentro desse grupo há uma divisão entre os subgrupos de ataques diretos e por reflexão. Nos ataques **diretos** os envolvidos enviam o tráfego diretamente para a vítima, sem passar por intermediários. Opcionalmente, os atacantes podem se utilizar de uma técnica chamada mascaramento de endereço IP (*IP Spoofing*), onde é modificado o endereço de origem nos cabeçalhos IP dos pacotes com o objetivo de dificultar a rastreabilidade da origem do ataque.

Outro subgrupo é o dos ataques por **reflexão**, onde diferentemente dos ataques diretos, os pacotes são enviados para um grupo intermediário de dispositivos. Esse tipo de ataque utiliza-se do *IP Spoofing* para inserir o endereço da vítima no campo do endereço de origem dos pacotes, fazendo com que os dispositivos intermediários enviem suas respostas à vítima, acreditando que o tráfego recebido tenha se originado dela.

Esses dispositivos intermediários são chamados de refletores [21], e além de serem utilizados para dificultar a identificação dos ataques, também podem ser usados para amplificar o tráfego enviado às vítimas. Quando isso acontece, o ataque é chamado de Ataque de Negação de Serviço por Reflexão Amplificada (AR-DDoS). O funcionamento de um ataque como esse é demonstrado na Figura 2.2.

A principal característica de um dispositivo refletor é responder aos pacotes recebidos com uma resposta que seja maior que o pacote de origem. Muitas vezes isso é possível devido aos protocolos da camada de aplicação que permite que sejam enviadas respostas a solicitações sem a necessidade de autenticação, seja por característica do protocolo ou seja

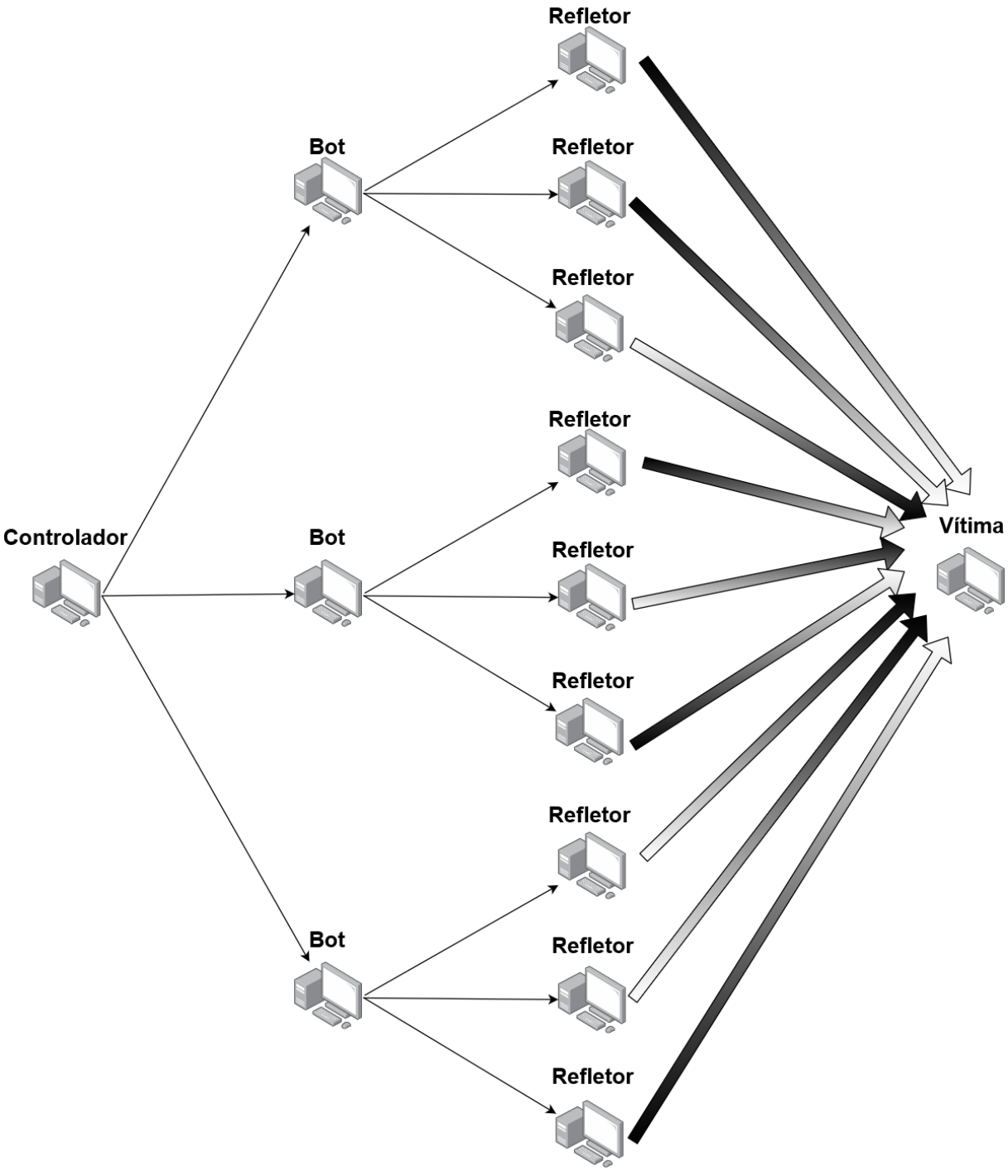

Figura 2.2: Exemplo do funcionamento de um ataque AR-DDoS.

por uma má configuração do serviço. Sendo assim, é interessante observar que num ataque AR-DDoS não é necessário invadir os refletores e implantar nenhum *malware*. É necessário apenas que eles possuam uma configuração do sistema que permita amplificação e que estejam numa rede que aceite *IP Spoofing* [22]. Assim, o esforço de preparação por parte do atacante é muito menor do que aquele necessário em ataques que utilizam somente *botnets*.

2.1.1 Carpet Bombing

Uma das melhorias incluídas no Linderhof é a possibilidade de execução do ataque utilizando a técnica do *Carpet Bombing*. O *Carpet Bombing* é uma nova variante do DDoS que consiste em atacar um bloco de endereços IP ao invés de um único endereço [23]. Segundo [24], o *Carpet Bombing* dificulta a mitigação pois muitas dessas técnicas envolvem a detecção de grandes volumes de ataque acima de um certo limiar em um hospedeiro específico, o que não acontece ao dividir o ataque entre diversos hospedeiros de uma rede. A distinção do *Carpet Bombing* para um ataque comum é demonstrada na Figura 2.3. Como se vê, ao invés do tráfego saturar apenas sobre o endereço do alvo, ele se espalha pela faixa de endereçamento da rede ande ele está.

Uma das consequências do uso do *Carpet Bombing* é que mitigações mais simples tornam-se ineficazes, como o *black-hole routing*, onde parte da rede têm seu tráfego descartado para se evitar o ataque. Essa técnica torna-se impotente no momento em que seria necessário descartar o tráfego de grande parte da rede, ou até mesmo de toda ela devido ao tráfego chegar em múltiplos endereços [25].

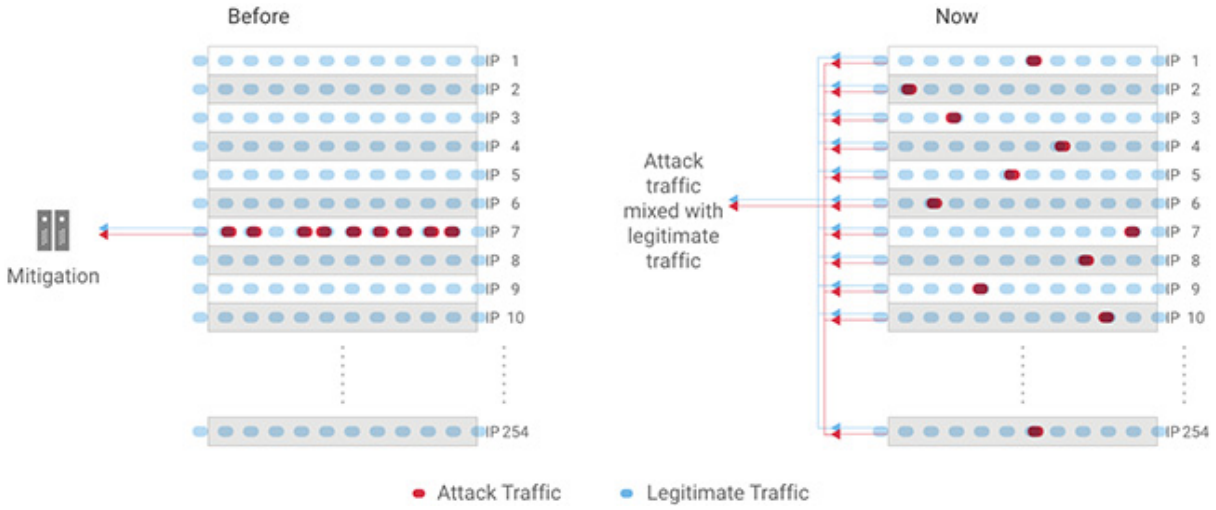

Figura 2.3: Comparação entre ataques comuns e o Carpet Bombing (Fonte: [26]).

2.1.2 Pulse Wave

Outra técnica que passa a ser possível de utilizar devido as melhorias feitas é o *Pulse Wave*. Trata-se de uma técnica utilizada para burlar sistemas de mitigação, que segundo [27], consiste em enviar pulsos curtos de tráfego para a vítima que se repetem em intervalos regulares.

Esse método é eficiente contra sistemas de mitigação híbridos, que se baseiam tanto em hardware quanto em nuvem. Esses sistemas possuem hardwares de mitigação que agem como a linha de frente diante de ataques. Ao atingir o limite do dispositivo, todo o tráfego é redirecionado para a nuvem. Essa transição leva alguns minutos para completar.

Diferentemente de um ataque normal onde sua intensidade aumenta ao longo do tempo, o *Pulse Wave* caracteriza-se por enviar pulsos de grande intensidade com um determinada frequência. O objetivo é ativar essa transição para a nuvem, tornando o serviço oferecido pela vítima indisponível por algum tempo, para então encerrar o pulso. Após um tempo suficiente para o serviço de mitigação retornar para sua versão local, é enviado um novo pulso, reiniciando o ciclo [28]. Esse ataque explora portanto os tempos de convergência do roteamento quando é utilizado uma mitigação hibrida com o que é chamado de *appliance first* em conjunto com o processamento na nuvem. O gráfico de um ataque como esse é mostrado na Figura 2.4.

Figura 2.4: Exemplo de um ataque utilizando o Pulse Wave (Fonte: [29]).

2.2 Protocolo CLDAP

O CLDAP (Connection-less Lightweight Directory Access Protocol) [30] é um protocolo baseado no LDAP. Ele foi definido com o objetivo de aumentar a performance ao realizar pequenas consultas. O CLDAP permite executar consultas ao diretório sem a necessidade de se estabelecer uma conexão, e para isso ele utiliza o protocolo UDP na porta 389. A característica de não requerer autenticação para responder as requisições é fundamental para sua utilização em ataque DDoS.

Por sua vez, o LDAP (Lightweight Directory Access Protocol) é um protocolo da camada de aplicação que fornece acesso a serviços de diretórios utilizando o modelo cliente-servidor, seguindo o padrão X.500 [31]. O cliente faz uma requisição contendo a operação que se deseja realizar e o servidor executa as operações necessárias para responder a solicitação com os dados requisitados.

Um diretório é um serviço utilizado para armazenar dados de determinados objetos, como pessoas ou computadores. Diferencia-se de um banco de dados por realizar leituras com mais frequência que escritas, portanto é otimizado para tal tarefa [32].

Segundo [31], o modelo que determina o formato das informações contidas em um diretório X.500, é composto de entradas, que por sua vez são compostas de atributos. Cada atributo possui um ou mais valores de determinado tipo. Ainda, as entradas possuem um atributo chamado *ObjectClass*, que determina qual o tipo dessa entrada, definindo assim quais atributos ela possui.

Os dados do diretório são armazenados em uma estrutura de árvore chamada DIT (Directory Information Tree), onde o caminho até um objeto torna-se seu identificador único, chamado de Distinguished Name (DN, Nome Distinto) [33]. O DN por sua vez é composto por uma sequência separada por vírgula de RDNs (Relative Distinguished Name, Nome Distinto Relativo), que são os ramos da árvore que levam o elemento e questão até a raiz da árvore.

A Figura 2.5 apresenta um exemplo de uma árvore DIT. Nesse exemplo, o DN do elemento cujo RDN é `uid=babs` é dado por `uid=babs,ou=People,dc=example,dc=com`.

O LDAP foi desenvolvido como uma alternativa ao DAP, protocolo de comunicação utilizado pelo X.500. Sua principal diferença é rodar em cima do protocolo TCP/IP, diferentemente do OSI utilizado pelo seu antecessor. Isso permite que o protocolo seja mais leve e possa ser rodado em ambientes menos robustos. Além disso foi removido o suporte a alguns comandos pouco utilizados pelo DAP. As operações definidas na RFC4511 [34] estão listadas na Tabela 2.1.

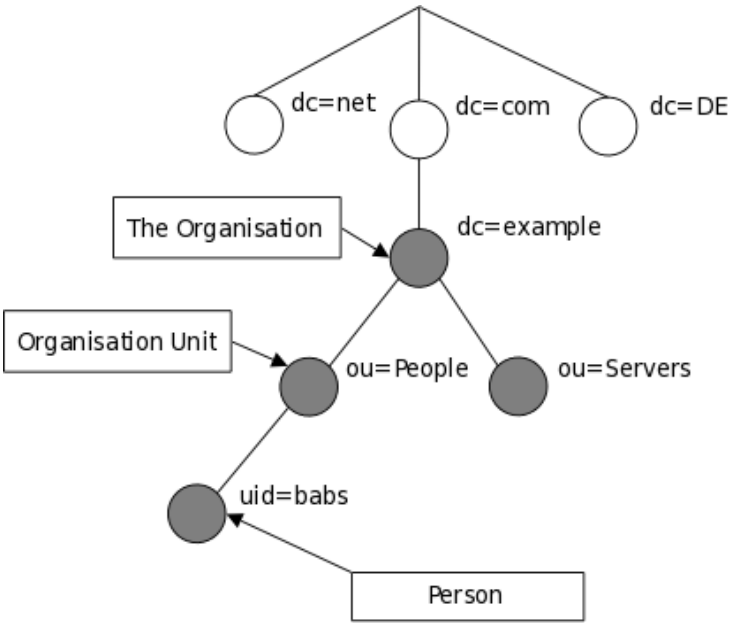

Figura 2.5: Exemplo da estrutura da árvore DIT (Fonte: [29]).

Tabela 2.1: Operações do LDAP.

Operação	**Ação**
Bind	Autenticar usuário
Unbind	Fechar conexão com o servidor
Search	Buscar entradas que correspondem a determinado critério
Modify	Alterar conteúdo de uma entrada
Add	Adicionar uma nova entrada
Delete	Remover uma entrada
Modify DN	Alterar o DN de uma entrada
Compare	Verificar se o atributo de uma entrada possui determinado valor
Abandon	Solicitar ao servidor para parar de processar uma requisição anterior
Extend	Executar uma operação não definida originalmente

Tabela 2.2: Valores dos campos definidos na implementação do ataque sobre CLDAP.

Campo	**Valor**
Operação	searchRequest
Filter	objectClass=*
Attribute	*

2.2.1 Uso do protocolo CLDAP para o ataque AR-DDoS

Para realizar o ataque, os pacotes CLDAP gerados utilizam a operação *searchRequest*, que realiza a requisição por um objeto. No campo *Filter* é utilizado o filtro *objectClass=**, que faz com que os objetos que possuam qualquer classe sejam retornados na pesquisa, que é o caso de todos os objetos da árvore. Por fim, o campo *Attribute* é definido com o valor ***, solicitando que todos os atributos dos objetos encontrados sejam retornados. Essas características visam maximizar o tamanho da resposta dos refletores. Esses campos e seus valores estão resumidos na Tabela 2.2.

2.2.2 Modus operandi do ataque

O funcionamento de um ataque utilizando o CLDAP como protocolo para reflexão segue os seguintes passos:

1. Escanear a rede em busca de dispositivos rodando CLDAP que respondam as requisições enviadas;
2. Criar uma mensagem contendo os campos e valores da Tabela 2.2 a fim de se obter uma maior amplificação;
3. Gerar um cabeçalho UDP com a mensagem gerada no passo anterior e com a porta de destino 389, padrão do protocolo;
4. Montar um cabeçalho IP com o endereço de origem mascarado como o IP da vítima e o endereço de destino como o endereço do refletor;
5. Enviar o pacote gerado para o refletor, que por sua vez responderá a requisição recebida para a vítima;

A busca por refletores da etapa 1 é feita da mesma forma que um ataque, com a diferença que não há o mascaramento da etapa 4. O endereço de origem em um pacote de busca é o endereço do próprio atacante, para que a resposta seja enviada a ele.

2.3 Resumo do capítulo

Este capítulo definiu os conceitos de ataque de negação de serviço e suas evoluções DDoS e AR-DDoS. Foi descrito a taxonomia desses ataques, ou seja, como esses ataques são classificados. Foram citados duas variantes dos ataques DDoS que buscam burlar os métodos de mitigação utilizados atualmente, o *Carpet Bombing* e o *Pulse Wave*.

Em seguida, o protocolo LDAP e sua variante CLDAP foram apresentados, explicando seu funcionamento e operações disponíveis.

Por fim, foi descrita a implementação do protocolo CLDAP feita no Linderhof para a realização de ataques de negação de serviço, mostrando quais características foram utilizadas para se obter a maior eficiência na amplificação dos pacotes recebidos pelo amplificador e um resumo de todas as etapas envolvidas em um ataque.

Capítulo 3

Linderhof

O Linderhof é uma ferramenta desenvolvida com o objetivo de auxiliar os estudos sobre ataques AR-DDoS. Ela é capaz de gerar ataques com taxas controladas de envio de pacotes e sua evolução, além da duração, vítima e refletores do ataque.

Os *mirrors*, como são chamados os protocolos que podem ser explorados nos refletores, podem ser facilmente adicionados à ferramenta com pequenas modificações no código. A ferramenta possui suporte aos *mirrors* CoAP, DNS, Memcached, NTP e SSDP.

3.1 Histórico

O Linderhof é a evolução de uma ferramenta chamada Striker, desenvolvida em 2015 [35]. O Striker possuía suporte somente ao SNMP, porém já possuía uma interface gráfica, dez níveis de ataque e um módulo de *scanner* que realizava uma busca por possíveis refletores.

Em 2018 o Striker foi aprimorado [36], recebendo suporte ao protocolo SSDP e melhorias na interface. Essa versão trouxe aprimoramentos visando facilitar a implementação de novos protocolos na ferramenta.

O Linderhof foi descrito no ano seguinte em [13]. Inicialmente foi adicionado a *engine* de geração de pacotes e a implementação somente do protocolo Memcached, porém a ferramenta foi criada com um foco em ser modular, permitindo com que novos protocolos sejam adicionados com facilidade. Em paralelo, foram desenvolvidos as implementações dos protocolos DNS [11], NTP [12], SSDP e CoAP [14], sendo que para cada um desses protocolos foi gerada uma versão específica do Linderhof.

Essa versão inicial do Linderhof não possuía uma interface gráfica, podendo ser utilizando somente via linha de comando. Além disso ela não possuía a capacidade de escanear a rede em busca de refletores.

Assim inicia-se o esforço de refatoração e unificação das varias versões em uma única ferramenta, culminando no lançamento da versão 1.0.0 do Linderhof [15]. A refatora-

ção foi apresentada em [37], com o objetivo de simplificar sua arquitetura, facilitando seu entendimento e promovendo uma melhor distribuição de responsabilidades, além de padronizar as implementações dos diversos *mirrors*.

3.2 Arquitetura

Como descrito em [37], o Linderhof é composto por 5 módulos: Interface, *Commander*, *Hall of Mirrors*, *Injector* e o *Scanner*. Este último sendo adicionado neste trabalho. A arquitetura pode ser visualizada na Figura 3.1 e os módulos originais são descritos a seguir.

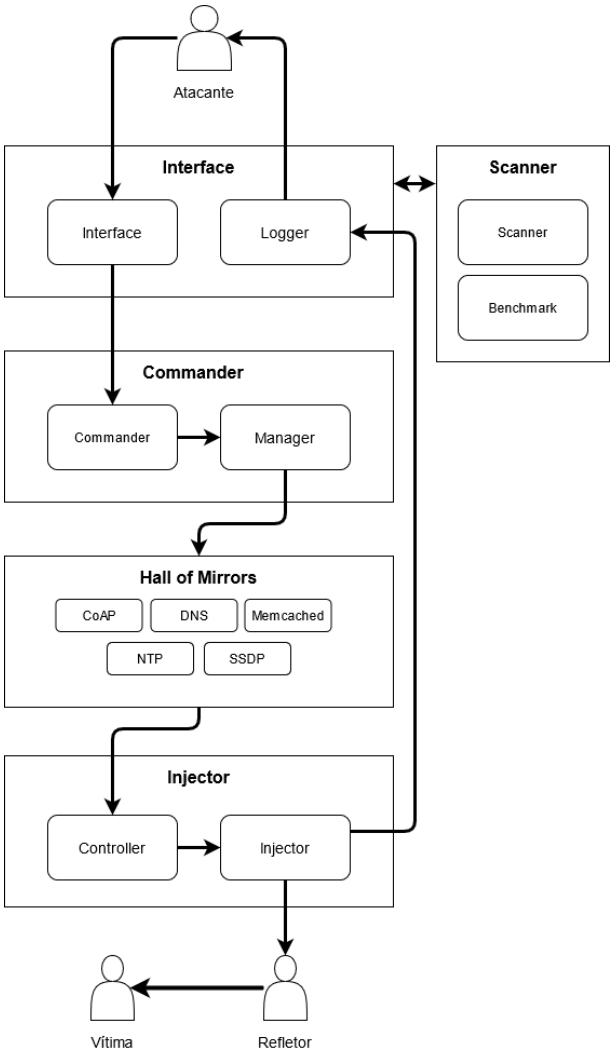

Figura 3.1: Arquitetura do Linderhof.

- **Interface:** Módulo com que o usuário interage. Responsável por receber os parâmetros do ataque e retornar informações do seu andamento através do submódulo *Logger*;
- **Commander:** Responsável por inicializar os sinais de erro da ferramenta e através do submódulo Manager realizar a chamada ao *mirror* definido para o ataque;
- **Hall of Mirrors (HOM):** É onde estão implementadas as funções que geram os pacotes de cada um dos protocolo;
- **Injector:** É o modulo responsável por realizar o envio dos pacotes gerados ao refletor. O submódulo *Controller* define e controla a taxa de injeção de acordo com o nível do ataque.

3.3 Adição de um novo protocolo

Uma das características mais importantes do Linderhof é a sua modularidade. Um dos objetivos iniciais era que ele fosse capaz de agregar facilmente novos protocolos para a realização do ataque.

Sendo assim, na implementação do CLDAP foi necessário adicionar dois novos arquivos no módulo HOM:

- um que monta os dados do protocolo no pacote:
- e outro que define as funções que montam o restante do pacote e chamam o injetor.

Fora esses dois arquivos principais, é necessário adicionar o novo protocolo em algumas estruturas, tais como:

- no módulo de interface onde o nome do protocolo é recebido através da chamada do ataque;
- na interface gráfica onde é necessário adicionar o protocolo na lista de opções disponibilizadas;
- no submódulo *Manager* do *Commander*; e
- nos submódulos *Scanner* e *Becnhmark* do módulo *Scanner*, onde as funções de montagem dos pacotes e execução do ataque definidas no HOM são chamadas.

3.4 Funcionalidades

Os parâmetros aceitos pelo Linderhof enumeram bem as suas funcionalidades. Os parâmetros antigos e os recém adicionados estão listados na Tabela 3.2.

Tabela 3.1: Níveis de ataque.

Nível	**Taxa (pacotes / s)**	**Taxa (pacotes / s)**
1	10^0	1
2	10^1	10
3	10^2	100
4	10^3	1 mil
5	10^4	10 mil
6	10^5	100 mil
7	10^6	1 milhões
8	10^7	10 milhões
9	10^8	100 milhões
10	10^9	1 bilhão

O parâmetro *mirror* deve ser preenchido com o nomo do protocolo a ser utilizado no ataque, podendo ser os valores COAP, DNS, MEMCACHED_GETSET, MEMCACHED_STAT, NTP, SNMP ou SSDP. Esse parâmetro é obrigatório e deve ser informado em todas as execuções da ferramenta.

O parâmetro *target* também é obrigatório e informa o endereço IP da vítima do ataque. Opcionalmente é possível informar a porta por onde a vítima receberá os pacotes, sendo escolhida uma aleatoriamente caso o valor não seja explicitado.

Os endereços dos refletores podem ser informados através do parâmetro *reflector*, sendo possível inserir um único refletor ou um arquivo de texto que contenha uma lista de refletores, com um endereço IP em cada linha. A porta dos refletores pode ser informada através do *reflecport*. Caso a porta não seja informada, será utilizada a porta padrão utilizada pelo protocolo e definida em sua implementação.

O nível do ataque é definido através do parâmetro *level*, tendo o valor 1 por padrão. O nível define a quantidade de pacotes que serão enviados pelo atacante a cada segundo durante a duração do ataque. Os níveis variam de 1 a 10 e a taxa de injeção é definida pela Equação 3.1. Logo, a quantidade de pacotes enviados por segundo em cada nível é listada na Tabela 3.1.

$$Taxa = 10^{nível-1} \tag{3.1}$$

Por padrão, a quantidade de pacotes definidas pelo nível para serem enviadas é dividida entre todos os refletores informados. Caso haja sobra de pacotes, o restante é divididos entre os primeiros refletores listados.

A duração do ataque, especificada pelo parâmetro *duration*, define o tempo, em segundos, que o ataque ficará ativo. Caso não seja informado um valor, o ataque terá duração indeterminada.

O parâmetro *inc* ativa o modo incremental de ataque. Nesse modo, os níveis são incrementados em uma unidade a cada valor especificado pelo parâmetro *inc*. Por exemplo, definindo o valor 5 para o parâmetro, o ataque aumentará sua taxa de injeção a cada 5 segundos até chegar no nível máximo ou acabar a duração do ataque.

Outro modo de ataque é *flooding*, ativado ao utilizar o parâmetro *flood*, que não requer nenhum valor. Nesse modo, o Linderhof envia o máximo de pacotes possível, sem se preocupar com as limitações impostas pelos níveis.

O último modo de ataque da ferramenta é o agressivo, ativado através do parâmetro *aggressive*, também sem valor. Ao ativar esse modo, a quantidade total de pacotes a ser injetada, seguindo o nível especificado, é enviado para cada refletor, não havendo, portanto, uma divisão como ocorre no modo normal. Sendo assim, num cenário onde foi definido um ataque de nível 3 com 4 refletores, por exemplo, cada um dos refletores receberia 100 pacotes ao invés dos 25 do modo normal.

O parâmetro *config* recebe o caminho de um arquivo de texto contendo os parâmetros que se deseja utilizar no ataque. Com isso, o usuário pode salvar as configurações de determinados ataques em arquivos separados e reutilizá-los, sem ser preciso digitar todos os parâmetros na linha de comando. Ao utilizar o parâmetro *config* sem especificar um arquivo, a ferramenta buscará um arquivo chamado *linderhof.conf* no diretório atual e o utilizará caso seja encontrado. A Figura 3.2 mostra um exemplo de um arquivo válido.

Por fim, existem os parâmetros que são específicos de cada protocolo, permitindo alterar campos dos seus pacotes gerados. O DNS possui o parâmetro *domain-name* que define o domínio a ser buscado nas requisições. O SSDP possui o *upnp-version*, que informa a versão do UPnP a ser utilizada e o *unicast*, que define o campo *host* para endereço unicast. O SNMP define os parâmetros *community-string* e *max-repetitions*, onde o primeiro funciona como uma senha do protocolo e o segundo está ligado ao tamanho do pacote que será enviado pelo refletor. Por último, o CoAP possui os campos *szx* e *uri-path*, informando o valor do campo *szx* no pacote e o caminho do recurso a ser consultado, respectivamente.

A tabela Tabela 3.2 sintetiza as informações referentes ao parâmetros da ferramenta e inclui os novos parâmetros incluídos por este trabalho e que serão apresentados na seção 3.5.

Tabela 3.2: Parâmetros aceitos pelo Linderhof.

Parâmetro	**Descrição**	**Argumento**	**Novo**
mirror	Nome do mirror	Obrigatório	Não
target	Endereços IP das vítimas	Obrigatório	Não
reflector	Endereços IP dos refletores	Obrigatório	Não
reflecport	Porta do refletor	Obrigatório	Não
targport	Porta da vítima	Obrigatório	Não
level	Nível do ataque	Obrigatório	Não
duration	Duração do ataque (em segundos)	Obrigatório	Não
inc	Intervalo entre incrementos do nível do ataque	Obrigatório	Não
help	Mostrar ajuda	Não	Sim
flood	Ativar modo flooding	Não	Sim
rate	Arquivo contendo as taxas do ataque	Obrigatório	Sim
aggressive	Ativar modo agressivo	Não	Sim
scanner-cidr	Buscar por refletores em determinada sub-rede	Obrigatório	Sim
scanner-path	Arquivo para salvar refletores encontrados	Obrigatório	Sim
benchmark	Testar parâmetros do mirrors	Não	Sim
shuffle	Aleatorizar ordem das vítimas	Não	Sim
config	Utilizar arquivo de configuração	Opcional	Sim
domain-name	DNS - Domínio	Obrigatório	Sim
upnp-version	SSDP - Versão do UPnP	Obrigatório	Sim
unicast	SSDP - Definir campo host para endereço unicast	Não	Sim
community-string	SNMP - Campo community string	Obrigatório	Sim
max-repetitions	SNMP - Campo max-repetitions	Obrigatório	Sim
szx	CoAP - Campo SZX (0-7)	Obrigatório	Sim
uri-path	CoAP - Campo uri-path	Obrigatório	Sim

3.5 Novas Funcionalidades

3.5.1 IPv6

A fim de estudar as diferenças na utilização dos protocolos IPv4 e IPv6 para ataques de negação de serviço, adicionou-se suporte ao protocolo IPv6. Essa funcionalidade aumenta a compatibilidade com diversos refletores e vítimas que utilizam-se desse protocolo. Testes e comparação entre os ataques utilizando os protocolos IPv4 e IPv6 foram descritos em [2].

3.5.2 Taxa customizável

Ataques DDoS podem ser executados com diferentes padrões de taxas além da taxa constante e por degraus, que são suportadas pelo Linderhof. Com isso em mente adicionou-se a funcionalidade que permite a customização dessa taxa ao longo do ataque, gerando diferentes formas de onda de ataque. Para isso, basta inserir no parâmetro rate o caminho de um arquivo que contenha um número inteiro por linha representando a quantidade de pacotes a serem enviados a cada segundo. Ao chegar no final do arquivo a leitura e reiniciada e o padrão se repete. Com isso torna-se possível utilizar padrões diversos como por exemplo uma onda quadrada, triangular ou senoidal.

Essa funcionalidade permite que seja executado ataques do tipo *Pulse Wave*, onde para isso basta informar a quantidade de pacotes que se deseja enviar durante o pico da onda quadrada e repetir esse valor tantas vezes quanto se queira que a taxa se mantenha nesse valor, e repetir o valor 0 tantas vezes quanto se queira que o ataque fique em repouso. Caso não seja definida uma duração para o ataque, esse padrão se repetirá indefinidamente.

3.5.3 Ataque em múltiplos endereços na sub-rede

Para permitir que a técnica *Carpet Bombing* seja utilizada adicionou-se suporte a endereços de sub-rede no campo do endereço da vítima. Com isso, a ferramenta passou a ter suporte a múltiplas vítimas.

Ao utilizar a notação CIDR (Classless Inter-Domain Routing) [38], o Linderhof a converte em um intervalo de endereços e circula por esses endereços ao longo do ataque, utilizando um endereço diferente a cada segundo.

Adicionou-se também o parâmetro *shuffle*, que faz com que os endereços sejam utilizados de forma aleatória, aumentando a dificuldade de se detectar o ataque.

3.5.4 Scanner

A implementação da interpretação de endereços CIDR tornou possível desenvolver a funcionalidade de *scanner*, que busca em uma sub-rede por dispositivos que executam os protocolos implementados na ferramenta de forma que possam ser utilizados como refletores.

Como mostrado na Figura 3.1, o *scanner* foi desenvolvido em um módulo separado do restante da arquitetura, comunicando-se somente com o módulo de interface. Ele se difere de um ataque pela baixa quantidade de pacotes enviados e por não precisar fazer o *IP spoofing*, já que o próprio atacante precisa receber a resposta dos dispositivos encontrados.

Também foi adicionado ao *scanner* o submódulo *benchmark*, que executa um teste de performance dos protocolos, verificando a amplificação que determinado protocolo é capaz de realizar. Para isso foi preciso alterar cada um dos *mirrors* do módulo *Hall of Mirrors* para adicionar os comandos de geração de pacotes sem o *IP spoofing* e os cenários que serão testados no *benchmark*. No caso do SNMP, por exemplo, foi adicionada uma função para encontrar o valor ideal para o campo *max-repetitions* a modo de se obter a maior amplificação.

O *scanner* é executado em um CIDR definido no parâmetro *scanner-cidr* e os refletores encontrados são então salvos em um arquivo de texto especificado em no campo *scanner-path*, o que permite que ele seja utilizado como o parâmetro *reflector* posteriormente.

3.5.5 Arquivo de configuração

Para facilitar a utilização da ferramenta foi adicionado um submódulo chamado *Config-Parser* dento do módulo Interface, que permite que as configurações utilizadas no ataque sejam lidas de um arquivo de configuração. Isso permite com que não seja necessário preencher os parâmetros a cada novo ataque. Além disso, permite também salvar as configurações desejadas em arquivos diferentes para serem reutilizadas com maior rapidez.

O parâmetro *config* deve conter o caminho do arquivo de configuração desejado. Caso seja utilizado esse parâmetro sem nenhum valor, a ferramenta buscará pelo arquivo de configuração padrão na raiz na aplicação, chamado de *linderhof.conf*, e o utilizará caso seja encontrado. A Figura 3.2 mostra um exemplo de um arquivo válido.

3.5.6 Interface gráfica

Com o objetivo de melhorar a usabilidade da aplicação foi desenvolvida uma interface gráfica (GUI, Graphical User Interface). A GUI foi feita com o conjunto de bibliotecas do *GTK* 3.22 [39] juntamente com o software *Glade* na versão 3.38.2 [40]. Ela foi escrita em C, assim como o restante do Linderhof.

linderhof.conf

```
[General]
# Mirror type (string)
MIRROR=DNS
# Target IP (string)
TARGET=192.168.1.2
# Target port (integer)
TARGET_PORT=
# Reflector IP (string)
REFLECTOR=192.168.1.1
# Reflector port (integer)
REFLECTOR_PORT=
# Attack level (integer)
LEVEL=1
# Attack duration (integer)
DURATION=3
# Increment attack delay (integer)
INCREMENT=
# File containing rates at which packets should be sent (string)
CUSTOM_RATE=rate-sample.txt
# Set aggressive mode on (boolean)
AGGRESSIVE=false
# Set flood mode on (boolean)
FLOOD=false
# Scan for reflectors at given CIDR (string)
SCANNER_CIDR=192.168.1.0/24
# File path to save reflectors found (string)
SCANNER_PATH=scan_output.txt
# Shuffle victims (boolean)
SHUFFLE=true

[DNS]
# Domain (string)
DOMAIN_NAME=ddos.dns.com
```

Figura 3.2: Exemplo de um arquivo de configuração.

A GUI possui as mesmas configurações da versão em linha de comando, porém as opções ficam mais fáceis de serem visualizadas e alteradas.

A interface gráfica permite selecionar entre as opções *Attack*, *Scan* e *Benchmark*, exibindo suas devidas configurações ao selecionar cada uma delas.

Essas opções ficam na lateral esquerda da interface juntamente com a seleção do protocolo, comum aos três modos citados. Ao selecionar um protocolo, suas opções específicas são exibidas na parte inferior da interface.

Na tela *Attack*, exibido na Figura 3.3, estão todas os parâmetros suportados nos ataques, com campos para digitar os valores de cada opção ou somente uma caixa de seleção para os parâmetros sem valores.

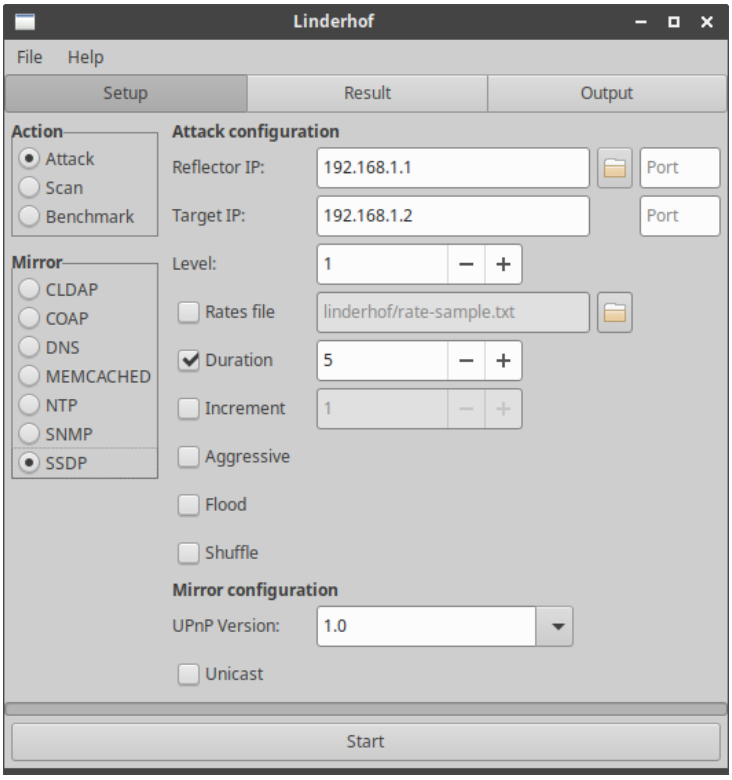

Figura 3.3: Tela Attack da interface gráfica.

Na tela *Scan* somente são apresentadas as opções de selecionar o CIDR e porta a serem escaneados e um campo para escolher um caminho para salvar o arquivo de endereços encontrados. Essa tela é mostrada na Figura 3.4.

Já na tela *Benchmark* da Figura 3.5 as opções se resumem ao endereço e a porta do dispositivo que se deseja testar a amplificação.

Na parte superior da interface gráfica estão posicionadas três abas, *Setup*, *Result* e *Output*. A primeira abrange as telas *Attack*, *Scan* e *Benchmark* já descritas. A aba *Result* da Figura 3.6 apresenta o gráfico da quantidade de pacotes enviados pelo atacante bem

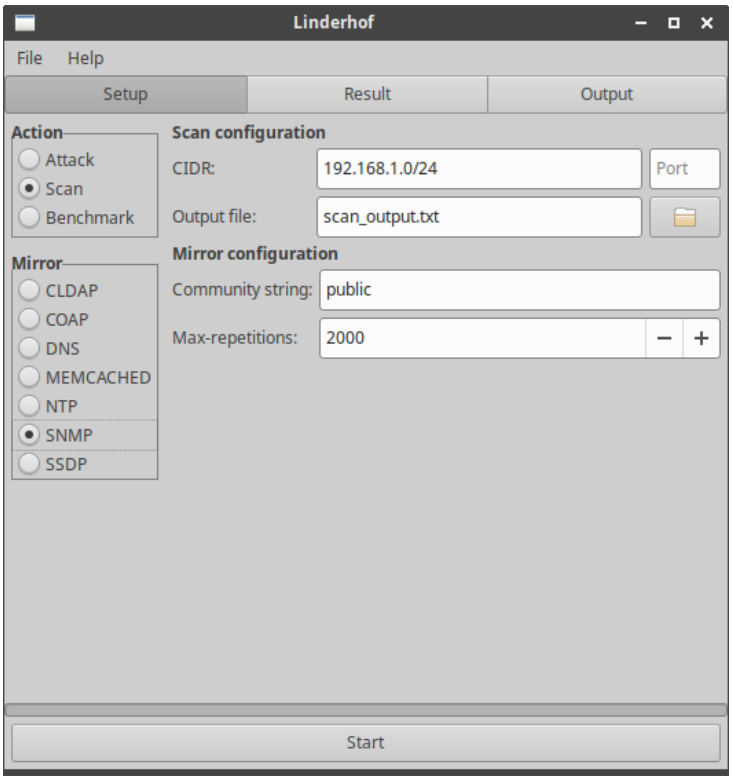

Figura 3.4: Tela Scan da interface gráfica.

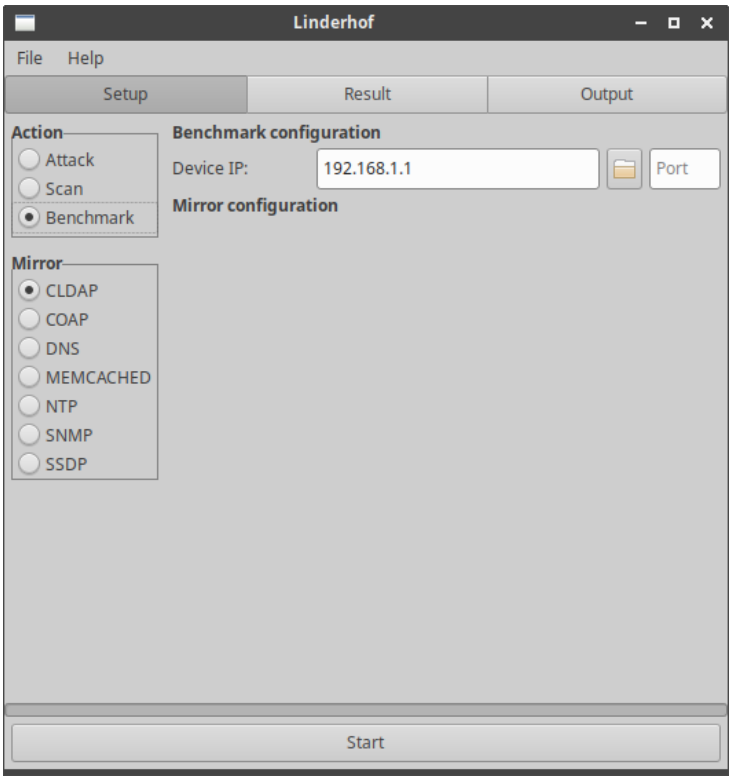

Figura 3.5: Tela Benchmark da interface gráfica.

como o total de pacotes enviados e o nível atual do ataque. Já na aba *Output* está presente a mesma saída que se obtém ao executar a ferramenta pela linha de comando, mostrando a quantidade de pacotes enviados para cada refletor ao longo do ataque, como exibido na Figura 3.7.

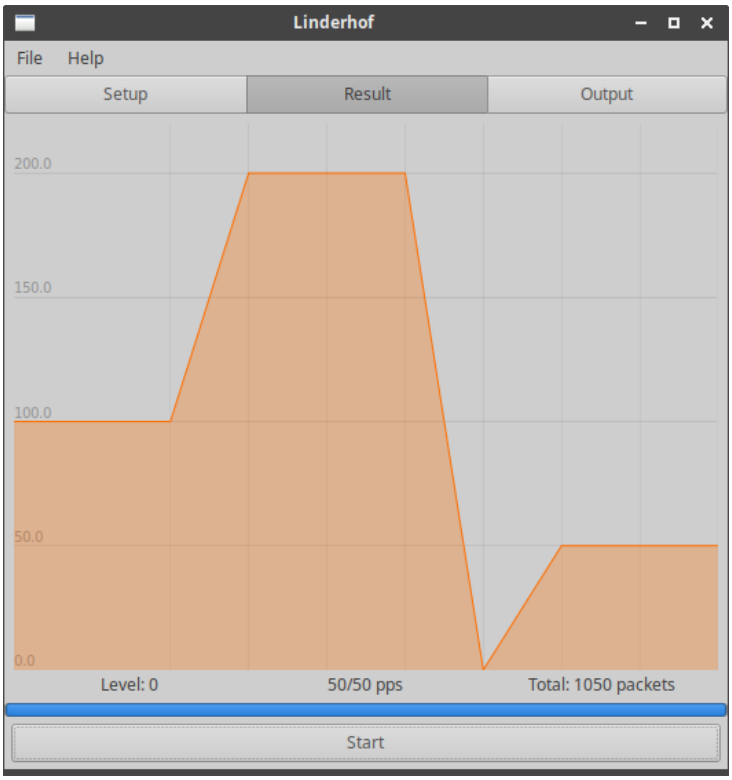

Figura 3.6: Aba Result da interface gráfica.

Apesar da criação da GUI, a utilização por linha de comando ainda está presente. Ao executar o Linderhof sem nenhum parâmetro a interface gráfica é exibida. Porém, ao adicionar qualquer parâmetro em sua execução, a versão CLI é executada normalmente.

3.5.7 Implementação de protocolos

A padronização facilitou a adição de novos *mirrors* à ferramenta. O trabalho fica por conta de somente montar os pacotes e adicionar as customizações de seus parâmetros. Fora as modificações do HOM ainda é necessário realizar algumas outras, como adicionar os parâmetros à linha de comando, atualizar a interface gráfica e adicionar o protocolo em algumas listagens e importações nos módulos *Commander* e *Scanner*. A utilização do protocolo CLDAP implementado neste trabalho seguindo as características citadas no capítulo Capítulo 2 será demonstrada no capítulo Capítulo 4.

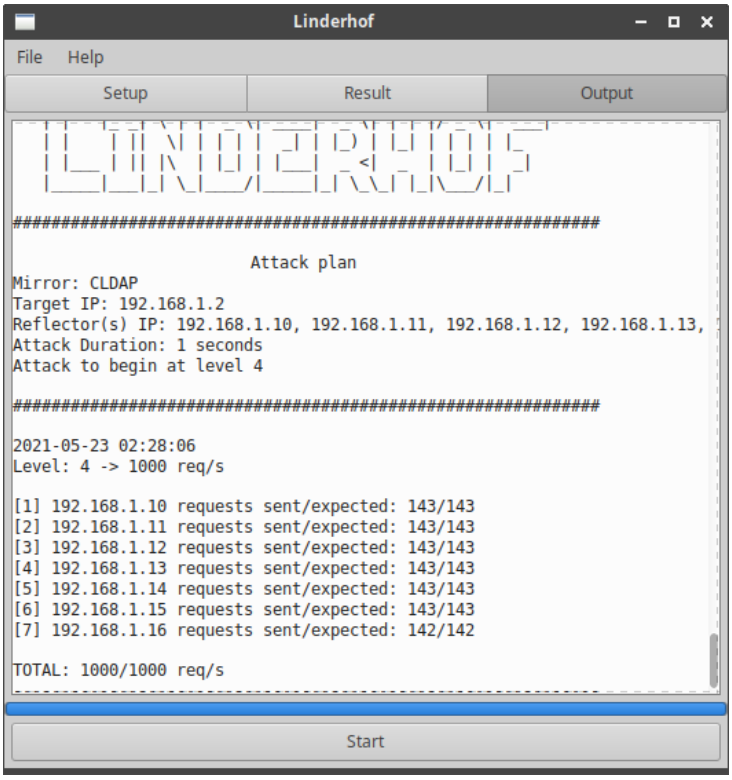

Figura 3.7: Aba Output da interface gráfica.

3.6 Resumo do capítulo

Neste capítulo foi apresentado a ferramenta Linderhof. Inicialmente foi descrito seu histórico e motivações para sua evolução, até chegar na versão 1.0.0, ponto inicial deste trabalho. Em seguida foi descrito brevemente sua arquitetura, contendo a responsabilidades de cada módulo existente na ferramenta. Na seção seguinte foram explicadas todas as suas funcionalidades e parâmetros aceitos, incluindo as contribuições feitas neste trabalho.

Capítulo 4

Testes e resultados

Nesta seção são apresentados os resultados dos testes realizados com a adição das novas funcionalidades na ferramenta Linderhof. Além das funcionalidades, há também o teste de desempenho que reflete a forma como a ferramenta é usada para a avaliação de um refletor.

4.1 Testes de funcionalidades

4.1.1 Múltiplos refletores

A Figura 4.1 mostra o funcionamento da aplicação ao inserir um arquivo contendo uma lista de endereços. O arquivo utilizado é mostrado na Figura 4.2.

Os pacotes foram distribuídos entre todos os refletores listados de modo que seja cumprida a taxa estabelecida na configuração do ataque.

4.1.2 Taxa de injeção e forma de onda de ataque customizável

O usuário pode definir uma taxa de injeção diferente das definidas pelos níveis. Com isso, pode ser criado diversas formas de onda de ataque.

Para demostrar as possibilidades de customização da taxa de injeção foram criados quatro arquivos de texto contendo em cada linha valores que possam gerar as formas de onda quadrada, triangular, dente de serra e senoidal, como podem ser vistas na Figura 4.3.

4.1.3 Ataque em múltiplos endereços na sub-rede

A Figura 4.4 mostra os endereços das vítimas capturados pelo software *Wireshark* ao utilizar os endereços CIDR 10.0.0.0/30, 10.0.0.0/27 e 10.0.0.0/24. Os prefixos utilizados definem que a notação representa, respectivamente, 4, 32 e 256 endereços IP. Porém,

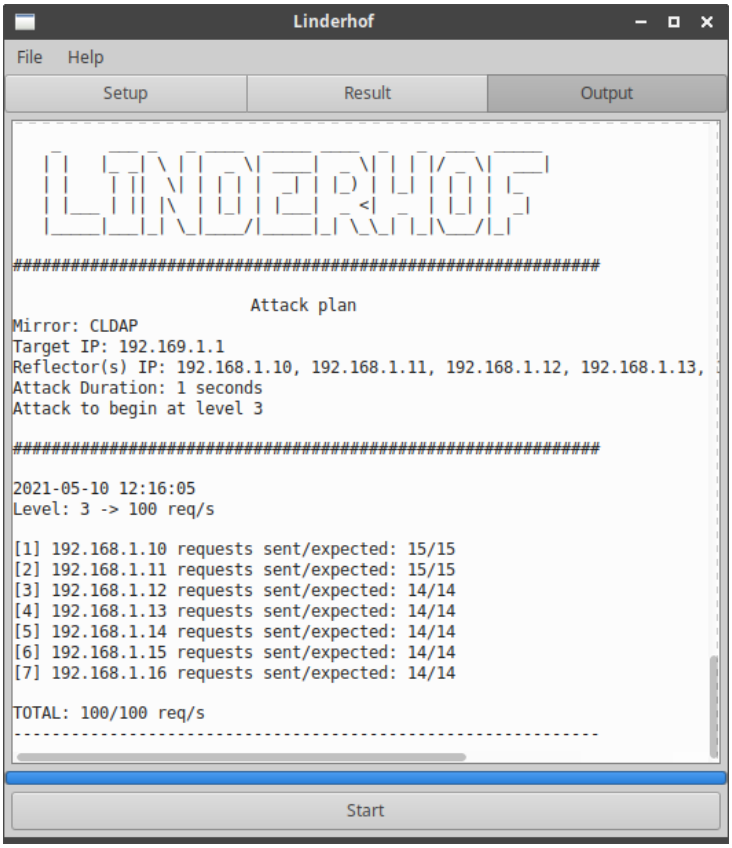

Figura 4.1: Utilização de múltiplos refletores.

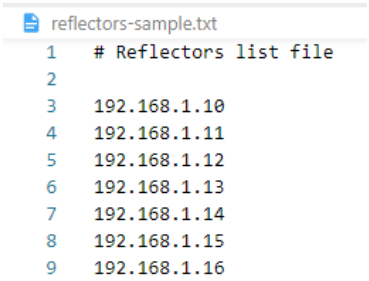

Figura 4.2: Arquivo contendo a lista de refletores.

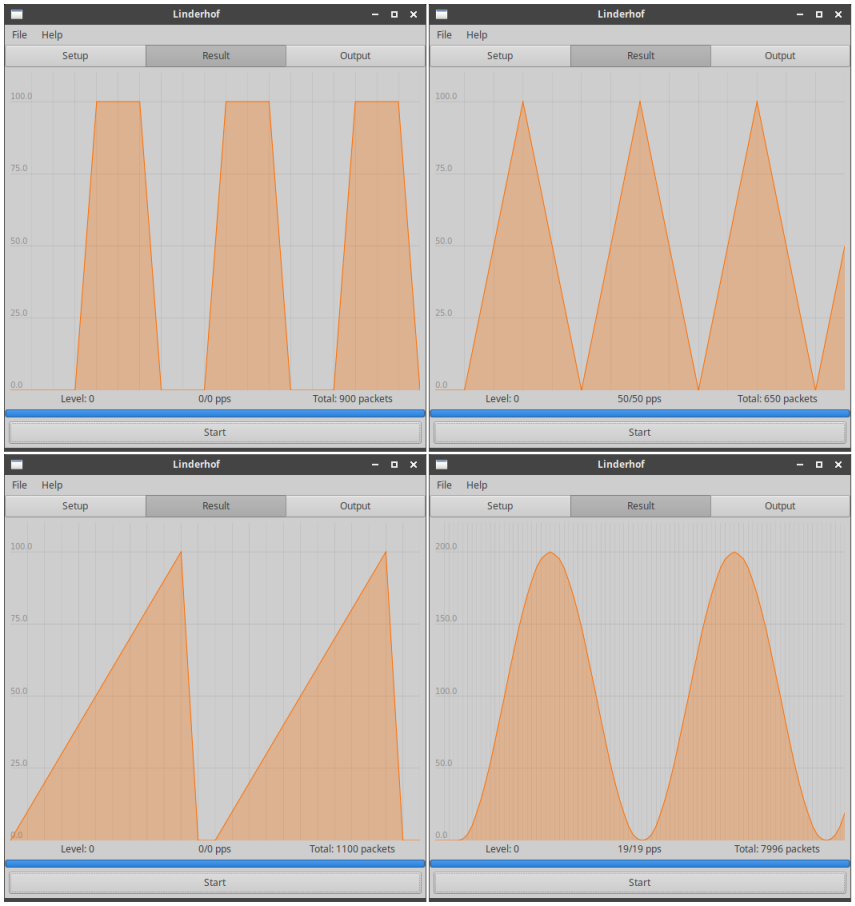

Figura 4.3: Taxa de ataque diversas formas de onda.

o primeiro e o último endereço são reservados para especificar a rede e o domínio de broadcast, sobrando 2, 30 e 254 endereços utilizáveis para atribuir aos hosts. As vítimas são colocadas em uma lista que é percorrida ciclicamente, alterando o endereço IP de destino do ataque a cada segundo. Há ainda a possibilidade de ordenar essa lista de forma aleatória utilizando a opção *shuffle*, o que diminui a previsibilidade do ataque.

Source	Source	Source
10.0.0.2	10.0.0.19	10.0.0.244
10.0.0.1	10.0.0.20	10.0.0.245
10.0.0.2	10.0.0.21	10.0.0.246
10.0.0.1	10.0.0.22	10.0.0.247
10.0.0.2	10.0.0.23	10.0.0.248
10.0.0.1	10.0.0.24	10.0.0.249
10.0.0.2	10.0.0.25	10.0.0.250
10.0.0.1	10.0.0.26	10.0.0.251
10.0.0.2	10.0.0.27	10.0.0.252
10.0.0.1	10.0.0.28	10.0.0.253
10.0.0.2	10.0.0.29	10.0.0.254
10.0.0.1	10.0.0.30	10.0.0.1
10.0.0.2	10.0.0.1	10.0.0.2
10.0.0.1	10.0.0.2	10.0.0.3
10.0.0.2	10.0.0.3	10.0.0.4
10.0.0.1	10.0.0.4	10.0.0.5
10.0.0.2	10.0.0.5	10.0.0.6

Figura 4.4: Endereços utilizados no ataque com múltiplas vítimas.

4.2 Teste de desempenho do protocolo CLDAP

Apesar de serem implementados os protocolos SNMP e CLDAP, os resultados dos testes utilizando o SNMP já foram apresentados em [2]. Sendo assim, somente os resultados dos testes realizados com o CLDAP serão apresentados neste trabalho.

4.3 Cenário utilizado nos testes

Foram utilizados quatro dispositivos durante a realização dos testes, sendo eles três computadores atuando como atacante, refletor e vítima, juntamente com um switch que os interconectava. Todos eles foram conectados por cabos CAT 5e.

Os papéis dos dispositivos são comparáveis aos descritos na Figura 2.2 do capítulo 2, com a diferença que nos testes não há um agente no papel do controlador. O que chamamos de atacante e que executa a ferramenta é representado como pelos *bots* da figura.

O switch utilizado é do modelo TP-LINK TL-SG1008D e conta com 8 portas 10/100/1000Mbps, interconectando os demais dispositivos cujas especificações estão listadas nas Tabelas 4.1 a 4.3.

Tabela 4.1: Especificações do atacante.

Característica	**Vítima**
Processador	Intel i5 8250U @ 1.6 GHz
Memória	8GB DDR4 @ 2400MHz
Placa de rede	ASIX AX88179 USB 3.0 Gigabit
Sistema Operacional	Xubuntu 20.04.1

Tabela 4.2: Especificações do refletor.

Característica	**Refletor**
Processador	AMD Ryzen 3700X @ 3.6 GHz
Memória	16GB DDR4 @ 3000MHz
Placa de rede	Realtek RTL8125B PCI-E 2,5 Gigabit
Sistema Operacional	Windows Server 2012 R2

Os pacotes foram capturados em todos os computados com o software *dumpcap*, enquanto a contagem dos pacotes e bytes trafegados a cada segundo durante a duração do ataque foi realizada com o software *tshark* em cima dos arquivos de captura gerados.

Os valores obtidos foram então separados em grupos de dez segundos, que representa a duração em cada nível, e calculado a média para obter o desempenho do ataque por segundo em cada nível. Esses dados são mostrados e discutidos na seção seguinte.

4.3.1 Metodologia e procedimentos

A metodologia utilizada neste trabalho baseia-se nos trabalhos [41], [1] e [2] e é descrita a seguir nesta seção.

Os testes foram realizados com a ferramenta Linderhof em um ambiente controlado com dispositivos sem acesso à internet.

Os ataques utilizaram os níveis 1 a 7 do Linderhof descritos pela Equação 3.1. Os níveis foram escolhidos por ter sido observado nos trabalhos citados que já ocorre saturação dos dispositivos dentro desse intervalo. Esses níveis foram então incrementados a cada 10 segundos, totalizando 70 segundos de ataque.

O comando do Linderhof abaixo foi utilizado para gerar o ataque:

Tabela 4.3: Especificações da vítima.

Característica	**Atacante**
Processador	Intel i5 4210U @ 1.7 GHz
Memória	6GB DDR3L @ 1600MHz
Placa de rede	Realtek RTL8168/8111 PCI-E Gigabit
Sistema Operacional	Xubuntu 20.04.1

```
./bin/lhf -m cldap -t 10.0.0.1 -r 10.0.0.2 -l 1 -i 10 -d 70
```

Ele informa o protocolo do refletor, os endereços da vítima e do refletor, o nível inicial do ataque como sendo 1, um incremento de nível a cada 10 segundos e a duração total de 70 segundos.

Para se verificar a eficácia de determinado protocolo como refletor realiza-se o cálculo de sua amplificação. Ela pode ser tanto da quantidade de pacotes quanto da quantidade de bits que a vítima recebe em comparação com a quantidade que o atacante envia por segundo. Esses cálculos são descritos pela Equação 4.1 e Equação 4.2.

$$Amplificação_{pacotes} = \frac{Pacotes\ de\ resposta}{Pacotes\ de\ requisição} \tag{4.1}$$

$$Amplificação_{bits} = \frac{Tamanho\ da\ resposta}{Tamanho\ da\ requisição} \tag{4.2}$$

No cálculo da amplificação dos bits considerou-se apenas os cabeçalhos IP e UDP, desconsiderando assim o cabeçalho Ethernet como demonstrado na Equação 4.3. Já no cálculo da amplificação por pacotes considerou-se também os fragmentos gerados devido ao tamanho do pacote de resposta enviados pelo refletor exceder os 1500 bytes do MTU (Maximum Transmission Unit).

$$Amplificação_{bits} = \frac{Cabeçalho_{IP} + Cabeçalho_{UDP} + Dados\ (resposta)}{Cabeçalho_{IP} + Cabeçalho_{UDP} + Dados\ (requisição)} \tag{4.3}$$

4.3.2 Resultados

Os resultados do ataque são mostrados abaixo. A Figura 4.5 e a Tabela 4.4 mostram o gráfico e a tabela, respectivamente, da quantidade de pacotes enviados e recebidos a cada segundo por nível do ataque, enquanto a Figura 4.6 e a Tabela 4.5 mostram essas mesmas informações em bits por segundo. Já a Tabela 4.6 apresenta os pacotes enviados e recebidos por segundo em cada dispositivo em comparação com o esperado segundo as especificações da ferramenta e do fator de amplificação de obtido. Por fim, a Tabela 4.7 e a Figura 4.7 mostra as amplificações obtidas em termos de pacotes e bits.

Com os resultados dos primeiros níveis nota-se que o fator de amplificação referente aos pacotes é de 2x, ou seja, para cada pacote enviado pelo atacante, a vítima recebe dois. Já em bits, o fator de amplificação obtido foi de 31,36x, segundo o cálculo da Equação 4.4.

$$Amplificação_{bits} = \frac{22832\ bits}{728\ bits} = 31,36 \tag{4.4}$$

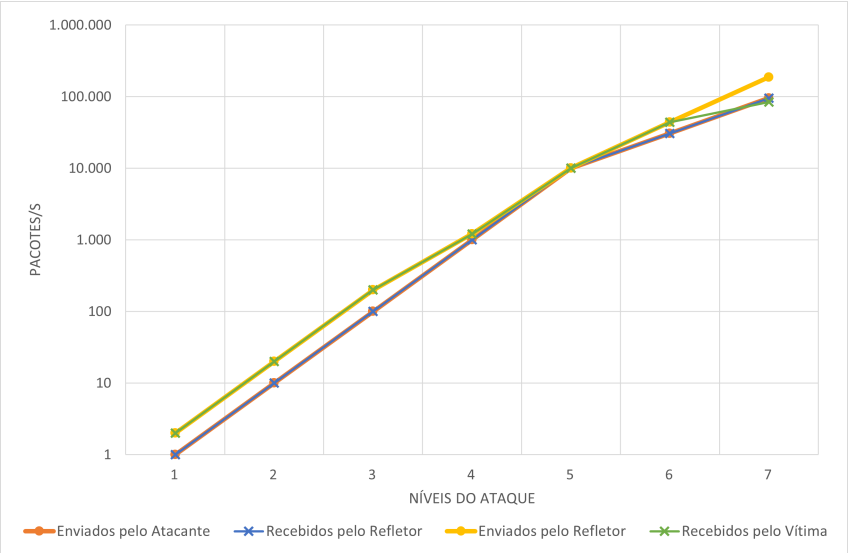

Figura 4.5: Pacotes enviados e recebidos por segundo.

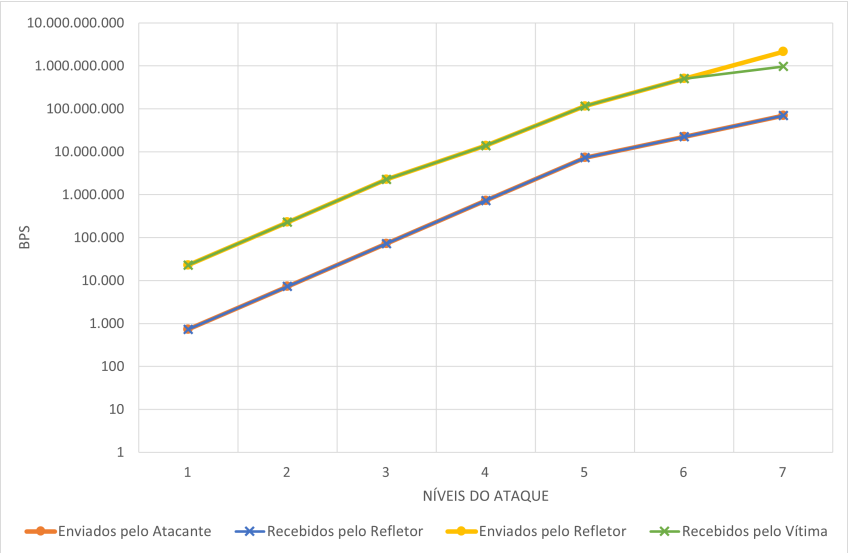

Figura 4.6: Bits enviados e recebidos por segundo.

Tabela 4.4: Pacotes enviados e recebidos por segundo.

Nível	**Atacante**	**Entrada Refletor**	**Saída Refletor**	**Vítima**
1	1	1	2	2
2	10	10	20	20
3	100	100	200	200
4	1000	1000	1212	1212
5	10000	10000	10075	10075
6	30633	30632	43949	43949
7	95437	95398	187217	84621

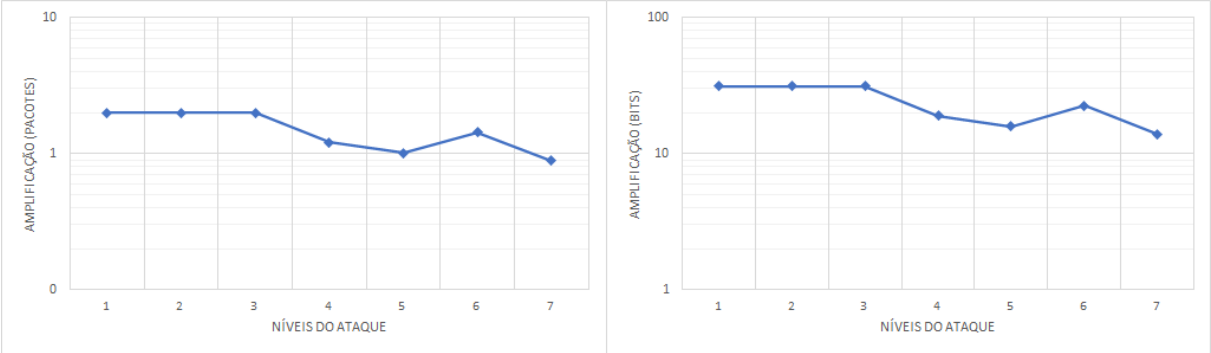

Figura 4.7: Amplificação em pacotes e bits por segundo.

Tabela 4.5: Bits enviados e recebidos por segundo.

Nível	**Atacante**	**Entrada Refletor**	**Saída Refletor**	**Vítima**
1	728	728	22832	22832
2	7280	7280	228320	228320
3	72800	72800	2283200	2283200
4	728000	728000	13836192	13836192
5	7280000	7280000	115011634	115011634
6	22300502	22300314	501717405	501717405
7	69478354	69450035	2137264706	966036819

Tabela 4.6: Pacotes trafegados em comparação com o esperado.

Nível	**Atacante**	**Entrada Refletor**	**Saída Refletor**	**Vítima**
1	100,00%	100,00%	100,00%	100,00%
2	100,00%	100,00%	100,00%	100,00%
3	100,00%	100,00%	100,00%	100,00%
4	100,00%	100,00%	60,60%	60,60%
5	100,00%	100,00%	50,37%	50,37%
6	30,63%	30,63%	21,97%	21,97%
7	9,54%	9,54%	9,36%	4,23%

Tabela 4.7: Amplificação por nível.

Nível	**Pacotes**	**Bits**
1	2,00	31,36
2	2,00	31,36
3	2,00	31,36
4	1,21	19,01
5	1,01	15,80
6	1,43	22,50
7	0,89	13,90

Observou-se no nível 4 a saturação do refletor. Houve uma queda na quantidade de pacotes enviados por ele e consequente diminuição na quantidade de pacotes recebidos pela vítima. Com isso, o fator de amplificação de pacotes passou de 2x para 1,21x, com o refletor enviando 60% dos pacotes que deveria para a vítima.

No nível 5 nota-se um aumento na saturação no refletor. Neste nível praticamente não houve amplificação, com o fator caindo para 1,01x.

Já no nível 6 ocorre a saturação do atacante, com uma queda expressiva de quase 70% na quantidade de pacotes enviados em comparação com o esperado para este nível, enviando uma taxa total de cerca de 22,3 Mbps e 30633 pacotes/s. Com isso, o refletor conseguiu aumentar a quantidade de pacotes enviados em comparação com os recebidos, elevando assim o fator de amplificação para 1,43x.

No nível 7, último nível observado, os pacotes enviados pelo atacante sofreram outro corte, chegando a quase 70 Mbps, menos de 10% da quantidade esperada. Observou-se também uma saturação na vítima, recebendo menos da metade dos pacotes enviados pelo refletor.

4.3.3 Discussão

Como esperado, o refletor foi o primeiro dispositivo a sofrer uma saturação, já no nível 4, devido a grande quantidade de pacotes que trafegam por ele. Desse momento em diante ocorre a diminuição da eficiência do ataque, pois parte dos pacotes enviados pelo atacante não são convertidos em pacotes recebidos pela vítima.

Somente foi possível observar a saturação da vítima no nível 7, quando alcançou a marca de 966 Mbps e 84,6 mil pacotes por segundo.

Os resultados obtidos são semelhantes aos dos trabalhos anteriores, como [1] e [2], onde foi constatado que ataques a partir do nível 4, ou até mesmo 3 em alguns casos, perdem sua eficiência, evidenciando assim a necessidade de uma grande quantidade de atacantes e refletores com baixas taxas de injeção de pacotes para realizar um ataque de grande escala.

Em comparação com os dados obtidos em [1] e [2] mostrados na Tabela 4.8, o CLDAP, com amplificação máxima de 2x pacotes e 31,36x bits, possui as menores taxas de amplificação tanto em pacotes quando em bits por segundo dentre os protocolos SNMP, SSDP, NTP, DNS e CoAP estudados.

O CLDAP também possui taxas menores que o CoAP, que segundo os testes feitos em [2], obteve o fator de amplificação em bits de 58x, mas o mesmo fator em quantidade de pacotes.

Apesar da menor taxa de amplificação em comparação com outros protocolos, mostrou-se viável a utilização do protocolo CLDAP em ataques AR-DDoS, fornecendo uma taxa de

Tabela 4.8: Comparação de taxas de amplificação por protocolo (Adaptado de [1] e [2]).

Protocolo	**Taxa de bits**	**Taxa de pacotes**
SNMP	609,03	33,40
SSDP	38,23	10,00
NTP	422,81	100,00
DNS	43,81	3,00
CoAP	58,44	2,00
CLDAP	31,36	2,00

amplificação considerável e aumentando o número de dispositivos que podem ser utilizados em tais ataques.

Deve-se notar ainda que quanto maior a quantidade de protocolos que possuam essa característica de amplificação, maior também é a quantidade de dispositivos propensos a serem utilizados como refletores. Com isso, aumenta-se a capacidade dos atacantes de gerarem grandes ataques, dificultado qualquer tipo de defesa adotada pelas suas vítimas.

4.4 Síntese

Os resultados obtidos com os testes relativos à implementação do AR-DDoS sobre CLDAP são consistentes com os resultados obtidos anteriormente para os outros protocolos com a saturação dos refletor ocorrendo na transição do nível 3 para 4. O motivo para essa recorrência ainda é motivo de investigação, sem uma explicação conclusiva.

Capítulo 5

Conclusão e trabalhos futuros

5.1 Conclusão

Esta monografia apresentou melhorias na ferramenta Linderhof, consolidadas na versão 2.0.0, no sentido de expandir a abrangência sobre ataques DDoS mais populares.

Primeiro foram discutidos os conceitos que envolvem um ataque de negação de serviço e a sua forma refletida e amplificada. Citamos alguns dos métodos que estão sendo utilizados para burlar sistemas de mitigação e falamos sobre o funcionamento e características do protocolo CLDAP.

Em seguida foi apresentada a ferramenta Linderhof, que este trabalho se propôs a melhorar. Foi apresentado seu histórico contendo inspirações e melhorias sofridas ao longo de diversos trabalhos. Sua arquitetura foi explicada brevemente e suas funcionalidades até então implementadas foram comentadas. Além disso, foram apresentadas as contribuições deste trabalho à ferramenta, incluindo a adição da interface gráfica, personalizações dos ataques e novos protocolos.

Com isso, foram exibidos os resultados obtidos a partir das novas funcionalidades inseridas no Linderhof. Foi mostrado o correto funcionamento da utilização de múltiplos refletores, customização das taxas de injeção de pacotes e ataques a múltiplas vítimas em uma sub-rede. Essas duas últimas contribuições abrem a possibilidade de estudos das formas de mitigação de técnicas como o *Pulse Wave* e *Carpet Bombing*.

Por fim, foi analisada a utilização do protocolo CLDAP em um ataque controlado. Utilizando três computadores nos papéis de atacante, refletor e vítima, foi gerado um ataque com níveis incrementais e de duração de 70 segundos, seguindo a metodologia utilizada em outros trabalhos que se utilizaram da mesma ferramenta.

Esse ataque permitiu observar o mesmo comportamento de trabalhos que analisaram outros protocolos. O refletor sofre saturação com uma baixa taxa de injeção de pacotes, com cerca de 22 Mbps enviados pelo atacante, que sofre saturação em seguida.

5.2 Trabalhos futuros

Com o objetivo de dar continuidade na evolução o Linderhof, são propostos os seguintes trabalhos:

- **Ataque de múltiplos vetores:** Os maiores ataques atualmente não se limitam a explorar somente um protocolo para reflexão. Um mesmo ataque abusa de diversos protocolos a fim de dificultar a mitigação e possibilitar a utilização da maior quantidade de refletores possíveis.
- **Orquestração:** Como foi observado nesse trabalho um único atacante não é capaz de gerar um grande volume de tráfego, mesmo com uma grande quantidade de refletores. Para gerar ataques volumosos é preciso uma grande rede de dispositivos que lançam os ataques a partir de diferentes origens simultaneamente e são controlados por um dispositivo central.
- **Novos protocolos:** Para possibilitar o estudo de ataques atuais é necessário que a ferramenta possua suporte aos protocolos que sejam mais utilizados, como é o caso do RDP que observou-se um recente aumento em sua utilização para ataques DoS.

Referências

[1] Gondim, João J. C., Robson de Oliveira Albuquerque e Orozco Ana Lucila Sandoval: *Mirror saturation in amplified reflection distributed denial of service: A case of study using snmp, ssdp, ntp and dns protocols*. `https//doi.org/10.1016/j.future.2020.01.024`, 2020. Future Generation Computer Systems. x, 4, 30, 34, 35

[2] Vasques, Alan T.: *Análise de Saturação de Dispositivos IoT Atuando como Refletores em Ataques Distribuído de Negação de Serviço por Reflexão Amplificada*, 2020. `https://repositorio.unb.br/handle/10482/40089`. x, 3, 19, 29, 30, 34, 35

[3] Shield, AWS: *Threat landscapereport - q1 2020*. https://aws-shield-tlr.s3.amazonaws.com/2020-Q1_AWS_Shield_TLR.pdf, 2020. 1

[4] Menscher, Damian: *Exponential growth in ddos attack volumes*. https://cloudblog.withgoogle.com/products/identity-security/identifying-and-protecting-against-the-largest-ddos-attacks/, 2020. 1

[5] Krebs, Brian: *Akamai on the record krebsonsecurity attack*, 2016. `https://krebsonsecurity.com/2016/11/akamai-on-the-record-krebsonsecurity-attack/`, acesso em 2021-05-21. 1

[6] Camargo, Camila Imbuzeiro: *Mirai : um estudo sobre botnets de dispositivos IoT*, 2019. `https://bdm.unb.br/handle/10483/21936`. 1

[7] Anna-senpai: *Mirai source code*, 2016. `https://github.com/jgamblin/Mirai-Source-Code`, acesso em 2021-05-21. 1

[8] Kolias, Constantinos, Georgios Kambourakis, Angelos Stavrou e Jeffrey Voas: *Ddos in the iot: Mirai and other botnets*. Computer, 50(7):80–84, 2017. 1

[9] Antonakakis, Manos, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J. Alex Halderman, Luca Invernizzi, Michalis Kallitsis, Deepak Kumar, Chaz Lever, Zane Ma, Joshua Mason, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas e Yi Zhou: *Understanding the mirai botnet*. Em *26th USENIX Security Symposium (USENIX Security 17)*, páginas 1093–1110, Vancouver, BC, agosto 2017. USENIX Association, ISBN 978-1-931971-40-9. `https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis`. 1

[10] Kenig, Ronen: *How much can a ddos attack cost your business?* https://blog.radware.com/security/2013/05/how-much-can-a-ddos-attack-cost-your-business/, 2013. 1

[11] Saldanha, Rodrigo de S.: *Ataque Distribuído de Negação de Serviço por Reflexão Amplificada Explorando o Protocolo Domain Name System*, 2019. 1, 13

[12] Vieira, Alexander A. de S.: *Ataque Distribuído de Negação de Serviço por Reflexão Amplificada usando Network Time Protocol*, 2019. `https://bdm.unb.br/handle/10483/25261`. 1, 13

[13] Miranda, Igor F.: *Ataque de negação de serviço por reflexão amplificada explorando Memcached*, 2019. `https://bdm.unb.br/handle/10483/25262`. 1, 13

[14] Pereira, Pedro H. M.: *Internet das coisas e seus riscos: uma análise da exploração de servidores CoAP como refletores de ataques de negação de serviço amplificados*, 2019. 1, 13

[15] Dantas, Amanda Lopes, Matheus de Oliveira Vieira, Alan Tamer Vasques e Joao José Costa 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. SBC, 2020. 2, 3, 13

[16] Vasques, Alan Tamer e João J. C. Gondim: *Amplified reflection ddos attacks over iot mirrors: A saturation analysis*. Em *2019 Workshop on Communication Networks and Power Systems (WCNPS)*, páginas 1–6, 2019. 3

[17] Vasques, Alan Tamer e João J. C. Gondim: *Amplified reflection ddos attacks over iot reflector running coap*. Em *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*, páginas 1–6, 2020. 3

[18] Gligor, Virgil D.: *A note on denial-of-service in operating systems*. IEEE Transactions on Software Engineering, SE-10(3):320–324, 1984. 4

[19] Mirkovic, Jelena e Peter Reiher: *A taxonomy of ddos attack and ddos defense mechanisms*. SIGCOMM Comput. Commun. Rev., 34(2):39–53, abril 2004, ISSN 0146-4833. `https://doi.org/10.1145/997150.997156`. 4

[20] Wainwright, Polly e Houssain Kettani: *An analysis of botnet models*. Em *Proceedings of the 2019 3rd International Conference on Compute and Data Analysis*, ICCDA 2019, página 116–121, New York, NY, USA, 2019. Association for Computing Machinery, ISBN 9781450366342. `https://doi.org/10.1145/3314545.3314562`. 4

[21] Paxson, Vern: *An analysis of using reflectors for distributed denial-of-service attacks*. Computer Communication Review, 31, julho 2001. 5

[22] Rossow, Christian: *Amplification hell: Revisiting network protocols for ddos abuse*. janeiro 2014, ISBN 1-891562-35-5. 7

[23] Bjarnason, Steinthor: *Ddos defences in the terabit era: Attack trends, carpet bombing.* https://blog.apnic.net/2018/12/04/ddos-defences-in-the-terabit-era-attack-trends-carpet-bombing/, acesso em 2021-05-21. 7

[24] NETSCOUNT: *Defending against carpet bombing attacks.* https://www.netscout.com/sites/default/files/2020-06/SECUC_005_EN-2002%20-%20Defending%20Against%20Carpet%20Bombing%20Attacks.pdf, acesso em 2021-05-21. 7

[25] Cimpanu, Catalin: *'carpet-bombing' ddos attack takes down south african isp for an entire day*, 2019. https://www.zdnet.com/article/carpet-bombing-ddos-attack-takes-down-south-african-isp-for-an-entire-day/, acesso em 2021-05-21. 7

[26] Help Net Security: *Ssdp amplification attacks rose 639%*, 2019. https://www.helpnetsecurity.com/2019/01/22/ssdp-amplification-attacks/, acesso em 2021-05-21. 7

[27] Chugunkov, Ilya V., Leonid O. Fedorov, Bela Sh. Achmiz e Zarina R. Sayfullina: *Development of the algorithm for protection against DDoS-attacks of type pulse wave.* 2018. 8

[28] DDoS-GUARD: *Hidden threat of pulse wave ddos attacks*, 2019. https://ddos-guard.net/en/info/blog-detail/hidden-threat-of-pulse-wave-ddos-attacks, acesso em 2021-05-21. 8

[29] Zeifman, Igal: *Attackers use ddos pulses to pin down multiple targets*, 2017. https://www.imperva.com/blog/pulse-wave-ddos-pins-down-multiple-targets/, acesso em 2021-05-21. 8, 10

[30] Zeilenga, Kurt: *Connection-less Lightweight Directory Access Protocol (CLDAP) to Historic Status.* RFC 3352, 2003. https://rfc-editor.org/rfc/rfc3352.txt. 9

[31] Howes, Timothy A: *The lightweight directory access protocol: X. 500 lite.* Relatório Técnico, Center for Information Technology Integration, 1995. 9

[32] Tuttle, Steven, Ami Ehlenberger, Ramakrishna Gorthi, Jay Leiserson, Richard Macbeth, Nathan Owen, Sunil Ranahandola, Michael Storrs, Chunhui Yang *et al.*: *Understanding LDAP-design and implementation.* IBM Redbooks, 2006. 9

[33] Zeilenga, Kurt: *Lightweight Directory Access Protocol (LDAP): Directory Information Models.* RFC 4512, 2006. https://rfc-editor.org/rfc/rfc4512.txt. 9

[34] Sermersheim, Jim: *Lightweight Directory Access Protocol (LDAP): The Protocol.* RFC 4511, 2006. https://rfc-editor.org/rfc/rfc4511.txt. 9

[35] Medeiros, Tiago F.: *Ataque distribuído de negação de serviço por reflexão amplificada usando Simple Network Management Protocol*, 2015. https://bdm.unb.br/handle/10483/11152. 13

[36] Hirata, Henrique S.: *Ataque de Negação de Serviço por Reflexão Amplificada usando Simple Service Discovery Protocol*, 2018. https://bdm.unb.br/handle/10483/22128. 13

[37] Dantas, Amanda L.: *Refatoração e Recomposição da Integridade Estrutural: O Caso do Linderhof*, 2019. 14

[38] Fuller, Vince, Tony Li, Jessica (Jie Yun) Yu e Kannan Varadhan: *Classless inter-domain routing (cidr): an address assignment and aggregation strategy*. Rfc 1519, RFC Editor, September 1993. http://www.rfc-editor.org/rfc/rfc1519.txt. 19

[39] GTK Team: *Gtk*, 2021. https://www.gtk.org. 20

[40] The Glade project: *Glade*, 2021. https://glade.gnome.org. 20

[41] Gondim, J. J. C., R. de Oliveira Albuquerque, A. Clayton Alves Nascimento, L. García Villalba e T. H. Kim: *A methodological approach for assessing amplified reflection distributed denial of service on the internet of things*. Sensors, 16(11):1855, 2016. 30

Apêndice A

Artigo SBRC

Linderhof: uma ferramenta para avaliação de sistemas de mitigação de ataques reflexivos volumétricos (DDoS)

Amanda Lopes Dantas[1], **Matheus de Oliveira Vieira**[1],
Alan Tamer Vasques[2], **João José Costa Gondim**[1,2]

[1]Departamento de Ciência da Computação – Universidade de Brasília (UnB)
Brasília – DF – Brasil

[2]Programa de Pós-Graduação Profissional em Engenharia Elétrica - PPEE
Departamento de Engenharia Elétrica – Universidade de Brasília (UnB)
Brasília – DF – Brasil

amandadantas19@gmail.com,{matheus.vieira,alan.tamer}@aluno.unb.br,gondim@unb.br

Abstract. *Denial of service attacks aim to disrupt legitimate users from accessing a particular service. Its amplified and reflected version is more commonly used and has become an increasing threat to the Internet stability. The Linderhof tool was created for the study of reflective and volumetric attacks mitigation systems. It enables the evaluation of the CoAP, DNS, Memcached, NTP, SSDP, SNMP protocols and is extensible to other protocols. In this paper the tool and its use will be presented in more detail.*

Resumo. *Os ataques de negação de serviço têm como objetivo interromper o acesso de usuários legítimos a um determinado serviço. A versão amplificada e refletida desse tipo de ataque é mais comumente utilizada e têm se tornado uma ameaça crescente à estabilidade da Internet. A ferramenta Linderhof foi criada para o estudo de sistemas de mitigação de ataques reflexivos volumétricos. Ela possibilita a avaliação dos protocolos CoAP, DNS, Memcached, NTP, SSDP, SNMP e é extensível a outros protocolos. Nesse artigo a ferramenta e seu uso serão apresentadas em mais detalhes.*

1. Introdução

Um ataque de Negação de Serviço (DoS, *Denial of Service*) tem como objetivo impedir o acesso de usuários legítimos aos serviços fornecidos pela vítima [Riza et al. 2019]. Existem vários contextos onde um ataque DoS pode ocorrer, como em sistemas operacionais e serviços baseados em rede. Um ataque DoS por reflexão/amplificação tem o objetivo de mascarar a fonte do ataque, usando terceiros para repassar tráfego ilegítimo para a vítima. Esses terceiros são denominados refletores [Peng et al. 2007]. O atacante manda para o refletor pacotes com o endereço de origem igual ao IP da vítima. Assim, quando o refletor responder aos pacotes, ele direciona o tráfego de resposta à vítima. Nesse ataque, geralmente são usados protocolos onde o pacote de *reply* é maior que o pacote de *request*, assim, o ataque também gera um efeito de amplificação [Paxson 2001]. O uso de refletores dificulta o rastreamento da origem do ataque [Rossow 2014].

Um ataque Distribuído de Negação de Serviço (DDoS, *Distributed Denial of Service*) ocorre quando vários dispositivos coordenados atacam uma ou mais vítimas, com o

intuito de exaurir os recursos de processamento ou conectividade dessas vítimas. Assim, os ataques Distribuídos de Negação de Serviço por Reflexão Amplificada (AR-DDoS, *Amplified Reflection Distributed Denial of Service*) são ataques de negação de serviço distribuídos, refletidos e amplificados.

O volume do tráfego dos ataques DDoS têm crescido nos últimos anos [Mahjabin et al. 2017]. Um exemplo desse crescimento é o ataque ao provedor DNS Dyn [Hilton 2016], que ocorreu em 2016, e afetou empresas como Amazon, Spotify, Netflix, Twitter e Github. Esse ataque chegou a magnitude de 1.2 Tbps, causando horas de indisponibilidade e perdas financeiras para as empresas afetadas, além da perda de clientes.

Visando a melhoria das formas de mitigação, o estudo desses ataques motivou a necessidade de implementações de referência. Entre outros requisitos, foram considerados: implementação do ataque sobre diferentes protocolos; controle de condução do ataque no que diz respeito à sua dinâmica; instrumentação e *logging*.

Esse trabalho apresenta a ferramenta Linderhof, que foi desenvolvida para servir de apoio em pesquisas relacionadas à mitigação de ataques AR-DDoS. Ela implementa ataques AR-DDoS abusando vários protocolos com a finalidade de estudar a dinâmica desses ataques em um ambiente controlado e acadêmico para que técnicas de defesa e de controle de danos sejam propostas e estudadas.

Nesse contexto, a ferramenta também suporta a avaliação de soluções de mitigação para tais ataques. [Gondim and Albuquerque 2019], [Gondim et al. 2020], [Gondim et al. 2016] e [Vasques and Gondim 2019] utilizaram versões iniciais que implementavam isoladamente os ataques para diversos protocolos compartilhando um *core* de geração de pacotes. Estas várias versões foram unificadas levando à versão apresentada nesse trabalho, que foi utilizada em [Vasques and Gondim 2020]. Nesses trabalhos, a ferramenta foi usada para a caracterização do comportamento de saturação de refletores sobre diversos protocolos. A ferramenta também tem sido utilizada no desenvolvimento de técnicas de detecção e identificação de refletores, além de avaliação de capacidade de sistemas de mitigação.

O restante desse artigo está organizado da seguinte forma: a Seção 2 apresenta a arquitetura e os detalhes de implementação do Linderhof, a Seção 3 explica como a demonstração será feita no salão, assim como informa onde o código-fonte e o vídeo estão disponíveis, a Seção 4 analisa o desempenho da ferramenta em um caso específico e, por fim, na Seção 5, as conclusões são expostas.

2. Ferramenta Linderhof

Linderhof é o nome de um palácio real alemão, construído por Ludwig II no século XIX, no sudoeste da região da Baviera. A ferramenta, desenvolvida em C, possui esse nome em alusão à Galeria de Espelhos (*Hall of Mirrors*) do palácio, cujos espelhos refletiam e amplificavam a luz das velas dos candelabros que ali existiam, de forma similar ao que ocorre com os pacotes nos ataques AR-DDoS. Nesse sentido, cada protocolo implementado na ferramenta representa um espelho (*mirror*) diferente da sua galeria.

A versão da ferramenta apresentada nesse artigo é a 1.0.0. Ela implementa o ataque AR-DDoS para os seguintes protocolos: CoAP, DNS, Memcached, NTP, SNMP e SSDP. A ferramenta gera os *probes* que são enviados aos refletores, que por sua vez

os amplificam e enviam ao alvo. No Linderhof, para um dado protocolo, além de suas características específicas, controla-se a taxa de geração de *probes*, a duração dos ataques, a evolução temporal da intensidade do ataque, a quantidade de refletores utilizados e seu regime de emprego. A ferramenta relata a quantidade de *probes* efetivamente enviados e os volumes de bytes transmitidos.

2.1. Arquitetura

A ferramenta é composta por quatro módulos principais (Fig. 1) descritos a seguir:

- ***Interface*:** é responsável pela interação com o usuário, recebendo como entrada os parâmetros do ataque e devolvendo na saída um resumo dos pacotes gerados e enviados;
- ***Commander*:** é o módulo onde ocorre o planejamento do ataque, de acordo com o protocolo (espelho) selecionado e os parâmetros escolhidos pelo usuário;
- ***Hall of Mirrors*:** é encarregado pela geração dos pacotes e pela iniciação dos ataques. Também é onde os espelhos são adicionados à ferramenta;
- ***Injector*:** é onde o ataque de fato ocorre, realizando o envio dos pacotes para o(s) refletor(es) e a geração de um resumo dos pacotes gerados e enviados. Esse módulo também regula a quantidade e a taxa de injeção de pacotes.

Entre as principais funcionalidades estão a possibilidade de enviar tráfego a múltiplos refletores, personalização dos parâmetros do ataque, como portas e opções específicas dos protocolos implementados e o suporte ao protocolo IPv6.

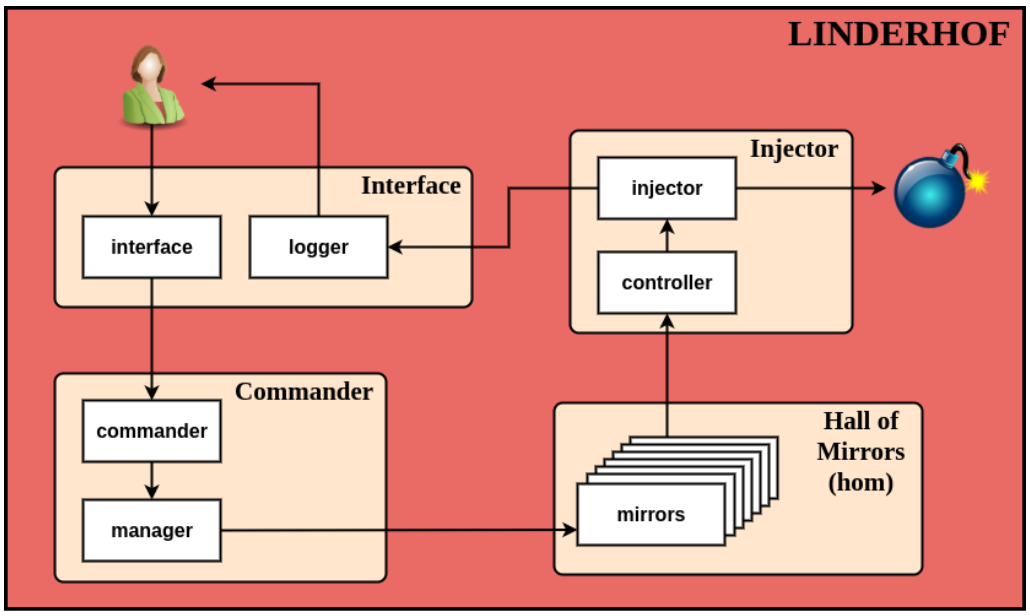

Figura 1. Arquitetura do Linderhof

2.2. Modos de Ataque

A ferramenta disponibiliza quatro modos de ataques que diferem no processo de injeção de pacotes. A injeção de pacotes segue níveis de intensidade, baseados em potência de 10.

O Linderhof possui 10 níveis disponíveis (1 a 10) e cada um possui uma intensidade diferente, que vai crescendo de forma exponencial, conforme a Eq. 1 (PpS significa Pacotes por Segundo):

$$PpS = 10^{\text{nível-1}} \tag{1}$$

Assim, por padrão, é gerado um número fixo de pacotes, distribuídos igualmente entre cada refletor, por nível. Já no modo de ataque incremental, a ferramenta aumenta o nível do ataque de acordo com a frequência passada pelo usuário. O modo agressivo, por sua vez, extrapola o nível do ataque para os refletores. Ou seja, cada refletor recebe a quantidade total de pacotes desejada para aquele nível. E, por fim, para o modo *flood* a ferramenta envia o máximo de pacotes possível para todos os refletores.

2.3. Implementação

A ferramenta foi desenvolvida em linguagem C com uma preocupação de deixá-la o mais modular possível, para que sua expansão acontecesse de forma simplificada. Desse modo, além dos seis protocolos atualmente suportados (CoAP, DNS, NTP, Memcached, SSDP e SNMP), é possível a implementação de outros protocolos sem que seja necessária uma modificação na estrutura já existente da ferramenta.

Os parâmetros e argumentos para inicialização da ferramenta e preparo do ataque podem ser passados via linha de comando ou via arquivo de configuração. A tabela 1 contém uma lista com todos os parâmetros globais do Linderhof, ou seja, os que podem ser utilizados com qualquer espelho, enquanto que a tabela 2 possui a lista de parâmetros específicos para cada espelho implementado na ferramenta, ambos para utilização via linha de comando.

3. Demonstração

A demonstração no Salão de Ferramentas será realizada utilizando um conjunto de máquinas reais e/ou virtuais, e seguirá o roteiro idêntico ao do vídeo enviado. Ao todo são necessárias pelo menos três máquinas: um atacante, um refletor e uma vítima, sendo que o Linderhof precisa ser instalado apenas na máquina do atacante. Em caso de serem utilizadas três máquinas físicas, elas serão interligadas por um *switch* e cabos. Essa rede poderá ser expandida com mais máquinas para demonstrar o uso de múltiplos refletores. Dependendo das máquinas envolvidas, será demonstrado o custo do ataque e sua consequente vantagem.

Serão explorados os diferentes protocolos e modos de ataques disponibilizados pela ferramenta e o resultado da demostração será exemplificado pelos logs da ferramenta e por análise do trafego de rede nas máquinas envolvidas. A demonstração poderá ser realizada *in loco* ou acessando o CyberSecLab via Internet.

O código-fonte da ferramenta foi disponibilizado na URL **https://cyberseclab.gigacandanga.net.br/CyberSecLab/linderhof-sbrc2020** e, para acessar o repositório, tanto o usuário quanto a senha são "**sbrc2020**". A documentação da ferramenta se encontra na pasta **docs** e na Wiki disponível no repositório. O vídeo de demonstração do Linderhof está disponível na URL **https://www.youtube.com/watch?v=M0ss3MAIRo8**, onde a instalação e as funcionalidades da ferramenta são detalhadas.

Tabela 1. Parâmetros globais do Linderhof

Parâmetro Curto	Parâmetro Longo	Parâmetro Obrigatório	Argumento Obrigatório	Argumento Padrão	Descrição
-m	--mirror	Sim	Sim	Nenhum	Espelho a ser utilizado
-t	--target	Sim	Sim	Nenhum	Endereço IP da vítima
-r	--reflector	Sim	Sim	Nenhum	Endereço IP do refletor ou arquivo com refletores
-g	--targport	Não	Sim	Aleatório (40000-60000)	Porta de origem da requisição
-p	--reflecport	Não	Sim	Padrão do Espelho	Porta de destino da requisição
-l	--level	Não	Sim	1	Nível do ataque
-d	--timer	Não	Sim	Ilimitado	Duração do ataque (em segundos)
-i	--inc	Não	Sim	Ilimitado	Duração de cada nível (em segundos)
-c	--config	Não	Sim	linderhof.conf	Arquivo de configuração
-a	--aggressive	Não	Não	-----	Modo agressivo
-f	--flood	Não	Não	-----	Modo *flooding*
-h	--help	Não	Não	-----	Ajuda

Tabela 2. Parâmetros dos espelhos do Linderhof

Parâmetro Curto	Parâmetro Longo	Parâmetro Obrigatório	Argumento Obrigatório	Argumento Padrão	Descrição
-D	--domain-name	Não	Sim	ddos.dns.com	DNS - Nome de domínio
-V	--upnp-version	Não	Sim	1.0	SSDP - Versão do UPnP
-U	--unicast	Não	Não	-----	SSDP - M-SEARCH no formato unicast
-C	--community-string	Não	Sim	public	SNMP - Comunidade
-R	--max-repetitions	Não	Sim	2000	SNMP - Campo max-repetitions
-Z	--szx	Não	Sim	6	CoAP - Campo SZX
-P	--uri-path	Não	Sim	/.well-known/core	CoAP - Caminho da URI

4. Caso de uso

Para demonstrar um caso real de utilização do Linderhof, foi utilizado um computador pessoal com as seguintes configurações, que atuou como um atacante, gerando tráfego a um determinado refletor:

- **Processador:** Intel Core i9-9900KS @ 4.00GHz;
- **Memória:** 32GB DDR4 @ 2666MHz RAM;
- **Interface de rede:** GigabitEthernet Intel I219-V;
- **Sistema Operacional**: Ubuntu 18.04 LTS x64.

Foram gerados ataques utilizando os seis protocolos disponíveis na ferramenta, nos seus 10 níveis, sendo 5 segundos para cada nível, com destino aos endereços IPv4 e IPv6 do refletor. Foi utilizado o comando abaixo, variando-se apenas as opções ”mirror”, ”ip-vitima”e ”ip-refletor”:

linderhof-sbrc2020$ bin/lhf -m *mirror* -t *ip-vitima* -r *ip-refletor* -d 50 -i 5 -l 1

Com o auxílio do *tcpdump*, foram analisadas as quantidades de tráfego e pacotes gerados pela ferramenta em cada um dos protocolos testados, tanto IPv4 quanto IPv6, e o resultado está exibido nas figuras 2, 3, 4 e 5.

Nota-se que a quantidade de pacotes gerados via IPv6, a partir do nível 5, é ligeiramente menor. Isso se deve ao tamanho do cabeçalho IPv6, que é de 40 bytes, contra 20 bytes do cabeçalho padrão do IPv4. Essa sobrecarga de 20 bytes em cada um dos pacotes faz com que nos níveis mais altos, onde uma grande quantidade de pacotes é gerada, o computador e o meio de comunicação não comporte todos os pacotes daqueles níveis.

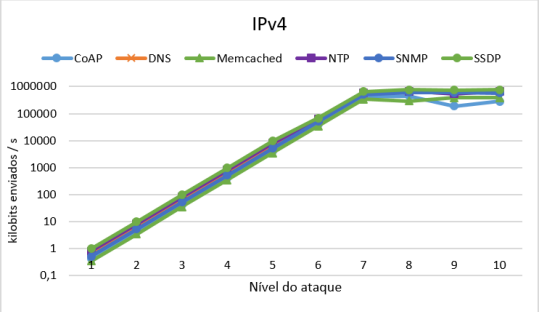

Figura 2. Kbps gerados - IPv4

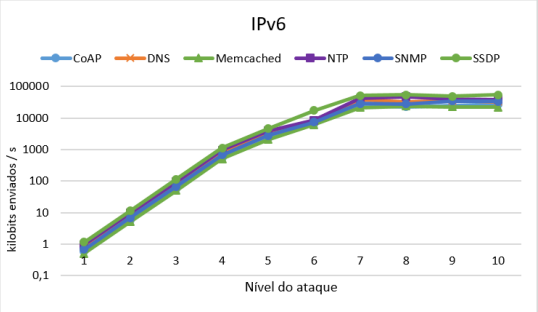

Figura 3. Kbps gerados - IPv6

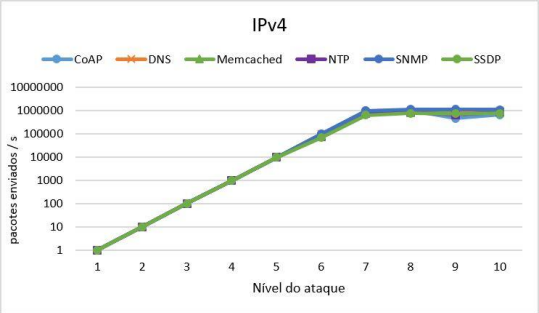

Figura 4. pkt/s gerados - IPv4

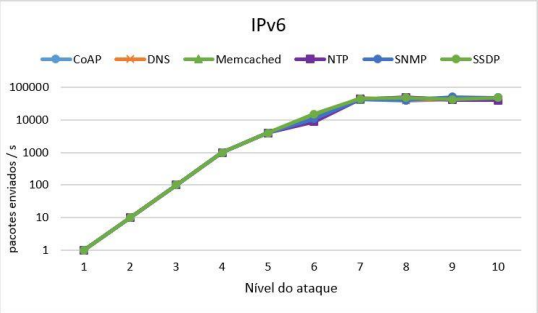

Figura 5. pkt/s gerados - IPv6

Também é possível notar, que a partir do nível 7, não foi possível gerar a quantidade máxima de pacotes destes níveis. Uma possível explicação é que a capacidade máxima do barramento do computador foi atingida. Dessa forma, a ferramenta se mostra eficiente utilizando ao máximo a capacidade oferecida pelo hardware hospedeiro.

5. Conclusão

A incidência de ataques DDoS no cenário atual da tecnologia têm crescido exponencialmente. Assim, o estudo da mitigação e controle de danos desse tipo de ataque é essencial. A ferramenta Linderhof não só representa uma boa alternativa para o estudo dessa mitigação, como também é uma boa alternativa para o estudo da natureza de ataques DDoS podendo ajudar, assim, a expandir as técnicas de detecção desse tipo de ataque.

Como trabalhos futuros, espera-se a inclusão de diferentes táticas de ataque como *carpet bombing* e *pulse attack*, além do avanço da ferramenta com a elaboração de uma interface gráfica e a inclusão de novos protocolos. Também espera-se criar um *scanner* de refletores, o que tornaria a ferramenta mais próxima do contexto de ataques DDoS reais. Já está sendo desenvolvido um *fork* para DDoS por *flooding*.

Agradecimentos

Os autores agradecem a Igor Miranda, Alexander Vieira, Rodrigo Saldanha e Pedro Henrique Pereira por colaborarem nas versões iniciais da ferramenta, e à GigaCandanga REDECOMEP-DF por seu suporte tecnológico.

Referências

Gondim, J. and Albuquerque, R. d. O. (2019). Mirror saturation in amplified reflection ddos. Actas de las V Jornadas Nacionales de Ciberseguridad; Junio 5-7, 2019, Cáceres,

España. Ed. Caro Lindo, Andrés and García Villalba, Luis Javier and Sandoval Orozco, Ana Lucila, Universidad de Extremadura, Servicio de Publicaciones.

Gondim, J. J. C., de Oliveira Albuquerque, R., Clayton Alves Nascimento, A., García Villalba, L., and Kim, T. H. (2016). A methodological approach for assessing amplified reflection distributed denial of service on the internet of things. *Sensors*, 16(11):1855.

Gondim, J. J. C., de Oliveira Albuquerque, R., and Sandoval, O. A. L. (2020). Mirror saturation in amplified reflection distributed denial of service: A case of study using snmp, ssdp, ntp and dns protocols. `https//doi.org/10.1016/j.future.2020.01.024`. Future Generation Computer Systems.

Hilton, S. (2016). Dyn analysis summary of friday october 21 attack. `https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/`. visitado em: 08/04/2020.

Mahjabin, T., Xiao, Y., Sun, G., and Jiang, W. (2017). A survey of distributed denial-of-service attack, prevention, and mitigation techniques. *International Journal of Distributed Sensor Networks*, 13(12):1550147717741463.

Paxson, V. (2001). An analysis of using reflectors for distributed denial-of-service attacks. *ACM SIGCOMM Computer Communication Review*, 31(3):38–47.

Peng, T., Leckie, C., and Ramamohanarao, K. (2007). Survey of network-based defense mechanisms countering the dos and ddos problems. *ACM Comput. Surv.*, 39(1).

Riza, A., Yusof, R., Udzir, N., and Selamat, A. (2019). Systematic literature review and taxonomy for ddos attack detection and prediction. *International Journal of Digital Enterprise Technology*, 1:292.

Rossow, C. (2014). Amplification hell: Revisiting network protocols for ddos abuse. In *21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26, 2014*.

Vasques, A. T. and Gondim, J. J. C. (2019). Amplified reflection ddos attacks over iot mirrors: A saturation analysis. 2019 Workshop on Communication Networks and Power Systems (WCNPS). IEEE.

Vasques, A. T. and Gondim, J. J. C. (2020). Ataques ddos por reflexão amplificada sobre refletor iot rodando coap. 2020 15th Iberian Conference on Information Systems and Technologies (CISTI). Submetido e Aceito.

Apêndice B

Registro de Software do Linderhof

REPÚBLICA FEDERATIVA DO BRASIL

MINISTÉRIO DA ECONOMIA

INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL

DIRETORIA DE PATENTES, PROGRAMAS DE COMPUTADOR E TOPOGRAFIAS DE CIRCUITOS INTEGRADOS

Certificado de Registro de Programa de Computador

Processo Nº: **BR512020001705-3**

O Instituto Nacional da Propriedade Industrial expede o presente certificado de registro de programa de computador, válido por 50 anos a partir de 1° de janeiro subsequente à data de 15/04/2020, em conformidade com o §2°, art. 2° da Lei 9.609, de 19 de Fevereiro de 1998.

Título: Linderhof versão 1.0.0 - Ferramenta para avaliação de sistemas de mitigação de ataques reflexivos volumétricos (DDoS)

Data de publicação: 15/04/2020

Titular(es): FUNDAÇÃO UNIVERSIDADE DE BRASILIA

Autor(es): JOÃO JOSÉ COSTA GONDIM; PEDRO HENRIQUE MORAIS PEREIRA; ALEXANDER ANDRÉ DE SOUZA VIEIRA; MATHEUS DE OLIVEIRA VIEIRA; RODRIGO DE SOUSA SALDANHA; AMANDA LOPES DANTAS; ALAN TAMER VASQUES; IGOR FERNANDES MIRANDA

Linguagem: C

Campo de aplicação: IF-07

Tipo de programa: PD-01

Algoritmo hash: SHA-512

Resumo digital hash:
17DF9F923C71FE8C51FF8A10225D20C5B5AEE71AF40C2805B49E5ACF7329E8F68A295E314331465AA181DE881
55B86B6A24EBEA91C51DBE3043D86610D033192

Expedido em: 01/09/2020

Aprovado por:
Helmar Alvares
Chefe da DIPTO - Portaria/INPI/DIRPA Nº 09, de 01 de julho de 2019