Linderhof v2.0.0

Matheus de O. Vieira[1], **Amanda L. Dantas**[1]
Alan T. Vasques[2], **João J. C. 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

{matheusov,amandadantas19,alantamer}@gmail.com, gondim@unb.br

Abstract. *This paper describes a dual-application tool for studying volumetric attacks and also evaluation and benchmarking volumetric distributed denial of service attack mitigation systems, specifically amplified reflection attacks. It implements amplification attacks abusing several protocols, under customized attack conduction tactics and controlled intensity, providing a user friendly interface.*

Resumo. *Este artigo descreve uma ferramenta de aplicação dual voltada ao estudo da dinâmica de ataques volumétricos e também para avaliação e benchmarking de soluções de mitigação contra ataques volumétricos distribuídos de negação de serviço, em especial os por reflexão amplificada. Ela implementa os ataques por reflexão abusando vários protocolos, disponibilizando táticas de condução de ataque e controle de intensidade de forma customizada, provendo uma interface gráfica amigável.*

1. Introdução

Linderhof é uma ferramenta de uso dual que tem por finalidade o estudo da dinâmica de ataques volumétricos distribuídos de negação de serviço (DDoS), especificamente os ataques DDoS por reflexão amplificada e também para a avaliação e *benchmarking* de soluções de mitigação contra ataques volumétricos DDoS. Trata-se de uma ferramenta de geração de ataques AR-DDoS abusando diversos protocolos e possibilitando o controle total da condução do ataque. A ferramenta tem sido utilizada para o estudo do comportamento da saturação dos refletores e permite também a avaliação de soluções de mitigação dos ataques em questão.

A versão inicial do Linderhof foi apresentada em [Dantas et al. 2020], gerando Registro de Programa de Computador (RPC), junto ao Instituto Nacional de Propriedade Industrial - INPI (processo "BR512020000389-3", solicitado em 15/04/2020 e expedido em 01/09/2020), e foi utilizada em vários trabalhos ([Gondim et al. 2016], [Gondim and de Oliveira Albuquerque 2019] e [Vasques and Gondim 2019]). Este trabalho apresenta sua nova versão, a v2.0.0, que mesmo durante desenvolvimento deu suporte ao estudos [Gondim et al. 2020], [Vasques 2020] e [Vasques and Gondim 2020]. Entre as melhorias estão a criação de uma interface gráfica que ajuda na usabilidade da aplicação,

a funcionalidade de procurar na rede por dispositivos vulneráveis que possam ser utilizados como refletores e a adição de novas formas de controle do ataque. Além disso, foi adicionado o CLDAP à lista de protocolos suportados pela ferramenta.

Este artigo está organizado da seguinte maneira: a Seção 2 apresenta a arquitetura e as funcionalidades da ferramenta, tanto as presentes na versão v1.0.0 quanto as adicionadas na versão atual, a v2.0.0. Na Seção 3 é explicada como será feita a demonstração da utilização da ferramenta no salão bem como é mostrado o local onde o código da ferramenta está disponível. Na Seção 4 é realizado um teste de desempenho da ferramenta em um pequeno caso de uso. Finalmente, a Seção 5 fecha com as conclusões.

2. Arquitetura e Funcionalidades

A ferramenta Linderhof foi desenvolvida na linguagem C e até o momento possui suporte aos protocolos CoAP, DNS, Memcached, NTP, SNMP, SSDP e CLDAP, sendo este último adicionado nesta segunda versão. Os pacotes destes protocolos são gerados pela ferramenta e enviados aos refletores que utilizam de determinados serviços que funcionam sobre esses protocolos e geram uma resposta ainda maior para as vítimas. Dado esse comportamento de reflexão, os refletores também são chamados de *mirrors*, característica que dá nome a ferramenta devido à Galeria de Espelhos presente no palácio Linderhof na Alemanha.

A versão atual do LInderhof não sofreu grandes mudanças em sua arquitetura em comparação com a versão v1.0.0 apresentada em [Dantas et al. 2020], sendo somente acrescentado o módulo de *Scanner*. Nas funcionalidades é onde estão as maiores novidades, e devido a isso elas serão apresentadas de maneira separada das originais.

2.1. Arquitetura

O Linderhof é composto por 5 módulos: Interface, *Commander*, *Hall of Mirrors*, *Injector* e o *Scanner*. A arquitetura é mostrada na Figura 1 e os módulos descritos a seguir.

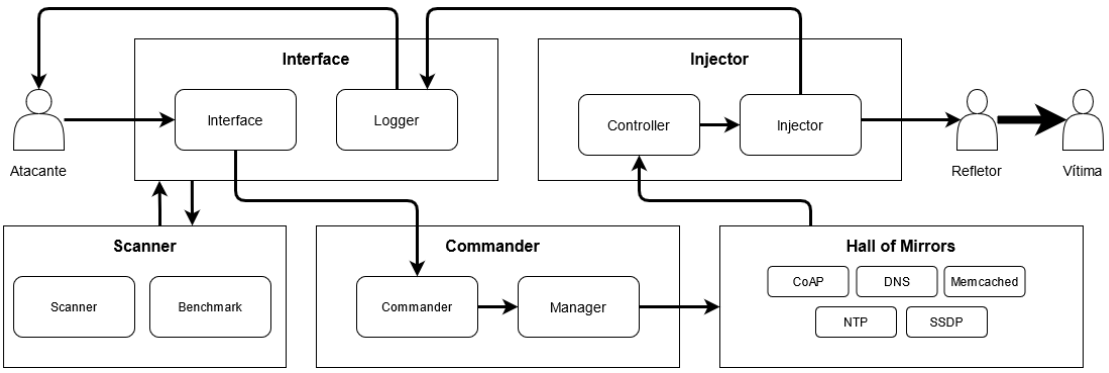


Figura 1. Arquitetura do Linderhof

- **Interface:** Módulo de interação com o usuário. 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):** Local estão implementadas as funções que geram os pacotes de cada um dos protocolos;
- **Injector:** 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.
- **Scanner:** Realiza a busca por refletores em uma sub-rede especificada. O submódulo *Benchmark* realiza um teste para verificar a amplificação realizada pelo refletor.

2.2. Funcionalidades

A Figura 2 lista todos os parâmetros aceitos pelo Linderhof através da linha de comando (Figura 2a), que também podem ser designados por arquivo de configuração (Figura 2b). As principais funcionalidades serão detalhadas nas subseções seguintes, divididas entre as funções originais e as adicionadas nesta nova versão.

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

(a) Em linha de comando

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
```

(b) Em arquivo de configuração

Figura 2. Parâmetros de configuração

2.2.1. Funcionalidades da v1.0.0

Os parâmetros obrigatórios do Linderhof são o *mirror* que define o protocolo que se deseja explorar no ataque, o endereço e opcionalmente a porta dos refletores e da vítima, com o qual é feito o *spoofing*. Para controlar a taxa de injeção dos pacotes é possível informar o nível do ataque, valor de 1 a 10 que define a quantidade de pacotes que serão enviados por segundo de acordo com a Equação 1. É possível definir para que essa nível seja elevado gradualmente e também definir a duração total do ataque.

$$Pacotes\ por\ segundo = 10^{\text{nível-1}} \tag{1}$$

Os pacotes gerados pela ferramenta são divididos igualmente entre todos os refletores envolvidos no ataque. Porém, é possível fazer com que a taxa definida na Eq. 1 seja enviada integralmente para cada um dos refletores ao utilizar o modo agressivo. O modo *flooding* remove a limitação imposta pelos níveis, enviando assim o máximo de pacotes suportados pela ferramenta e máquina utilizada, sem que seja feito nenhum controle de taxa de envio presentes na ferramenta.

Existem também os parâmetros responsáveis por definir características específicas de cada protocolo e pacote gerados pelo Linderhof, que podem possibilitar que os pacotes sejam aceitos pelo refletor ou até mesmo permitir uma maior amplificação. A configurações podem ser salvas em um arquivo de texto para reuso em diferentes execuções (Figura 2b).

2.3. Funcionalidades da v2.0.0

As funcionalidades introduzidas na versão 2.0.0 são descritas a seguir.

2.3.1. Taxa customizável

Ataques de negação de serviço comumente são vistos com diferentes taxas de intensidade ao longo do ataque. Para replicar esse comportamento adicionou-se a funcionalidade que permite que essa taxa seja variável ao longo do tempo, fazendo com que o gráfico de sua intensidade forme diferentes padrões. A intensidade é informada em um arquivo onde cada linha informa a quantidade de pacotes que deve ser enviado a cada segundo. Ao final do arquivo a leitura é reiniciada, repetindo o padrão, seja ele uma onda quadrada, triangular ou senoidal, por exemplo.

Um tipo de ataque que pode ser executado com essa funcionalidade é o *Pulse Wave* [DDoS-GUARD 2019], onde suas ondas podem ser descritas no arquivo informando a quantidade de pacotes a serem enviados durante o pulso e repetindo esse valor tantas vezes se queira que ele dure, seguindo dos valores zero para representar seu repouso. A ferramenta então passará a enviar pulsos de ataque repetidamente, assemelhando-se ao *Pulse Wave*.

2.3.2. Ataque em múltiplos endereços na sub-rede

Adicionou-se também o suporte a múltiplas vítimas em um mesmo ataque. Ao inserir uma faixa de endereços IP na notação CIDR [Fuller et al. 1993], é feita a expansão dos endereços representados e o ataque passa a utilizá-los. As vítimas são alternadas a cada segundo, podendo ser escolhidas de forma sequencial ou aleatória. Essa funcionalidade abre espaço para o estudo de técnicas de ataque como o *Carpet Bombing* [NETSCOUT 2020], que busca evitar detecções basedas em limites e estatística ao dividir o ataque entre diversos endereços IP dente de um mesmo bloco.

2.3.3. Scanner

Parte do trabalho de realizar ataques DDoS é a busca por refletores. Implementou-se então a funcionalidade de realizar esta busca por dispositivos que executam os protocolos

suportados pela ferramenta e possuam configurações que permitem serem utilizados para tal finalidade.

Criou-se o módulo *Scanner* para comportar a funcionalidade de buscar por refletores em um CIDR informado. Essa busca difere-se de um ataque propriamente dito por pela baixa taxa de injeção de pacotes e por não ser feito o *IP spoofing*, já que os dispositivos devem responder diretamente para a máquina que está realizando o *scan*. Os dispositivos encontrados são salvos em um arquivo de texto, que pode ser informado posteriormente no parâmetro *reflector*.

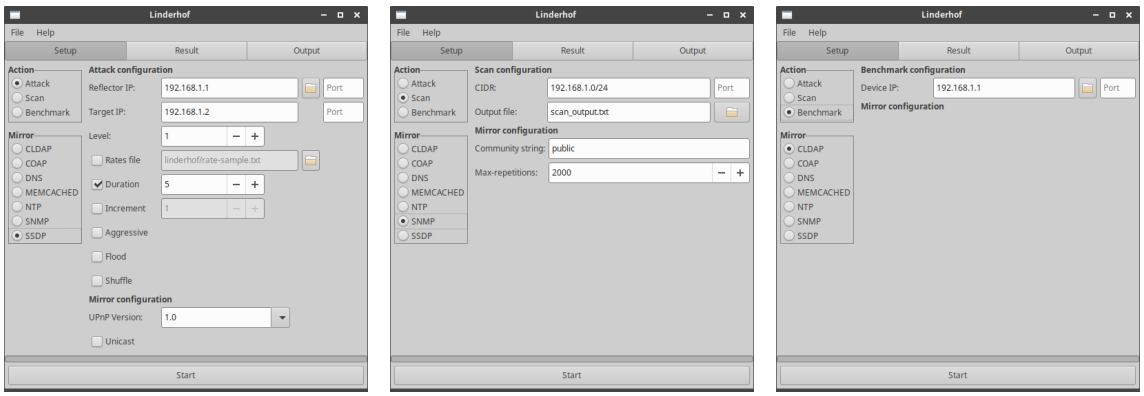
Também foi adicionado ao *scanner* o submódulo *benchmark*, que realiza uma verificação para testar a capacidade de um dispositivo em amplificar um pacote de determinado protocolo. Assim como o *Scanner*, não é feito o *IP spoofing*. Adicionou-se então, em cada um dos mirrors no *Hall of Mirrors*, comandos de geração dos pacotes para cada cenário testado no *benchmark*.

2.3.4. Interface gráfica

Uma interface gráfica foi adicionada à ferramenta com o objetivo de facilitar o seu uso. A GUI foi desenvolvida com o conjunto de bibliotecas do *GTK* [GTK Team 2021] juntamente com o software *Glade* [The Glade project 2021]. Seu funcionamento foi escrito em C, e seus elementos são descritos por um arquivo XML, gerado pelo *Glade*.

O funcionamento do Linderhof através da linha de comando continua presente, a interface gráfica é somente uma alternativa ao seu uso. Todas os parâmetros presentes através da linha de comando também estão presentes no GUI. As configurações estão presentes na Aba *Setup* e divididas entre os módulos *Attack*, *Scan* e *Benchmark*, e todos eles contam com a opção de selecionar o protocolo que deseja-se utilizar.

No módulo *Attack*, exibido na Figura 3a, estão presentes todas as opções disponíveis para customizar o ataque, bem como as opções relativas ao *mirror* escolhido.



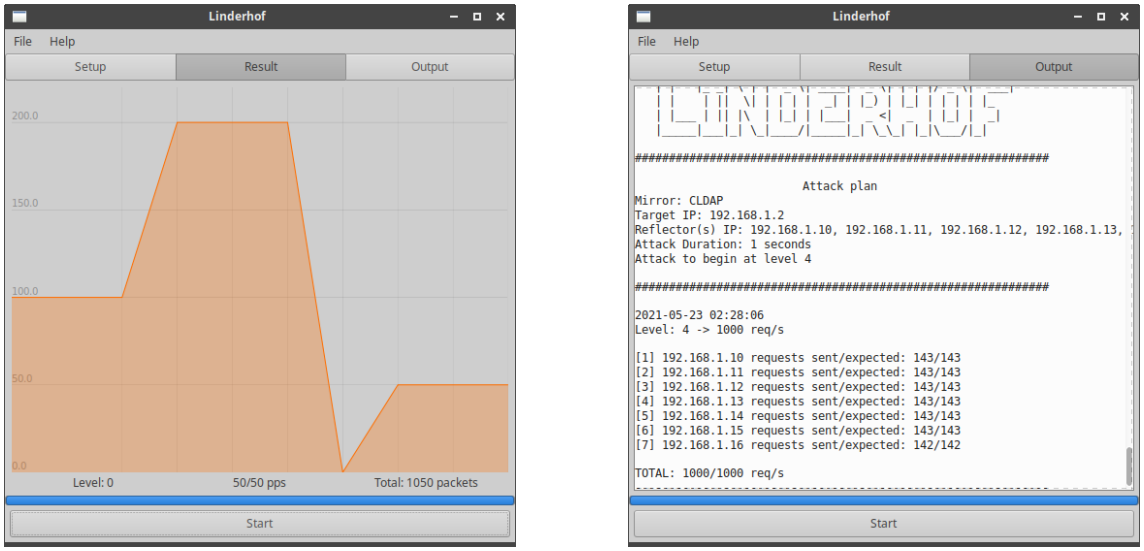
(a) Attack **(b) Scan** **(c) Benchmark**

Figura 3. Telas da interface gráfica

No módulo *Scan* da Figura 3b está presente somente as opções de inserir o CIDR e porta que deseja-se escanear, além do campo para escolher o caminho onde os endereços dos dispositivos encontrados devem ser salvos.

Por fim, o módulo *Benchmark* da Figura 3c exibe a opção de inserir o endereço e a porta do dispositivo a ser testado.

As abas *Result* e *Output* exibem o andamento do ataque. A aba *Result* da Figura 4a apresenta o gráfico da quantidade de pacotes enviados pelo atacante ao longo do ataque, o total de pacotes enviados e o nível em que encontra-se o ataque. A aba *Output* contém a saída da ferramenta em forma textual, assim como acontece ao executá-la através da linha de comando, mostrando o nível, a quantidade total de pacotes enviados no último segundo e quantidade de pacotes enviados para cada refletor, como exibido na Figura 4b.



(a) Result **(b) Output**

Figura 4. Abas da interface gráfica

O Linderhof possui então duas formas de utilização. Ao executá-lo sem nenhum parâmetro a interface gráfica é exibida. Porém, adicionando qualquer parâmetro à execução, a versão de linha de comando é utilizada.

2.3.5. Implementação de ataque abusando de outros protocolos

A implementação do ataque abusando um protocolo é totalmente modular, com uma padronização facilita a adição de novos *mirrors* à ferramenta. Esforço de implementação consiste em montar os pacotes e adicionar as customizações de seus parâmetros. Fora as modificações do HOM, ainda é necessário 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 implementação do ataque abusando do protocolo CLDAP seguiu este padrão.

3. Desempenho

Os resultados de um teste de ataque são mostrados na Tabela 1. Os dados se referem ao protocolo CLDAP, sobre um computador com a seguinte especificação: CPU Intel i5 8250U @ 1.6 GHz com 8GB DDR4, @ 2400MHz, placa de rede ASIX AX88179 USB

3.0 Gigabit e sistema operacional Xubuntu 20.04.1. Foi gerado um ataque com níveis incrementais e de duração de 70 segundos (10s por nível).

Como pode-se verificar, a geração de pacotes satura deixando de acompanhar a taxa especificada a partir do nível 6 com cerca de 30,6 kpps e 22 Mbps enviados, e atinge um pico de pouco mais de 95 kpps e quase 70 Mbps no nível 7. Essa saturação ocorre devido às limitações da máquina e rede utilizadas. A ferramenta informa a quantidade de pacotes que foi capaz de gerar, não necessariamente a quantidade de fato enviada. Por isso, essa medição é realizada com um software específico para esse propósito, como o*Wireshark*.

Tabela 1. Pacotes e bits enviados por segundo.

Nível	Pacotes	bit/s	% do esperado
1	1	728	100,00
2	10	7280	100,00
3	100	72800	100,00
4	1000	728000	100,00
5	10000	7280000	100,00
6	30633	22300502	30,63
7	95437	69478354	9,54

4. Demonstração

A demonstração apresentará as principais funcionalidades da ferramenta executando uma sequência de ataque com intensidade incremental, incluindo como a intensidade pode ser modulada gerando diferentes ondas de ataque (Figura 5), como o ataque pulsado (*pulse attack*). Também será apresentada a técnica de *carpet bombing*.

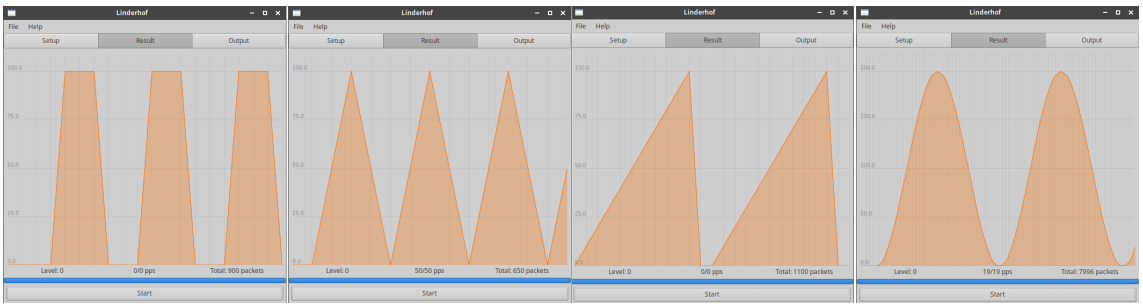


Figura 5. Formas de onda geradas pelo Linderhof

A saturação do atacante ocorre no nível 6, 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. 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.

5. Código e manuais

O código-fonte do Linderhof encontram-se no endereço **https://cyberseclab.gigacandanga.net.br/linderhof/sbseg21**. Ambos o usuário e

senha para acessar o repositório é "**sbseg2021**". O arquivo README contém as instruções para realizar a compilação e execução do código. A documentação da ferramenta está contida no repositório, dentro da pasta **docs**.

6. Conclusão

Este trabalho apresentou melhorias na ferramenta Linderhof, consolidadas na versão 2.0.0, no sentido de expandir a abrangência sobre ataques DDoS mais populares, novos protocolos, provendo maior controle na condução dos ataques, de forma customizada e facilitando o uso por meio de uma interface gráfica.

Entre as novas funcionalidades inseridas no Linderhof, vale ressaltar a 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*.

Referências

Dantas, A. L., de Oliveira Vieira, M., Vasques, A. T., and Gondim, J. J. C. (2020). Linderhof: uma ferramenta para avaliação de sistemas de mitigação de ataques reflexivos volumétricos (ddos). In *Anais Estendidos do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 25–32. SBC.

DDoS-GUARD (2019). Hidden threat of pulse wave ddos attacks. `https://ddos-guard.net/en/info/blog-detail/hidden-threat-of-pulse-wave-ddos-attacks`.

Fuller, V., Li, T., Yu, J. J. Y., and Varadhan, K. (1993). Classless inter-domain routing (cidr): an address assignment and aggregation strategy. RFC 1519, RFC Editor.

Gondim, J. J. and de Oliveira Albuquerque, R. (2019). Mirror saturation in amplified reflection ddos. In *Actas de las V Jornadas Nacionales de Ciberseguridad: junio 5-7, 2019. Cáceres*, pages 185–190. 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.

GTK Team (2021). Gtk. `https://www.gtk.org`.

NETSCOUT (2020). Defending against carpet bombing attacks. `https://www.netscout.com/use-case/carpet-bombing-attacks`.

The Glade project (2021). Glade. `https://glade.gnome.org`.

Vasques, A. T. (2020). Análise de saturação de dispositivos iot atuando como refletores em ataques distribuído de negação de serviço por reflexão amplificada. `https://repositorio.unb.br/handle/10482/40089`. Dissertação (mestrado) — Universidade de Brasília, Faculdade de Tecnologia.

Vasques, A. T. and Gondim, J. J. C. (2019). Amplified reflection ddos attacks over iot mirrors: A saturation analysis. In *2019 Workshop on Communication Networks and Power Systems (WCNPS)*, pages 1–6.

Vasques, A. T. and Gondim, J. J. C. (2020). Amplified reflection ddos attacks over iot reflector running coap. In *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6.