



**MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA INSTITUTO
FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO TOCANTINS -
CAMPUS ARAGUATINS
CURSO DE LICENCIATURA EM COMPUTAÇÃO**

LUCAS ERBERT SALES DE SOUZA

**Challenger DBN - botnet detection: uma ferramenta para detecção de botnet
com base na análise de comportamento do tráfego de rede**

**ARAGUATINS
2022**



**MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA INSTITUTO
FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO TOCANTINS -
CAMPUS ARAGUATINS
CURSO DE LICENCIATURA EM COMPUTAÇÃO**

LUCAS ERBERT SALES DE SOUZA

**Challenger DBN - botnet detection: uma ferramenta para detecção de botnet
com base na análise de comportamento do tráfego de rede**

Trabalho de Conclusão de Curso apresentado à
Coordenação do Curso de Licenciatura em
Computação do Instituto Federal do Tocantins –
Campus Araguatins, como exigência à obtenção
do grau de graduação em Licenciatura em
Computação.

Orientador (a): Prof. Me. Ancelmo Frank Coelho
Castro Melo.

**ARAGUATINS
2022**

Dados Internacionais de Catalogação na Publicação (CIP)
Bibliotecas do Instituto Federal do Tocantins

- S719c Souza, Lucas Erbert Sales de
Challenger DBN - botnet detection : uma ferramenta para detecção de botnet com base na análise de comportamento do tráfego de rede / Lucas Erbert Sales de Souza. – Araguatins, TO, 2022.
32 p. : il. color.
- Trabalho de Conclusão de Curso (Licenciatura em Computação) – Instituto Federal de Educação, Ciência e Tecnologia do Tocantins, Campus Araguatins, Araguatins, TO, 2022.
- Orientador: Me. Ancelmo Frank Coelho Castro
1. Segurança de rede. 2. Botnet. 3. Ataque DDoS. I. Castro, Ancelmo Frank Coelho. II. Título.

CDD 004

A reprodução total ou parcial, de qualquer forma ou por qualquer meio, deste documento é autorizada para fins de estudo e pesquisa, desde que citada a fonte.

Elaborado pelo sistema de geração automática de ficha catalográfica do IFTO com os dados fornecidos pelo(a) autor(a).



Ministério da Educação
Secretaria de Educação Profissional e Tecnológica
Instituto Federal de Educação, Ciência e Tecnologia do Tocantins
Campus Araguatins

FOLHA DE APROVAÇÃO

TÍTULO: CHALLENGER DBN - BOTNET DETECTION: UMA FERRAMENTA PARA DETECÇÃO DE BOTNETCOM BASE NA ANÁLISE DE COMPORTAMENTO DO TRÁFEGO DE REDE

AUTOR: LUCAS ERBERT SALES DE SOUSA

ORIENTADOR: Prof. Me. ANCELMO FRANK COELHO CASTRO

Trabalho de Conclusão de Curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia do Tocantins, *Campus Araguatins*, como parte das exigências para a conclusão do Curso de Licenciatura em Computação.

Aprovada em 07 de dezembro de 2022.



Documento assinado eletronicamente por **Rogério Pereira de Sousa, Servidor**, em 07/12/2022, às 16:00, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Ancelmo Frank Coelho Castro, Servidor**, em 07/12/2022, às 16:06, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Adeilson Marques da Silva Cardoso, Servidor**, em 13/02/2023, às 15:53, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ifto.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1827577** e o código CRC **8F7837DE**.

Criado por 2489616, versão 2 por 2489616 em 07/12/2022 16:00:18.

AGRADECIMENTOS

Agradeço ao prof. Me. Ancelmo Frank Coelho Castro, orientador que auxiliou no processo final, disponibilizando seu tempo, recurso e conhecimentos essenciais para tornar esse sonho possível.

Agradeço ao prof. Me. Adeilson Marques da Silva Cardoso, responsável pelas orientações precisas no período de pré-projeto.

Agradeço ao senhor Nelson Brito, desenvolvedor da ferramenta de ataque DDoS, que prestou esclarecimentos importantes sobre a ferramenta ao qual foi utilizada.

Agradeço a equipe de suporte do *framework BYOB* no *Discord* que prestou apoio durante todo o processo de configuração tirando dúvidas e apontando soluções.

Agradeço a Flaviana de Moura pelo suporte prestado pelo curto período de coorientação.

Um agradecimento especial a prof. Me. Marinara Cabral, que sempre me deu a maior força tanto psicológica quanto estrutural, cedendo seus recursos para conclusão do projeto e aconselhou e torceu muito pelo meu sucesso.

No ensejo, gostaria de aproveitar a oportunidade para mandar aquele salve para a galera do G.A.R.O.D.Y. É isso, rapaziada, finalmente formei.

“O que sabemos é uma gota; o que ignoramos é um oceano.”
(Isaac Newton)

RESUMO

Apesar das *botnets* terem sido criadas há mais de 25 anos, nunca houve uma solução definitiva para o problema devido à complexidade organizacional que explora os meios legais de tráfego de rede. Ao infectar os mais variados dispositivos conectados a internet, um exército de robôs zumbis torna-se escravo das vontades do seu mestre, sejam elas quais forem. Muitos utilizam o poder que lhes é adquirido para explicar falhas em sistemas diversos a fim de saná-las. Outros têm intenções não tão nobres e utilizam de tal estrutura para indisponibilizar serviços através da chama inundação, os motivos são os mais variados e podem ser insatisfação com o serviço alvejado, concorrência, puro *status* ou simplesmente por fazê-lo. Para tanto, torna-se essencial o desenvolvimento de alternativas para evitar esse problema, sistemas de detecção tornam-se cada dia mais necessários visto que a democratização das *botnets* é um processo real. Assim sendo, este trabalho tem por objetivo viabilizar a projeção de uma aplicação que seja capaz de detectar *botnets* a fim de evitar prejuízos.

Palavras-chave: Segurança de rede. Ataque DDoS. Detecção. Botnet.

ABSTRACT

Although botnets have been created for over 25 years, there has never been a definitive solution to the problem due to the organizational complexity that exploits the legal means of network traffic. By infecting the most varied devices connected to the internet, an army of zombie robots becomes a slave to their master's will, whatever they may be. Many use the power acquired to explain failures in different systems in order to remedy them. Others have not so noble intentions and use such a structure to make services unavailable through flooding. Therefore, it is essential to develop alternatives to avoid this problem, detection systems become more and more necessary since the democratization of botnets is a real process. Therefore, this work aims to enable the design of an application that is capable of detecting botnets in order to avoid losses.

Keywords: Network security. DDoS attack. Detection. Botnet.

SUMÁRIO

1 INTRODUÇÃO	10
2 PROBLEMATICA	11
3 JUSTIFICATIVA.....	12
4 OBJETIVOS.....	13
4.1 Objetivo Geral.....	13
4.2 Objetivos Específicos	13
5 REVISÃO DA LITERATURA	14
5.1 Botnet.....	14
5.2 Biblioteca JPCAP	15
5.3 Ataque de Negação de Serviços Distribuído	16
6 PROCEDIMENTOS METODOLÓGICOS.....	19
6.1 Área de estudo	19
6.2 Etapa um: análise de requisitos.....	19
6.3 Etapa dois: estruturação de uma botnet <i>BYOB</i>.....	21
6.4 Etapa três: engenharia e elaboração de requisitos do Challenger DBN	23
6.5 Etapa quatro: interface Challenger DBN	23
6.6 Etapa cinco: materiais e métodos	24
7 RESULTADOS E DISCUSSÕES.....	26
8 CONSIDERAÇÕES FINAIS	30

1 INTRODUÇÃO

Com o passar dos anos, a informação evoluiu de diversas formas no que diz respeito ao armazenamento, porém, as formas de assegurar a integridade desta passaram a exigir métodos mais sofisticados para garantir sua eficiência. A cada novo passo dessa evolução, os mecanismos de invasão paralelamente aprimoram-se tornando tais evoluções um percurso infinito (SILVA, 2016).

Os sistemas de proteção à informação são imprescindíveis para as grandes empresas e demais organizações, pois garantem o sigilo e a harmonia nas suas negociações, transações no geral e executam-se de forma transparente ao usuário comum. A informatização desses sistemas é importante, pois reduz potencialmente as probabilidades de prejuízos das mais variadas naturezas devido à possibilidade de falha humana.

Dentre os vários tipos de ataque, encontram-se os ataques provocados por meio de *botnet* (*Robot Network*) que são popularmente conhecidos como “*DDoS* (*Distributed Denial of Service*)” ou ataque de negação de serviços distribuído. Esse tipo de atividade tem por finalidade interromper um serviço por meio de ataque utilizando uma grande quantidade inserção de dados a partir do envio de inúmeras requisições de conexão. As *Botnet's* podem ser detectáveis através de análise de dados obtidos por meio do tráfego da rede (BARBOSA et al., 2014).

JPCAP (*Network Packet Capture Facility for Java*) é uma biblioteca escrita em Java que fornece uma interface e um sistema para captura de pacotes de rede (...). Utiliza o *libpcap*, uma biblioteca do sistema amplamente implementada para captura de pacotes (SOURCEFORGE, 2013). Desse modo, a *Jpcap* pode viabilizar a identificação de *botnet's* através da análise do tráfego capturado.

Em função disso, o presente trabalho tem por objetivo projetar e desenvolver a ferramenta *Challenger* DBN (*Challenger detection Botnet* ou Desafiante detecção de robôs de rede), uma ferramenta para análise de dados do tráfego de rede e detecção de *botnet's* com base nos pacotes de rede coletados por meio da biblioteca *JPCAP*, através de uma interface que opera em modo promiscuo em três etapas de execução, no Campus Araguatins do Instituto Federal de Educação, Ciência e tecnologia do Tocantins.

2 PROBLEMATICA

A segurança da informação é um tema de grande relevância, tendo em vista o crescente avanço na utilização de novas tecnologias, sendo assim vários questionamentos são levantados quanto aos métodos e técnicas utilizadas para proteção dessas informações. Partindo-se desse contexto, o presente trabalho aponta os seguintes questionamentos: Como detectar *botnet's* com base na análise de comportamento do tráfego da rede?

3 JUSTIFICATIVA

Ao longo dos anos foi crescente e significativo o emprego das redes de computadores, como afirma Silva (2016, pag. 97), “Atualmente não é possível pensar em desenvolver qualquer tipo de aplicação, seja para computador, celular, televisão, ou quaisquer outros dispositivos, que não utilize algum tipo de rede”, dessa forma destaca-se como crescente e significativa também a vulnerabilidade dessas redes e as ameaças existentes. Assim, é necessário desenvolver ferramentas, métodos e técnicas em prol de detectar essas ameaças, pois, na medida em que evoluem, geram novos desafios para a segurança dos computadores.

Para Silva (2016), a segurança pode ser compreendida como um processo cíclico, iniciando com a descoberta de novas ameaças, em seguida, a realização dos estudos e análises necessárias e, por fim, surgem as medidas de prevenção.

Um ataque de negação de serviço distribuído bem coordenado pode conter um “dano destrutivo” inimaginável. Segundo Cardoso (2018, pag. 27): “As ameaças provocadas por ataques a serviços disponíveis na rede podem causar prejuízos, visto que podem comprometer a segurança da informação” uma vez que “(...) uma informação pode ser valiosa” (OLIVEIRA e BEM, 2015, pag. 2). Um ataque pode interromper um serviço por um determinado período com uma quantidade exorbitante de requisições falsas, esgotando os recursos do servidor em questão e impedindo conexões reais que podem significar perda de recursos financeiros.

Diante desse contexto, o trabalho justifica-se pela necessidade de analisar os dados do tráfico da rede e detectar possíveis *botnet's* por meio da utilização do *Challenger DBN* para análise de dados obtida através da ferramenta *JPCAP*.

O trabalho demonstra a efetividade dos mecanismos de detecção mencionados bem como beneficiará a instituição com a implementação dos mesmos. Contudo, as informações e resultados obtidos com o desenvolvimento do projeto poderão servir de base para outras pesquisas e para o aprimoramento da segurança da rede utilizada.

4 OBJETIVOS

4.1 Objetivo Geral

Estruturar uma *botnet* para testar uma ferramenta capaz de constatar ataques DDoS para viabilizar o Challenger DBN – uma aplicação capaz de detectar *botnet*'s com base na análise de comportamento do tráfego da rede utilizando a biblioteca JPCAP.

4.2 Objetivos Específicos

- Implementar uma *botnet* para testes de detecção de ataques DDoS
- Criar regras para detecção de *botnet*'s com base no comportamento no tráfego;
- Analisar o tráfego de rede através de ferramentas para detecção de *botnet* disponíveis;
- Avaliar a projeção do *Challenger DBN* através de simulações em ambientes de teste controlado, e
- Identificar possíveis *botnet*'s e emitir alertas;

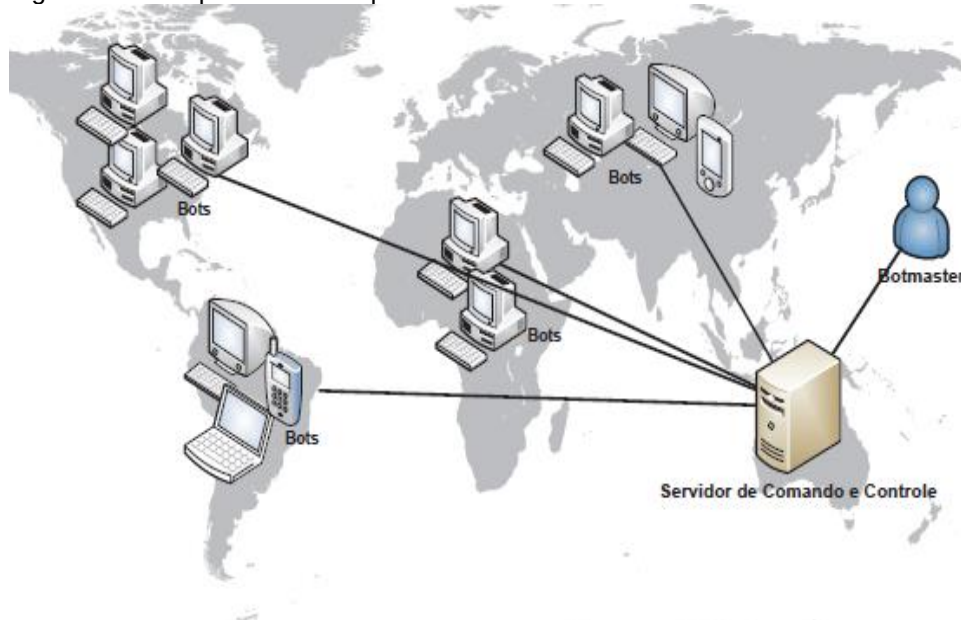
5 REVISÃO DA LITERATURA

5.1 Botnet

Uma *botnet* possui várias utilidades e detém um poder que pode ser inimaginável dependendo do tamanho da estrutura, segundo Silva (2016), pode ser conceituada como uma rede de máquinas infectadas por um software malicioso denominado “*bot*” oriunda da palavra *robot* (robô) e controlada remotamente via internet (*network*) por um ou mais atacantes (*botmaster*). Entretanto, as máquinas contaminadas não sofrem quaisquer alterações em seu padrão comportamental devido esse vírus, porém, tornam-se “zumbis” e estão paralelamente sujeitas as ordens do atacante.

Barbosa et. al (2014) descreve que essa infraestrutura permite que os *bots* se conectem a uma entidade de controle, que pode ser centralizada (Figura 1) ou distribuída.

Figura 1 - Exemplo de uma arquitetura centralizada de *botnet*.

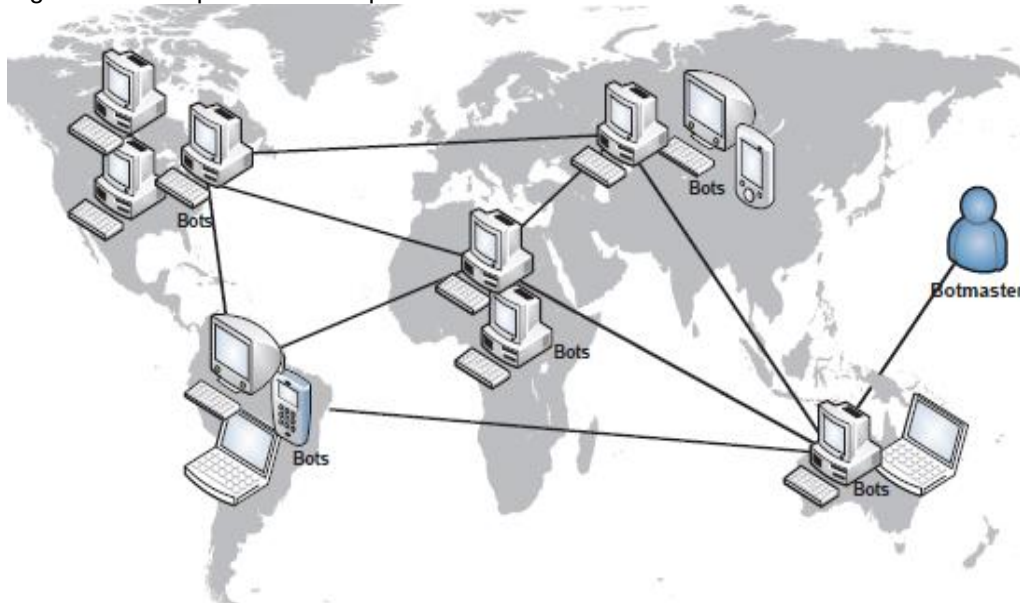


Fonte: Barbosa et.al. (2014)

As *botnet's* baseadas em arquiteturas centralizadas oferecem menor latência de comunicação das mensagens trocadas entre o *botmaster* e os *bots*, no entanto o principal problema é que o servidor de infraestrutura de comando e controle por si próprio é um ponto central de falhas (Barbosa et.al., 2014).

Quanto às *botnet's* baseadas em arquiteturas descentralizadas (Figura 2) Barbosa et.al. (2014, p. 107) destacam que as informações sobre os membros ou da própria *botnet* estão distribuídas ao longo de toda rede maliciosa. Por isso, quando as *botnet's* são “baseadas em arquiteturas descentralizadas são mais difíceis de serem desarticuladas, pois mesmo a descoberta de muitos *bots* não necessariamente significa a perda da rede da *botnet* inteira, visto que não existe um servidor central”.

Figura 2 - Exemplo de uma arquitetura descentralizada de *botnet*.



Fonte: Barbosa et al. (2014)

Apesar disso, Silva (2016) ressalta que as *botnet's* evoluíram bastante ao longo dos anos, tanto com modificações na arquitetura, no uso dos protocolos de rede, nos tipos de ação realizadas como na forma de comunicação, todavia a evolução em suas características tem como objetivo principal torná-las de difícil detecção o que se justifica pelo fato do mesmo não precisar obedecer nenhum protocolo oficial ou qualquer norma vigente, adaptando-se conforme surjam novos mecanismos de defesa e detecção.

5.2 Biblioteca JPCAP

JPCAP é uma biblioteca capaz de analisar diferentes protocolos possibilitando detecção dos tipos de ataque pela definição dos protocolos. Ainda, possibilita

desenvolvimentos de aplicações de captura de pacotes em Java (SOURCEFORGE, 2013). Essas aplicações fazem chamadas da biblioteca permitindo operação dos pacotes obtidos por meio do tráfego de rede.

Cardoso (2018, pag. 66) conceitua *JPCAP* como:

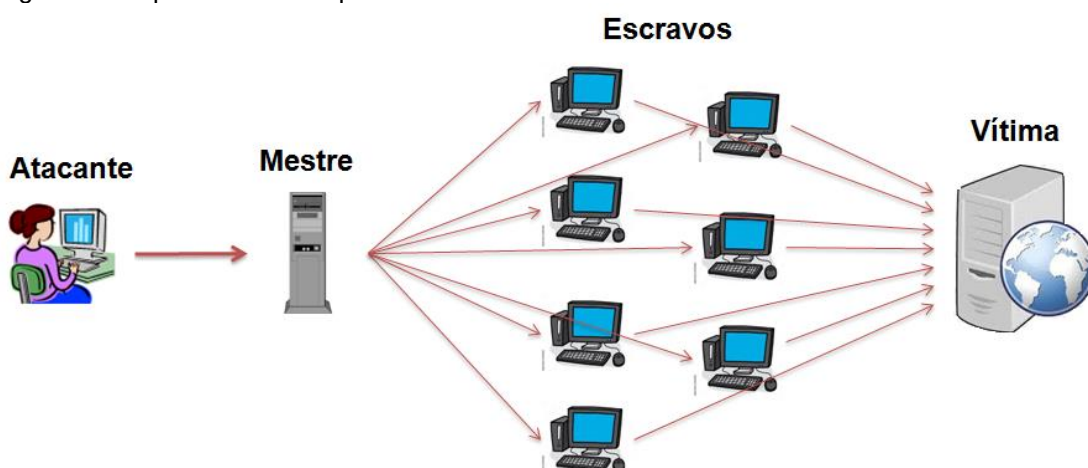
(...) uma biblioteca utilizada para capturar o tráfego de rede, e consiste em um conjunto de classes Java e interfaces que implementam chamadas para uma biblioteca do sistema libpcap. O Jpcap oferece captura de pacotes em tempo real, e uma opção para armazenar e ler pacotes capturados em arquivos offline, também permite a filtragem de pacotes por tipo de protocolo (IPv4, IPv6, ARP / RARP, TCP, UDP e ICMPv4).

5.3 Ataque de Negação de Serviços Distribuído

DDoS (*Distributed Denial of Service*) é um tipo de ataque cometido por uma rede de computadores ou qualquer dispositivo que utiliza protocolos de conexão, infectados por um vírus conhecido como “*bot*”, chamada de *botnet* (*Robot Network*) (Barbosa et al., 2014). A princípio, esse vírus, diferentemente dos demais, não tem função diretamente maliciosa, mas torna seu hospedeiro um servo do *botmaster* estando sujeito aos seus comandos.

No momento que o atacante executa o comando para realizar um ataque, toda a rede começa a enviar um número grande de requisições na intenção de criar falsas conexões com o servidor/vítima, saturando, assim, todo o sistema (Figura 2). Essa saturação promove o esgotamento dos recursos ou da banda do servidor dependendo do tipo de ataque (CARDOSO, 2018).

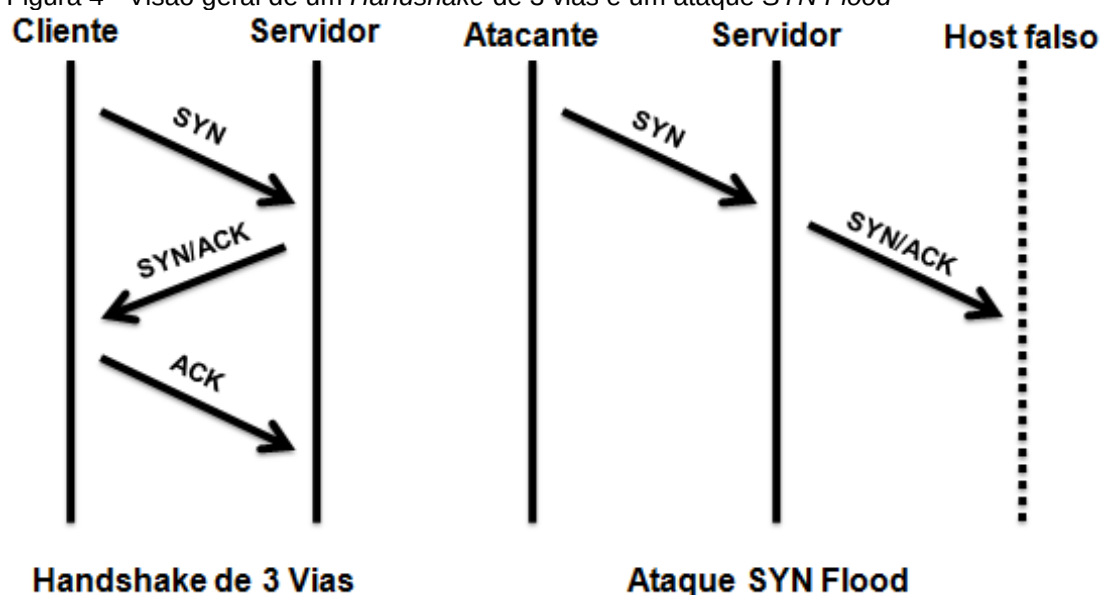
Figura 3 - Arquitetura de ataque *DDoS*



Fonte: clipart-library/mylovview/gratispng/

As ferramentas para esse tipo de processo, geralmente, contam com um recurso de mascaramento de *IP (Internet Protocol)*, desse modo, no momento do ataque, as requisições enviadas têm a finalidade de estabelecer a conexão por um computador com *IP* falso e, devido a isso, este não recebe a confirmação de sua solicitação, deixando a requisição em aberto (CARDOSO, 2018). Conforme o esquema (Figura 4):

Figura 4 - Visão geral de um *Handshake* de 3 vias e um ataque *SYN Flood*



Fonte: Adaptação de Cardoso (2018)

Para infelicidade dos administradores de sistemas de proteção, os ataques *DDoS* não se baseiam somente em um tipo de ataque, muito menos, uma única solução. Basicamente existem 3 (três) tipos de ataque *DDoS*: *SYN Flood*, *UDP Flood* e *ICMP Flood* (CERT.BR, 2016).

Os ataques *SYN Flood*, demonstrado na figura 4, funcionam durante o processo de *Handshake*, como destaca Cardoso (2018, p. 31):

Ataques por inundação TCP SYN, conhecido também como SYN Flood, é um dos ataques de negação de serviço mais usados. Nesse ataque, os invasores enviam tantos pedidos de conexão para um servidor que os usuários legítimos não podem se conectar a esse servidor.

De acordo com Cert.Br (2016) os ataques *UDP Flood* baseiam-se no protocolo *UDP*, estes tentam exaurir a banda disponível enviando ao alvo grande

volume de tráfego. Cardoso (2018) estipula uma taxa maior que 30 pacotes por segundo para confirmação de um ataque do tipo *UDP Flood*.

O *ICMP Flood* é discriminado por Sab et al. (2013) como inundação *ping*, ou *ping flood*. Este tipo de ataque se baseia no envio constante a partir de um endereço IP mascarado de uma grande quantidade de pacotes *echo request (ping)* até que o limite de *requests* por segundo seja ultrapassado.

É importante ressaltar que, neste último caso, o atacante necessita de certo acesso privilegiado ao sistema alvo e ainda contar com recursos mais relevantes. A finalidade desse tipo de ataque é esgotar a banda de sua vítima.

6 PROCEDIMENTOS METODOLÓGICOS

6.1 Área de estudo

A pesquisa será desenvolvida no *campus* Araguatins do Instituto Federal de Educação, Ciência e Tecnologia do Tocantins, localizado no município de Araguatins (figura 5), no extremo norte do Tocantins, com uma área demográfica de 2.625,286 km² e população de 31.329 habitantes (IBGE, 2010).

O ambiente de estudo se dará através de laboratório virtual, portanto, controlado. As máquinas contarão com um sistema operacional Linux/GNU Ubuntu 22.04 versão de servidor, em um total de 10 máquinas.

Figura 5 - Localização geográfica da cidade de Araguatins



Fonte: Google Maps (2022)

6.2 Etapa um: análise de requisitos

A maior esmagadora das grandes empresas e organizações utilizam a rede mundial de computadores para manutenção de sites, serviços, bancos de dados e etc. Um grande avanço no quesito praticidade, mas, por outro lado, surge o grande risco que são os golpes cibernéticos como exemplo do *DDoS*.

Segundo Duarte et al., 2015 (pag. 2):

Ataques de negação de serviços existem desde o surgimento das redes de computadores, mas eles não recebiam muita importância até atingirem provedores de serviços de Internet, grandes empresas e órgãos do governo. Com o uso crescente da Internet, obteve-se acesso aos servidores dessas entidades e os ataques tomaram maior visibilidade. Isso atraiu mais pessoas interessadas e métodos cada vez mais sofisticados foram sendo desenvolvidos.

Acerca dos primeiros ataques de negação de serviços, Duarte et al. (2015, pag. 2) prosseguem:

Em setembro de 1996, um ataque *DoS* do tipo *SYN Flood* derrubou o provedor de serviços de Internet de Nova Iorque, *Panix*, por uma semana, e ataques subsequentes derrubaram também os servidores do Internet *Chess Club* e do jornal *The New York Times*.

O primeiro *DDoS* reconhecido veio alguns anos depois, Duarte et al. (2015, pag. 2):

Em 1999, foi executado o primeiro ataque *DDoS* de destaque: *trinoo* (também conhecido como *trin00*). O ataque foi direcionado à Universidade de Minnesota em agosto de 1999, durou dois dias e envolvia o envio de uma quantidade maciça de pacotes UDP, originados de milhares de computadores.

Em termos de poder de ataque, o alvo foi o *GitHub*, a plataforma de desenvolvimento de software, fundada em 2008 com sede em São Francisco, Califórnia, EUA. De acordo com uma matéria do site *InfoMoney*, (2019):

Em fevereiro de 2018, o site sofreu um golpe com nada mais, nada menos que 1.35 Tbps de capacidade – recorde “oficial” até hoje, já que algumas empresas de segurança reportaram ataques maiores, mas não forneceram dados conclusivos ou mais detalhes sobre as ocorrências.

Os motivos que levam um ataque de negação distribuído de serviços a tornar um servidor inacessível são vários, tais como: “motivações ideológicas, comerciais, financeiras ou somente por demonstração de poder e prestígio e elevação do status do atacante” (BARDAL, 2014, pag. 9).

É imprescindível que os servidores e serviços estejam ativos e disponíveis permanentemente para atender às suas demandas, pois “a cada segundo que se passa sem restaurar as operações equivale a um custo de montagem, insatisfação do usuário e oportunidades perdidas” (FASTNETMON, 2022).

Em um levantamento realizado por Vieira (2019): “Os ataques *DDoS* se popularizaram na última década: estudos indicam que esse tipo de golpe quase

triplicou entre 2014 e 2017”. Devido a esse crescente número de incidentes perigosos à segurança de rede, principalmente dos sistemas financeiros em todo o mundo, se faz necessário a aquisição de ferramentas modernas e eficientes no combate e detecção de tais ofensivas.

6.3 Etapa dois: estruturação de uma botnet *BYOB*

Diferentemente de um ataque de negação de serviços, para o sucesso de um disparo de um ataque de negação de serviços distribuído se faz indispensável a utilização de uma *botnet*, uma vez que, nesse caso, o conceito do termo “distribuído” refere-se propriamente a estrutura da *botnet*.

O *framework BYOB (build your own botnet)* é uma estrutura de código aberto baseada na linguagem de programação Python para fins acadêmicos, que como o próprio nome sugere, permite criar e configurar sua *botnet*. Possui uma série de módulos e possibilidades e está disponível na GitHub – uma plataforma para desenvolvedores com um vasto acervo – de fácil acesso e configuração. O site oficial (*byob.dev*) disponibiliza um rápido tutorial de instalação e ainda é possível acessar o suporte no *Discord* da ferramenta para tirar dúvidas.

De acordo com a descrição na plataforma GitHub (2022), a ferramenta foi projetada para permitir que estudantes e desenvolvedores implementem facilmente sua própria estrutura e adicionem novos recursos interessantes sem precisar desenvolver um novo servidor C2 ou uma ferramenta de administração remota do zero.

Os requisitos para a instalação são: *Python*, o sistema de gerenciamento de pacotes em projetos *Python*, *PIP* e o automatizador a implantação de aplicativos como containers, Docker. As versões são, respectivamente: 3.6.9; 20.3.1 e 20.10

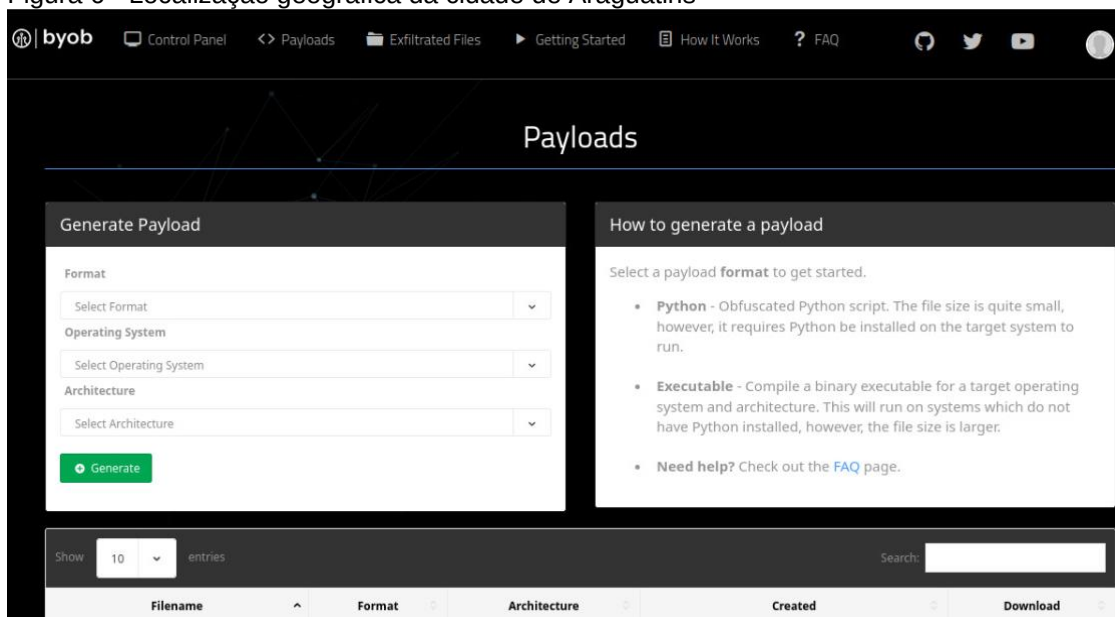
Após o *download*, a instalação, caso o *Python* já esteja instalado, segue os seguintes passos:

1. Abrir o terminal
2. \$ sudo apt-get update
3. \$ cd /byob-master/webgui
4. \$ python -m pip install -r requirements.txt
5. \$ sudo ./startup.sh

6. Ao final do processo, o *script* solicitará confirmação de reinicialização, caso opte por não reinicializar, o servidor C2 será habilitado automaticamente no endereço <http://0.0.0.0:5000>.

O servidor C2 simula um *site* e permite ao usuário controlar todos os recursos da *botnet* através de uma página na *web*. Possui diversos módulos nativos do sistema que podem ser acessados e executados por meio de um clique do *mouse* na aba painel de controle.

Figura 6 - Localização geográfica da cidade de Araguatins



Fonte: autoria própria (2022)

Acessando o endereço supracitado, na aba *payloads* (figura 6) é possível gerar uma “carga útil”, nomenclatura dada aos arquivos usados na infecção dos dispositivos/*bots*, que pode ser tanto no formato executável padrão do *Windows* quanto na linguagem de programação *Python*. Também é possível gerar uma carga útil através do terminal acessando o diretório */byob/byob* sob o comando: *python3 client.py -host- -porta*. No qual o *host* é o *IP* da máquina de comando e controle e a porta a ser utilizada, no caso a 8080. Vale ressaltar a necessidade de gerar a carga no mesmo *IP* e porta indicados na inicialização do servidor C2.

Após todos os passos de configuração e geração de carga útil, o próximo passo se resume a executar o arquivo gerado na máquina destino adicionando, assim, os *bots* ao seu servidor, que estarão imediatamente aptos a receber os comandos do painel de controle.

6.4 Etapa três: engenharia e elaboração de requisitos do Challenger DBN

Para detecção de uma *botnet* se faz necessário a elaboração de uma aplicação, realizada na linguagem de programação Java, em um sistema operacional *Linux/GNU Ubuntu 22.04*, versão servidor, que será desenvolvida por meio da biblioteca *JPCAP*, que deverá ser acionada no primeiro momento.

Os parâmetros para configuração de um ataque realizado a partir de uma *botnet* serão estabelecidos no código fonte da aplicação, esta será integrada a biblioteca e farão captura e análise em tempo real do tráfego de rede.

No momento em que uma possível *botnet* é detectada por meio dos parâmetros estabelecidos na aplicação, o sistema notificará o administrador. A detecção será realizada a partir das regras que serão estabelecidas no código do aplicativo, considerando a extração das características do tráfego da rede, tais como: endereço *IP* destino e origem dos pacotes trafegados, tamanho dos pacotes, taxa de pacotes por segundo, entre outros.

A aplicação será multiplataforma podendo ser executada em sistemas operacionais como *Windows* e *Linux/GNU*. O sistema atuará em modo promiscuo, podendo analisar toda a rede e sub-rede na qual estiver conectada.

6.5 Etapa quatro: interface Challenger DBN

A arquitetura da ferramenta *Challenger DBN* se organizará da seguinte maneira:

Figura 7 - Interface *Challenger DBN*



Fonte: autoria própria (2022)

Conforme o esquema (Figura 6), a arquitetura da ferramenta (*Challenger DBN*) se comporá em três camadas. A primeira camada conterá uma classe que fará chamadas da biblioteca *JPCAP* e terá por finalidade a captura dos pacotes de tráfego a partir da placa de rede que operará em modo promíscuo.

Seguindo o processo, na segunda camada, a ferramenta será capaz de analisar, tanto em tempo real quanto em modo *offline*, o tráfego da rede por diferentes protocolos tais quais: *TCP*, *UDP* e *ICMP*.

Na última etapa, o sistema contará com a detecção de possíveis *botnet's* permitindo sua verificação e retendo informações, tais como: tipo de operação, veracidade do ataque e, ainda, com uma função de alerta no momento em que uma possível *botnet* for detectada por meio de uma caixa de diálogo.

6.6 Etapa cinco: materiais e métodos

Nessa etapa, faz-se necessário o levantamento de uma ferramenta similar e preexistente para testes e validação de requisitos de sistema levantados anteriormente no intuito de obterem-se resultados e discussões acerca do que se espera para o produto em desenvolvimento.

A ferramenta *Fastnetmon* é capaz de detectar ataques de *botnet's* rapidamente em um processamento volumétrico de dados. A ferramenta conta com um centro de depuração, que também pode ser entendido como captura de pacotes de rede, um filtro de dados para uma análise mais eficiente.

Para comprovar a eficiência e eficácia da ferramenta, usar-se-ão ferramentas para projetar uma *botnet*. Para isso, utilizar-se-á a ferramenta *T50 Sukhoi PAK FA Mixed Packet Injector*, Tacio (2011) a define como:

Um *packet injection* (injeção de pacote) livre, criado pelo brasileiro Nelson Brito, capaz de fazer ataques *DoS* e *DDoS* usando o conceito de *stress testing*. Com ele você pode enviar um número altíssimo de requisições de pacotes, de tal forma que o alvo não consiga atender a todas as requisições ou as atenda de forma lenta, dessa forma o alvo pode cair ou ficar lento.

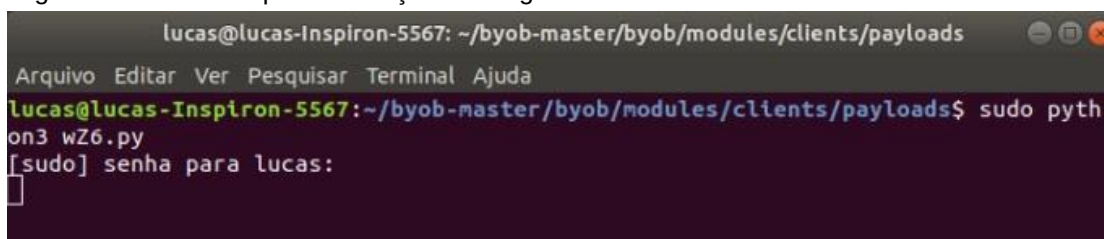
A ferramenta é capaz de executar vários tipos de ataque, são eles: *ICMP*, *TCP* e *UDP*. Para realizar esse ataque, criar-se-á *botnet* com as máquinas virtuais disponíveis em ambiente virtual controlado. Será utilizada uma máquina real como *botmaster*, ambas contarão com um sistema operacional *Linux/GNU Ubuntu 22.04*, versão de servidor. O comando será expedido por meio do *botmaster* e será observado o alerta de possível ataque.

7 RESULTADOS E DISCUSSÕES

Os testes duraram aproximadamente 6 horas e foram realizados em um ambiente laboratorial localizado no Instituto Federal de Educação, Ciências e Tecnologia do Tocantins – *Campus Araguatins*, com aproximadamente 50 metros² e 35 máquinas com sistema operacional Linux Ubuntu 22.04 e configuração de hardware padronizada.

Para iniciar os testes foi necessário configurar toda a *botnet* de acordo com o ambiente e estrutura disponíveis. Um total de dez máquinas sequenciais dentre as disponíveis no laboratório foram selecionadas e adicionadas a *botnet* pelo terminal através do comando: “python3 wZ6.py” conforme a figura 7. O comando mencionado executa a carga útil denominada wZ6.py gerada por meio do *script* “cliente.py”.

Figura 8 – Comando para execução da carga útil através do terminal



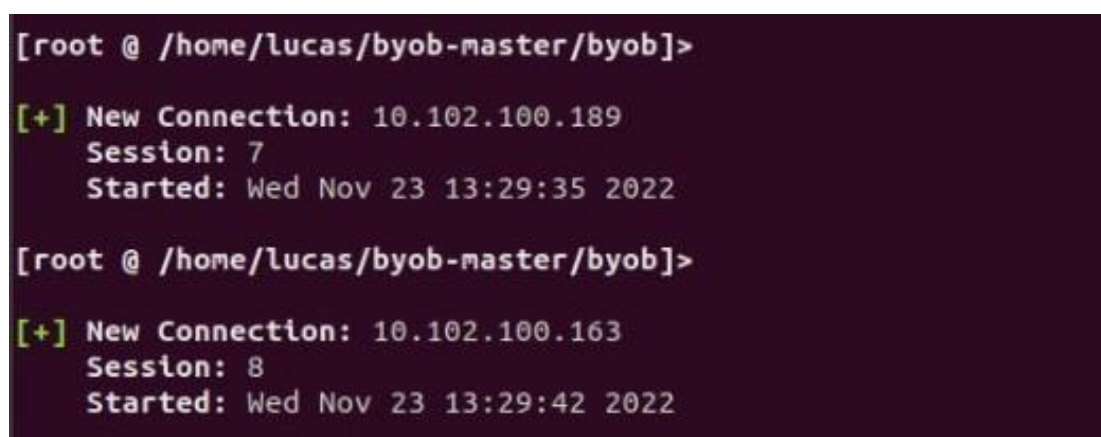
```
lucas@lucas-Inspiron-5567: ~/byob-master/byob/modules/clients/payloads
Arquivo Editar Ver Pesquisar Terminal Ajuda
lucas@lucas-Inspiron-5567:~/byob-master/byob/modules/clients/payloads$ sudo python3 wZ6.py
[sudo] senha para lucas:

```

Fonte: autoria própria (2022)

Na medida que a carga útil é executada nos dispositivos destinados a serem *bots*, é possível verificar no servidor de comando e controle e no servidor administrador via terminal as novas conexões. Cada *bot* adicionado gera um novo cliente na *botnet* e uma nova sessão (figura 8).

Figura 9 – Novas conexões de *bots* registradas na *botnet*



```
[root @ /home/lucas/byob-master/byob]>
[+] New Connection: 10.102.100.189
    Session: 7
    Started: Wed Nov 23 13:29:35 2022

[root @ /home/lucas/byob-master/byob]>
[+] New Connection: 10.102.100.163
    Session: 8
    Started: Wed Nov 23 13:29:42 2022

```

Fonte: autoria própria (2022)

A máquina vítima foi selecionada aleatoriamente dentre as disponíveis no laboratório, com as mesmas configurações das máquinas *bots*. Iniciou-se, então, o processo de instalação da ferramenta pré-selecionada de teste denominada *Fastnetmon* (figura 9), viabilizando a configuração que partiu do tutorial presente no site oficial fastnetmon.com. As métricas para detecção dos ataques de negação de serviços distribuídos seguiram as recomendadas pelo tutorial, diferenciando a largura de banda com base em 500 megabytes em taxa de amostragens para 100 megabytes de largura de banda.

Figura 10 – Status de serviço ativo da ferramenta *Fastnetmon*

```

Aluno@AGT-LABHP: ~
Aluno@AGT-LABHP:~$ sudo service fastnetmon status
● fastnetmon.service - FastNetMon - DoS/DDoS analyzer with sFlow/Netflow/mirror
   Loaded: loaded (/lib/systemd/system/fastnetmon.service; enabled; vendor pr>
   Active: active (running) since Wed 2022-11-23 17:12:33 -03; 2min 46s ago
   Process: 754 ExecStart=/opt/fastnetmon/app/bin/fastnetmon --daemonize (code>
  Main PID: 874 (fastnetmon)
    Tasks: 18 (limit: 4536)
   Memory: 25.7M
    CGroup: /system.slice/fastnetmon.service
            └─874 /opt/fastnetmon/app/bin/fastnetmon --daemonize

nov 23 17:11:49 AGT-LABHP systemd[1]: Starting FastNetMon - DoS/DDoS analyzer w>
nov 23 17:11:57 AGT-LABHP fastnetmon[754]: We will run in daemonized mode
nov 23 17:11:57 AGT-LABHP systemd[1]: fastnetmon.service: Can't open PID file />
nov 23 17:12:33 AGT-LABHP systemd[1]: Started FastNetMon - DoS/DDoS analyzer wi>
lines 1-14/14 (END)

```

Fonte: autoria própria (2022)

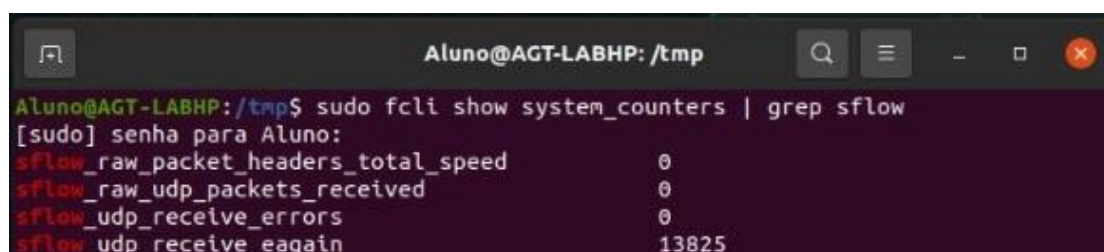
Com todos os *bots* devidamente *setados*, partiu-se, então, efetivamente para os testes com o servidor de comando e controle via terminal. Para disparar um ataque, foi necessário acessar a primeira sessão da *botnet* que representa a máquina mestre por meio do comando “*shell 0*”. Esse passo se fez necessário pelo fato da configuração da própria *botnet*.

O próximo passo se deu ao ataque propriamente através do comando “*broadcast sudo t50 10.102.100.125 –flood --turbo –protocol TCP*”. O módulo “*broadcast*” é responsável por destinar a atividade a ser executada – o ataque – as demais sessões. O argumento “*sudo*” coloca o comando em modo *root* dando-o poder de administrador. O argumento “*t50*” refere-se a ferramenta utilizada para

efetuar um ataque tipo *DDoS*, instalada na máquina mestre que corresponde a sessão zero. Os parâmetros restantes indicam, respectivamente um ataque de inundação, um modo turbo propriamente dito e protocolo a ser utilizado indicando o tipo do ataque, no caso *SYN flood*.

Para os demais tipos de ataques, o comando foi o mesmo alternando apenas o protocolo. Ao todo utilizados foram: *TCP*, *UDP* e *ICMP*. De modo que os disparos rápidos de aproximadamente 30 - 60 segundos foram sucessivos e as capturas mostraram-se constantes.

Figura 11 – Registro da captura de pacotes no protocolo UDP



```

Aluno@AGT-LABHP: /tmp
Aluno@AGT-LABHP:/tmp$ sudo fcli show system_counters | grep sflow
[sudo] senha para Aluno:
sflow_raw_packet_headers_total_speed      0
sflow_raw_udp_packets_received            0
sflow_udp_receive_errors                  0
sflow_udp_receive_eagain                  13825
  
```

Fonte: autoria própria (2022)

O comando “`sudo fcli show system_counter | grep sflow`” é responsável por exibir os registros da aplicação de teste na rede, uma vez que a versão utilizada não possui interface gráfica disponível. É possível, ainda, verificar a captura do ataque do tipo *UDP flood* registrado na tabela do sistema totalizando 13825 pacotes de inserção de datagramas no turno em questão (figura 10).

Durante os testes, foi possível observar também no gerenciador de tarefas da máquina alvo um uso exagerado de recursos computacionais no que diz respeito ao tráfego de rede visto que não havia justificativa plausível para isso (figura 11).

Figura 12 – Registro do tráfego de rede capturado através do gerenciados de tarefas



Fonte: autoria própria (2022)

Na figura um é possível perceber um intenso recebimento de dados totalizando 1,2 gigabytes que representa praticamente o triplo de pacotes enviados se justificando evidentemente um ataque de *botnet*. O gráfico cruza tempo em segundos por quantidade de dados por segundo. Em média levou-se apenas pouco mais de dois minutos para a inserção total exibida na tela.

A ferramenta foi capaz de registrar toda a inserção de datagramas fazendo a somatória de todos os turnos de disparos totalizando 275.349 pacotes.

Figura 13 – Registro da captura de pacotes no protocolo UDP

A screenshot of a terminal window with a dark background and light-colored text. It displays five lines of statistics for UDP packet capture. The first line shows 'global_system_ignoredmulti' with a value of 5324. The second line shows 'global_system_incsumerrors' with a value of 0. The third line shows 'global_system_indatagrams' with a value of 275349. The fourth line shows 'global_system_inerrors' with a value of 209. The fifth line shows 'global_system_memerrors' with a value of 0.

global_system_ignoredmulti	5324
global_system_incsumerrors	0
global_system_indatagrams	275349
global_system_inerrors	209
global_system_memerrors	0

Fonte: autoria própria (2022)

8 CONSIDERAÇÕES FINAIS

Este trabalho tem por finalidade analisar o funcionamento de uma ferramenta capaz de detectar ataques de negação de serviço distribuído para viabilizar requisitos indispensáveis e a projeção de uma ferramenta, a *Challenger DBN*, que seja capaz de realizar as mesmas atividades com diferenciais. Para isso utilizou a ferramenta *Fastnetmon*.

A aplicação selecionada mostrou-se razoavelmente intuitiva e com um bom suporte para adaptação a rede a ser monitorada, conta com tutorial para manejo via terminal em sua página oficial e interface gráfica disponível para as versões *premium* e dá autonomia ao próprio usuário para determinar suas métricas de análise do tráfego de rede.

De modo geral, a ferramenta se mostrou satisfatoriamente eficaz, sendo capaz de detectar ataques de negação de serviços distribuídos com certa rapidez, o que se espera para uma aplicação em uma área tão sensível. Pode vir a ser útil como base para a estruturação e arquitetura da *Challenger DBN*.

Os pontos de melhorias inicialmente observados podem ser interface gráfica intuitiva mesmo nas versões comunitárias o que produziria certa facilidade no manejo, bloqueio de acessos de atacantes padronizado e alertas significativos, uma vez que no ambiente real, é estritamente necessário que a administração do sistema que venha a servir de alvo para um ataque malicioso seja informada o quanto antes para que se possa tomar as devidas providências.

Para uma ferramenta de detecção de *botnets*, o alerta de ataque é primordial para contenção rápida de possíveis prejuízos, principalmente em se tratando de âmbito empresarial. Para isso, pode-se ser adotada a metodologia de alertas via mídias sociais, como observada na aplicação estudada que utiliza a mídia *Telegram* para notificação seu supervisor. Este recurso é imprescindível para sistemas que não estão sob supervisão fora de horário comercial, por exemplo.

REFERÊNCIAS

BARBOSA, Kaio R. S.; MARTINS, Gilbert B.; SOUTO, Eduardo; FEITOSA Eduardo. **Botnets: Características e Métodos de Detecção Através do Tráfego de Rede**. In: Minicursos do XIV Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais. **Anais...** Belo Horizonte, 2014.

BARDAL, Rafael Machado. **ESTUDO SOBRE ATAQUES DE NEGAÇÃO DE SERVIÇO E UMA ABORDAGEM PRÁTICA**. Curitiba - PR. 2015. Disponível em: <http://repositorio.utfpr.edu.br/jspui/bitstream/1/17315/2/CT_GESER_VI_2015_3.pdf>. Acesso em: 07 de outubro de 2022.

CARDOSO, Adeilson Marques da Silva. **CEPIDS: Um IDS baseado no Processamento de Eventos Complexos para Internet de Coisas**. Dissertação (Mestrado) Ciência da Computação/ccet. São Luis, 2018.

CERT.BR - **Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil**. Recomendações para Melhorar o Cenário de Ataques Distribuídos de Negação de Serviço (DDoS). Versão 1.0, 2016. Disponível em: <<https://www.cert.br/docs/whitepapers/ddos/#4>>. Acesso em: 25 de agosto de 2018.

DUARTE, Otto Carlos Muniz Bandeira et al. **Negação de Serviço e Botnets**. 2015. Disponível em: <https://www.gta.ufrj.br/grad/15_1/dos/index.html>. Acesso: 17 de setembro de 2022.

FASTNETMON. **Fast, Reliable, Automated DDoS Detection, Prevention, and Mitigation**. Traduzido por: Google tradutor. c2022. Disponível em: <<https://fastnetmon.com/ddos-detection-and-mitigation/>>. Acesso em: 23 de setembro de 2022.

IBGE. **Instituto Brasileiro de Geografia e Estatística**. Dados Estatísticos do município de Araguatins. Disponível em: <<https://www.ibge.gov.br/>>. Acesso em: 27 de agosto 2018.

INFOMONEY. InfoMoney. **Os mais famosos ataques DDoS da história – e o que podem nos ensinar sobre segurança na web**. 2019. Disponível em: <<https://www.infomoney.com.br/patrocinados/prnewswire/os-mais-famosos-ataques-ddos-da-historia-e-o-que-podem-nos-ensinar-sobre-seguranca-na-web/>>. Acesso em: 18 de setembro de 2022.

OLIVEIRA, Lucas Vinícius de; BEM, Ricardo Orige de. **PROTEÇÃO NA REDE: Uma análise informativa do atual cenário da segurança da informação**. ARARANGUÁ – SC, 2015. Disponível em: <http://www.academia.edu/14512556/PROTE%C3%87%C3%83O_NA_REDEUma_an%C3%A1lise_informativa_do_atual_cen%C3%A1rio_da_seguran%C3%A7a_da_informa%C3%A7%C3%A3o>. Acesso em: 27 de agosto 2018.

SAB, Gabriel Augusto Amim; FERREIRA, Rafael Cardoso; ROZENDO, Rafael Gonsalves. **Negação de Serviço, Negação de Serviço Distribuída e Botnets**. Rio de Janeiro, 2013. Disponível em: <https://www.gta.ufrj.br/grad/13_1/dos/ataques.html#2.1>. Acesso em: 25 de agosto de 2018.

SILVA, Daisy Cristine Albuquerque da. **Análise da eficiência dos algoritmos de aprendizado de máquinas para a detecção de botnets**. Dissertação (mestrado) - Instituto Militar de Engenharia. Rio de Janeiro, 2016.

SOURCEFORGE. **Follow Network Packet Capture Facility for Java**. 2013. Disponível em: <<https://sourceforge.net/projects/jpcap>>. Acesso em: 27 de agosto 2018.

TACIO, Paulo. **FERRAMENTA PARA DOS/DDOS T50**. 2011. Disponível em: <<http://www.mundodoshackers.com.br/ferramenta-para-dosddos-t50>>. Acesso em: 30 de agosto de 2018.

VIEIRA, Nathan. **Ataques DDoS geram preocupação no setor financeiro**. 2019. Disponível em: <<https://canaltech.com.br/seguranca/ataques-ddos-geram-preocupacao-no-setor-financeiro-156861/>>. Acesso em: 23 de setembro de 2022.