UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

BRUNO RICCELLI DOS SANTOS SILVA

IMPLEMENTAÇÃO E ANÁLISE DE UM FRAMEWORK DE DETECÇÃO DE ATAQUES DISTRIBUÍDOS DE NEGAÇÃO DE SERVIÇO

FORTALEZA

2017

BRUNO RICCELLI DOS SANTOS SILVA

IMPLEMENTAÇÃO E ANÁLISE DE UM FRAMEWORK DE DETECÇÃO DE ATAQUES DISTRIBUÍDOS DE NEGAÇÃO DE SERVIÇO

Monografia apresentada ao Curso de Engenharia de Computação da Universidade Federal do Ceará, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Computação.

Orientador: Prof. Msc. Ricardo Jardel Nunes da Silveira

Co-Orientador: Prof. Msc. Marcelo Araújo Lima

FORTALEZA

2017

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

S233i Santos Silva, Bruno Riccelli dos.
Implementação e análise de um framework de detecção de ataques distribuídos de negação de serviço / Bruno Riccelli dos Santos Silva. – 2017.
39 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Tecnologia, Curso de Engenharia de Computação, Fortaleza, 2017.
Orientação: Prof. Me. Ricardo Jardel Nunes da Silveira.
Coorientação: Prof. Me. Marcelo Araújo Lima.

1. Redes. 2. Segurança. 3. Tempo real. I. Título.

CDD 621.39

BRUNO RICCELLI DOS SANTOS SILVA

IMPLEMENTAÇÃO E ANÁLISE DE UM FRAMEWORK DE DETECÇÃO DE ATAQUES DISTRIBUÍDOS DE NEGAÇÃO DE SERVIÇO

Monografia apresentada ao Curso de Engenharia de Computação da Universidade Federal do Ceará, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Computação.

Aprovada em: 18/12/2017.

BANCA EXAMINADORA

Prof. Msc. Ricardo Jardel Nunes da Silveira (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Msc. Marcelo Araújo Lima (Co-Orientador)
Instituto Federal do Ceará (IFCE)

Prof. Dr. Jarbas Aryel da Silveira
Universidade Federal do Ceará (UFC)

Prof. Msc. Daniel Alencar Barros Tavares
Instituto Federal do Ceará (IFCE)

Dedico este trabalho à minha família e namorada, pessoas que fizeram de tudo para que eu chegasse onde cheguei.

AGRADECIMENTOS

Agradeço primeiramente a Deus, que iluminou meu caminho durante essa jornada, me dando saúde e força para superar as dificuldades.

À minha namorada, Luéline Elias, pelo amor, paciência, dedicação e companheirismo em todos os momentos.

À minha família, por sua capacidade de acreditar e investir em mim. Mãe, sua dedicação foi o que deu, em alguns momentos, a esperança para seguir.

Ao meu orientador, Prof. Ricardo Jardel Nunes da Silveira, pelo acompanhamento e estreitamento da relação professor-aluno e exemplo de profissional bem como pelo apoio, incentivo, sugestões e comentários durante a supervisão dos meus estudos.

Ao meu coorientador, Prof. Marcelo Araújo Lima, pelo apoio, incentivo, sugestões e tempo dedicado para me ajudar durante meus estudos.

Aos meus amigos da Universidade Federal do Ceará, 8086FC e 8086Team pela amizade e pelos momentos de descontração e estudo.

E a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

Fortaleza, Dezembro de 2017.

Bruno Riccelli dos Santos Silva

”A persistência é o caminho do êxito.”
(Charles Chaplin)

RESUMO

Ataques Distribuídos de Negação de Serviço (DDoS) figuram uma das principais ameaças a redes de computadores atualmente. Sistemas de Detecção de Intrusão (IDS) são responsáveis por detectarem pacotes maliciosos e proverem medidas cabíveis para que a ameaça não cause maiores danos ao servidor/*host*. O presente trabalho visa o estudo, implementação e validação de um *framework* de detecção de ataques DDoS publicado este ano em um periódico cientificamente reconhecido. O *framework* foi testado e validado, utilizando-se duas bases de dados (*Data Mining* e DARPA), no qual obteve alto desempenho medido em termos de taxa de acertos.

Palavras-chave: Redes. Segurança. Tempo real.

ABSTRACT

In these days, Distributed Denial of Service (DDoS) attacks are one of the main threats for network connected computers. This way, Intrusion Detection Systems (IDS), that are systems for detection of DDoS attacs trough the identification malicious packages, are a fundamental tool for protecting servers or hosts against the these threats. The present work aims at the study, implementation and validation of a DDoS attack detection framework, recently published this year in a scientifically recognized journal. The framework has been tested and validated, by using two databases (Data Mining and DARPA) of DDoS attacks, in which our framework has obtained high performance, measured terms of accuracy rate, as of 100% of correctness identification for both databases.

Keywords: Network. Security. Real-time.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

DDoS	*Distributed Denial of Service*
IoT	*Internet of Things*
DoS	*Denial of Service*
IDS	*Intrusion Detection System*
IP	*Internet Protocol*
DNS	*Domain Name Service*
ICMP	*Internet Control Message Protocol*
LOIC	*Low Orbit Ion Cannon*
UDP	*User Datagram Protocol*
SQL	*Structured Query Language*
MIB	*Management Information Base*

SUMÁRIO

1 INTRODUÇÃO

No início da década de noventa, a *internet* era uma ferramenta disponível apenas em universidades, para uso por professores, pesquisadores e alunos, os quais utilizavam computadores caros, pesados e conectados por um grande número de fios. No entanto, nos últimos trinta anos, devido a avanços tecnológicos de fabricação de semicondutores, houve uma miniaturização do computador, de maneira a caber na palma de nossas mãos, além de uma redução drástica do custo, e da facilidade de uso com a tecnologia *touch screen* e de muitos outros avanços.

Em paralelo com as melhorias dos dispositivos, também foi ampliada a abrangência e velocidade de comunicação de dados na *internet*, principalmente pela evolução das tecnologias de comunicação sem fio, tais como ADSL (*Asymmetric Digital Subscriber Line*), Wi-Fi e 4G/5G. Essa evolução concomitante de dispositivos e de tecnologias de suporte à comunicação fez a *internet* deixar de ser uma ferramenta restrita a universidades para atingir todas as pessoas, de todas as idades, e de todas as classes sociais, que hoje fazem o uso da *internet* em seu cotidiano para atividades, tais como, comunicar-se com amigos e familiares por mensagens instantâneas, chamadas de voz e fazer uso de aplicações *online*, o que tem sido cada vez mais comum no contexto atual na rede mundial de computadores. Além disso, o número de serviços e comodidades fornecidos para os usuários apenas aumentam seja por aplicações *web*, aplicativos de celular, ou até mesmo dispositivos IoT (*Internet of Things*). Com essa difusão dessas tecnologias, atividades maliciosas na rede vêm surgindo em proporção cada vez maior, colocando em risco os serviços, bem como a integridade dos dados dos usuários.

De acordo com Barford *et al.* (2002), redes que não possuem nenhum mecanismo de análise de tráfego, não podem garantir segurança aos clientes que a utilizam, pois não têm garantia de que podem operar eficientemente. Assim, redes que não possuem sistemas de detecção de atividades maliciosas estão sujeitos a terem suas funcionalidades comprometidas, ou mesmo invadidas por ataques produzidos por um agente malicioso na rede.

Um ataque consiste de uma atividade maliciosa que explora uma vulnerabilidade na rede ou em um nó da rede. Dentre os ataques em redes de computadores, o DoS (*Denial of Service*) é o que ocasiona uma maior perturbação da qualidade de serviço da rede (BADISHI; KEIDAR; SASSON, 2006). Um ataque DoS tenta tornar um nó na rede (geralmente servidor *web*) indisponível, inundando-o com falsas requisições, ocupando a largura de banda e/ou consumindo seus recursos computacionais (CHEN, 2006). Uma evolução deste tipo de ataque trata-se do DDoS, possuindo as mesmas características do anterior, porém sendo executado de forma distribuída, ou seja, mais de um atacante ou mesmo um atacante controlando remotamente outras vítimas, com o objetivo de tornar indisponível um servidor/serviço alvo. O primeiro registro de atividades DDoS aconteceu em Julho de 1999, na universidade de Minnesota - Estados Unidos, onde a rede da vítima ficou indisponível por mais que dois dias (SRIVASTAVA *et al.*, 2011). No ano seguinte,

um ataque DDoS tornou-se conhecido mundialmente, quando um grande número de sites como o *Yahoo, EBay, Amazon e CNN* ficaram inoperantes devido a ataques dessa natureza (CALCE; SILVERMAN, 2008).

Assim, sistemas de detecção de intrusão (IDS) são necessários para monitorar a rede em busca de pacotes maliciosos e assim identificá-los para tomar as medidas corretivas que impeçam o avanço do atacante.

O presente trabalho faz o estudo, implementação e validação de um *framework* de detecção de ataques DDoS, o qual pode ser encontrado em (HOQUE; KASHYAP; BHATTACHARYYA, 2017). Tal arcabouço é capaz de detectar um tráfego malicioso utilizando uma correlação proposta pelos autores que considera poucos parâmetros de tráfego durante a análise, utilizando uma janela de tráfego de 1 segundo. Assim, neste trabalho é realizada a implementação em MATLAB para o *framework* e a validação é mostrada a partir de duas bases de dados estudadas e testadas no *framework*.

1.1 Objetivos

O presente trabalho tem por objetivos gerais estudar, implementar e validar um *framework* de detecção de ataques DDoS encontrado na literatura. No decorrer do desenvolvimento deste trabalho, os seguintes objetivos específicos foram perseguidos e atingidos:

- Entender o funcionamento do *framework* estudado.
- Implementar em MATLAB o *framework* estudado.
- Simular o *framework* bases de dados para validar os resultados com o artigo de origem.

1.2 Organização da monografia

Os estudos deste trabalho estão organizados da seguinte forma: No Capítulo 2 é apresentado um estudo bibliográfico sobre ameaças de rede, ataques DDoS e sistemas IDS (*Intrusion Detection System*). No Capítulo 3, mostramos a modelagem do ambiente de simulação utilizado neste trabalho é descrita. No Capítulo 4, apresentamos o desempenho obtido pelo *framework* estudado por meio da taxa de acerto para cada janela de tráfego. Por fim, o último capítulo deste trabalho apresenta as conclusões realizadas a partir dos resultados obtidos, além de algumas perspectivas para a continuação deste trabalho.

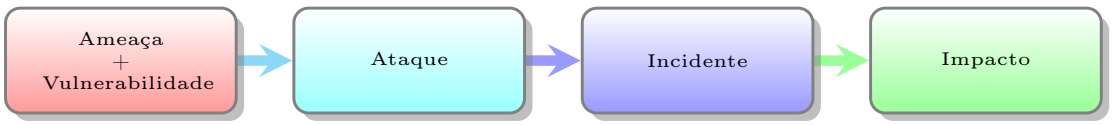
2 REVISÃO BIBLIOGRÁFICA

Nesse capítulo, os conceitos acerca deste trabalho são apresentados, abordando definições na área de segurança em redes tais como ataques DoS, DDoS, sistemas IDS, além de uma revisão da literatura acerca de *frameworks* de detecção de ataques em redes de computadores.

2.1 Segurança da Informação

Uma ameaça trata-se de um potencial para violação de segurança quando há uma circunstância que pode quebrá-la, causando danos a um serviço/*host*. Exemplos de ameaças são: *malwares*, ataques de negação de serviço e envio de pacotes com falso endereço origem. Uma ameaça explora uma vulnerabilidade no alvo para obter as informações que deseja ou mesmo tornar o serviço indisponível, ou seja, ocorrendo a violação da segurança no alvo que é o ataque. Um ataque pode ocasionar, por exemplo, a destruição dos dados, perda da integridade, dentre outros incidentes. Um incidente, por sua vez pode causar prejuízo financeiro para a imagem do alvo, além de causar indisponibilidade do serviço fornecido (KUROSE; ROSS, 2010). A Figura 2.1 mostra uma síntese desses conceitos.

Figura 2.1 – Fluxo explicativo de uma ameaça



Fonte: Elaborada pelo autor.

2.2 Ataques DoS e DDoS

Um ataque de negação de serviço (DoS - Denial-of-Service) torna um componente de rede inutilizável por usuários que estejam consumindo o serviço fornecido. A maioria dos ataques DoS na *internet* pode ser dividida em três categorias: (KUROSE; ROSS, 2010)

- Ataque de vulnerabilidade: Mensagens são enviadas a uma aplicação vulnerável ou a um servidor, sendo executado em um hospedeiro alvo.
- Inundação na largura de banda: O atacante envia um grande número de pacotes maliciosos ao hospedeiro alvo até que o enlace de acesso do alvo fique cheio, impedindo os pacotes legítimos de alcançarem o servidor.

- Inundação na conexão: O atacante estabelece um grande número de conexões TCP semi-abertas ou abertas no hospedeiro alvo.

Já segundo Douligeris e Mitrokotsa (2004), os ataques DoS são classificados em cinco categorias baseadas no protocolo cujo é atacado: dispositivo, sistema operacional, aplicação, inundação de dados e características do protocolo. O primeiro inclui ataques que podem ser causados ao tirar vantagem de *bugs* ou vulnerabilidades em *software*. O segundo leva em consideração, ataques que aproveitam-se da forma como os protocolos são implementados pelos sistemas operacionais. Ataques baseados na aplicação infectam o alvo por meio de *bugs* específicos da rede e tentam drenar os recursos da vítima. Em ataques baseados em inundação de dados, um atacante tenta usar a largura de banda disponível para mandar grandes quantidades de dados, fazendo com que o alvo processe todo esse volume de informação. Por fim, ataques baseados em características do protocolo são caracterizados por tirarem vantagens de certos padrões de protocolo. Por exemplo, vários ataques exploram o fato de que os endereços de origem IP podem ser falsificados. Além disso, vários tipos de ataques DoS são focados no protocolo DNS (*Domain Name Service*), onde eles atacam o *cache* de DNS em servidores de nomes. Um dos principais ataques por exploração de protocolos é o TCP SYN, no qual o atacante faz a requisição, o alvo responde e o agente malicioso não responde, não completando o *handshake* em três vias. Assim, realizadas várias requisições desse tipo, o alvo terá todos os seus recursos ocupados e nenhuma nova conexão poderá ser feita.

Um ataque distribuído de negação de serviço (do inglês, DDoS) utiliza as propriedades de um ataque DoS de forma distribuída. Em outras palavras, a indisponibilidade de um serviço é causada por ataques oriundos de um ou mais IPs origem, tornando mais complexo o tratamento e a busca pelo atacante que está propagando a ameaça. De acordo com Wang *et al.* (2015), atualmente os atacantes podem lançar vários ataques DDoS, focando-se nos recursos: largura de banda da memória e CPU, e em aplicativos (aplicações *web*, serviços de banco de dados). Alguns tipos de ataques DDoS podem ser citados:(ALKASASSBEH *et al.*, 2016)

- *Smurf*
- HTTP *Flood*
- UDP Flood
- SIDDoS

2.2.1 Smurf

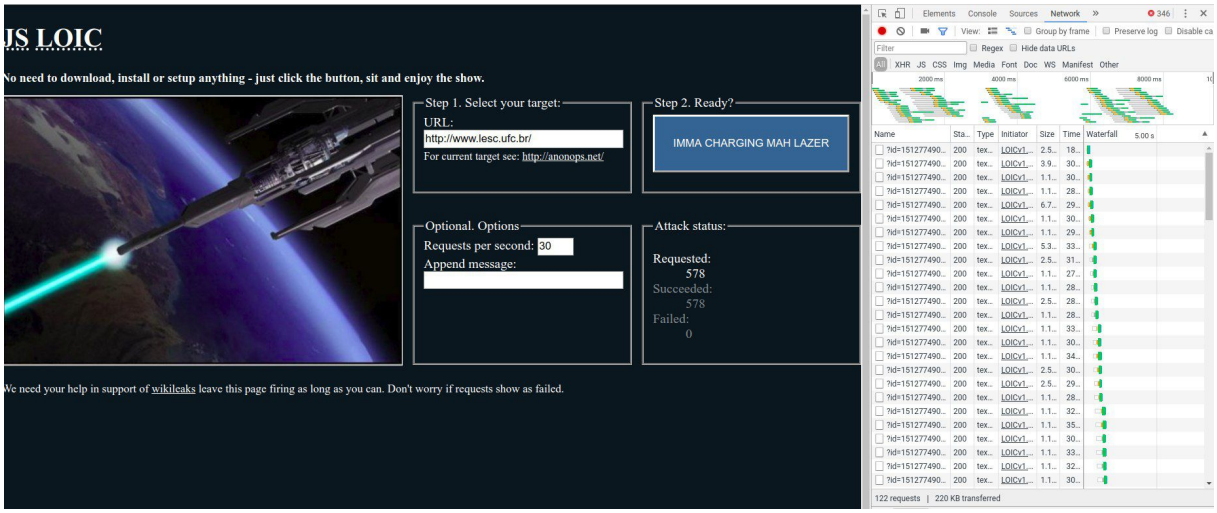
Ataques DDoS do tipo *Smurf* possuem dois componentes principais que são o uso de requisições ICMP (*Internet Control Message Protocol*) forjadas e a direção dos pacotes para um endereço *broadcast*. O protocolo ICMP é usado para troca de mensagens de controle e pode ser

usado para determinar se uma máquina na *internet* está respondendo. Um exemplo prático desse comando é o *ping*, o qual envia mensagens para um IP(*Internet Protocol*) e recebe uma resposta, sendo do IP alvo, ou por tempo de requisição excedido (*timeout*). Além disso, um IP *broadcast* serve para comunicar-se com todos os *hosts* em um segmento de rede. Assim, os invasores usam pacotes de solicitação de eco ICMP direcionados para endereços de IP *broadcast* para gerar ataques de negação de serviço. Note que esse ataque possui três participantes: o atacante, um intermediário(que também pode ser uma vítima) e o alvo. Esse intermediário recebe um pacote ICMP direcionado ao IP *broadcast* de sua rede. Se esse intermediário não filtrar seu tráfego, muitas máquinas na rede receberão esse pacote de requisição e responderão ao mesmo por meio de uma resposta eco ICMP, provocando um congestionamento na rede (CENTER, 1998).

2.2.2 HTTP flood

Esse tipo de ataque, diferentemente do *Smurf* é realizado na camada de aplicação (protocolo HTTP). Trata-se de um ataque volumétrico, ou seja, torna um recurso indisponível por meio de uma grande quantidade de informação em uma pequena janela de tempo. Tal ataque pode ser explorado por usar uma grande quantidade de conexões concorrentes, ou por meio de um grande consumo de banda (como por exemplo vários *hosts* fazendo *download* de um arquivo grande) em uma pequena janela de tempo. Assim, um atacante pode infectar vários *hosts* e comandá-los a realizar uma quantidade excessiva de conexões em um alvo. Além disso, dentre as várias ferramentas existentes, pode-se citar o LOIC (*Low Orbit Ion Cannon*). Tal ferramenta realiza um número de requisições por segundo definido pelo usuário a um alvo. A Figura 2.2 mostra seu uso

Figura 2.2 – Exemplo de ataque utilizando a ferramenta LOIC



Fonte: http://metacortexsecurity.com/tools/anon/LOIC/LOICv1.html

A Figura 2.2 mostra requisições à direita sendo respondidas com código 200 (OK) pelo servidor alvo, sendo que 30 requisições por segundo são enviadas ao mesmo. Vale ressaltar que

o uso de apenas um *host* realizando esse processo não configura um ataque DDoS, visto que seriam necessárias várias máquinas enviando esse tipo de requisição em uma mesma janela de tempo para causar algum dano ao servidor.

2.2.3 UDP flood

Diferentemente do HTTP *flood*, a versão UDP (*User Datagram Protocol*) atua no protocolo da camada de transporte e tem como objetivo congestionar o *link* do alvo. Assim, em um ataque UDP *flood*, uma grande quantidade de pacotes UDP são enviados para as portas do alvo. Para ocultar a identidade do atacante, o mesmo falsifica o endereço IP de origem dos pacotes de ataque. Os ataques de inundação UDP também podem esgotar a largura de banda da rede em torno do sistema da vítima. Por isso, os sistemas em torno da vítima também são impactados devido ao ataque de inundação UDP (XIAOMING; SEJDINI; CHOWDHURY, 2010). Segundo Douligeris e Mitrokotsa (2004), um ataque UDP *flood* é possível quando um atacante envia um pacote UDP para uma porta aleatória do sistema da vítima. Após isso, quando a vítima recebe esse pacote, ela irá determinar que aplicação está aguardando na porta destino e quando ela percebe que não tem nenhuma aplicação esperando, ela envia um pacote ICMP contendo a mensagem de ”destino inalcançável” para o destino cujo IP foi forjado e devido a grande quantidade de pacotes UDP, o sistema alvo irá cair. Uma ferramenta comumente utilizada para esse tipo de ataque é o Trin00 (CRISCUOLO, 2000), a qual é responsável por lançar massas de dados UDP para um ou mais endereços IP (DITTRICH, 2002).

2.2.4 SIDDoS

Outro tipo de ataque DDoS é o SQL *Injection* DDoS, no qual atacantes inserem requisições SQL (*Structured Query Language*) para o banco de dados da vítima através, por exemplo, de um formulário o qual possui vulnerabilidades (falta de validação das entradas). Então, ilegalmente, permitindo acesso aos recursos ou mesmo aos dados guardados pelo servidor (ALKASASSBEH *et al.*, 2016). A maior parte desse tipo de ataque ocorre em telas de *login*, as quais o atacante tem acesso indireto ao banco de dados, através do preenchimento malicioso desses elementos com códigos SQL. Uma ferramenta utilizada para tal fim é o SQLMap, no qual o uso consiste em apenas definir um alvo e o tipo de ataque SQL *Injection*.

2.3 IDS

Para tratar esse tipo de problema, são utilizados sistemas do tipo IDS (*Intrusion detection System*). Tratam-se de *softwares* responsáveis por detectar anomalias na rede, como por exemplo acessos não autorizados e tráfegos mal intencionados. Os IDS monitoram o tráfego, buscando anomalias e em caso positivo, alerta aos administradores da rede para que estes tomarem as medidas corretivas, bloqueando as portas, negando serviço a um IP específico que esteja enviando

requisições maliciosas ou fechando serviços que são geralmente utilizados para ataques. Segundo Ashoor e Gore (2011), os sistemas IDS possuem 3 categorias:

- Sistemas de detecção baseados em assinatura.
- Sistemas de detecção baseados em anomalias.
- Sistemas de detecção baseados em especificação.

O sistema de detecção baseado em assinaturas representa um ataque conhecido através de um padrão ou assinatura para que os ataques descritos e suas variações sejam identificados. Assim, esse tipo de sistema depende de listas atualizadas com padrões de ataques e assim, será impossível detectar uma ameaça desconhecida ou atualizada. No caso de sistemas baseados em anomalias, é importante que haja uma classificação da rede como normal ou anômala, além de conhecer o comportamento normal da rede. Por fim, um sistema de detecção baseados em especificação é responsável por monitorar os processos e caso detecte qualquer comportamento anormal, emitirá um alerta e deve ser mantido e atualizado sempre que houver alguma alteração.

Assim, os sistemas IDS utilizam alguns conceitos para realizar seus serviços. Pode-se destacar a definição de objeto de tráfego, o qual significa um conjunto de pacotes em uma determinada janela de tempo. A partir de um objeto de tráfego, pode-se calcular métricas de avaliação da rede. Vale ressaltar que para a obtenção de objetos de tráfego, faz-se necessário o uso de *sniffer*, um analisador de rede que captura o tráfego de entrada e saída. Desta forma, os pacotes são capturados e calculam-se os parâmetros do objeto de tráfego.

Alguns IDS podem ser encontrados na literatura. O trabalho em (CABRERA *et al.*, 2001) utiliza dados da MIB (*Management Information Base*) vinda do roteador para realizar a detecção. Esses dados incluem parâmetros que indicam diferentes estatísticas de pacotes e rotas, focando na identificação de padrões estatísticos de diferentes formas, com o objetivo de realizar a detecção o mais cedo possível. Outro mecanismo chamado CTPS/PF (*Congestion-Triggered Packet Sampling/Packet Filtering*) foi proposto por (HUANG; PULLEN, 2001). De acordo com essa abordagem, um subconjunto de pacotes descartados devido ao congestionamento são selecionados para análise estatística. Se uma anomalia é indicada pelos resultados estatísticos, um sinal é enviado ao roteador para filtrar os pacotes maliciosos. No trabalho proposto por Gil e Poletto (2001) é apresentada uma heurística chamada MULTOPS, que postula que se a detecção de endereços IP que participam de um ataque DDoS é possível, então são tomadas medidas para bloquear apenas esses endereços específicos. Cada dispositivo de rede mantém uma árvore de vários níveis que contém estatísticas de taxa de pacotes para prefixos de sub-rede em diferentes níveis de agregação. MULTOPS usa taxas desproporcionais de *hosts* e sub-redes para detectar ataques. Quando armazena as estatísticas com base em endereços de origem, é dito que ele opera em modo orientado a ataques, caso contrário, atua no modo orientado à vítima. Uma estrutura de dados MULTOPS pode assim ser usada para manter o controle de ataques em *hosts*. Quando a

taxa de pacote de uma sub-rede atinge um determinado limite, um novo sub-nó é criado para acompanhar as taxas de pacotes mais finas. Esse processo prossegue até que sejam mantidas as taxas de pacotes de endereços IP. Portanto, a partir de uma granularidade grosseira, pode-se detectar mais precisamente a fonte de ataque exata ou os endereços de destino. Já o IDS proposto em Hoque, Kashyap e Bhattacharyya (2017), no qual esse trabalho é baseado, propõe-se um *framework* de detecção de ataques DDoS baseado em uma correlação proposta pelos mesmos autores chamada NaHiD, a qual é capaz de verificar a cada segundo, se uma instância de tráfego é normal ou maliciosa. O *framework* possui três componentes: Pré-processamento, Módulo de detecção e gerenciador de segurança. O primeiro é responsável por capturar e filtrar os pacotes em uma janela de um segundo. No módulo de detecção, a correlação será calculada entre um perfil normal pré-estabelecido e o tráfego em análise. Se o valor da correlação for maior que o limiar também pré-estabelecido, significa que o tráfego analisado tem similaridade alta com um tráfego normal, sendo julgado dessa forma e o *framework* irá atualizar esse perfil normal. Caso contrário, o tráfego será atribuído como um ataque. O gerenciador de segurança salva as estatísticas de tráfego de rede, bem como os valores calculados em *logs*. A medida de correlação NaHiD leva em consideração três parâmetros: Entropia e Variação de IPs origem e taxa de pacotes. A entropia é calculada baseada na fórmula de Shannon, a qual mede o grau de desorganização de um conjunto. Já na variação de IPs origem, tem-se que cada variação de um IP origem em um conjunto de pacotes deve ser incrementada. Por tratar-se de detecção em tempo real, os autores propuseram o cálculo da correlação NaHiD em *hardware* (FPGA), pois o tempo de detecção cairia drasticamente com essa modificação. Assim, tal correlação apresentou baixo tempo de processamento entre a detecção em *hardware* e *software*, sendo da ordem de 1 microssegundo para identificar um tráfego.

3 METODOLOGIA

Nesse capítulo são apresentadas a medida de correlação utilizada no trabalho, além das principais características do *framework*, mostrando como a correlação é aplicada para a detecção de ataques DDoS e quais bases de dados são utilizadas para a avaliação do *framework*, destacando sua estrutura e ferramentas utilizadas para o tratamento desses dados.

3.1 Modelo de correlação NaHiD

Neste trabalho, o *framework* utilizado baseia-se na correlação proposta por (HOQUE; KASHYAP; BHATTACHARYYA, 2017) chamada NaHiD, cujo objetivo é distinguir objetos de tráfego normais e maliciosos. Tal medida leva em consideração o desvio padrão e a média de cada objeto, ponderando cada elemento como mostrado na equação a seguir:

$$NaHiD(X,Y) = 1 - \frac{1}{n}\sum_{i=1}^{n}\frac{(|X(i) - Y(i)|)}{||\mu X - sX| - X(i)| + ||\mu Y - sY| - Y(i)|} \tag{3.1}$$

onde

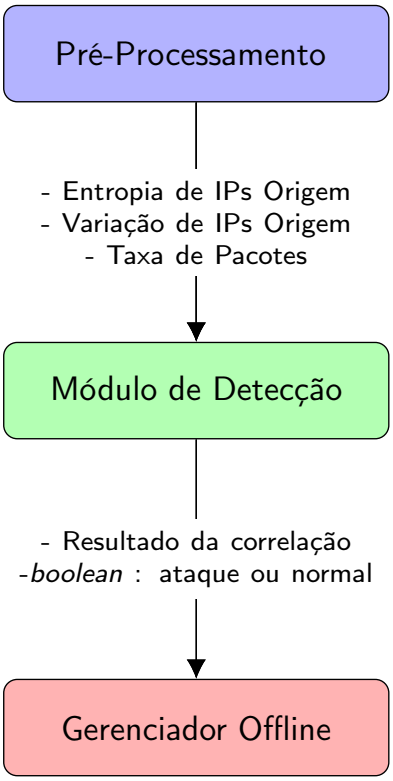
- μX: Média aritmética do objeto de tráfego X.
- μX: Média aritmética do objeto de tráfego Y.
- sX: Desvio padrão do objeto de tráfego X.
- sY: Desvio Padrão do objeto de tráfego Y.

As provas de simetria e identidade da correlação podem ser encontradas em (HOQUE; KASHYAP; BHATTACHARYYA, 2017).

3.2 *Framework* de detecção de ataques DDoS

O *framework* tem como objetivo, detectar ataques DDoS em tempo real na rede monitorada, a partir de dados trafegados na rede com uma taxa aceitável de erros. Tal arcabouço possui três módulos: pré-processamento, detecção e um de segurança. A Figura 3.1 mostra o fluxo de funcionamento do *framework*. Amostras de tráfego são capturadas de uma porta do roteador na forma de um pacote TCP/IP e enviadas ao módulo de pré-processamento. Nessa fase, a cada segundo, os pacotes recebidos são agrupados e essa instância de tráfego é enviada para o módulo de detecção de ataques, que irá classificar a instância como normal ou maliciosa. O gerente de segurança manterá um perfil normal, como referência, e um valor limiar de correlação em sua base de perfis, para ser usado pelo módulo de detecção. Incrementalmente, o gerente

Figura 3.1 – Estrutura do *framework* analisado



Fonte: Elaborada pelo autor.

recalcula o perfil normal baseado nos valores anteriores. Se o tráfego anterior for considerado normal, este será o tráfego referencial para a correlação na janela seguinte.

3.2.1 Pré-Processamento

Nessa etapa, os dados são coletados por um *sniffer* da rede, o qual analisa todos os pacotes trafegados e, a cada segundo, as métricas desejadas são calculadas para servirem de entrada para a correlação NaHiD.

3.2.1.1 Entropia de IPs origem

A entropia de IPs origem é uma medida do grau de desordem, onde ela é máxima caso todos os elementos sejam diferentes e o tamanho da entrada seja máximo, e será mínima (igual a 0) quando todos os elementos forem iguais, independentemente do tamanho. Assim, a entropia é dada pela seguinte fórmula:

$$H(X) = -\sum_{i}^{n} p(x_i) log_2(x_i) \tag{3.2}$$

Onde X é a entrada e representa os IPs origem das requisições e n é o número total de valores possíveis para o IP origem. A Tabela 3.1 mostra exemplos com valores de entrada para entropia,

bem como o resultado do cálculo da função.

Note que a entropia é mínima quando todos os IPs origem são iguais (primeira linha da tabela) e

Tabela 3.1 – Exemplo de IPs origem com respectivos valores de entropia

IPs origem					Entropia
192.168.8.8	192.168.8.8	192.168.8.8	192.168.8.8	192.168.8.8	0
192.168.15.129	192.168.8.5	192.168.8.8	192.168.10.16	192.168.20.22	2.3219
192.168.8.8	192.168.8.8	192.168.8.5	192.168.10.16	192.168.20.22	1.9219
192.168.20.22	192.168.20.22	192.168.20.22	192.168.20.22	192.168.8.8	0.7219

Fonte: Elaborada pelo autor.

máxima quando todos os os IPs origem são diferentes (segunda linha).

3.2.1.2 Variação de IPs Origem

Essa medida, diferentemente da entropia, trata-se da taxa de mudança dos IPs origem e é calculada da seguinte forma:

$$V_{Ip}(X) = \frac{\delta}{N} \tag{3.3}$$

Onde δ é o número de mudanças de IPs origem e N é o número total de IPs de entrada. Neste trabalho consideramos uma variação cada troca de valores como no exemplo:

$$X = 1, 2, 1, 2, 3 \tag{3.4}$$

Assim, nesse vetor consideram-se 4 variações ainda que sejam para um valor que repetiu-se. Assim se os IPs origem mudarem frequentemente, a variação será alta (HOQUE; KASHYAP; BHATTACHARYYA, 2017).

A observação do comportamento de ataques por *flood* mostra que esse tipo de ameaça pode ser gerada por atacantes reais como zumbis. Se endereços de IP origem falsificados forem utilizados durante um ataque DDoS TCP SYN, a entropia e variação de IPs origem serão altas e esse comportamento também ocorre em um tráfego normal (HOQUE; KASHYAP; BHATTACHARYYA, 2017). Assim faz-se necessário o uso da taxa de pacotes em *bits* como terceiro parâmetro de entrada para o módulo de detecção.

3.2.2 Módulo de Detecção

O modulo de detecção consiste na aplicação da correlação NaHiD, utilizando os três parâmetros de entrada fornecidos pelo módulo de pré-processamento:

- Variação de IPs origem
- Entropia de IPs origem
- Taxa de pacotes

Por tratar-se de uma medida de correlação, é necessário manter um valor de referência para o cálculo. Assim, os parâmetros (Variação e entropia de IPs origem, além da taxa de pacotes) de um tráfego normal devem ser fixados para a comparação com a instância de tráfego a ser analisada. Além disso, define-se um limiar do resultado da correlação para distinguir instâncias normais de maliciosas. Caso a correlação calculada seja menor que esse limiar, o tráfego analisado não possui semelhança suficiente com o perfil normal comparado, sendo, portanto, considerado um ataque.

3.2.3 Gerenciador de Segurança

Nesse módulo, os valores de correlação, IPs origem, destino, taxas de pacotes, além dos resultados do módulo de detecção são salvos para a janela de tráfego em análise e caso o módulo de detecção identifique que o tráfego em questão é normal, este será atualizado com os valores do mesmo para a próxima janela.

3.3 Aplicação do *framework* de detecção em bases de dados reais

Para a avaliação do trabalho, duas bases de tráfegos de rede foram escolhidas: DARPA e *DataMining*, os quais são mais detalhados a seguir.

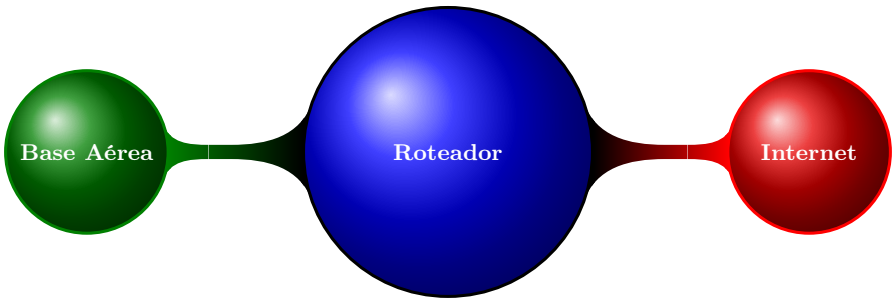
3.3.1 DARPA 2000

A base de dados DARPA foi produzida por pesquisadores do *Lincoln Laboratory* do Instituto de Tecnologia de Massachusetts (MIT) nos Estados Unidos e tem por objetivo coletar dados de tráfego de rede da Força Aérea do país para encontrar vulnerabilidades em seu sistema bem como ser utilizado para avaliações futuras. Os dados foram coletados e passaram por uma fase de treinamento de 5 semanas com 19 tipos de ataques para simular ameaças internas a rede. O ambiente de rede era composto por duas partes: a rede interna da Força aérea e a rede externa que representava a *Internet*; ambos conectados por meio de um roteador como mostra a Figura 3.2. Assim, um *sniffer* de rede foi instalado no roteador e todas as requisições para os computadores da Força aérea foram capturadas em um arquivo *tcpdump*, o qual pode ser encontrado em (LIPPMANN *et al.*, 2000a).
A partir desse *dataset* é possível extrair informações acerca de cada pacote transmitido durante o período de aquisição dos dados como mostra o exemplo na Tabela 3.2.

No presente trabalho a ferramenta *Wireshark* foi utilizada para o tratamento desse *dataset* no módulo de processamento. Assim, algumas considerações devem ser feitas:

- Janela de um segundo de tráfego.
- Cálculo de entropia, variação de IPs origem e taxa de pacotes média.
- Cálculo da correlação NaHiD com base no item anterior.

Figura 3.2 – Estrutura de rede Base Aérea dos EUA



Fonte: Elaborada pelo autor.

Tabela 3.2 – Exemplo base de dados DARPA

Número	Tempo	Origem	Destino	Protocolo	Tamanho
2710	09:12:42	194.27.251.21	172.16.112.100	TCP	60
2711	09:12:42	3comFast_9c:b2:54	*Broadcast*	ARP	60
2712	09:12:42	Cisco_38:46:33	3comFast_9c:b2:54	ARP	60
2713	09:12:42	172.16.112.100	194.27.251.21	TCP	60
2714	09:12:42	194.27.251.21	172.16.112.100	TCP	60
2715	109:12:42	3comFast_9c:b2:54	*Broadcast*	ARP	60
2716	09:12:42	Dell_a3:57:db	3comFast_9c:b2:54	ARP	60
2717	09:12:42	172.16.112.100	172.16.112.20	DNS	86

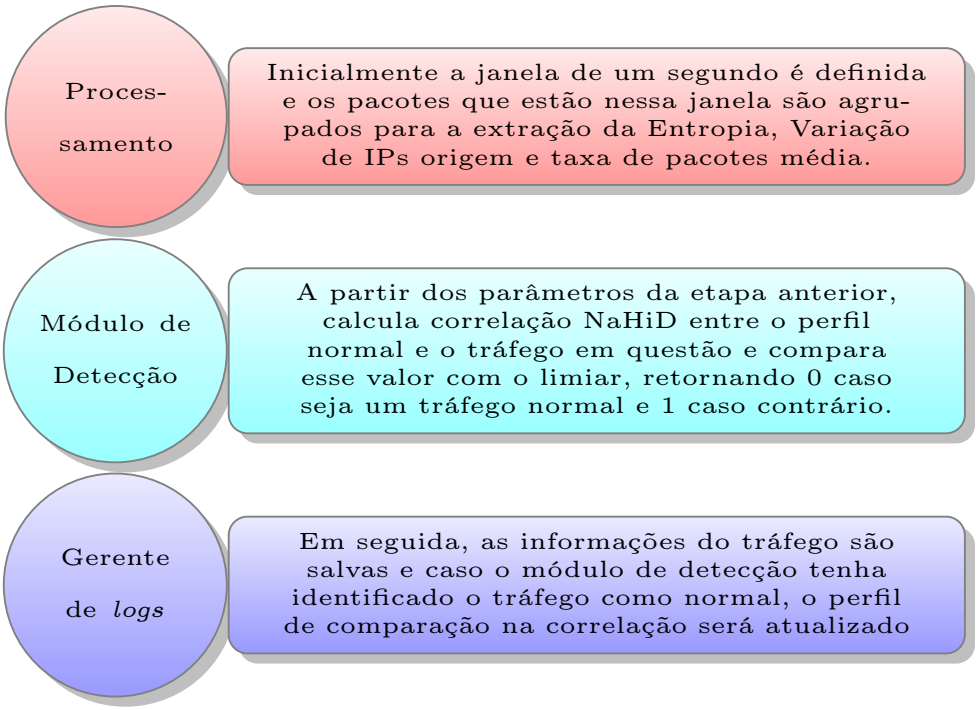
Fonte: Elaborada pelo autor, baseada em (LIPPMANN *et al.*, 2000b).

Note que de acordo com a estrutura do *dataset* mostrada na Tabela 3.2, a implementação do *framework* segue o seguinte fluxo mostrado na Figura 3.3

A documentação da base de dados DARPA, fornece informações de quais fluxos capturados são ataques. Possuindo a estrutura da Tabela 3.3

Com base nas informações da documentação, cada ataque ocorre no período descrito conforme Tabela 3.3. Ao abrir o *dataset* no *Wireshark* percebe-se também que existe a variação de fuso-horário que deve ser convertida, sendo no caso uma hora adiantado. Assim, aplica-se um

Figura 3.3 – Diagrama descritivo análise e aplicação do *framework* na base de dados DARPA



Fonte: Elaborada pelo autor.

Tabela 3.3 – Exemplo da estrutura do arquivo de respostas do *dataset* DARPA 2000

Tempo	**Duração H:Min:Seg**	**Protocolo**	**Origem**	**Destino**
08:12:42	00:00:05	SMTP	194.27.251.21	172.16.112.100
08:49:17	00:00:06	SMTP	135.8.60.182	172.16.112.100
08:55:10	00:01:05	SMTP	206.48.044.018	172.16.112.100
09:44:17	00:00:40	FTP	172.16.113.204	172.16.112.100
09:55:27	00:00:03	SMTP	153.107.252.061	72.16.112.100
10:06:42	00:03:43	TELNET	172.016.118.070	172.16.112.100
10:40:29	00:00:03	FTP	205.160.208.190	172.16.112.100
10:52:28	000:02:56	TELNET	172.016.118.070	172.16.112.100

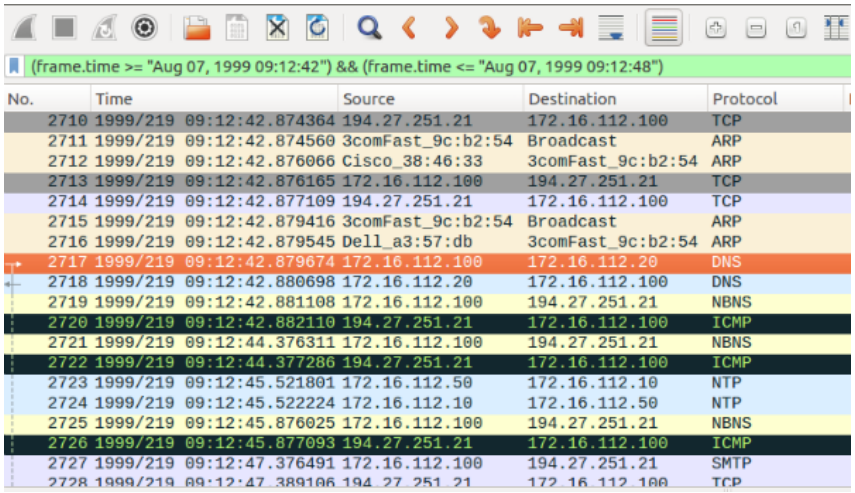
Fonte: Elaborada pelo autor, baseada em (LIPPMANN *et al.*, 2000b).

filtro no *Wireshark* conforme o exemplo abaixo.

$$(frame.time >= "Aug07, 199909:12:42")\&\&(frame.time <= "Aug07, 199909:12:48") \quad (3.5)$$

Desta forma, estamos filtrando os pacotes transmitidos entre o horário de $09:12:42$ e $09:12:47$, conforme primeiro elemento da Tabela 3.3. Assim, resultando nos pacotes filtrados conforme Figura 3.4.

Figura 3.4 – Resultado do filtro aplicado no *Wireshark*



Fonte: Elaborada pelo autor

Vale ressaltar que o ataque não consta em toda a janela de duração, podendo ser encontrado dentro de tal intervalo. Assim, para meios de detecção, o *framework* irá buscar um ataque dentro do intervalo dado pelo arquivo de respostas. Se houver pelo menos um ataque dentro dessa duração, o módulo de detecção terá acertado.

3.3.2 DataMining

Outra base de dados estudada no trabalho foi a desenvolvida por (ALKASASSBEH *et al.*, 2016), a qual consta em sua totalidade por ataques DDoS de quatro tipos:

- SIDDoS
- HTTP *Flood*
- UDP *Flood*
- *Smurf*

Esse *dataset* foi gerado em um simulador de rede chamado NS2, um *software* que representa uma rede de computadores de forma realista. A Tabela 3.4 mostra os campos do *dataset* avaliado.

Algumas considerações foram tomadas para a análise dessa base de dados:

- Para construir a janela de um segundo, considerou-se a soma de todos os atrasos por pacote:

– Atraso de nó do pacote.

– Atraso de pacote.

Tabela 3.4 – Estrutura base de dados *Data Mining*

Número	**Tempo**
1	Endereço IP origem
2	Endereço IP destino
3	Id do pacote
4	Nó origem
5	Nó destino
6	Tipo de pacote
7	Tamanho do pacote
8	*Flags*
9	Id da *flag*
10	Número de sequência
11	Número de pacotes
12	Número de *bytes*
13	Nome do nó origem
14	Nome do nó destino
15	Entrada de pacote
16	Saída de pacote
17	Taxa de pacotes Recebidos
18	Atraso de nó do pacote
19	Taxa de pacotes
20	Taxa de *bytes*
21	Tamanho médio do pacote
22	Utilização
23	Atraso de pacote
24	Tempo de envio do pacote
25	Tempo de pacote reservado
26	Primeiro pacote enviado
27	Último pacote reservado

Fonte: Elaborada pelo autor, baseada em (ALKASASSBEH *et al.*, 2016).

  - Tempo de pacote reservado.

  Essa consideração foi tomada devido a natureza do *dataset*, o qual não tem outra abordagem para a recepção.

- A média das taxas dos pacotes foi considerada dentro da janela de um segundo.
- Por ser um *dataset* composto apenas por ataques, a comparação com o limiar inverte-se para denotar o quanto dois pacotes são parecidos na correlação.

A base de dados é disponibilizada no formato *Weka Attribute-relation*(extensão arff), o qual é utilizado geralmente para compactar grandes massas de dados e processá-las utilizando técnicas de *machine learning*. Assim, para o processamento dos mesmos as ferramentas Weka e MATLAB foram utilizadas. Inicialmente, no módulo de Pré-Processamento, converteu-se o arquivo .arff para um .mat por meio de *script* e em seguida, com o *dataset* carregado, a janela de um segundo é aplicada, conforme as considerações já mencionadas. Em seguida, os pacotes da janela são agrupados para a extração dos IPs origem, a partir de onde serão calculadas a entropia e variação, além da taxa de pacotes média. Posteriormente, a partir dos parâmetros da etapa anterior, o módulo de detecção calcula a correlação NaHiD entre o perfil ataque e o tráfego em questão e compara com o limiar, retornando 0 caso seja um tráfego normal e 1 caso contrário. Após isso, no gerente *offline*, as informações do tráfego analisado são salvas e caso o módulo de detecção tenha identificado o tráfego como ataque, o perfil de comparação na correlação será atualizado.

3.4 Método de avaliação do *framework*

Como forma de avaliação são consideradas as taxas de acertos, de falsos positivos e negativos. Em outras palavras, os *datasets* avaliados possuem arquivos de respostas, onde os tráfegos normais e ataques são descriminados. Assim, a resposta do *framework* estudado será comparada com esse arquivo e a taxa de acertos será calculada da seguinte forma

$$T_a = \left(1 - \frac{N_e}{N_t}\right) * 100, \tag{3.6}$$

onde T_a é a taxa de acertos, N_e é o número tráfegos julgados erroneamente, N_t é o número de tráfegos analisados pelo *framework*. Além disso, define-se como métrica a taxa de falsos positivos, que denota a percentagem dos tráfegos normais que foram classificados como ataques e a taxa de falsos negativos, que expressa a percentagem dos tráfegos maliciosos que foram classificados como normais.

4 EXPERIMENTOS E RESULTADOS

Nesse capítulo, são apresentados detalhes dos experimentos realizados e dos resultados obtidos a partir de simulações utilizando as bases de dados mostradas no Capítulo 3.

4.1 Análise *dataset DataMining*

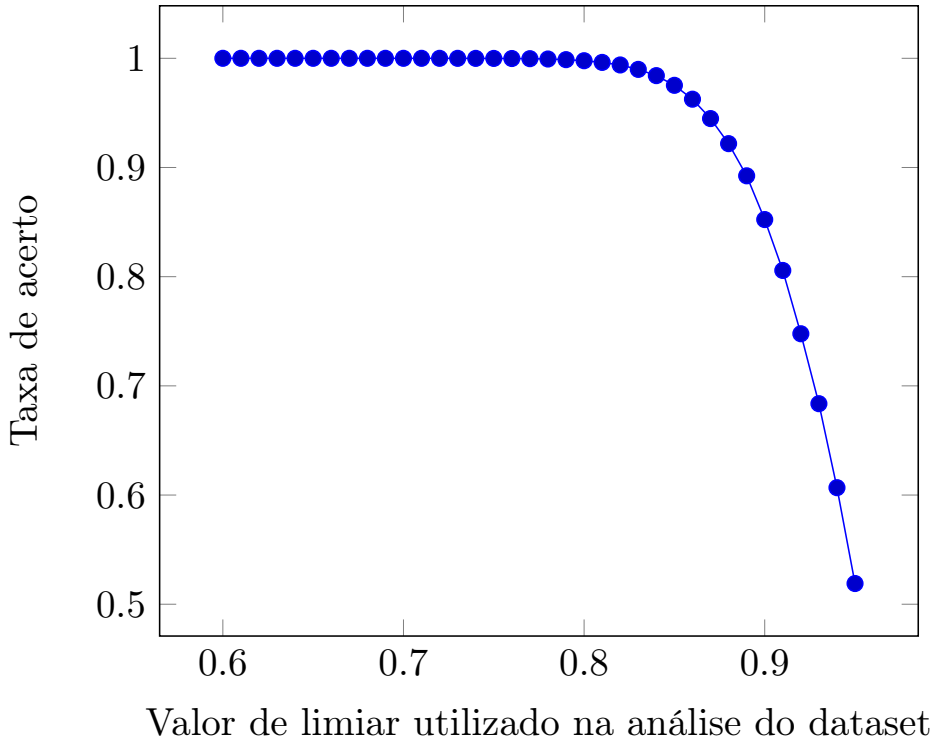
Na Figura 4.1 tem-se o gráfico da taxa de acerto em função do valor do limiar, o qual foi obtido aplicando o *dataset* proposto por (ALKASASSBEH *et al.*, 2016) ao *framework*. Para valores de limiar entre 0.8 e 0.84 a taxa de acerto permanece acima de 98.4%, e conforme o aumento do limiar, além de 0.84, a taxa de acerto vai decrescendo. Tal comportamento é esperado, visto que mesmo um tráfego não malicioso, porém suavemente descorrelacionado do tráfego padrão de referência, seria classificado como um ataque. Nesse sentido, vale ressaltar que existem diferentes tipos de ataques DDoS, os quais geralmente possuem abordagens singulares para a realização dos mesmos. Dessa forma, a escolha de limiares próximos a um não representa uma boa escolha para este tipo de problema, uma vez que a granularidade dos ataques não seria abrangida pelo *framework*. Na tabela Tabela 4.1 são mostrados as taxas de acertos, falsos positivos e falsos negativos na análise do *dataset*, complementando o gráfico apresentando na Figura 4.1.

Como pode ser visto na tabela Tabela 4.1, a coluna ”Taxa de falsos positivos” não apresentou nenhuma ocorrência (NA). Esse fato, deve-se especialmente à natureza do *dataset*, o qual é integralmente composto de tráfegos maliciosos e, dessa forma, não se aplica o conceito de falso positivo. Já na coluna ”Taxa de falsos negativos” (tráfego de ataque que foi considerado normal), vemos que dependendo do valor de limiar, temos um número considerável de eventos. Para um valor elevado de limiar, a taxa de acertos baixa e consequentemente, o número de falsos negativos aumenta.

4.2 Análise *dataset* DARPA 2000

A segunda base de dados usada para avaliar o *framework* é o DARPA, conforme descrito na Capítulo 3. A Figura 4.2 apresenta os gráficos das taxas de acertos em função dos limiares simulados. Para limiares entre 0.64 e 0.68, as taxas de acertos ficam entre 60 a 85%. Tal comportamento é diferente da base de dados anterior, pois as bases diferem em termos de tratamento temporal, modo de detecção e arquivo de respostas, já que no DARPA é mostrada uma janela de tempo onde provavelmente o ataque nomeado irá ocorrer, enquanto no *Datamining* tem-se uma base constituída apenas por ataques. Nesse sentido, espera-se que os valores de taxa de acerto de detecção sejam diferentes para ambos os *datasets*. A Tabela 4.2 mostra as taxas de acertos, falsos positivos e falsos negativos resultantes da análise do *dataset* DARPA 2000. Verifica-se

Figura 4.1 – Análise do *dataset Data Mining*



Fonte: Elaborada pelo autor.

uma baixa taxa de falsos positivos e negativos, tendo em vista que a taxa de detecção é máxima para a maioria dos limiares simulados, mostrando a eficiência do método de detecção.

A Figura 4.3 mostra os resultados obtidos por (HOQUE; KASHYAP; BHATTACHARYYA, 2017) para a análise do *dataset* DARPA. Para limiares acima de 70%, ambos os gráficos têm 100% de acerto. Assim, ao escolher um intervalo válido de limiares para realizar a detecção deve-se levar em conta os valores para os quais a detecção teve maiores taxas de acerto.

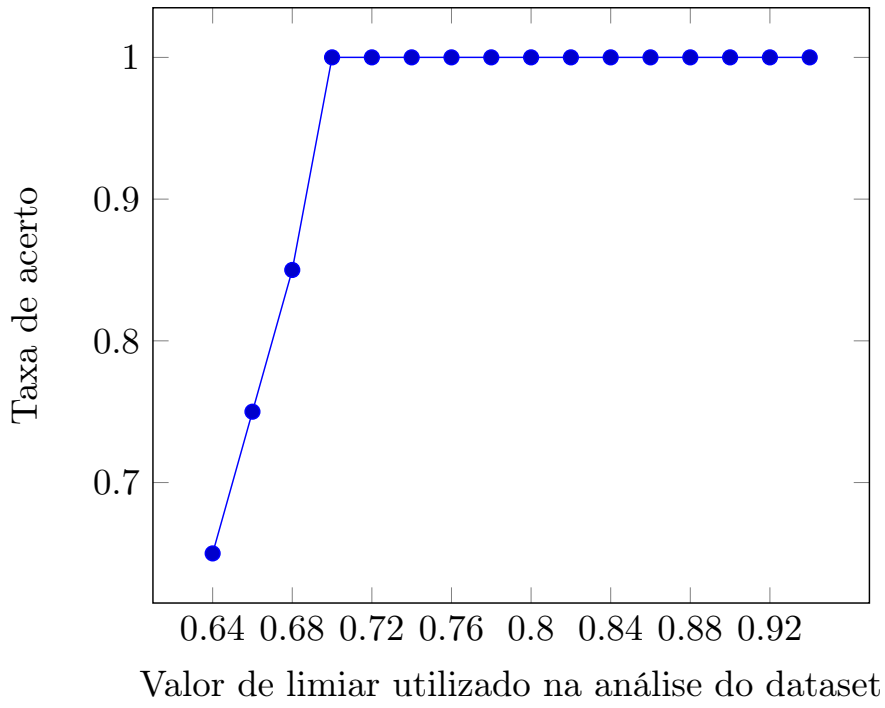
Infelizmente (HOQUE; KASHYAP; BHATTACHARYYA, 2017) não explicita a versão da base de dados DARPA utilizada em seu trabalho, tampouco respondeu a nossa tentativa de contato por *e-mail*. Portanto, não podemos afirmar categoricamente que usamos a mesma versão do *dataset* DARPA usado por Hoque, Kashyap e Bhattacharyya (2017). Ao comparar os gráficos resultantes, verificamos grande semelhança em seus valores de taxa de acerto, confirmando que o *framework* implementado está funcionando corretamente.

Tabela 4.1 – Exemplo base de dados Data Mining

Limiar	**Taxa de acerto**	**Taxa de falsos positivos**	**Taxa de falsos negativos**
0.60	100%	NA	0%
0.62	99.9996%	NA	0.0003996%
0.64	99.9996%	NA	0.0003996%
0.66	99.9996%	NA	0.0003996%
0.68	99.9996%	NA	0.0003996%
0.70	99.9992%	NA	0.0007993%
0.72	99.9992%	NA	0.0007993%
0.74	99.9992%	NA	0.0007993%
0.76	99.9800%	NA	0.0.01998%
0.78	99.9664%	NA	0.03357%
0.80	99.7790%	NA	0.2210%
0.82	99.3905%	NA	0.6095 %
0.84	98.4137%	NA	1.5863%
0.86	96.2556%	NA	3.7444%
0.88	92.1842%	NA	7.8158%
0.90	85.2309%	NA	14.7691%
0.92	74.7722%	NA	25.2278%
0.94	60.6753%	NA	39.3247%

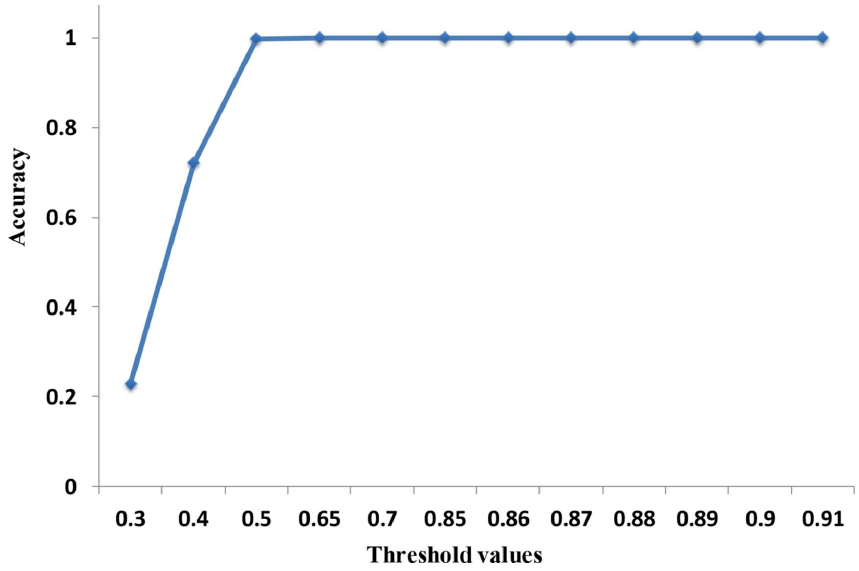
Fonte: Elaborada pelo autor.

Figura 4.2 – Análise do *dataset* DARPA



Fonte: Elaborada pelo autor.

Figura 4.3 – Resultados artigo (HOQUE; KASHYAP; BHATTACHARYYA, 2017)



Fonte: (HOQUE; KASHYAP; BHATTACHARYYA, 2017).

Tabela 4.2 – Exemplo base de dados DARPA

Limiar	**Taxa de acerto**	**Taxa de falsos positivos**	**Taxa de falsos negativos**
0.64	65%	10%	25%
0.66	75%	7.5%	17.5 %
0.68	82.5%	5%	12.5%
0.70	100%	0%	0%
0.72	100%	0%	0%
0.74	100%	0%	0%
0.76	100%	0%	0%
0.78	100%	0%	0%
0.80	100%	0%	0%
0.82	100%	0%	0%
0.84	100%	0%	0%
0.86	100%	0%	0%
0.88	100%	0%	0%
0.90	100%	0%	0%
0.92	100%	0%	0%
0.94	100%	0%	0%

Fonte: Elaborada pelo autor.

5 CONCLUSÕES E TRABALHOS FUTUROS

Esse trabalho apresentou um estudo de um *framework* de detecção de ataques DDoS, o qual recebe como entrada um arquivo contendo um dataset e analisa todos os tráfegos, fornecendo como resposta se o tráfego analisado é normal ou malicioso. O *framework* foi implementado na linguagem MATLAB, o qual recebe um arquivo de entrada contendo o tráfego a ser analisado. Para cada janela de um segundo, as métricas de entropia, variação de IPs origem e taxa de pacotes são calculadas e baseado nesses três valores calcula-se a correlação (NaHiD) que comparando com as métricas de um tráfego normal, retorna uma resposta entre 0 e 1, que denota o quanto o tráfego analisado é parecido com um normal. Vários valores de limiar são testados para avaliarmos o comportamento do *framework* em termos de taxa de acerto, falsos positivos e negativos.

5.1 Contribuições

A partir da implementação do *framework*, foram avaliadas duas bases de dados DataMining e DARPA, as quais continham tráfegos maliciosos. Os gráficos mostraram um comportamento esperado para ambas as bases de dados, sendo que para cada valor de limiar escolhido, o comportamento da curva de taxas de acerto no primeiro *dataset* possui valores próximos a 100% para limiares de até 0.82, e no caso da base de dados DARPA, a detecção é máxima para uma larga faixa de limiar (entre 0.7 e 0.94). Ao comparar com (HOQUE; KASHYAP; BHATTACHARYYA, 2017), observou-se um comportamento muito semelhante para a base de dados DARPA. Conclui-se portanto que o sistema implementado é capaz de detectar ataques DDoS, sendo portanto, válido para detecção desse tipo de ataques, mostrando comportamento excelente ao ser avaliado segundo taxa de acertos.

Como uma última contribuição deste trabalho, todos os códigos-fonte do *framework* implementado, bem como os resultados e o código-fonte em LaTeX desse documento monográfico, estão disponíveis para acesso sob a licença GPL(*Gnu Public License*) versão 3 no link : <https://github.com/jardelufc/RTDDoS>. Adicionalmente, acrescentamos que o código pode ser portado para rodar em Octave, visto que não foi utilizado nenhum *toolbox* proprietário.

5.2 Trabalhos Futuros

Como perspectivas futuras para o trabalho, as seguintes abordagens poderão ser adotadas:

- Estudar algumas técnicas de *machine learning* para aprimorar a detecção.
- Desenvolver o *framework* com limiares adaptativos para maximizar os acertos em outras bases de dados e em situações reais.
- Comparar o *framework* implementado com outros na literatura.

- Estudar modificações na correlação utilizada.
- Criar um ambiente controlado, com alguns *hosts* e um servidor para simular ataques reais e testar o *framework*.
- Criar uma aplicação a ser instalada no servidor, a qual ficaria analisando os pacotes e detectando possíveis ataques em quaisquer nós da rede.

BIBLIOGRAFIA

ALKASASSBEH, M. *et al.* Detecting Distributed Denial of Service Attacks Using Data Mining Techniques. v. 7, 01 2016. 16, 18, 27, 28, 30

ASHOOR, A. S.; GORE, S. Importance of intrusion detection system (ids). **International Journal of Scientific and Engineering Research**, v. 2, n. 1, p. 1–4, 2011. 19

BADISHI, G. *et al.* Exposing and eliminating vulnerabilities to denial of service attacks in secure gossip-based multicast. **IEEE Transactions on Dependable and Secure Computing**, IEEE, v. 3, n. 1, p. 45–61, 2006. 13

BARFORD, P. *et al.* A signal analysis of network traffic anomalies. In: ACM. **Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment**. [S.l.], 2002. p. 71–82. 13

CABRERA, J. B. *et al.* Proactive detection of distributed denial of service attacks using mib traffic variables-a feasibility study. In: IEEE. **Integrated Network Management Proceedings, 2001 IEEE/IFIP International Symposium on**. [S.l.], 2001. p. 609–622. 19

CALCE, M.; SILVERMAN, C. **Mafiaboy: How I Cracked the Internet and why It's Still Broken**. [S.l.]: Penguin Group Canada, 2008. 14

CENTER, C. C. **CERT advisory CA-1998-01 smurf IP denial-of-service attacks**. [S.l.]: January, 1998. 17

CHEN, E. Y. Detecting dos attacks on sip systems. In: **1st IEEE Workshop on VoIP Management and Security, 2006.** [S.l.: s.n.], 2006. p. 53–58. 13

CRISCUOLO, P. J. **Distributed denial of service: Trin00, tribe flood network, tribe flood network 2000, and stacheldraht ciac-2319**. [S.l.], 2000. 18

DITTRICH, D. **The DoS ProjectÕs "trinoo"Distributed Denial of Service attack tool, University of Washington, October 21, 1999**. 2002. 18

DOULIGERIS, C.; MITROKOTSA, A. Ddos attacks and defense mechanisms: classification and state-of-the-art. **Computer Networks**, v. 44, n. 5, p. 643 – 666, 2004. ISSN 1389-1286. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128603004250>. 16, 18

GIL, T. M.; POLETTO, M. Multops: A data-structure for bandwidth attack detection. In: **USENIX Security Symposium**. [S.l.: s.n.], 2001. p. 23–38. 19

HOQUE, N. *et al.* Real-time DDoS Attack Detection Using FPGA. **Computer Communications**, v. 110, n. Supplement C, p. 48 – 58, 2017. ISSN 0140-3664. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0140366416306442>. 9, 14, 20, 21, 23, 31, 33, 35

HUANG, Y.; PULLEN, J. M. Countering denial-of-service attacks using congestion triggered packet sampling and filtering. In: IEEE. **Computer Communications and Networks, 2001. Proceedings. Tenth International Conference on**. [S.l.], 2001. p. 490–494. 19

KUROSE, J. F.; ROSS, K. W. **Redes de Computadores e a Internet: Uma abordagem top-down**. Trad. 5 ed. São Paulo: Pearson, 2010. 15

LIPPMANN, R. *et al.* **DARPA INTRUSION DETECTION EVALUATION**. 2000. <https://www.ll.mit.edu/ideval/index.html>. Acessado em 18/08/2017. 24

LIPPMANN, R. P. *et al.* Evaluating intrusion detection systems: the 1998 darpa off-line intrusion detection evaluation. In: **DARPA Information Survivability Conference and Exposition, 2000. DISCEX '00. Proceedings**. [S.l.: s.n.], 2000. v. 2, p. 12–26 vol.2. 25, 26

SRIVASTAVA, A. *et al.* A recent survey on ddos attacks and defense mechanisms. **Advances in Parallel Distributed Computing**, Springer, p. 570–580, 2011. 13

WANG, B. *et al.* Ddos attack protection in the era of cloud computing and software-defined networking. **Computer Networks**, v. 81, n. Supplement C, p. 308 – 319, 2015. ISSN 1389-1286. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128615000742>. 16

XIAOMING, L. *et al.* Denial of service (dos) attack with udp flood. **School of Computer Science, University of Windsor, Canada**, 2010. 18