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, Ceará

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, Ceará
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.

Aprovada em: ___/___/_____

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.

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

Resumo

Palavras-chaves: Detecção de ataques DDoS. Segurança em redes. Tempo real. Framework .

Abstract

Key-words: DDoS attack detection. Network Security. Real-time. Framework.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

DDOS	Distributed Denial of Service

Lista de símbolos

X	Vetor de entrada para correlação NaHiD

Sumário

1 Introdução

A rede mundial de computadores tem sido amplamente difundida nos ultimos anos

Ataques Distribuídos de Negação de Serviço (do inglês, DDoS) são uma ameaça a servidores de redes online, tais como servidores de sites web e servidores em nuvem. O objetivo desse tipo de ataque intencional é inundar o alvo com requisições e assim deixá-lo indisponível na rede. Existem essencialmente três tipos de ataques: Negação distribuída, Handshake e UDP. O primeiro caracteriza-se por requisições abertas por um grande número de computadores infectados. No segundo, faz-se uma comunicação inicial com o alvo que não é completada, mantendo assim o servidor esperando indefinidamente. Já no terceiro, fluxos falsos UDP são criados com o mesmo objetivo de tornar o serviço inoperante. Os métodos estatísticos existentes na literatura para análise de ataque DDoS falham principalmente devido às correlações de deslocamento, escala e deslocamento-escala ao longo de tráfegos de rede, gerando assim uma grande ocorrência de falsos positivos. Além disso, métodos estatísticos impõem alto overhead computacional quando um grande número de objetos é incluído para análise. Consequentemente, tais métodos falham em realizar detecção de ataque DDoS em tempo real. Algumas medidas de correlação tais como Pearson, Spearman e Kendall falham em identificar a diferença entre um pacote normal e um malicioso quando há valores correlacionados entre os pacotes. De fato, um método de detecção de ataques DDoS precisa considerar poucos parâmetros de tráfego durante a análise, tal como o método chamado NaHiDVERC (HOQUE; KASHYAP; BHATTACHARYYA, 2017), o qual analisa apenas entropia de IPs e taxa de pacote. Tendo em vista uma implementação em software e hardware, este método será utilizado em nosso trabalho, visto que é facilmente implementável em hardware. O método computa dois valores: a distância absoluta e o desvio entre A e B a partir da média e do desvio padrão. Se a entropia de IPs origem em um pequeno intervalo de tempo é alta e a taxa de pacote é também muito alta, a probabilidade de ataque é alta. Se a variação entre IPs origem é muito alta e a taxa de pacote também é alta, a probabilidade de ataque é alta. O framework tem como objetivo detectar ataques DDoS em tempo real no computador alvo. Trata-se de uma combinação entre aplicações em software e hardware, para classificar um tráfego como normal ou ataque com uma taxa aceitável de acertos. Tal arcabouço possui três componentes: pré-processamento, um módulo de hardware dedicado para detecção e um gerente de segurança. Neste trabalho os componentes um e três serão trabalhados. Além disso, é necessária a presença de um roteador para capturar tráfego e duas bases de dados. Amostras de tráfego serão capturadas de uma porta do roteador como um pacote TCP/IP, que são 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 ataque, que irá classificar a instância como normal ou ataque. O gerente de segurança manterá um perfil normal e um valor limiar em sua base de perfis, para ser usado pelo módulo de detecção. Incrementalmente, o gerente recalcula o perfil normal e o limiar baseado nos valores anteriores. Existem duas abordagens durante a análise do tráfego: uma considerando apenas a informação no cabeçalho do pacote ou se o cabeçalho e dados estarão juntos. Nas duas formas os campos dos pacotes são analisados para detectar alguma anomalia na rede. IP e porta origem/destino, protocolos e flags do cabeçalho TCP são úteis para detectar pacotes maliciosos. Assim, a entropia e a variação entre IPs origem e taxa de pacotes são calculados para cada amostra de tráfego. O último módulo, que é o módulo de segurança irá operar offline e fará análises detalhadas dos logs de detecção usando técnicas de machine learning e estatística. Além disso, feedbacks de especialistas podem ser utilizados para validar os resultados. Inicialmente, o gerente vai calcular um perfil de tráfego normal que melhor representa instâncias desse tráfego para treinamento. Esses valores serão carregados na base de dados. Vale ressaltar que esses valores serão modificados dinamicamente de acordo com as amostras de tráfego.

1.1 Objetivos

1.2 Organização da monografia

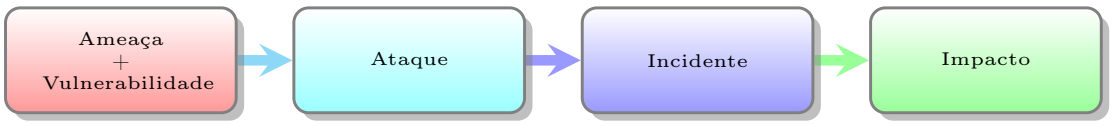
Os estudos deste trabalho estão organizados da seguinte forma: No próximo capítulo será apresentado um estudo bibliográfico sobre ameaças de rede e ataques DDoS. No Capítulo 3, a modelagem do ambiente de simulação utilizado neste trabalho é descrita. No Capítulo 4, apresentamos o desempenho obtido pelo *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 e algumas perspectivas para a continuação deste trabalho.

2 Revisão Bibliográfica

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, quebra de equipamentos, 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 semiabertas ou abertas no hospedeiro alvo.

Já segundo Douligeris e Mitrokotsa (2004), os ataques DoS são classificados em 5 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. Vários tipos de ataques DoS foram focados em DNS, e muitos deles envolvem atacar cache de DNS em servidores de nomes. Um invasor que possui um servidor de nomes pode coagir um servidor de nome alvo para armazenar de registros falsos ao receber uma requisição da vítima.

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

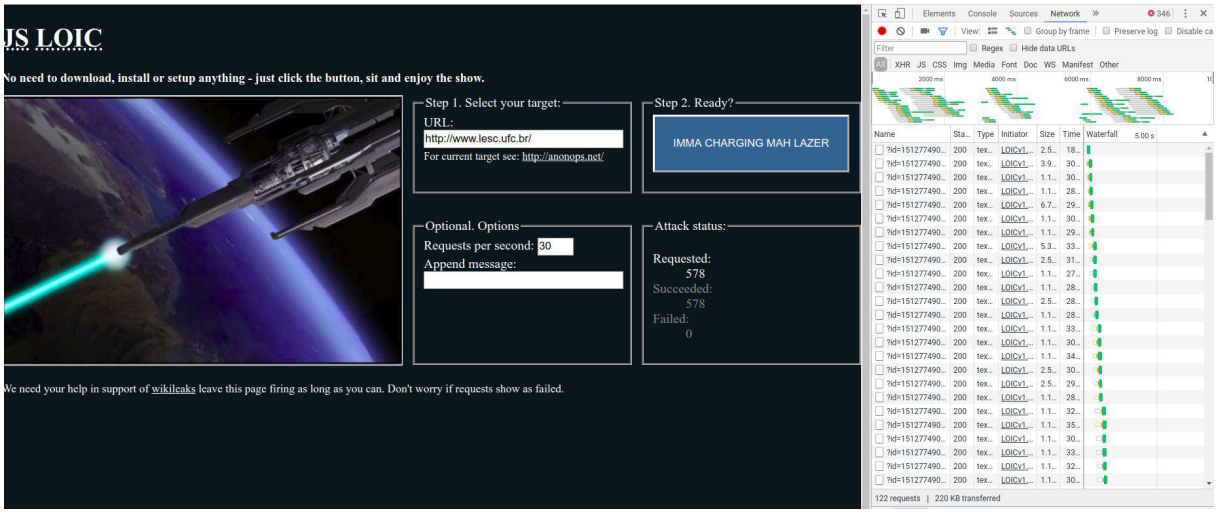
Ataques DDoS do tipo *Smurf* possuem dois componentes principais que são o uso de requisições ICMP (*Internet Control Message Protocol*) forjadas e a direção dos pacotes para um endereço *broadcast*. O protocolo ICMP é usado para troca de mensagens de controle e pode ser usado para determinar se uma 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 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 necessários 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 ou para portas aleatórias ou específicas no alvo. Para determinar o pedido requisitado, a vítima processa os dados recebidos. Em caso de ausência do pedido solicitado na porta solicitada, o sistema da vítima envia uma mensagem "Destino inacessível"ao remetente (invasor). 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; MITROKOTSA, 2004), um ataque UDP *flood* é possível quando um atacante envia um pacote UDP para uma porta aleatória do sistema dd 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 uma um pacote ICMP de destino inalcançável para o destino de origem forjado e com 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 um código malicioso SQL como uma string que irá passar para um banco de dados de um site web como uma equação. 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, pois o atacante tem acesso indireto ao banco por meio do mesmo, facilitando as consultas ao banco de dados por meio de códigos maliciosos. Em complemento a isso, uma vez que o atacante inseriu o código malicioso, ele utiliza-se a técnica do *proxy* reverso, que faz com que o servidor alvo estabeleça comunicação com a máquina do atacante e o mesmo passa a possuir controle total do alvo por meio do *prompt* de comando ou terminal.

2.3 IDS

Para tratar esse tipo de problema, sistemas de detecção de intrusos (do inglês, IDS) são utilizados. 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 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; GORE, 2011), 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 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, depende-se de 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, IDS utilizam alguns conceitos para realizarem 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 que entra e sai. 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; POLETTO, 2001) é proposta uma heurística chamada MULTOPS, que postula 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 pode ir até finalmente por taxas de pacotes de endereço IP serem mantidas. 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; BHATTACHARYYA, 2017), no qual esse trabalho é baseado, propõe um *framework* de detecção de ataques DDoS baseado em uma correlação proposta pelos mesmos autores chamada NaHiD, o 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 que está sendo analisada e enviar para o módulo de detecção, no qual a correlação será calculada entre um perfil normal pré-estabelecido e o tráfego será julgado baseado no valor da correlação e no limiar que também é pré-estabelecido. Se o valor da correlação for maior que o limiar, 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 n 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 calculo 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 (nome que possivelmente provém a partir das iniciais de cada autor), cujo objetivo é distinguir objetos de tráfego normais e maliciosos. Tal medida leva em consideração principalmente 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

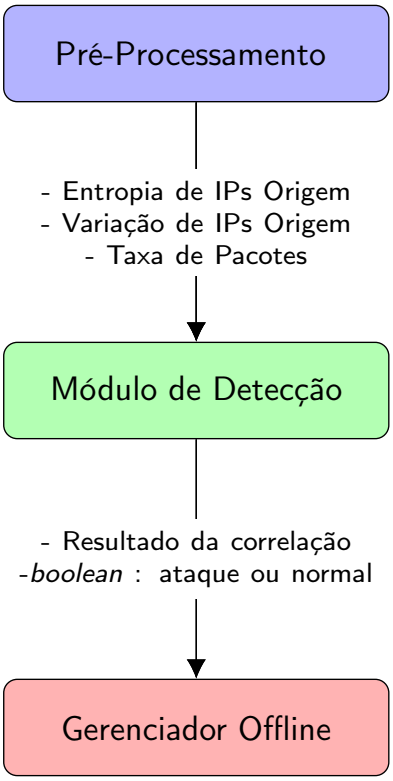
- μX: Média aritmética do objeto de tráfego X.
- μX: Média aritmética do objeto de tráfego Y.
- sX: Desvio padrão do objeto de tráfego X.
- sY: Desvio Padrão do objeto de tráfego Y.

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

Figura 3.1 – Estrutura do *framework* analisado



Fonte: Elaborada pelo autor.

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 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 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 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.

tabela) e 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 numero 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, tal tráfego em analisado não tem semelhança suficiente com o perfil normal comparado, sendo considerado um ataque.

3.2.3 Gerenciador Offline

Nessa 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 se o módulo de detecção identificar que o tráfego em questão é normal, este será atualizado com os valores do mesmo para a próxima janela.

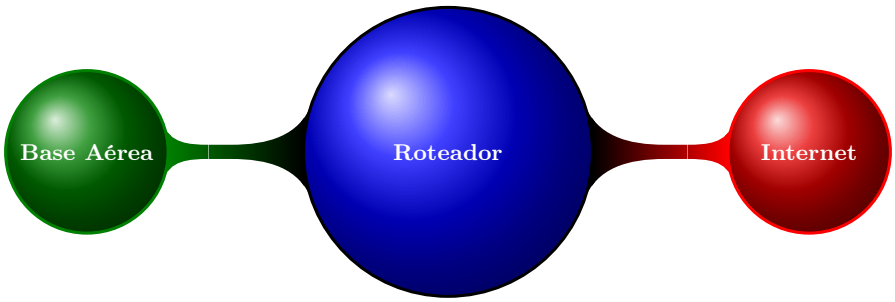
3.3 Aplicação do *framework* de detecção em bases de dados reais

Para a avaliação do trabalho, duas bases de tráfegos de rede foram escolhidas: DARPA e DataMining[escolher melhor esse nome] os quais são mais detalhados a seguir

3.3.1 DARPA - MIT

A base de dados DARPA foi produzida por pesquisadores do *Lincoln Laboratory* do Instituto de Tecnologia de Massachusetts 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 7 semanas com 38 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 (DARPA...,).
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 2.

Figura 3.2 – Estrutura de rede Base Aérea dos EUA



Fonte: Elaborada pelo autor.

Tabela 2 – Exemplo base de dados DARPA

Número	**Tempo**	**Origem**	**Destino**	**Protocolo**	**Tamanho**[bytes]
1	18:56:12.1386	192.168.0.20	192.168.0.30	TCP	60
2	18:56:12.1391	192.168.0.30	192.168.0.20	TCP	60
3	18:56:12.1588	192.168.0.30	192.168.0.20	TELNET	84
4	18:56:12.2099	192.168.0.20	192.168.0.30	TCP	60
5	18:56:13.0567	192.168.0.20	192.168.0.30	TELNET	69
6	18:56:13.0584	192.168.0.30	192.168.0.20	TELNET	66
7	18:56:13.0626	192.168.0.20	192.168.0.30	TELNET	72
8	18:56:13.0821	192.168.0.30	192.168.0.20	TCP	60

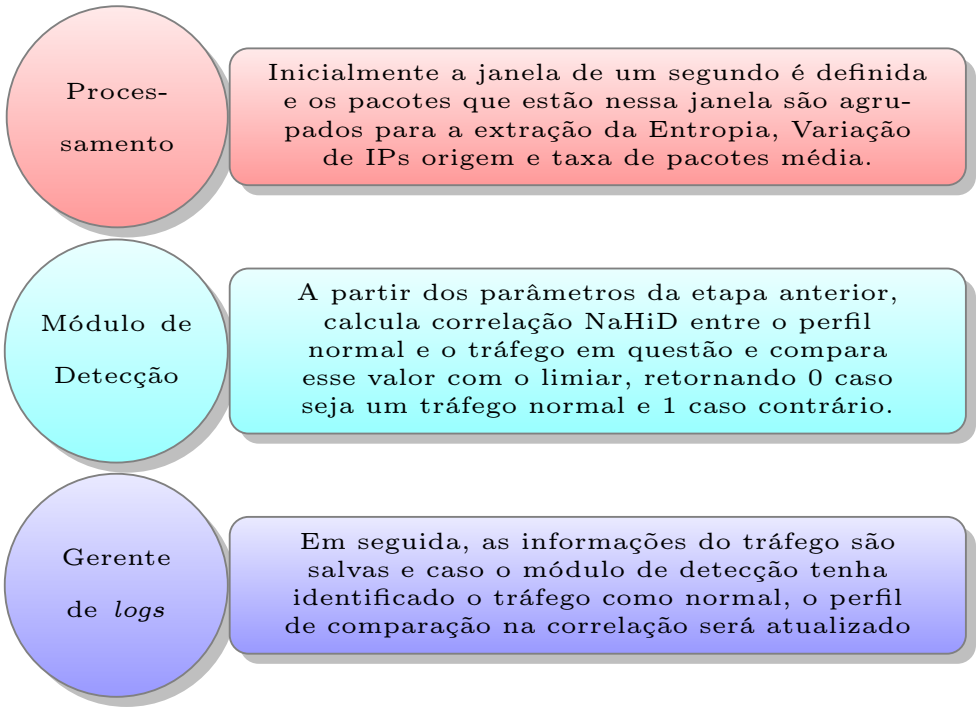
Fonte: Elaborada pelo autor, baseada em (LIPPMANN *et al.*, 2000).

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.

Note que de acordo com a estrutura do *dataset* mostrada na Tabela 2, a implementação do *framework* segue o seguinte fluxo mostrado na Figura 3.3

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



Fonte: Elaborada pelo autor.

.

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
- UDDP 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 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 – Estrutura base de dados (ALKASASSBEH *et al.*, 2016)

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.

- 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 normal 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 normal, 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

$$T_a = \left(1 - \frac{N_e}{N_t}\right) * 100, \tag{3.5}$$

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, as taxas de falsos positivos e negativos tem formulação semelhante.

4 Experimentos e Resultados

Nessa seção, detalhes dos experimentos realizados, bem como os resultados experimentais são mostrados a partir de simulações utilizando as bases de dados apresentadas na Seção 3.

4.1 Análise *dataset DataMining*

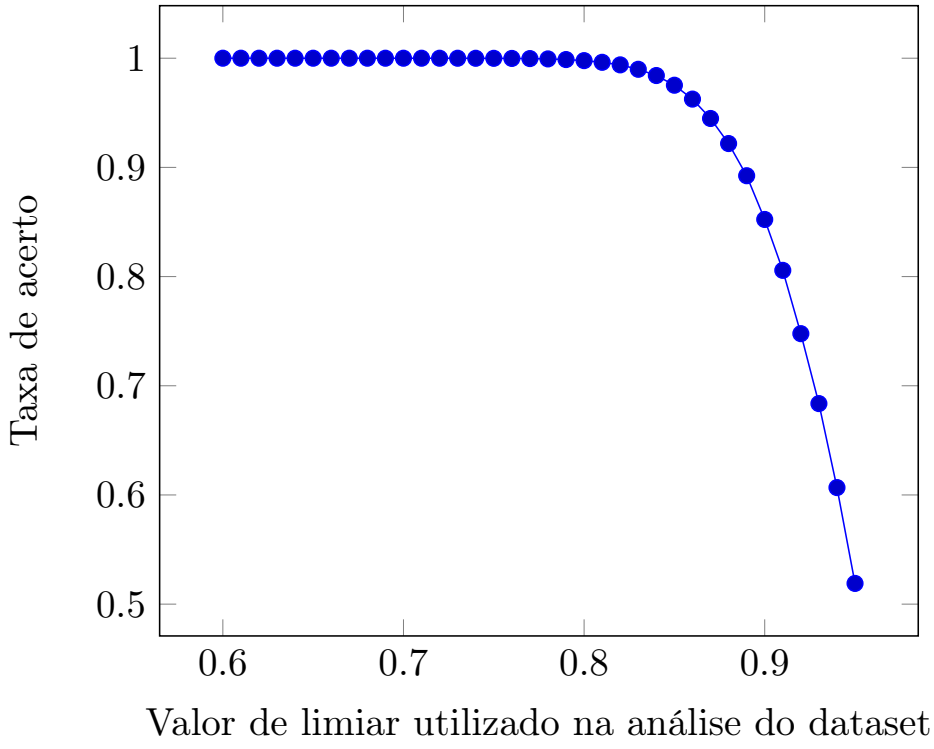
Na Figura 4.1 têm-se os gráficos de taxa de acerto em função dos diferentes limiares simulados para o *dataset* proposto por (ALKASASSBEH *et al.*, 2016), o qual possui apenas ataques e está descrito na Seção 3. Para valores de limiar entre 0.8 e 0.84 a taxa de acerto permanece acima de 98.4% e conforme o aumento do limiar, a taxa de acerto vai decrescendo. Tal comportamento é esperado, visto que para valores altos de limiar, o tráfego analisado deve ter propriedades (entropia, variação de IPs origem e taxa de pacotes) muito próximas do perfil normal para não ser considerado um ataque. Desta forma, vale ressaltar que existem diferentes tipos de ataques DDoS e geralmente possuem abordagens singulares para os ataques. Assim, seria uma escolha equivocada utilizar valores de limiar próximos a 1, pois a granularidade dos ataques não seria abrangida pelo *framework*. A Tabela 4 complementa o gráfico apresentado na Figura 4.1, pois mostra o número de acertos, falsos positivos e falsos negativos na análise do *dataset*.

Note que a coluna da taxa de falsos positivos consta com 0% devido a natureza do *dataset* ser apenas de ataques. Logo, não é possível o *framework* detectar um tráfego como ataque e ele na realidade ser um fluxo normal. Por outro lado, é possível detectar um tráfego como normal, sendo que ele trata-se de um ataque, como mostrado na tabela.

4.2 Análise *dataset* DARPA

Outra base de dados avaliada pelo *framework* é o DARPA conforme descrito na Seção 3. A Figura 4.2 apresenta os gráficos da taxa de acertos em função dos limiares simulados. Para limiares entre 0.64 e 0.68, a taxa de acertos fica entre 60 a 85%. Tal comportamento é diferente da base de dados anterior, pois tratam-se de bases que 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. Além disso, conforme explanado na Seção 3, o *framework* teve que ser adaptado para analisar o *dataset*, sendo possível analisar a base a qual contém apenas ataques. Logo, espera-se que os valores sejam diferentes para ambos os *datasets*. A Tabela 5 mostra as taxas de acertos,

Figura 4.1 – Análise do *dataset* avaliado



Fonte: Elaborada pelo autor.

falsos positivos e falsos negativos resultantes da análise. O *framework* possui baixa taxa de falsos 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. Vale ressaltar que na referência, a versão exata do *dataset* não foi explicitada. No entanto, os gráficos possuem semelhança em seus valores de taxa de acerto, sendo possível considerar validado o *framework*.

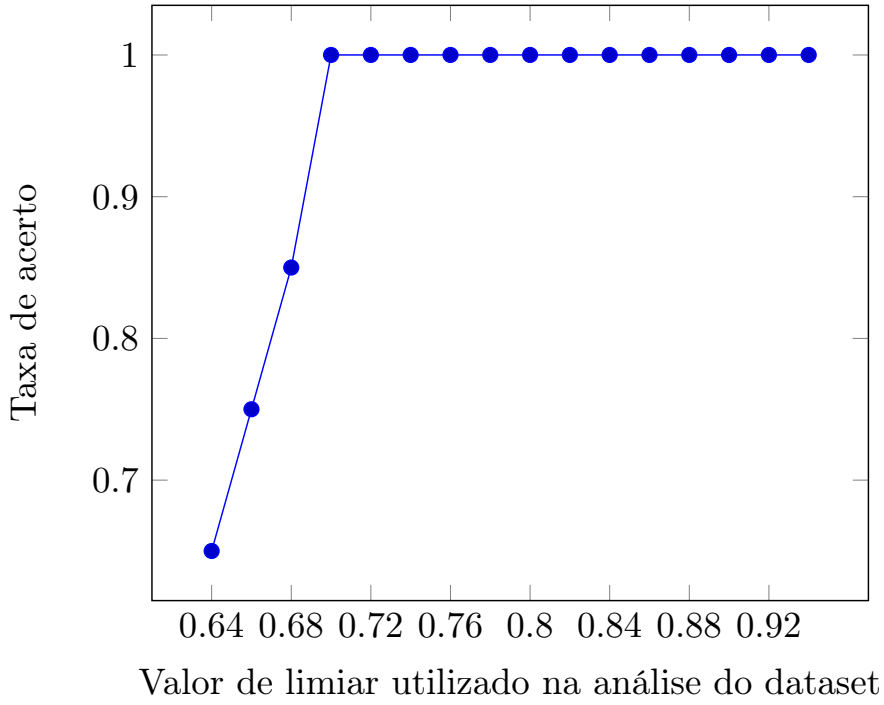
Para limiares acima de 70%, ambos os gráficos têm 100% de acerto. Assim, ao escolher um intervalo válido de limiares para realizar a detecção deve-se levar em conta os valores para os quais a detecção teve maiores taxas de acerto.

Tabela 4 – Exemplo base de dados DARPA

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

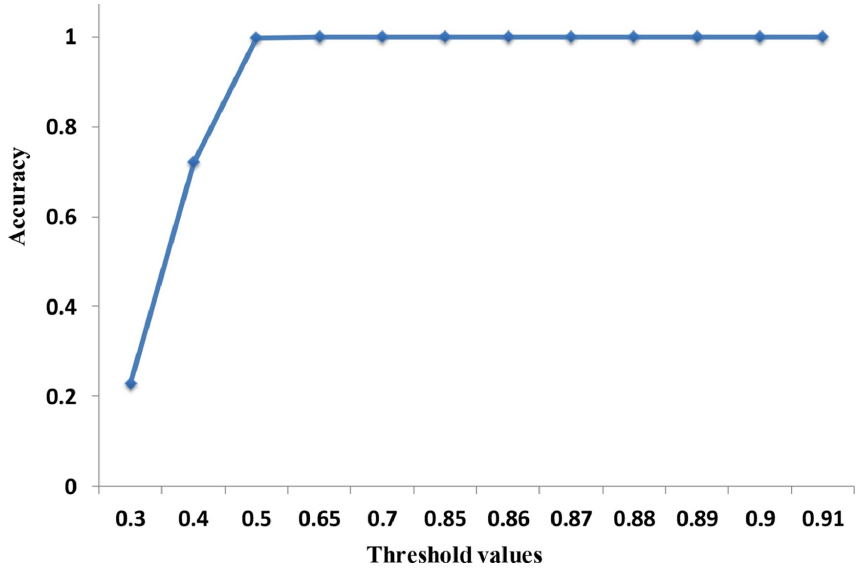
Fonte: Elaborada pelo autor.

Figura 4.2 – Análise do *dataset* avaliado



Fonte: Elaborada pelo autor.

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



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

Tabela 5 – Exemplo base de dados DARPA

Limiar	**Taxa de acerto**	**Taxa de falsos positivos**	**Taxa de falsos negativos**
0.64	65%	0%	35%
0.66	75%	0%	25 %
0.68	85%	0%	15%
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

Referências

ALKASASSBEH, M. *et al.* Detecting Distributed Denial of Service Attacks Using Data Mining Techniques. v. 7, 01 2016. Citado 6 vezes nas páginas 9, 16, 18, 26, 27 e 29.

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.

Citado na página 19.

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.

Citado na página 19.

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

Citado na página 17.

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

Citado na página 18.

DARPA INTRUSION DETECTION EVALUATION. <https://www.ll.mit.edu/ideval/index.html>. Acessado em 18/08/2017.

Citado na página 24.

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

Citado na página 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>. Citado 2 vezes nas páginas 16 e 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.

Citado na página 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>. Citado 7 vezes nas páginas 8, 13, 20, 21, 23, 30 e 32.

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.

Citado na página 19.

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

Citado na página 15.

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.

Citado na página 25.

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>.

Citado na página 16.

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

Citado na página 18.