PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Bacharelado em Ciência da Computação

Rodrigo Caetano de Oliveira Rocha

DETECÇÃO EM TEMPO-REAL DE ATAQUES DE NEGAÇÃO DE SERVIÇO NA REDE DE ORIGEM USANDO UM CLASSIFICADOR BAYESIANO SIMPLES

Belo Horizonte
2012

Rodrigo Caetano de Oliveira Rocha

DETECÇÃO EM TEMPO-REAL DE ATAQUES DE NEGAÇÃO DE SERVIÇO NA REDE DE ORIGEM USANDO UM CLASSIFICADOR BAYESIANO SIMPLES

Monografia apresentada ao programa de Bacharelado em Ciência da Computação da Pontifícia Universidade Católica de Minas Gerais, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Humberto Torres Marques Neto

Belo Horizonte
2012

Rodrigo Caetano de Oliveira Rocha

DETECÇÃO EM TEMPO-REAL DE ATAQUES DE NEGAÇÃO DE SERVIÇO NA REDE DE ORIGEM USANDO UM CLASSIFICADOR BAYESIANO SIMPLES

Monografia apresentada ao programa de Bacharelado em Ciência da Computação da Pontifícia Universidade Católica de Minas Gerais, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Humberto Torres Marques Neto

Fátima de Lima Procópio Duarte Figueiredo

Marco Aurélio de Souza Birchal

Belo Horizonte, 25 de Junho de 2012

RESUMO

Ataque de negação de serviço é um dos ataques mais comumente realizados por *botnets*. Implantando o mecanismo de defesa na rede de origem, tem-se que o fluxo de ataque poderá ser bloqueado antes de entrar no núcleo da Internet e ser agregado a outros fluxos, o que poderia causar congestionamento. O baixo grau de agregação de fluxos existente na origem permite usar estratégias de defesa mais complexas e com maior precisão. Este trabalho propõe um mecanismo de detecção em tempo-real de ataques de negação de serviço, utilizando um classificador bayesiano simples, implantado na rede de origem. Como a proposta do classificador é detectar ataques executados por agentes de uma *botnet*, o conjunto de amostras para treinamento deve representar ao máximo o comportamento de uma vítima secundária. Com base em uma avaliação matemática, obtiveram-se um limiar para cada ataque sendo detectado, de maneira que se um determinado fluxo apresentar características tais que estejam além desse limiar, o mesmo será classificado como um tráfego de ataque. Para casos críticos de ataque, sugere-se a utilização de mecanismos de detecção em diferentes locais de implantação, trabalhando em conjunto na detecção de ataques.

Palavras-chave: Ataque de negação de serviço. Tempo-real. Detecção. Classificador bayesiano simples. Segurança de redes.

ABSTRACT

Denial-of-service attack is one of the most common attacks performed by botnets. By deploying defense mechanism at the source network, one has that the attack flows can be stopped before entering the Internet core and blend with other flows, thereby creating possible congestion. The low degree of flow aggregation that there exists at the source allows the use of more complex defense strategies with higher accuracy. This paper proposes a mechanism for real-time detection of denial-of-service attacks, using a naive Bayesian classifier and that must be deployed at the attack source network. As the classifier proposal is to detect attacks carried out by botnet agents, the training dataset should represent the behavior of a secondary victim to the maximum. Based on a mathematical evaluation, one has obtained a threshold for each attack being detected, in such a way that if a given traffic flow have such a characteristic that is above the threshold, it will be classified as an attack traffic flow. For cases of critical attacks, it is suggested using detection mechanisms deployed at different locations working together for detecting the attacks.

Keywords: Denial-of-service. DoS attack. DDoS attack. NBC. Naive bayesian classifier. Real-time. Detection. Network security.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

DoS – Negação de Serviço (*Denial-of-Service*)

DDoS – Negação de Serviço Distribuído (*Distributed Denial-of-Service*)

IRC – *Internet Relay Chat*

C&C – *Command-and-Control*

IP – Protocolo de Internet (*Internet Protocol*)

TCP – *Transmission Control Protocol*

UDP – *User Datagram Protocol*

ICMP – *Internet Control Message Protocol*

DNS – Sistema de Nomes de Domínios (*Domain Name System*)

PPPoE – Protocolo Ponto-a-Ponto sobre Ethernet (*Point-to-Point Protocol over Ethernet*)

ARP – *Address Resolution Protocol*

VoIP – Voz sobre IP (*Voice over IP*)

CUSUM – Soma Acumulativa (*Cumulative Sum*)

ISP – Provedor de Servicos da Internet (*Internet Service Provider*)

CAT – *Change Aggregation Trees*

DCD – *Distributed Change-point Detection*

KDD – *Knowledge Discovery and Data Mining*

ID3 – *Iterative Dichotomiser 3*

CGI – *Common Gateway Interface*

D-WARD – *DDoS Network Attack Recognition and Defense*

SUMÁRIO

1 INTRODUÇÃO

Botnets representam uma ameaça crescente à segurança cibernética e têm sido uma das ameaças mais danosas na Internet, pois mesclam grande parte das ações maliciosas geralmente encontradas em *worms*, *rootkits* e cavalos de Tróia (ZHANG et al., 2011; FEILY; SHAHRESTANI; RAMADASS, 2009). *Botnet* é uma plataforma computacional distribuída, sendo predominantemente usada para realização de atividades ilegais como ataque distribuído de negação de serviço (DDoS, *distributed denial-of-service*), envio de *spams*, envio de e-mails de *phishing*, disseminação de cavalos de Tróia, distribuição ilegal de mídias e *softwares* piratas, roubo de informações e recursos computacionais, extorsão de *e-bussiness*, disseminação de *malwares*, fraude do clique, e roubo de identidade (ZHANG et al., 2011; FEILY; SHAHRESTANI; RAMADASS, 2009).

Ataque de negação de serviço (DoS, *denial-of-service*) é um dos ataques mais comumente realizados por *botnets* (SAHA; GAIROLA, 2005). Ataque DoS pode ser descrito como um ataque designado a tornar um recurso de rede específico incapaz de prestar seu serviço normalmente, impedindo que usuários legítimos façam uso desse recurso. Ataque DDoS é um ataque onde múltiplos sistemas comprometidos são usados para executar um ataque DoS coordenado contra um ou mais alvos, adicionando a dimensão de muitos para um ao ataque DoS. Por meio de uma tecnologia de cliente/servidor, o intruso é capaz de multiplicar a eficácia do ataque DoS significativamente aproveitando do recurso de múltiplos computadores recrutados involuntariamente (DOULIGERIS; MITROKOTSA, 2004).

Como visto no estudo feito por Douligeris e Mitrokotsa (2004), existem três classificações de mecanismos de defesa por local de implantação: mecanismos implantados na rede da vítima, na rede intermediária e na rede de origem. O estudo feito por Mirkovic, Prier e Reiher (2002) aponta que, idealmente, ataques DDoS devem ser parados o mais perto da origem quanto possível. Implantando o mecanismo de defesa na rede de origem, tem-se que o fluxo de ataque poderá ser bloqueado antes de entrar no núcleo da Internet e ser agregado a outros fluxos, o que poderia causar congestionamento. O baixo grau de agregação de fluxos existente na origem permite usar estratégias de defesa mais complexas e com maior precisão (DOULIGERIS; MITROKOTSA, 2004).

Mecanismos de defesa implantados na rede da vítima têm a necessidade de adotar sistemas de rastreamento de pacotes IP, uma vez que pacotes com endereço IP de origem forjado,

IP spoofing, podem ser usados no ataque para ocultar sua verdadeira origem (LAUFER et al., 2005). Essa necessidade de rastreamento de pacotes IP não existe em mecanismos de defesa implantados na origem, pois a detecção está sendo realizada na própria origem do ataque.

Este trabalho propõe um mecanismo de detecção em tempo-real de diferentes tipos de ataque de negação de serviço, e que deve ser implantado na rede de origem do ataque, ou seja, em máquinas sujeitas a serem possíveis agentes de uma *botnet* destinada a efetuar ataques DDoS. Esse mecanismo objetiva detectar fluxos com indícios de ataque antes que se agreguem aos demais fluxos, para que futuramente possam ser de fato bloqueados, evitando causar maiores danos à rede. O mecanismo proposto utiliza o classificador bayesiano simples para detectar tráfegos que possuem indícios de ataque.

A implantação na rede de origem do ataque deve ser feita o mais próximo do agente quanto possível, isto é, nos próprios sistemas de máquinas sujeitas a serem possíveis agentes de uma *botnet*, em *switches* de usuários finais, ou até mesmo em provedores de banda larga para Internet. O objetivo de se implantar próximo aos agentes é ter o mecanismo implantado em um ponto onde o fluxo de pacotes enviado por cada agente não esteja demasiadamente agregado, de maneira que o padrão de comportamento do sistema agente possa ser reconhecido.

O fato do classificador bayesiano simples ser capaz de suportar uma grande quantidade de atributos, sem tornar inviável a fase de treinamento, faz do classificador uma boa escolha quanto à detecção de ataques de negação de serviço (VIJAYASARATHY; RAGHAVAN; RAVINDRAN, 2011). Estudos comparativos mostram que o classificador bayesiano simples apresenta resultados equivalentes aos apresentados por outros algoritmos, como o classificador de redes neurais ou árvores de decisão, mesmo sendo um algoritmo de construção muito simples e possuindo um método de inferência de tempo linear. O classificador bayesiano simples, do ponto de vista computacional, é mais eficiente em ambas as etapas de aprendizagem e classificação (AMOR; BENFERHAT; ELOUEDI, 2004; HAN; KAMBER; PEI, 2006)

Este trabalho está organizado em sete capítulos. O Capítulo 2 apresenta conceitos básicos para um melhor entendimento teórico deste trabalho. Os trabalhos relacionados são discutidos no Capítulo 3. O Capítulo 4 apresenta o mecanismo de detecção proposto por este trabalho. No Capítulo 5, apresenta-se a metodologia utilizada para obtenção da base de treinamento do classificador bayesiano simples, bem como os resultados desse treinamento. O Capítulo 6 apresenta uma análise matemática do mecanismo juntamente com resultados experimentais. Por fim, o Capítulo 7 exibe conclusões e propostas para trabalhos futuros.

2 FUNDAMENTOS TEÓRICOS

Este capítulo apresenta os principais conceitos e fundamentos que servem como uma base para o desenvolvimento deste trabalho. A Seção 2.1 apresenta uma definição de *botnet* e dois dos principais modelos de arquitetura utilizados. Na Seção 2.2 é exibido os principais tipos de ataque de negação de serviço, juntamente com uma breve explicação dos mesmos. A Seção 2.3 apresenta a definição matemática do classificador Bayesiano simples que é utilizado pelo mecanismo de detecção apresentado por este trabalho.

2.1 Botnet

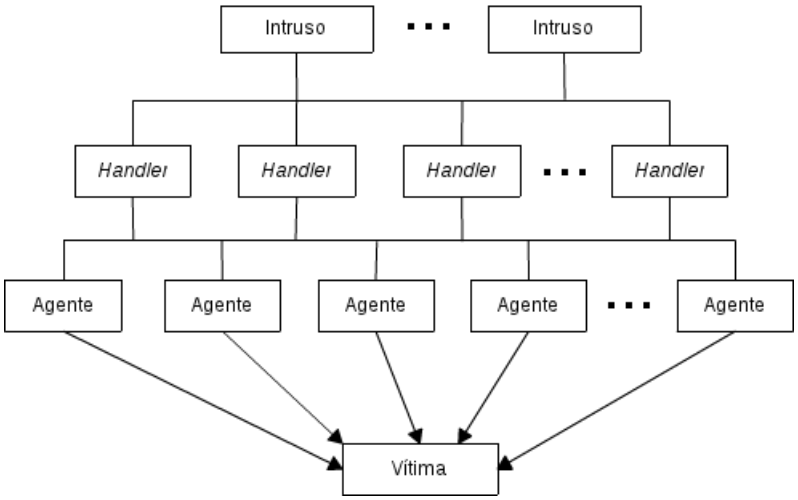
Botnet é uma plataforma computacional distribuída, composta por uma rede de sistemas comprometidos que são controlados remotamente por um intruso. *Botnets* são predominantemente usadas para realizar atividades ilegais como ataque distribuído de negação de serviço (DDoS), envio de *spams*, envio de e-mail de *phishing*, dissemina- ção de cavalos de Tróia, distribuição ilegal de mídias e *softwares* piratas, roubo de informações e recursos computacionais, extorsão de *e-bussiness*, disseminação de *malwares*, fraude do clique, e roubo de identidade (ZHANG et al., 2011; FEILY; SHAHRESTANI; RAMADASS, 2009).

A principal característica que diferencia uma *botnet* de outros *malwares* é a existência de uma rede de comunicação. Existem dois modelos principais de arquitetura utilizados em redes de comunicação para *botnets*, o modelo de *Agent-Handler* e o modelo baseado em *Internet Relay Chat* (IRC).

No modelo de *Agent-Handler*, os *handlers* são pacotes de *softwares* localizados pela Internet que são usados pelos intrusos para se comunicarem com os agentes, são os *handlers* que controlam diretamente os agentes. Os *softwares* dos agentes são implantados em sistemas comprometidos, chamados de "vítimas secundárias", que eventualmente são utilizados para efetuar algum ataque. De acordo com sua configuração, os agentes podem ser instruídos a se comunicarem com um ou mais *handlers*. O intruso se comunica com os *handlers* para identificar quais agentes estão disponíveis, para agendar os ataques ou para atualizar os agentes. Os *handlers* mantêm uma lista de todos os agentes que estão sob suas responsabilidades. Os usuários dos sistemas que hospedam os *softwares* dos agentes tipicamente não sabem que seus sistemas

foram comprometidos e que irão participar de ataques. Geralmente os softwares dos *handlers* são implantados em roteadores ou servidores de rede que lidam com grande volume de tráfego, dificultando a identificação de mensagens entre intrusos e *handlers* e entre *handlers* e agentes por sistemas de detecção de intrusão (SPECHT, 2004).

Figura 1 – Modelo de *Agent-Handler*

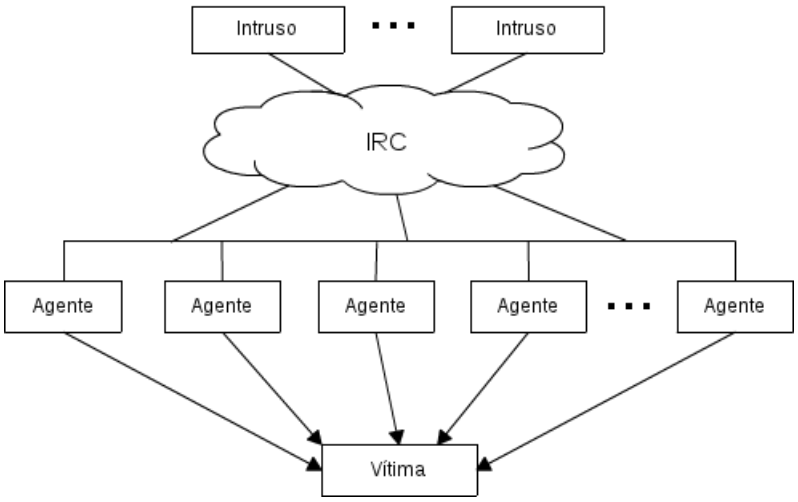


Fonte: Adaptada de (SPECHT, 2004)

O modelo baseado em IRC é semelhante ao modelo de *Agent-Handler* exceto pela utilização de canais de comunicação IRC ao invés de utilizar *handlers* em servidores de rede. Os canais de comunicação IRC são utilizados pelos intrusos para se comunicarem com os agentes. Os agentes se conectam à canais IRC pré-definidos, se colocando à disposição para receber comandos dos controladores da *botnet*, também chamado de *botmaster*. O uso do protocolo de comunicação IRC proporciona alguns benefícios para os intrusos como o uso de portas de IRC legítimas para enviar comandos aos agentes, dificultando o rastreamento de pacotes de comandos de controle (C&C, *command-and-control*) de *botnet*. Comumente, servidores IRC possuem um grande volume de tráfego, esta característica de servidores IRC facilita para que um intruso se esconda de uma possível detecção. Um outro benefício para os intrusos é que esse não necessitaria manter uma lista dos agentes, sendo que bastaria entrar em um servidor IRC e verificar a lista de todos os agentes disponíveis (SPECHT, 2004; SAHA; GAIROLA, 2005).

Em ambos os modelos, refere-se aos agentes como "vítimas secundárias" e às vítimas dos ataques DDoS como "vítimas primárias". É importante perceber que em ambos os modelos existe a presença de vítimas secundárias, realizando um papel essencial durante os ataques DDoS, pois são elas que realizam de modo efetivo os ataques.

Figura 2 – Modelo baseado em IRC



Fonte: Adaptada de (SPECHT, 2004)

2.2 Ataques de Negação de Serviço

O principal objetivo de um ataque de negação de serviço é deixar um recurso computacional inacessível para seus usuários legítimos. Esta seção apresenta uma descrição sobre os principais métodos de ataque, com base nas classificações apresentadas por Specht (2004) e Douligeris e Mitrokotsa (2004). As duas classes principais de métodos de ataque são diminuição de largura de banda e esgotamento de recursos. Ataques de diminuição de largura de banda são caracterizados pelos ataques por inundação e amplificação. Ataques de esgotamento de recursos são ataques que fazem uso indevido dos protocolos de comunicação de rede ou enviam pacotes de rede malformados (SPECHT, 2004; DOULIGERIS; MITROKOTSA, 2004).

2.2.1 Ataques por Inundação

Ataques por inundação se caracterizam por enviarem um grande volume de tráfego ao sistema da vítima primária de modo a congestionar sua banda. O impacto deste ataque pode variar entre deixar o sistema lento, derrubá-lo ou sobrecarregar a banda da rede da vítima. Ataques por inundação podem usar pacotes UDP (*User Datagram Protocol*) ou ICMP (*Internet Control Message Protocol*) (SPECHT, 2004).

As ferramentas de ataque DDoS geralmente utilizam endereços IP de origem forjados para efetuar os ataques, ocultando o endereço das vítimas secundárias, dificultando a identificação dos agentes da *botnet*, já que os pacotes de resposta da vítima primária não são enviados aos agentes. Caso os ataques sejam feitos utilizando endereços IP de origem forjados, para se localizar os agentes será necessário utilizar sistemas de rastreamento de pacotes IP (SPECHT, 2004; LAUFER et al., 2005).

Em ataques por inundação UDP envia-se um grande volume de pacotes UDP para portas fixas ou aleatórias da vítima primária. Tipicamente, inundação UDP é designada à atacar em portas aleatórias da vítima. Para cada pacote UDP recebido pela vítima, ela determinará qual aplicação está ouvindo na porta destino. Caso não haja nenhuma aplicação ouvindo na porta destino, será enviado um pacote ICMP de destino inacessível ao endereço de origem, possivelmente forjado. Ao tentar responder à um grande volume de requisições em diferentes portas, o sistema da vítima ficará eventualmente inacessível para os usuários legítimos (KUMARASAMY; GOWRISHANKAR, 2012; SPECHT, 2004; DOULIGERIS; MITROKOTSA, 2004).

Em ataques por inundação ICMP, também chamados de *Smurf Attack*, envia-se um grande volume de pacotes de ICMP ECHO REPLY para a vítima primária. Estes pacotes requerem uma resposta da vítima, causando uma saturação na banda da conexão de rede da vítima primária. Durante este ataque é comum que o IP de origem dos pacotes ICMP sejam forjados (KUMARASAMY; GOWRISHANKAR, 2012; SPECHT, 2004; DOULIGERIS; MITROKOTSA, 2004).

2.2.2 Ataques por Amplificação

Ataques por amplificação se caracterizam por enviarem requisições forjadas para uma grande quantidade de computadores ou para um endereço IP de *broadcast*, que por sua vez responderão às requisições. Forjando o endereço IP de origem das requisições para o endereço IP da vítima primária fará com que todas as respostas sejam direcionadas para o alvo do ataque. O endereço IP de *broadcast* é um recurso encontrado em roteadores. Quando uma requisição possui um endereço IP de *broadcast* como endereço de destino, o roteador replica o pacote e o envia para todos os endereços IP dentro do intervalo de *broadcast*. Em ataques por amplificação, endereços de *broadcast* são usados para amplificar e refletir o tráfego de ataque, reduzindo então a banda da vítima primária (SPECHT, 2004; DOULIGERIS; MITROKOTSA, 2004).

Um ataque por amplificação pode ser realizado diretamente pelo intruso ou por meio de agentes, vítimas secundárias, de uma *botnet*. Mesmo que o intruso efetue diretamente o ataque enviando mensagens de *broadcast*, o ataque por amplificação por característica proporciona, ao intruso, vítimas secundárias que estarão efetuando o ataque direcionando o fluxo à vítima primária, sem a necessidade de infiltrar nas vítimas secundárias para instalar *softwares* agentes, criando uma estrutura de *botnet* (SPECHT, 2004; DOULIGERIS; MITROKOTSA, 2004).

2.2.3 Ataques por Exploração de Protocolos

Ataques por exploração de protocolos se caracterizam por consumir excessivamente os recursos da vítima primária explorando alguma característica específica ou falha de implementação de algum protocolo instalado no sistema da vítima. Os principais ataques por exploração de protocolos são por uso indevido de pacotes TCP SYN (*Transfer Control Protocol Synchronize*) ou de pacotes TCP PUSH+ACK.

Ataques por inundação TCP SYN é um dos ataques de negação de serviço mais usados. Este ataque faz uso indevido do mecanismo de estabelecimento e finalização de conexões TCP, *three-way handshake*, e suas limitações para manter conexões semi-abertas. Em uma conexão TCP, quando o servidor recebe um pacote TCP SYN, o pacote é interpretado como uma requisição de um cliente para se iniciar uma conexão TCP. Ao receber a requisição, o servidor aloca recursos para guardar informações sobre o estado da conexão TCP, envia um pacote TCP SYN+ACK como resposta para o cliente e aguarda até que a conexão semi-aberta seja completada ou até que o tempo de espera seja expirado (SUN; FAN; LIU, 2007). Em ataques, o servidor recebe um grande volume de pacotes TCP SYN, mas não recebe o pacote TCP ACK finalizando o protocolo de conexão. Geralmente, os pacotes TCP SYN são enviados para a vítima primária com o endereço IP de origem forjado, de maneira que o sistema da vítima responderá com um pacote TCP SYN+ACK à um sistema que não tenha requisitado uma conexão TCP. Com um grande volume de conexões semi-abertas, o sistema da vítima primária terá seus recursos sobrecarregados, fazendo com que as novas requisições de conexão TCP sejam perdidas (KUMARASAMY; GOWRISHANKAR, 2012; SPECHT, 2004).

Ataques PUSH+ACK são caracterizados por enviar um grande volume de pacotes TCP PUSH+ACK. Esses pacotes fazem com que o sistema da vítima descarregue todos os dados do

buffer de TCP, independentemente do *buffer* estar cheio ou não, e enviar um TCP ACK quando completar a descarga do mesmo. Quando esse ataque é realizado em grande escala, de forma distribuída, o sistema da vítima primária não será capaz de processar um grande volume de requisições TCP PUSH+ACK, sobrecarregando-o (SPECHT, 2004).

2.3 Classificador Bayesiano Simples

Classificadores bayesianos são classificadores estatísticos, baseados no Teorema de Bayes, que classificam um objeto em uma determinada classe baseando-se na probabilidade desse objeto pertencer a essa classe. O classificador bayesiano simples (*Naive Bayesian Classifier*) assume uma independência condicional entre os atributos de uma dada classe, isto é, o valor de um atributo não influencia o valor dos demais atributos (RISH, 2001). Segundo Zhang e Su (2004), o classificador bayesiano simples está entre os algorítimos de classificação mais eficazes e eficientes. Estudos comparando algoritmos de classificação mostram que o classificador bayesiano simples é equivalente em eficiência com árvores de decisão e com o classificador de redes neurais, além de apresentar alta precisão e velocidade, mesmo quando usado com grandes bancos de dados (HAN; KAMBER; PEI, 2006).

O classificador bayesiano simples pode ser aplicado em uma grande variedade de problemas de classificação como em diagnósticos médicos, caracterização de textos e filtros colaborativos de e-mail. Quando comparado à outros métodos mais sofisticados, o classificador bayesiano comumente se mostra uma melhor solução (BOUCKAERT, 2005).

2.3.1 Descrição Matemática do Classificador Bayesiano Simples

Uma descrição matemática do classificador bayesiano simples, conforme apresentado por Rish (2001) e Han, Kamber e Pei (2006), é dada a seguir:

Seja D um conjunto de amostras para treinamento, onde cada amostra está associada com uma determinada classe. Cada amostra é representada por um vetor n-dimensional, $X = (x_1, x_2, ..., x_n)$, que descreve os valores de n atributos, $A_1, A_2, ..., A_n$, respectivamente.

Suponha que existam m classes distintas, $C_1, C_2, ..., C_m$. Dado uma amostra, X, o clas-

sificador irá predizer que X pertence a uma classe que tenha a maior *probabilidade posterior*. Isto é, X pertence à classe C_i se e somente se

$$P(C_i \mid X) > P(C_j \mid X) \quad \text{para } 1 \leq j \leq m, j \neq i.$$

Portanto, maximiza-se $P(C_i \mid X)$. A classe C_i é chamada de *hipótese posterior máxima*.

A probabilidade posterior pode ser calculada usando o Teorema de Bayes. Sabe-se que

$$P(X \cap C_i) = P(X \mid C_i) \cdot P(C_i) = P(C_i \mid X) \cdot P(X)$$

onde

$$P(C_i \mid X) = \frac{P(X \mid C_i) \cdot P(C_i)}{P(X)} \quad \text{(Teorema de Bayes)}$$

Como $P(X)$ é uma constante para todas as classes, para maximizar a probabilidade posterior é preciso maximizar apenas o numerador $P(X \mid C_i) \cdot P(C_i)$. Caso não se conheça a priori a probabilidade de cada classe, assume-se comumente que as classes possuem probabilidades iguais, isto é, $P(C_1) = P(C_2) = ... = P(C_m) = \frac{1}{m}$, bastando, portanto, maximizar $P(X \mid C_i)$. Caso as probabilidades sejam conhecidas, maximiza-se o produto $P(X \mid C_i) \cdot P(C_i)$. É possível, também, estimar a probabilidade posterior de uma classe fazendo $P(C_i) = |C_{i,D}|/|D|$, onde $|C_{i,D}|$ é o número de amostras da classe C_i no conjunto de treinamento D.

Para reduzir o custo computacional de se calcular $P(X \mid C_i)$, é feito a consideração de independência condicional entre classes, isto é, o valor de um atributo não influencia o valor dos outros. Portanto,

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i) = P(x_1 \mid C_i) \cdot P(x_2 \mid C_i) \cdot ... \cdot P(x_n \mid C_i)$$

As probabilidades de $P(X \mid C_i)$ podem ser calculadas a partir da base de amostras de treinamento da seguinte maneira:

- Se A_k é um atributo categórico, então $P(x_k \mid C_i)$ é o número de amostras da classe C_i em D tendo o valor de x_k para A_k, dividido pelo número de amostras da classe C_i em D.

- Se A_k é um atributo de valor contínuo, então o cálculo é feito com base na distribuição Gaussiana. Considera-se que um atributo de valor contínuo possui uma distribuição Gaussiana, sendo μ a média e σ o desvio padrão, definido por

  $$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

  de modo que

  $$P(x_k \mid C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})$$

Para classificar X, $P(X \mid C_i)P(C_i)$ é calculado para todas as classes C_i. A amostra X será classificada na classe C_i se e somente se

$$P(X \mid C_i)P(C_i) > P(X \mid C_j)P(C_j) \quad \text{para } 1 \leq j \leq m, j \neq i$$

ou seja, a amostra X será classificada na classe C_i cuja probabilidade posterior $P(X \mid C_i)P(C_i)$ seja máxima.

3 TRABALHOS RELACIONADOS

Este capítulo apresenta os principais trabalhos relacionados, organizados em seções. A Seção 3.1 apresenta trabalhos relacionados à detecção de *botnets*, incluíndo métodos de detecção baseados em características comportamentais de *botnets* ou em comunicação e comandos de controle. A Seção 3.2 apresenta diversos trabalhos que propõem mecanismos de detecção de ataques DDoS. Na Seção 3.3 são apresentados alguns trabalhos sobre sistemas de detecção de intrusos baseados principalmente em algoritmos de aprendizagem de máquina. Na Seção 3.4 são apresentados alguns trabalhos relacionados à mecanismos de mitigação de ataques DDoS.

3.1 Detecção de Botnets

Mecanismos de detecção de *botnet* basicamente se resumem em monitorar e analisar de forma passiva o tráfego de uma dada rede de computadores. Grande parte dos trabalhos relacionados à detecção de *botnets* apresentam mecanismos que são implantados próximo aos agentes ou mestres de *botnets*. Alguns destes trabalhos são descritos a seguir.

Cooke, Jahanian e McPherson (2005) estudam a eficácia da detecção de *botnets* monitorando diretamente a comunicação IRC ou outras atividades de comandos de controle. Um dos métodos de detecção apontado por esse estudo é observar portas IRC conhecidas e procurar por padrões de mensagens que casam com comandos previamente conhecidos. Um segundo método apresentado é baseado na observação de características comportamentais de *bots*. Uma última abordagem para detecção de *botnets* descrita pelos autores é utilizar *honeypots* para capturar *bots* e então observar suas características de comandos de controle que serão utilizadas posteriormente para detecção de *botnets*. Esse estudo aponta que maiores esforços serão necessários em outros métodos para parar *botnets*. Um trabalho feito em resposta à essa necessidade é o de Binkley e Singh (2006), que apresenta maior robustez em seu método de detecção de *botnet*.

O trabalho de Binkley e Singh (2006) apresenta um algoritmo baseado em anomalias para detecção de *botnets* IRC. Esse algoritmo combina detecção de anomalia baseada em TCP com estatísticas de simbologias e mensagens IRC. A detecção de anomalia baseada em TCP é feita por um monitor que observa o fluxo dos pacotes TCP e utiliza uma métrica chamada de

TCP *work weight*. Porém, esta abordagem de defesa poderia ser facilmente combatida criptografando os comandos IRC.

Goebel e Holz (2007) apresentam um método para detecção de máquinas infectadas por *bots* baseado na avaliação de *nicknames* IRC. Semelhante ao método de análise de mensagem IRC apresentado por Binkley e Singh (2006), o método apresentado por Goebel e Holz (2007) monitora o fluxo de pacotes IRC procurando e avaliando por características específicas nos *nicknames* usados por usuários de canais de comunicação IRC. A avaliação dos *nicknames* é baseada em pontuações de características comuns a *nicknames* usados por *bots*. Uma diferença entre o trabalho de Binkley e Singh (2006) e o trabalho de Goebel e Holz (2007) é que o segundo funciona em tempo real e geralmente é capaz de detectar máquinas infectadas mais cedo que outros sistemas de detecção de intrusos.

Karasaridis, Rexroad e Hoeflin (2007), diferentemente de Goebel e Holz (2007), apresentam uma metodologia de detecção e caracterização de *botnets* a partir de uma análise passiva baseada em anomalias. Como esta abordagem é baseada em uma análise feita principalmente na camada de transporte, algumas das vantagens apresentadas são que ela é capaz de detectar *botnets* que utilizam uma comunicação criptografada e não violam questões de privacidade. Este trabalho obteve uma taxa de falso-positivo de menos de 2% na detecção de *botnets*.

Strayer et al. (2006) oferecem uma abordagem para detecção de *botnets* baseadas em IRC que examina características de fluxo como largura de banda, duração e temporização de pacotes, procurando por evidências de atividades de comandos de controle (C&C, *command-and-control*) de *botnets*. Nessa abordagem, os pacotes são monitorados e então analisados. Pacotes que são improváveis de serem de comunicação de *botnets* são eliminados por um pré-filtro.

Fedynyshyn, Chuah e Tan (2011) apresentam um método baseado em *host* para detecção e classificação de diferentes tipos de infecção de *botnet* baseado em seus estilos de C&C. A abordagem apresentada por esse artigo se baseia em treinar um classificador utilizando uma base de dados coletada pelos próprios autores. A base de dados é composta pelos dados de tráfego legítimos de rede e pelos dados de tráfego de *botnets*. Esse estudo aponta que existe uma similaridade inerente entre as estruturas de C&C de diferentes tipos de *bots* e que as características de rede do tráfego de C&C de *botnet* são inerentemente diferentes de tráfegos legítimos de rede.

3.2 Detecção de Ataques de Negação de Serviço

A detecção de ataques de negação de serviço pode ser feita com o propósito de precaver-se que um computador *host* ou uma rede não seja a origem bem como a vítima de um ataque. Mecanismos de detecção de ataques DoS são baseados em bancos de dados de assinaturas conhecidas ou baseado no reconhecimento de anomalias no comportamento dos sistemas.

Wang, Zhang e Shin (2002) propõem um mecanismo simples e robusto para detectar ataques por inundação SYN. Esse mecanismo de detecção é baseado no comportamento de protocolo de conexão TCP. Wang, Zhang e Shin (2002) utilizam o método de soma acumulativa não paramétrica (CUSUM, *cumulative sum*) para não deixar o mecanismo de detecção sensível a padrões de acessos e sites, tornando a aplicação do mecanismo de detecção mais geral e sua implantação consideravelmente mais fácil. Esse estudo aponta que esse mecanismo de detecção tem baixa latência e alta precisão de detecção.

O trabalho de Sun, Fan e Liu (2007) é uma melhoria do mecanismo de detecção apresentado por Wang, Zhang e Shin (2002). Sun, Fan e Liu (2007) armazenam as informações de fluxo de pacotes SYN em um *Bloom Filter*, e contam a quantidade de pacotes FIN e RST de acordo com o *Bloom Filter*, deixando o mecanismo de defesa ainda mais robusto em relação à detecção de ataques por inundação TCP SYN mais complexos.

Mitrokotsa e Douligeris (2005) propõem uma abordagem que detecta ataques DDoS usando *Emergent Self-Organizing Maps*. *Self-Organizing Maps* é um método de redes de aprendizado competitivo ou não-supervisionado que tem sua base na biologia e que produz um mapa topológico 2D ilustrando os dados de entrada de acordo com suas similaridades. A abordagem apresentada é baseada na classificação de tráfegos “normais” contra tráfegos “anormais” no sentido de ataques DDoS. O método proposto permite uma classificação automática de eventos que estão contídos em *logs* e visualização do tráfego da rede. Simulações mostraram a eficácia dessa abordagem quando comparada com as abordagens propostas anteriormente em relação à falsos alarmes e probabilidades de detecção. Esse estudo aponta que a abordagem apresentada é extremamente poderosa ao produzir resultados eficientes.

Sen (2011) propõe um mecanismo para proteger um servidor web contra ataques DDoS. Em seu mecanismo, o tráfego de entrada é continuamente monitorado e qualquer anomalia que surgir no tráfego é imediatamente detectada. O algoritmo de detecção proposto por esse trabalho é baseado em análise estatística do tráfego de entrada e um *framework* de testes robusto

de hipóteses. Durante um ataque, sessões de fontes legítimas não são interrompidas e a carga do servidor é restabelecida ao nível normal bloquendo o tráfego vindo das fontes de ataque. O algoritmo de detecção é composto por vários módulos com diferentes níveis de custos computacionais e de uso de memória. Módulos mais precisos envolvem lógicas complexas de detecção, consequentemente envolvendo maior custo computacional e uso de memória. E módulos aproximados são rápidos em detectar um ataque, possuem baixo custo e baixa precisão

Wu et al. (2011) apresentam um sistema de detecção de ataque DDoS baseado em árvore de decisão. Ao detectar um ataque DDoS, o sistema rastreia a localização do intruso por uma técnica de casamento de padrões do fluxo de tráfego. Esse trabalho foca na detecção de ataques baseados em inundação. O sistema é composto por monitores implantados na vítima, chamados de *protection agents*, e nos roteadores intermediários, chamados de *sentinels*. Esse trabalho propõe uma detecção baseada na classificação do tráfego de entrada e saída entre situação sem ataque e situação com ataque. A classificação é feita aplicando a técnica de árvore de decisão utilizando 15 atributos como base para detecção de tráfego anormal. O sistema apresentado por esse artigo é capaz de detectar ataques DDoS com uma taxa de falso-positivo de 1,2% à 2,4% e uma taxa de falso-negativo de 2% à 10%, e rastreia o caminho de ataque com uma taxa de falso-negativo de 8% à 12% e uma taxa de falso-positivo de 12% à 14%.

O trabalho de Chen, Hwang e Ku (2007) apresenta um abordagem distribuída para detecção de ataques DDoS por inundação, baseado na arquitetura de *distributed change-point detection* (DCD) usando *change aggregation trees* (CAT). Esse sistema de defesa é implantado sobre o núcleo da rede e operado pelos provedores de serviços da Internet (ISP, *Internet service provider*). Esse estudo aponta que ao início de um ataque DDoS algumas variações do tráfego são detectáveis nos roteadores de Internet ou em *gateways*. Cada domínio ISP tem um servidor CAT para agregar os alertas por inundação reportados pelos roteadores, onde os servidores de domínios CAT colaboram entre si para obter uma decisão final. Esse trabalho apresenta resultados experimentais mostrando que quatro domínios de rede são suficientes para se obter uma precisão de 98% na detecção com uma taxa de 1% de falso-positivo. Esse estudo aponta que essa cobertura de segurança é abrangente o suficiente para garantir a segurança de grande parte das redes de ISP em ataques reais de DDoS por inundação.

Tariq et al. (2011) apresentam um mecanismo de defesa colaborativo baseado em *peer to peer* para ataques DDoS. O mecanismo proposto detecta ataques nos roteadores próximos à vítima, enviando uma mensagem de alerta para seus nodos vizinhos, permitindo que eles se

defendam proativamente. Esse estudo aponta que ao invés de desenvolver sistemas de defesa centralizados, é importante desenvolver sistemas distribuídos de detecção e defesa, onde nodos heterogêneos podem colaborar pela rede monitorando o tráfego de uma maneira cooperativa. A cooperação entre cada nodo de detecção é feita com um mecanismo de comunicação confiável. Esse trabalho aponta que os resultados das simulações mostraram a eficiência da solução com uma menor taxa de falso-positivo e menor dano à rede devido à abordagem proativa da defesa.

O trabalho de Vijayasarathy, Raghavan e Ravindran (2011) propõe uma abordagem prática baseada em anomalias para detecção de ataques DDoS, utilizando um classificador bayesiano simples. O sistema proposto por esse trabalho tem propósito de ser um sistema de tempo-real e que deve ser implantado próximo ao alvo primário. Apesar desse sistema ter como foco os protocolos TCP e UDP, soluções para outros protocolos podem ser facilmente integradas.

O trabalho de Kumarasamy e Gowrishankar (2012) apresenta um mecanismo de defesa ativo dedicado a detectar ataques por inundação de TCP SYN. Este mecanismo é baseado na colaboração entre os nós de defesa ao longo da rede intermediária, mas com o objetivo de realizar a detecção na vítima primária. A detecção dos ataques se baseia na combinação de quatro abordagens. O primeiro caso é monitorando os campos de *flags* do pacotes TCP, procurando por anomalias. O segundo caso se baseia na detecção de anomalias em portas de pacotes TCP, como portas nulas. No terceiro caso, o mecanismo busca detectar respostas ICMP devido ao uso de endereços IP forjados. O último caso é baseado em realizar a detecção rastreando a rota de uma mensagem ICMP. Esse trabalho aponta que o mecanismo apresentado é robusto e apresenta uma detecção com alta precisão e eficiência.

3.3 Sistemas de Detecção de Intrusos

Kruegel et al. (2003) apresentam um método de classificação de eventos que se baseia em redes bayesianas. Esse estudo aponta que redes bayesianas aumentam a agregação de modelos diferentes de saída e permitem que se incorpore informações adicionais sem maiores problemas. Tais características são apontadas, pelos autores Kruegel et al. (2003), como sendo as principais características que levam à uma alta quantidade de falso-positivo em sistemas de detecção de intrusos. Resultados experimentais apresentados por esse trabalho mostram que a precisão do processo de classificação de eventos foi melhorada significativamente com o uso da

abordagem proposta, apresentando uma redução significativa de falso-positivos.

O trabalho de Amor, Benferhat e Elouedi (2004) apresenta um estudo comparativo entre o classificador bayesiano simples e árvores de decisão. O estudo experimental é feito com a conjunto de dados de intrusão do *KDD Cup* de 1999 (KDD'99). Esse estudo aponta que mesmo o classificador bayesiano simples sendo um algoritmo de construção muito simples e possuir um método de inferência de tempo linear, o classificador bayesiano apresenta resultados próximos ou melhores que outros algoritmos de aprendizagem de máquina, como árvores de decisão. Esse estudo mostra que do ponto de vista computacional, o classificador bayesiano simples é mais eficiente em ambas as etapas de aprendizagem e classificação.

Farid, Harbi e Rahman (2010) apresentam um algoritmo híbrido de aprendizagem de máquina para detecção adaptativa de intrusões de rede usando o classificador bayesiano simples e árvore de decisão. O algoritmo híbrido utiliza o algoritmo *Iterative Dichotomiser 3* (ID3) que é um algoritmo para geração de árvode de decisão. Esse trabalho utiliza o conjunto de dados do *KDD Cup* de 1999 (KDD'99) como base para teste. Os ataques do conjunto de dados KDD'99 foram detectados com uma precisão de 99% usando o algoritmo híbrido proposto. Esse estudo aponta que o algoritmo híbrido minimizou a taxa de falso-positivos.

3.4 Mitigação de Ataques de Negação de Serviço

Mecanismos para mitigação de ataques de negação de serviço são comumente implantados próximos à vítima com o objetivo de permitir que a vítima seja capaz de responder às requisições dos usuários legítimos de forma adequada, mesmo estando sob ataque. A seguir são apresentados alguns trabalhos que propõem mecanismos para mitigação destes ataques.

Lau et al. (2000) examinaram como diferentes algoritmos de enfileiramento implementados em um roteador de rede executam durante um ataque, e se usuários legítimos foram capazes de obter a largura de banda desejada. O estudo foi feito simulando um ataque DDoS usando o simulador de rede ns-2. Lau et al. (2000) mediram a taxa de transferência fornecida aos usuários legítimos e aos intrusos quando usando os seguintes algoritmos de enfileiramento: *DropTail*, *Fair Queuing*, *Stochastic Fair Queuing*, *Deficit Round Robin*, *Random Early Detection*, e o *Class Based Queuing*. Esse estudo aponta que o algoritmo *Class Based Queuing* obteve o melhor desempenho e proporcionou toda largura de banda requisitada por usuários legítimos,

enquanto o algoritmo *Random Early Detection* foi o melhor dentre aqueles que não necessitam de um *overhead* adicional.

O trabalho de Kargl, Maier e Weber (2001) tem como objetivo principal apresentar um ambiente de proteção que seja capaz de manter um servidor web sob ataque respondendo apropriadamente às requisições feitas por usuários legítimos. O ambiente de proteção é composto por vários servidores web que são acessados por um balanceador de cargas. O algoritmo de enfileiramento utilizado por esse ambiente de proteção é o *Class Based Queuing*, o mesmo algoritmo que foi apontado como sendo o melhor algoritmo de enfileiramento por Lau et al. (2000). No teste utilizando um banco de dados de HTML estático, o intruso consumiu apenas uma quantidade limitada da largura de banda de entrada e saída, enquanto os usuários legítimos puderam acessar o servidor web de uma maneira praticamente normal. No teste utilizando documentos HTML gerados por *scripts* de CGI, o intruso consumiu a maior parte da largura de banda do servidor web.

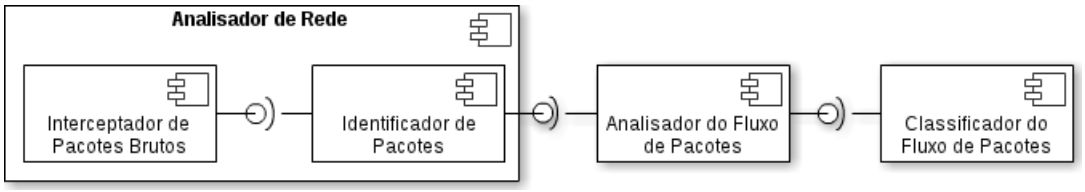
Mirkovic, Prier e Reiher (2002) propõem um sistema de defesa, chamado D-WARD, contra ataques DDoS que é implantado na rede de origem, com o objetivo de evitar que máquinas participem dos ataques DDoS. Esse sistema, de maneira autônoma, detecta e bloqueia ataques originados onde o sistema foi implantado. A detecção é feita por um monitoramento constante do fluxo de tráfego em ambos os sentidos, fazendo comparações periódicas com um modelo de fluxo normal. Esse estudo aponta que D-WARD oferece bom serviço aos tráfegos legítimos mesmo durante uma situação de ataque, reduzindo o tráfego de ataque a um nível insignificante.

Este trabalho se posiciona dentre os trabalhos relacionados como um mecanismo de detecção de ataques de negação de serviço. Como sugerido por Mirkovic, Prier e Reiher (2002) e como apresentado pelo estudo de Douligeris e Mitrokotsa (2004), a proposta deste mecanismo é para implantação na rede de origem do ataque de negação de serviço. O objetivo deste mecanismo é detectar os ataques em tempo-real e para isto é necessário que o fluxo gerado localmente seja analisado e classsificado eficientemente, de maneira que a sobrecarga de processamento necessária para realizar a classificação de um fluxo não degrade o desempenho do sistema no qual se implantou o mecanismo. Para isso, este trabalho propõe o uso do classificador bayesiano simples por possuir um método de inferência de tempo linear (RISH, 2001).

4 DESCRIÇÃO DO MECANISMO

Este capítulo descreve em detalhes o mecanismo de detecção de ataques de negação de serviço apresentado por este trabalho, bem como uma descrição da implementação de um protótipo experimental desenvolvido. O diagrama de componentes apresentado na Figura 3 ilustra a arquitetura desse mecanismo. Cada componente descrito na arquitetura está detalhado nas seções seguintes.

Figura 3 – Diagrama de Componentes do Mecanismo de Detecção de Ataques DoS



Fonte: Elaborada pelo autor

4.1 Interceptador de Pacotes Brutos

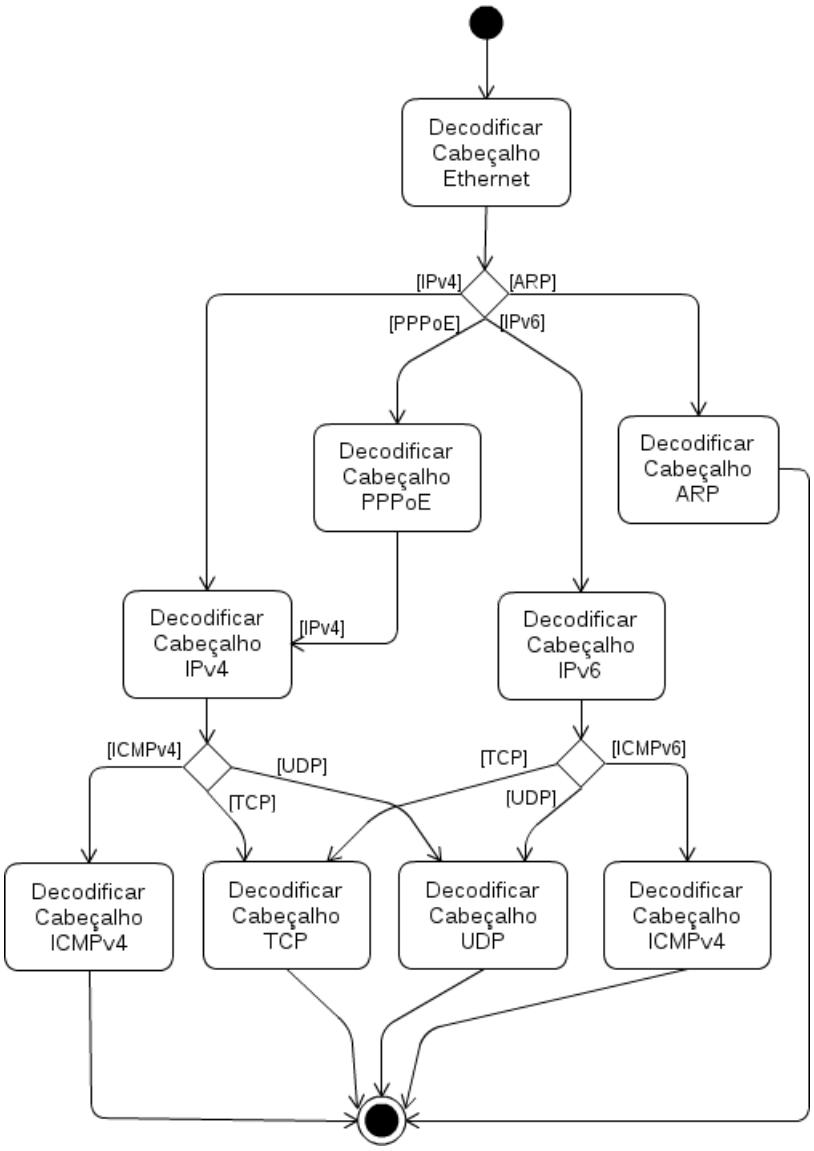
Esse componente é o módulo principal de um analisador de rede, também conhecido como *Ethernet Sniffer*, responsável por monitorar a interface de rede e interceptar o tráfego de pacotes, capturando cada pacote no formato bruto, isto é, como um conjunto de *bytes* sem a devida identificação dos cabeçalhos e protocolos. Para a implementação desse módulo foi utilizado a biblioteca de captura de pacotes, sendo que para ambientes *Unix* foi utilizado a biblioteca libpcap e para ambientes *Windows* a biblioteca WinPcap (JACOBSON; LERES; MCCANNE, 2004).

Essa biblioteca de captura de pacotes é amplamente utilizada por ferramentas de rede, como monitores de rede e sistemas de detecção de intrusos, proporcionando certa portabilidade, considerando que as implementações dessa biblioteca para os ambientes *Unix* e *Windows* são equivalentes.

4.2 Identificador de Pacotes

Como os pacotes são capturados no formato de pacotes brutos, existe a necessidade desses pacotes passarem por um processo de identificação, onde serão identificados quais protocolos de rede compõem os pacotes. O diagrama de atividades apresentado pela Figura 4 ilustra o algoritmo do componente identificador de pacotes implementado para o protótipo experimental, apresentando cada atividade de decodificação do pacote bruto.

Figura 4 – Diagrama de Atividades do Módulo de Identificação de Pacotes



Fonte: Elaborada pelo autor

Cada atividade do diagrama decodifica, a partir do pacote bruto, o cabeçalho do protocolo, identificando cada atributo e determinando qual protocolo está sendo utilizado na camada seguinte da pilha de protocolos. Os protocolos são identificados de acordo com as padronizações especificadas pelos *Requests for Comments* (RFC), publicadas pela *Internet Engineering Task Force* (IETF), e as padronizações do *Institute of Electrical and Electronics Engineers* (IEEE).

4.3 Analisador do Fluxo de Pacotes

Após identificados, os pacotes passarão pelo módulo de análise. Esse módulo é responsável por extrair e agrupar os atributos a partir do fluxo de pacotes. O agrupamento dos dados é realizado utilizando o conceito de janelas e separando os fluxos de pacote por endereço IP de destino. Os atributos são extraídos a partir dos pacotes identificados e são específicos para cada ataque.

4.3.1 Agrupamento dos dados

O agrupamento dos dados é realizado utilizando o conceito de janelas, que são responsáveis por particionar o fluxo de pacotes em subconjuntos. O uso de janelas permite ao mecanismo analisar o fluxo de pacotes com certo controle de tempo, o que não seria possível com a análise individual dos pacotes (VIJAYASARATHY; RAGHAVAN; RAVINDRAN, 2011). O conceito de janelas pode ser usado de duas maneiras:

Janela Temporal se baseia em particionar o fluxo de pacotes em subconjuntos de pacotes agrupados por um determinado período de tempo.

Janela por Contagem de Pacotes se baseia no agrupamento dos pacotes por contagem da ocorrência de uma determinada quantidade de pacotes.

Em uma situação de ataque, tem-se que a vítima secundária tende a concentrar todo o fluxo de ataque à uma única vítima primária, com o objetivo de maximizar a eficácia do ataque. Por este motivo, o fluxo de pacotes é separado por endereço IP de destino. Para isso,

é utilizado uma tabela *Hash*, tendo o endereço IP como chave e os dados agrupados como elementos da tabela *Hash*. Este trabalho adota o conceito de janelas temporais, agrupando os dados dos atributos por um intervalo de tempo, t_w. Entretanto, caso uma vítima secundária esteja realizando ataques à múltiplas vítimas primárias, estes serão detectados como ataques individuais.

4.3.2 Atributos para detecção de ataques por inundação TCP SYN

A extração de atributos de pacotes TCP para detecção de ataques por inundação TCP SYN é feita com base na característica do mecanismo de estabelecimento e finalização de conexões TCP, considerando que em uma conexão válida haverá um par de pacotes TCP SYN e TCP FIN, em alguns casos podendo terminar com um pacote TCP RST. Ataques por inundação TCP SYN exploram este mecanismo com conexões semi-abertas, de maneira que não haverá um equilíbrio entre a quantidade de pacotes TCP SYN e pacotes TCP FIN ou TCP RST (KUMARASAMY; GOWRISHANKAR, 2012; SUN; FAN; LIU, 2007).

O atributo utilizado para classificação do fluxo de pacotes é extraído com base na janela temporal onde é armazenado a quantidade de pacotes TCP SYN, P_{SYN}, a quantidade de pacotes TCP FIN, P_{FIN}, e a quantidade de pacotes TCP RST, P_{RST}, capturados a cada intervalo de tempo, t_w. O atributo é calculado como $X = (P_{SYN} - (P_{FIN} + P_{RST}))$, onde, em situações de conexões legítimas, a diferença $P_{SYN} - (P_{FIN} + P_{RST})$ será próxima de zero, enquanto em situações de ataques tem-se $P_{SYN} - (P_{FIN} + P_{RST}) > 0$, geralmente sendo $P_{SYN} - (P_{FIN} + P_{RST}) = P_{SYN}$ devido ao uso de endereços de origem forjados.

4.3.3 Atributos para detecção de ataques por inundação UDP

Comunicação por UDP não são orientadas por conexão, garantindo maior velocidade de comunicação. Cabeçalhos de pacotes UDP não possuem campos para determinar os estados da comunicação, como o campo de *flags* existentes em cabeçalhos de pacotes TCP. Com base nestas características de comunicação por UDP, ataques por inundação por UDP são ataques baseados somente na degradação da banda de rede, cujo objetivo principal é exaurir os recur-

sos de banda disponíveis da rede da vítima primária (KUMARASAMY; GOWRISHANKAR, 2012; VIJAYASARATHY; RAGHAVAN; RAVINDRAN, 2011).

O atributo utilizado pelo classificador é a quantidade de pacotes UDP, P_{UDP}, capturados a cada intervalo de tempo, t_w. Esse atributo, $X = (P_{UDP})$, é agrupado individualmente para cada endereço IP destino, durante o período de cada janela temporal.

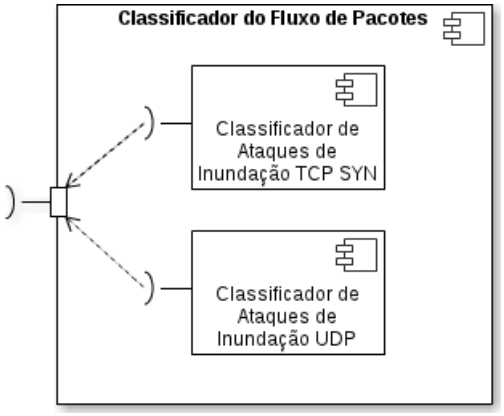
4.4 Classificador do Fluxo de Pacotes

Esse componente é responsável por classificar o fluxo de pacotes entre tráfego normal e tráfego suspeito de ataque. Para esse fim, é utilizado um classificador bayesiano simples com atributos de valor contínuo, como descrito na Seção 2.3. Para cada ataque DoS que se deseja classificar, deve-se treinar um classificador especificamente para esse ataque.

A fase de treinamento é responsável por calcular a média e o desvio padrão para cada classe e atributo, a partir de um conjunto de amostras com os atributos adequados que represente um tráfego normal e um outro conjunto de amostras que represente um tráfego desse novo ataque. Essa fase é realizada de modo *off-line* e os conjuntos de amostras são obtidos de maneira supervisionada.

Após treinado, no intervalo t_w de cada janela temporal, o classificador executa os cálculos de classificação para os atributos extraídos pelo analisador do fluxo de pacotes. Tais cálculos são executados por cada classificadores bayesianos simples, como ilustrado pelo diagrama de componentes apresentado na Figura 5. Cada componente emitirá um aviso de ataque detectado ao classificar um determinado fluxo de pacotes como ataque.

Figura 5 – Diagrama de Componentes do Classificador do Fluxo de Pacotes



Fonte: Elaborada pelo autor

4.5 Extensão do Mecanismo

Uma das vantagens desse mecanismo é a facilidade em estendê-lo. A modularização apresentada por esse mecanismo facilita sua extensão, de maneira que para acrescentar a detecção de um novo tipo de ataque DoS seja necessário apenas acrescentar sua extração e agrupamento de atributos no analisador de fluxo de pacotes, e acrescentar um classificador bayesiano simples treinado especificamente para esse novo ataque.

A extração dos atributos deve ser feita a partir da identificação dos pacotes, resultando em um vetor n-dimensional, $X = (x_1, x_2, ..., x_n)$, que descreva os atributos necessários para classificar o fluxo de pacotes como normal ou suspeito de ataque. Para os ataques por inundação TCP SYN e UDP os atributos extraídos são $X = (P_{SYN} - (P_{FIN} + P_{RST}))$ e $X = (P_{UDP})$, respectivamente.

Para cada novo ataque que se deseja detectar, deve-se acrescentar um classificador bayesiano simples específico para seus atributos extraídos pelo analisador do fluxo de pacotes. Esse classificador deve ser treinado exclusivamente para detectar esse novo ataque, recebendo um conjunto de amostras com os atributos adequados que represente um tráfego normal e um outro conjunto de amostras que represente um tráfego desse novo ataque. Esse classificador treinado será inserido ao componente do classificador do fluxo de pacotes.

5 METODOLOGIA DE TREINAMENTO

Este capítulo apresenta a metodologia utilizada para realizar o treinamento dos classificadores para o protótipo experimental. Esse treinamento é realizado utilizando um conjunto de amostras que represente cada classe, ou seja, o classificador deve ser treinado utilizando um conjunto de amostras de um fluxo de pacotes que represente um tráfego normal e um outro conjunto de amostras que represente um tráfego de ataque. Deve-se treinar um classificador bayesiano simples especificamente para cada ataque que se deseja detectar, considerando inclusive que os atributos geralmente serão diferentes entre os ataques.

O treinamento de um classificador bayesiano simples consiste em calcular, a partir dos conjuntos de amostras, a média, μ, e o desvio padrão, σ, para cada classe e atributo. Os conjuntos de amostras são obtidos de maneira supervisionada e controlada, ou seja, ao treinar um classificador, além de saber que um conjunto de amostras de fluxo representa um tráfego normal e o outro conjunto de amostras representa um tráfego de ataque, deve-se minimizar ao máximo a inserção de ruídos em tais conjuntos (HAN; KAMBER; PEI, 2006). Entre os ruídos a serem evitados estão amostras contraditórias, amostras classificadas erroneamente e desequilíbrio entre as amostras (FARID; HARBI; RAHMAN, 2010).

Como a proposta do classificador é detectar ataques executados por agentes de uma *botnet*, o conjunto de amostras para treinamento deve representar ao máximo o comportamento de uma vítima secundária. O objetivo da base de treinamento é representar um usuário comum de rede, que seria a vítima secundária, cujo uso da rede seja legítimo, e representar um agente instalado realizando ataques DoS.

A base de treinamento deve conter os atributos necessários para cada classificador, isto é, o classificador de ataques por inundação TCP SYN deve ser treinado com um conjunto de amostras do atributo $X = (P_{SYN} - (P_{FIN} + P_{RST}))$, enquanto o classificador de ataques por inundação UDP deve ser treinado com um conjunto de amostras do atributo $X = (P_{UDP})$.

A coleta do conjunto de amostras de tráfego normal, para treinar o protótipo experimental do mecanismo, foi realizada buscando abranger uma grande variedade de uso de rede, visando representar um usuário legítimo comum. Para isso, alguns dos padrões de uso de rede contidos na base de treinamento são:

- Uso do protocolo TCP
  - Acesso a sites de notícias por navegadores *web*;
  - Acesso a *webmails*;
  - Pesquisas em máquinas de busca na *web*;
  - Uso de redes sociais por navegadores *web*;
  - Transmissão de arquivos por BitTorrent;
  - Recepção de arquivos por BitTorrent;
  - Uso de serviços de armazenamento de arquivos, com base no conceito de computação em nuvem;
  - Acesso a sites de compartilhamento de videos.
- Uso do protocolo UDP
  - Envio de requisições DNS;
  - Recepção de respostas DNS;
  - Uso de videoconferências;
  - Uso de voz sobre IP (VoIP, *Voice over IP*).

A coleta do conjunto de amostras de tráfego de ataque, para treinar o protótipo experimental do mecanismo, foi realizada com base na simulção de ataques por inundação TCP SYN e UDP. Para realizar as simulações foi utilizado a ferramenta de segurança de rede hping (SANFILIPPO, 2006), que é uma ferramenta de geração de pacotes e analise de rede, comumente utilizada para realizar testes de *firewalls* e rede. Com essa ferramenta, pode-se gerar pacotes para os protocolos TCP, UDP, ICMP e RAW-IP, sendo possível especificar inclusive o intervalo entre as transmissões dos pacotes ou informar que a transmissão de pacotes se dará no modo de inundação. Ao coletar o tráfego de ataque, é necessário se assegurar de que não há tráfego de uso legítimo na rede, minimizando a inserção de ruídos na base de treinamento.

Após coletar todos os conjuntos de dados necessários, treina-se os classificadores bayesianos simples. Os resultados do treinamento do classificador de ataques por inundação TCP SYN estão apresentados na Tabela 1.

Tabela 1 – Tabela de Treinamento do Ataque por Inundação TCP SYN

Classe	Média (μ)	Desvio Padrão (σ)
Tráfego Normal (C_N)	1.367528	15.162268
Tráfego de Ataque (C_A)	33709.571429	22649.832694

Fonte: Elaborada pelo autor

Na Tabela 2 estão apresentados os resultados do treinamento do classificador de ataques por inundação UDP.

Tabela 2 – Tabela de Treinamento do Ataque por Inundação UDP

Classe	Média (μ)	Desvio Padrão (σ)
Tráfego Normal (C_N)	124.352941	344.498086
Tráfego de Ataque (C_A)	35768.285714	21922.441377

Fonte: Elaborada pelo autor

6 RESULTADOS

Este capítulo apresenta uma avaliação do mecanismo de detecção com base em um estudo matemático e experimental do mesmo. A Seção 6.1 expõe uma demonstração matemática da classificação dos ataques de inundação TCP SYN e UDP. A Seção 6.2 apresenta alguns resultados experimentais do mecanismo de classificação para ambos os ataques.

6.1 Avaliação Matemática

Pela definição matemática do classificar bayesiano simples, sabe-se que uma dada amostra, X, será classificada como tráfego de ataque, na classe C_A, se e somente se sua probabilidade posterior for maior que a probabilidade posterior para a classe de tráfego normal, C_N. Isto é,

$$P(X \mid C_A)P(C_A) > P(X \mid C_N)P(C_N) \tag{6.1}$$

Tendo que os atributos utilizados para detecção dos ataques por inundação TCP SYN e UDP são vetores unidimensionais, pode-se calcular o limiar entre a classificação como tráfego suspeito de ataque e tráfego normal resolvendo a inequação (6.1) em relação à amostra $X = (x)$. Por essa resolução, encontrando o limiar especificamente para os dois tipos de ataque estudados por este trabalho, é possível ter uma melhor compreensão de como o mecanismo se comportará em uma dada situação. Essa resolução é apresentada a seguir:

$$P(X \mid C_A)P(C_A) > P(X \mid C_N)P(C_N) \Leftrightarrow \tag{6.2}$$

$$P(X \mid C_A) > P(X \mid C_N), \quad P(C_A) = P(C_N) = \frac{1}{2} \tag{6.3}$$

Uma vez que os atributos são valores contínuos, tem-se

$$P(X \mid C_A) > P(X \mid C_N) \Leftrightarrow \tag{6.4}$$

$$g(x, \mu_A, \sigma_A) > g(x, \mu_N, \sigma_N) \Leftrightarrow \tag{6.5}$$

$$\frac{1}{\sqrt{2\pi\sigma_A}} e^{-\frac{(x-\mu_A)^2}{2\sigma_A^2}} > \frac{1}{\sqrt{2\pi\sigma_N}} e^{-\frac{(x-\mu_N)^2}{2\sigma_N^2}} \Leftrightarrow \tag{6.6}$$

$$\frac{1}{\sqrt{2\pi}\sqrt{\sigma_A}} e^{-\frac{(x-\mu_A)^2}{2\sigma_A^2}} > \frac{1}{\sqrt{2\pi}\sqrt{\sigma_N}} e^{-\frac{(x-\mu_N)^2}{2\sigma_N^2}} \Leftrightarrow \tag{6.7}$$

$$\frac{1}{\sqrt{\sigma_A}}e^{-\frac{(x-\mu_A)^2}{2\sigma_A^2}} > \frac{1}{\sqrt{\sigma_N}}e^{-\frac{(x-\mu_N)^2}{2\sigma_N^2}} \tag{6.8}$$

Aplicando logaritmo natural em ambos os lados da inequação, tem-se

$$\ln\left(\frac{1}{\sqrt{\sigma_A}}e^{-\frac{(x-\mu_A)^2}{2\sigma_A^2}}\right) > \ln\left(\frac{1}{\sqrt{\sigma_N}}e^{-\frac{(x-\mu_N)^2}{2\sigma_N^2}}\right) \Leftrightarrow \tag{6.9}$$

$$\ln\frac{1}{\sqrt{\sigma_A}} + \ln e^{-\frac{(x-\mu_A)^2}{2\sigma_A^2}} > \ln\frac{1}{\sqrt{\sigma_N}} + \ln e^{-\frac{(x-\mu_N)^2}{2\sigma_N^2}} \Leftrightarrow \tag{6.10}$$

$$\ln 1 - \ln\sigma_A^{\frac{1}{2}} - \frac{(x-\mu_A)^2}{2\sigma_A^2} > \ln 1 - \ln\sigma_N^{\frac{1}{2}} - \frac{(x-\mu_N)^2}{2\sigma_N^2} \Leftrightarrow \tag{6.11}$$

$$-\frac{\ln\sigma_A}{2} - \frac{(x-\mu_A)^2}{2\sigma_A^2} > -\frac{\ln\sigma_N}{2} - \frac{(x-\mu_N)^2}{2\sigma_N^2} \tag{6.12}$$

Multiplicando ambos os lados por -2, obtém-se

$$\ln\sigma_A + \frac{(x-\mu_A)^2}{\sigma_A^2} < \ln\sigma_N + \frac{(x-\mu_N)^2}{\sigma_N^2} \Leftrightarrow \tag{6.13}$$

$$\frac{\sigma_A^2\ln\sigma_A + (x-\mu_A)^2}{\sigma_A^2} < \frac{\sigma_N^2\ln\sigma_N + (x-\mu_N)^2}{\sigma_N^2} \tag{6.14}$$

Assim, multiplicando ambos os lados por $\sigma_N^2\sigma_A^2$, obtém-se

$$\sigma_N^2\sigma_A^2\ln\sigma_A + \sigma_N^2(x-\mu_A)^2 < \sigma_A^2\sigma_N^2\ln\sigma_N + \sigma_A^2(x-\mu_N)^2 \Leftrightarrow \tag{6.15}$$

$$\sigma_N^2\sigma_A^2\ln\sigma_A + \sigma_N^2(x^2 - 2\mu_A x + \mu_A^2) < \sigma_A^2\sigma_N^2\ln\sigma_N + \sigma_A^2(x^2 - 2\mu_N x + \mu_N^2) \Leftrightarrow \tag{6.16}$$

$$\sigma_N^2x^2 - 2\sigma_N^2\mu_A x + \sigma_N^2\sigma_A^2\ln\sigma_A + \sigma_N^2\mu_A^2 < \sigma_A^2x^2 - 2\sigma_A^2\mu_N x + \sigma_A^2\sigma_N^2\ln\sigma_N + \sigma_A^2\mu_N^2 \Leftrightarrow \tag{6.17}$$

$$(\sigma_A^2-\sigma_N^2)x^2 + (2\sigma_N^2\mu_A - 2\sigma_A^2\mu_N)x + (\sigma_A^2\sigma_N^2\ln\sigma_N + \sigma_A^2\mu_N^2 - \sigma_N^2\sigma_A^2\ln\sigma_A - \sigma_N^2\mu_A^2) > 0 \tag{6.18}$$

Dessa maneira, fazendo

$$\begin{cases} a = \sigma_A^2 - \sigma_N^2 \\ b = 2\sigma_N^2\mu_A - 2\sigma_A^2\mu_N \\ c = \sigma_A^2\sigma_N^2\ln\sigma_N + \sigma_A^2\mu_N^2 - \sigma_N^2\sigma_A^2\ln\sigma_A - \sigma_N^2\mu_A^2 \end{cases} \tag{6.19}$$

a inequação (6.18) pode ser simplificada para

$$ax^2 + bx + c > 0 \tag{6.20}$$

6.1.1 Classificador de Ataques por Inundação TCP SYN

Com base nos valores apresentados pela Tabela 1, pode-se calcular os valores do atributo $X = (P_{SYN} - (P_{FIN} + P_{RST}))$ para os quais o classificador assumirá como suspeito de ataque. Assim, assumindo os valores

$$\begin{cases} \mu_N = 1.367528 \\ \sigma_N = 15.162268 \\ \mu_A = 33709.571429 \\ \sigma_A = 22649.832694 \end{cases} \tag{6.21}$$

e aplicando-os às equações de (6.19), tem-se que

$$\begin{cases} a = 513014691.172 \\ b = -1387625256.52 \\ c = -1122307116050 \end{cases} \tag{6.22}$$

e, portanto

$$513014691.172x^2 - 1387625256.52x - 1122307116050 > 0 \tag{6.23}$$

Essa inequação pode ser resolvida encontrando as raizes da equação equivalente

$$ax^2 + bx + c = 0 \tag{6.24}$$

de maneira, que usando a fórmula de Bhaskara

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{6.25}$$

obtém-se as raizes

$$\begin{cases} x_1 = 48.144511249255494 \\ x_2 = -45.439666182639954 \end{cases} \tag{6.26}$$

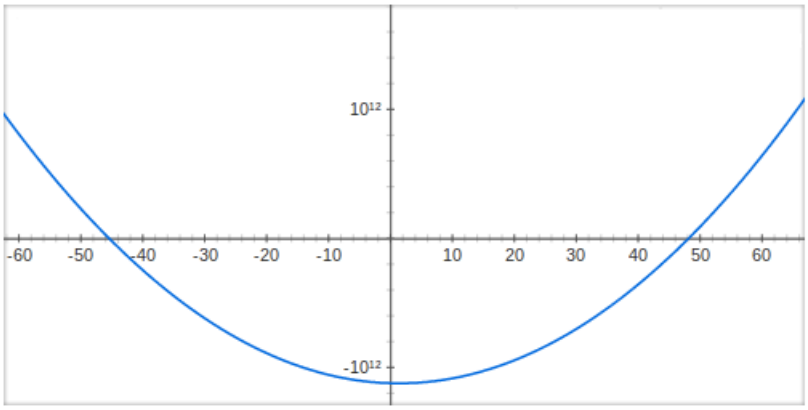
Realizando um estudo de sinais a partir do gráfico ilustrado pela Figura 6, observa-se que

$P(X \mid C_A) > P(X \mid C_N)$ sempre que $x < -45.4396661826399$ ou $x > 48.1445112492555$

A partir desse estudo matemático, observa-se que sempre que um fluxo de pacotes apresentar características tais que $P_{SYN} - (P_{FIN} + P_{RST}) \geq 49$ ou $P_{SYN} - (P_{FIN} + P_{RST}) \leq -46$, o mesmo será classificado como um fluxo suspeito de ataque. Assim, fluxos de pacotes onde $-46 < P_{SYN} - (P_{FIN} + P_{RST}) < 49$ serão classificados como fluxo normal, simbolizados graficamente por valores negativos no eixo das ordenadas. Esse limiar não é um valor fixo, mas sim relativo à base de treinamento, de maneira que basta modificá-la para se obter um novo limiar. Portanto, melhorando o conjunto de dados de treinamento, melhora-se o limiar de classificação.

Figura 6 – Gráfico do Classificador do Ataque TCP SYN

$y = P(X \mid C_A) - P(X \mid C_N), x = P_{SYN} - (P_{FIN} + P_{RST})$



Fonte: Elaborada pelo autor

Esse resultado mostra que para ataques realizados por uma *botnet* com um número de agentes suficientemente grande, é possível que cada agente envie uma quantidade de pacotes TCP SYN pequena o bastante para ser classificada como fluxo normal, $-46 < P_{SYN} - (P_{FIN} + P_{RST}) < 49$, e ainda ser capaz de tornar os recursos da vítima primária inacessíveis para os usuários legítimos. Para esses casos críticos de ataque, a detecção pode ser realizada em pontos com maior agregação de fluxos de ataque, possivelmente por mecanismos de detecção na rede intermediária ou na rede da vítima primária.

6.1.2 Classificador de Ataques por Inundação UDP

De forma semelhante, com base nos valores apresentados pela Tabela 2, pode-se calcular os valores do atributo $X = (P_{UDP})$ para os quais o classificador assumirá como suspeito de ataque. Assim, assumindo os valores

$$\begin{cases} \mu_N = 124.352941 \\ \sigma_N = 344.498086 \\ \mu_A = 35768.285714 \\ \sigma_A = 21922.441377 \end{cases} \tag{6.27}$$

e aplicando-os às equações de (6.19), tem-se que

$$\begin{cases} a = 480474756.997 \\ b = -111036530523 \\ c = -381284510876000 \end{cases} \tag{6.28}$$

e, portanto

$$480474756.997x^2 - 111036530523x - 381284510876000 > 0 \tag{6.29}$$

Essa inequação pode ser resolvida, pela fórmula de Bhaskara, encontrando as raizes da equação equivalente

$$ax^2 + bx + c = 0 \tag{6.30}$$

logo obtém-se as raizes

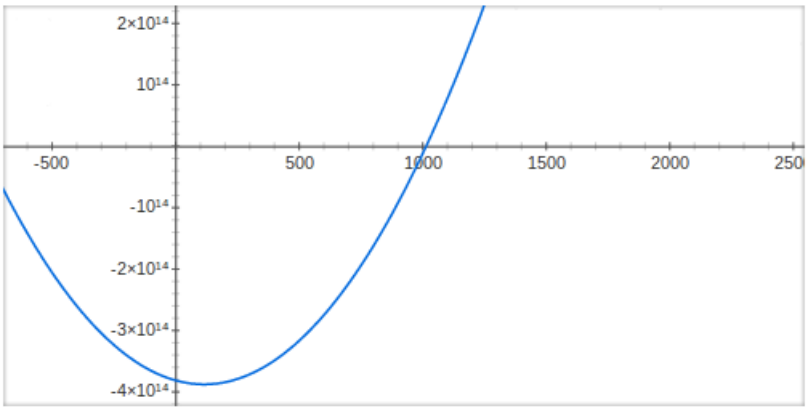
$$\begin{cases} x_1 = 1013.8301019021364 \\ x_2 = -782.7325699172792 \end{cases} \tag{6.31}$$

Com base em um estudo de sinais a partir do gráfico ilustrado pela Figura 7, observa-se que

$$P(X \mid C_A) > P(X \mid C_N) \quad \text{sempre que } x < -782.73256991728 \text{ ou } x > 1013.83010190214$$

Figura 7 – Gráfico do Classificador do Ataque UDP

$y = P(X \mid C_A) - P(X \mid C_N), x = P_{UDP}$



Fonte: Elaborada pelo autor

Por esse estudo matemático, verifica-se que um fluxo de pacotes será classificado como um fluxo suspeito de ataque, sempre que apresentar características tais que $P_{UDP} \geq 1014$ ou $P_{UDP} \leq -783$. Entretanto, $P_{UDP} < 0$ é um valor inválido, pois P_{UDP} é a quantidade de pacotes UDP enviados para um determinado endereço IP. Assim, fluxos de pacotes com $P_{UDP} < 1014$ serão classificados como fluxo normal. Contudo, esse limiar não é um valor fixo, mas sim relativo à base de treinamento. De maneira que melhorando o conjunto de dados de treinamento, melhora-se também o limiar de classificação.

Por esse resultado, nota-se que para ataques realizados por uma *botnet* com um número de agentes suficientemente grande, é possível que cada agente envie uma quantidade de pacotes UDP pequena o bastante para ser classificada como fluxo normal, e ainda ser capaz de tornar os recursos da vítima primária inacessíveis para os usuários legítimos. Para esses casos críticos

de ataque, a detecção pode ser realizada em pontos com maior agregação de fluxos de ataque, possivelmente por mecanismos de detecção na rede intermediária ou na rede da vítima primária.

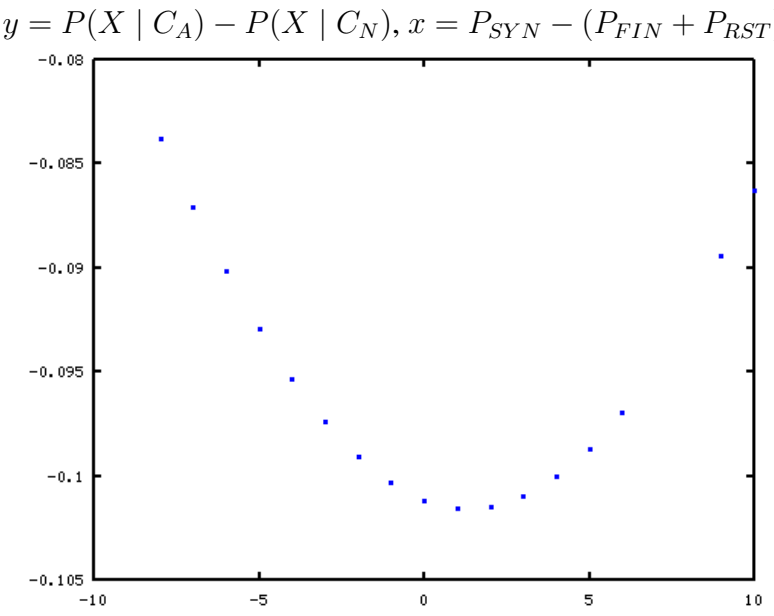
6.2 Resultados Experimentais Mediante Simulações

Com base no estudo matemático exposto pela Seção 6.1, é possível realizar uma série de experimentos e compará-los aos resultados obtidos matematicamente. Os experimentos foram realizados mediante simulações, seguindo os padrões de uso apresentados no Capítulo 5.

6.2.1 Classificador de Ataques por Inundação TCP SYN

Para os experimentos realizados simulando um uso legítimo da rede, o classificador apresentou o resultado ilustrado graficamente pela Figura 8. Por esse gráfico é possivel observar que o tráfego normal experimental se aproxima do gráfico teórico calculado na Seção 6.1, como apresentado pela Figura 6, mostrando experimentalmente que um fluxo de pacotes legítimo apresentará características tais que $-46 < P_{SYN} - (P_{FIN} + P_{RST}) < 49$, como calculado na Subseção 6.1.1.

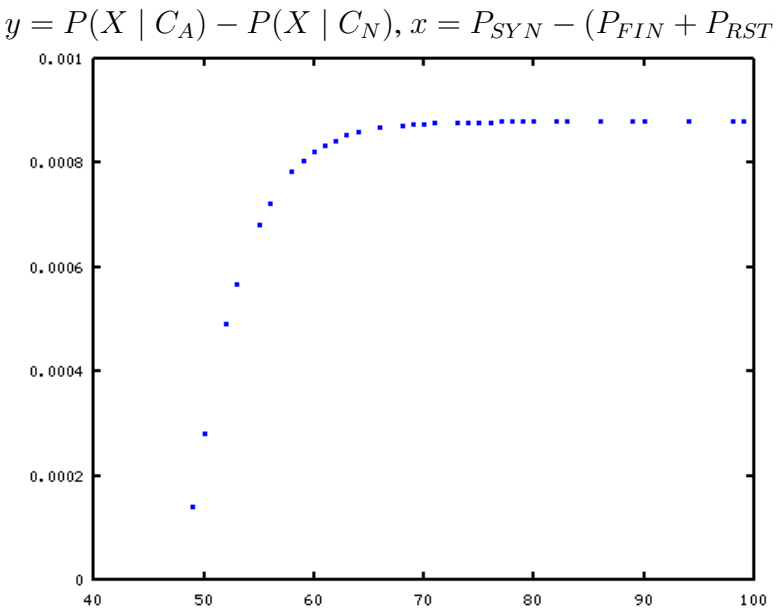
Figura 8 – Gráfico de Valores Experimentais de Tráfegos Normais



Fonte: Elaborada pelo autor

O gráfico apresentado pela Figura 9 ilustra os resultados do classificador para os experimentos de ataque por inundação TCP SYN, simulados com a ferramenta hping, variando o intervalo entre a transmissão de pacotes. Como esperado, fluxos de pacotes de ataque que apresentam características tais que $P_{SYN} - (P_{FIN} + P_{RST}) \geq 49$ ou $P_{SYN} - (P_{FIN} + P_{RST}) \leq -46$ são classificados como um tráfego de ataque por inundação TCP SYN.

Figura 9 – Gráfico de Valores Experimentais de Tráfegos de Ataque



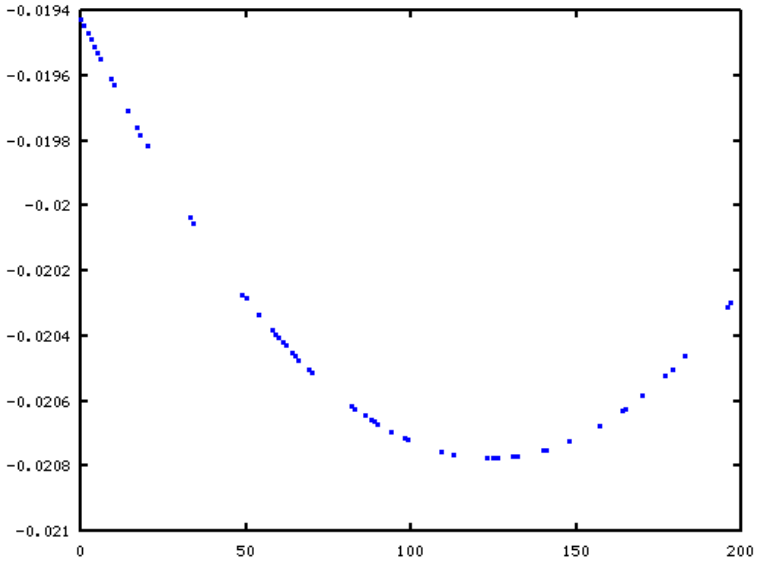
Fonte: Elaborada pelo autor

6.2.2 Classificador de Ataques por Inundação UDP

Os resultados dos experimentos de uso legítimo da rede, em relação ao protocolo UDP, apresentados pelo classificador são expostos pelo gráfico ilustrado na Figura 10. Por estes experimentos é possível observar que os resultados experimentais são equivalentes aos valores esperados com base na avaliação matemática apresentada na Subseção 6.2.1, mostrando experimentalmente que um fluxo de pacotes legítimo apresentará características que $0 \leq P_{UDP} < 1014$.

Figura 10 – Gráfico de Valores Experimentais de Tráfegos Normais

$$y = P(X \mid C_A) - P(X \mid C_N), x = P_{UDP}$$

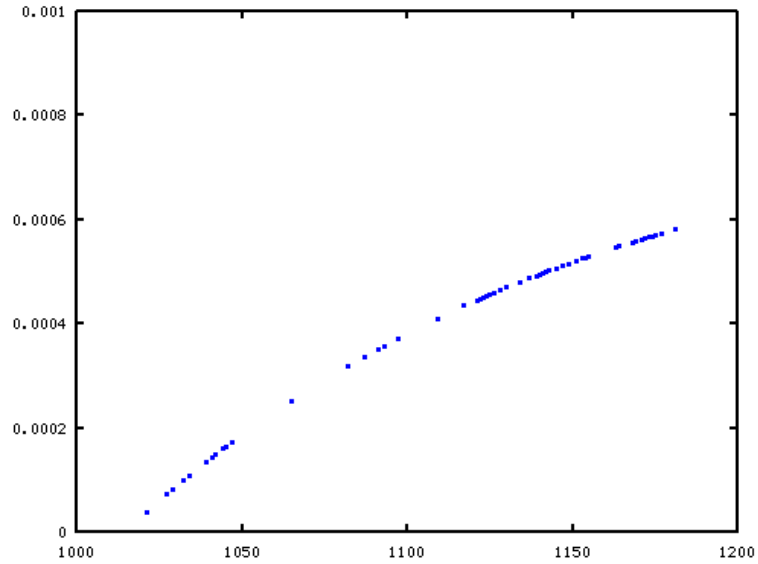


Fonte: Elaborada pelo autor

O gráfico apresentado pela Figura 11 ilustra os resultados do classificador para os experimentos de ataque por inundação UDP, simulados com a ferramenta hping, variando o intervalo entre a transmissão de pacotes. Como esperado, fluxos de pacotes de ataque que apresentam características que $P_{UDP} \geq 1014$ são classificados como um tráfego suspeito de ataque.

Figura 11 – Gráfico de Valores Experimentais de Tráfegos de Ataque

$$y = P(X \mid C_A) - P(X \mid C_N), x = P_{UDP}$$



Fonte: Elaborada pelo tor

7 CONCLUSÕES

Mecanismos de defesa de ataques DDoS implantados na rede da vítima têm a necessidade de adotar sistemas de rastreamento de pacotes IP, uma vez que pacotes com endereço IP de origem forjado podem ser usados no ataque para ocultar sua verdadeira origem. Estudos apontam que, idealmente, ataques DDoS devem ser parados o mais perto da origem quanto possível. Dessa maneira, implantando o mecanismo de defesa na rede de origem, temos que o fluxo de ataque poderá ser bloqueado antes de entrar no núcleo da Internet e ser agregado a outros fluxos, podendo causar congestionamento, além de que o baixo grau de agregação de fluxos permite usar estratégias de defesa mais complexas e com maior precisão.

Este trabalho propôs um mecanismo de detecção em tempo-real de diferentes tipos de ataque de negação de serviço, cuja implantação deve ser feita na rede de origem do ataque, ou seja, em máquinas sujeitas a serem possíveis agentes de uma *botnet* destinada a efetuar ataques DDoS. Esse mecanismo objetiva detectar fluxos com indícios de ataque antes que se agreguem aos demais fluxos, para que futuramente possam ser de fato bloqueados, evitando causar maiores danos à rede. O mecanismo proposto utiliza o classificador bayesiano simples para detectar tráfegos com indícios de ataque.

A modularização apresentada por esse mecanismo facilita sua extensão, de maneira que para acrescentar a detecção de um novo tipo de ataque DoS seja necessário apenas acrescentar sua extração e agrupamento de atributos no analisador de fluxo de pacotes, e acrescentar um classificador bayesiano simples treinado especificamente para esse novo ataque.

Utilizando um protótipo experimental e uma base de treinamento obtida simulando de maneira controlada o comportamento de um agente de uma *botnet*, foram realizados alguns experimentos. Os resultados experimentais foram então confrontados com os resultados de uma avaliação matemática. Com base nessa avaliação matemática, obteve-se um limiar para cada ataque sendo detectado, de maneira que se um determinado fluxo apresentar características tais que estejam além desse limiar, o mesmo será classificado como um tráfego de ataque. Mostrando que o mecanismo é eficaz quanto à detecção de ataques. Entretanto, tais limiares não são valores fixos, mas sim relativos às bases de treinamento.

Os resultados mostraram também que para ataques realizados por uma *botnet* com um número de agentes suficientemente grande, é possível que cada agente envie uma quantidade de pacotes de ataque pequena o bastante para ser classificada como fluxo normal e ainda ser

capaz de tornar os recursos da vítima primária inacessíveis para os usuários legítimos. Para esses casos críticos de ataque, a detecção pode ser realizada em pontos com maior agregação de fluxos de ataque, possivelmente por mecanismos de detecção na rede intermediária ou na rede da vítima primária. Portanto, se forem utilizados mecanismos de detecção de ataques de negação de serviço em conjunto, é possível tornar a detecção dos ataques mais efetiva, de maneira que, quando possível detectar os ataques ainda na origem, pode-se bloqueá-los e evitar que sejam agregados a outros fluxos, evitando causar congestionamentos e outros danos maiores à rede.

7.1 Trabalhos Futuros

Esta seção exibe alguns possíveis trabalhos futuros que podem ser realizados, complementando e incrementando este trabalho.

Como possíveis trabalhos futuros, pode-se realizar uma avaliação matemática e experimental utilizando dados de ataques reais, comparando com o que foi obtido por este trabalho utilizando dados obtidos por simulação. Desta maneira, será possível avaliar o limiar obtido com o treinamento utilizando dados reais e compará-lo, em situações reais, com o limiar obtido de maneira simulada.

Um segundo trabalho que pode ser feito é a comparação deste mecanismo com outros mecanismos de detecção de ataques de negação de serviço implantados na origem, equivalentes ao apresentado por este trabalho. Assim, será possível avaliar as vantagens de um ou de outro mecanismo, tendo a possibilidade de desenvolver mecanismos híbridos.

Pode-se também fazer uma extensão deste mecanismo para detectar outros tipos de ataque de negação de serviço. Entre as possibilidades estão o ataque por inundação TCP PUSH+ACK ou o ataqgue por inundação ICMP.

Por fim, pode-se estudar a utilização deste mecanismo de detecção apresentado, trabalhando em conjunto com outros mecanismos de detecção implantados em diferentes localizações, como na rede intermediária ou na vítima primária. Dessa maneira será possível extrair o melhor de cada mecanismo, tirando proveito de suas vantagens individuais.

REFERÊNCIAS

AMOR, N. B.; BENFERHAT, S.; ELOUEDI, Z. Naive Bayes vs decision trees in intrusion detection systems. In: *Proceedings of the 2004 ACM symposium on Applied computing*. New York, NY, USA: ACM, 2004. (SAC '04), p. 420–424. ISBN 1-58113-812-1. Disponível em: <http://doi.acm.org/10.1145/967900.967989>.

BINKLEY, J. R.; SINGH, S. An algorithm for anomaly-based botnet detection. In: *Proceedings of the 2nd conference on Steps to Reducing Unwanted Traffic on the Internet - Volume 2*. Berkeley, CA, USA: USENIX Association, 2006. p. 7–7. Disponível em: <http://dl.acm.org/citation.cfm?id=1251296.1251303>.

BOUCKAERT, R. Naive bayes classifiers that perform well with continuous variables. In: WEBB, G.; YU, X. (Ed.). *AI 2004: Advances in Artificial Intelligence*. Berlin / Heidelberg, Germany: Springer Berlin / Heidelberg, 2005, (Lecture Notes in Computer Science, v. 3339). p. 85–116.

CHEN, Y.; HWANG, K.; KU, W.-S. Collaborative Detection of DDoS Attacks over Multiple Network Domains. *Parallel and Distributed Systems, IEEE Transactions on*, v. 18, n. 12, p. 1649 –1662, dec. 2007. ISSN 1045-9219.

COOKE, E.; JAHANIAN, F.; MCPHERSON, D. The Zombie roundup: understanding, detecting, and disrupting botnets. In: *Proceedings of the Steps to Reducing Unwanted Traffic on the Internet on Steps to Reducing Unwanted Traffic on the Internet Workshop*. Berkeley, CA, USA: USENIX Association, 2005. p. 6–6. Disponível em: <http://portal.acm.org/citation.cfm?id=1251282.1251288>.

DOULIGERIS, C.; MITROKOTSA, A. DDoS attacks and defense mechanisms: classification and state-of-the-art. *Comput. Netw.*, Elsevier North-Holland, Inc., New York, NY, USA, v. 44, p. 643–666, April 2004. ISSN 1389-1286. Disponível em: <http://dl.acm.org/citation.cfm?id=987153.987158>.

FARID, D. M.; HARBI, N.; RAHMAN, M. Z. Combining Naive Bayes and Decision Tree for Adaptive Intrusion Detection. *CoRR*, abs/1005.4496, 2010.

FEDYNYSHYN, G.; CHUAH, M. C.; TAN, G. Detection and classification of different botnet C&C channels. In: *Proceedings of the 8th international conference on Autonomic and trusted computing*. Berlin, Heidelberg: Springer-Verlag, 2011. (ATC'11), p. 228–242. ISBN 978-3-642-23495-8. Disponível em: <http://dl.acm.org/citation.cfm?id=2035700.2035722>.

FEILY, M.; SHAHRESTANI, A.; RAMADASS, S. A Survey of Botnet and Botnet Detection. *Emerging Security Information, Systems, and Technologies, The International Conference on*, IEEE Computer Society, Los Alamitos, CA, USA, v. 0, p. 268–273, 2009.

GOEBEL, J.; HOLZ, T. Rishi: identify bot contaminated hosts by IRC nickname evaluation. In: *Proceedings of the first conference on First Workshop on Hot Topics in Understanding*

Botnets. Berkeley, CA, USA: USENIX Association, 2007. p. 8–8. Disponível em: <http://dl.acm.org/citation.cfm?id=1323128.1323136>.

HAN, J.; KAMBER, M.; PEI, J. *Data Mining: Concepts and Techniques, Second Edition (The Morgan Kaufmann Series in Data Management Systems)*. 2. ed. Morgan Kaufmann, 2006. Hardcover. ISBN 1558609016. Disponível em: <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/1558609016>.

JACOBSON, V.; LERES, C.; MCCANNE, S. *Packet Capture Library (libpcap) Manual*. Lawrence Berkeley National Laboratory, University of California, Berkeley, CA., January 15 2004. URL: www.tcpdump.org.

KARASARIDIS, A.; REXROAD, B.; HOEFLIN, D. Wide-scale botnet detection and characterization. In: *Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*. Berkeley, CA, USA: USENIX Association, 2007. p. 7–7. Disponível em: <http://dl.acm.org/citation.cfm?id=1323128.1323135>.

KARGL, F.; MAIER, J.; WEBER, M. Protecting web servers from distributed denial of service attacks. In: *Proceedings of the 10th international conference on World Wide Web*. New York, NY, USA: ACM, 2001. (WWW '01), p. 514–524. ISBN 1-58113-348-0. Disponível em: <http://doi.acm.org/10.1145/371920.372148>.

KRUEGEL, C. et al. Bayesian Event Classification for Intrusion Detection. In: *Proceedings of the 19th Annual Computer Security Applications Conference*. Washington, DC, USA: IEEE Computer Society, 2003. (ACSAC '03), p. 14–. ISBN 0-7695-2041-3. Disponível em: <http://dl.acm.org/citation.cfm?id=956415.956436>.

KUMARASAMY, S.; GOWRISHANKAR, A. An Active Defense Mechanism for TCP SYN flooding attacks. *CoRR*, abs/1201.2103, 2012.

LAU, F. et al. Distributed denial of service attacks. In: *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*. Nashville, TN: Systems, Man, and Cybernetics, 2000 IEEE International Conference on, 2000. v. 3, p. 2275 –2280 vol.3. ISSN 1062-922X.

LAUFER, R. P. et al. Negação de Serviço: Ataques e Contramedidas. In: *Livro Texto dos Mini-cursos do V Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*. Florianópolis, Brasil: (SBSeg'2005), 2005.

MIRKOVIC, J.; PRIER, G.; REIHER, P. L. Attacking DDoS at the Source. In: *Proceedings of the 10th IEEE International Conference on Network Protocols*. Washington, DC, USA: IEEE Computer Society, 2002. (ICNP '02), p. 312–321. ISBN 0-7695-1856-7. Disponível em: <http://dl.acm.org/citation.cfm?id=645532.656169>.

MITROKOTSA, A.; DOULIGERIS, C. Detecting denial of service attacks using emergent self-organizing maps. In: *Signal Processing and Information Technology, 2005. Proceedings of the Fifth IEEE International Symposium on*. Athens: IEEE, 2005. p. 375 –380.

RISH, I. An empirical study of the naive Bayes classifier. In: *Proceedings of IJCAI-01 workshop on Empirical Methods in AI*. Sicily, Italy: IJCAI, 2001.

SAHA, B.; GAIROLA, A. Botnet: An overview. In: *CERT-In White Paper (CIWP-2005-05)*. [S.l.]: CERT, 2005.

SANFILIPPO, S. *hping*. 2006. http://www.hping.org.

SEN, J. A Novel Mechanism for Detection of Distributed Denial of Service Attacks. *CoRR*, abs/1101.2715, 2011.

SPECHT, S. M. Distributed denial of service: taxonomies of attacks, tools and countermeasures. In: *Proceedings of the International Workshop on Security in Parallel and Distributed Systems, 2004*. [S.l.: s.n.], 2004. p. 543–550.

STRAYER, W. et al. Detecting Botnets with Tight Command and Control. In: *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*. Tampa, FL: IEEE, 2006. p. 195 –202. ISSN 0742-1303.

SUN, C.; FAN, J.; LIU, B. A Robust Scheme to Detect SYN Flooding Attacks. In: *Communications and Networking in China, 2007. CHINACOM '07. Second International Conference on*. Shanghai: IEEE, 2007. p. 397 –401.

TARIQ, U. et al. Collaborative Peer to Peer Defense Mechanism for DDoS Attacks. *Procedia Computer Science*, v. 5, n. 0, p. 157 – 164, 2011. ISSN 1877-0509. ¡ce:title¿The 2nd International Conference on Ambient Systems, Networks and Technologies (ANT-2011) / The 8th International Conference on Mobile Web Information Systems (MobiWIS 2011)¡/ce:title¿. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1877050911003474>.

VIJAYASARATHY, R.; RAGHAVAN, S. V.; RAVINDRAN, B. A system approach to network modeling for DDoS detection using a Naive Bayesian classifier. In: *COMSNETS'11, Third International Conference on Communication Systems and Networks*. Bangalore, India: IEEE, 2011. p. 1–10.

WANG, H.; ZHANG, D.; SHIN, K. G. Detecting SYN flooding attacks. In: *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. New York, NY, USA: IEEE, 2002. v. 3, p. 1530 – 1539. ISSN 0743-166X.

WU, Y. et al. DDoS detection and traceback with decision tree and grey relational analysis. *Int. J. Ad Hoc Ubiquitous Comput.*, Inderscience Publishers, Inderscience Publishers, Geneva, SWITZERLAND, v. 7, p. 121–136, March 2011. ISSN 1743-8225. Disponível em: <http://dx.doi.org/10.1504/IJAHUC.2011.038998>.

ZHANG, H.; SU, J. Naive Bayesian classifiers for ranking. In: *Proceedings of the 15th European Conference on Machine Learning (ECML2004*. [S.l.]: Springer, 2004.

ZHANG, J. et al. Boosting the scalability of botnet detection using adaptive traffic sampling. In: *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*. New York, NY, USA: ACM, 2011. (ASIACCS '11), p. 124–134. ISBN 978-1-4503-0564-8. Disponível em: <http://doi.acm.org/10.1145/1966913.1966930>.