Identificando Botnets Geradoras de Ataques DDoS Volumétricos por Processamento de Sinais em Grafos

Arthur E. G. Ferreira, Michele Nogueira

[1]Centro de Ciência de Segurança Computacional (CCSC)
Universidade Federal do Paraná (UFPR)
{aegferreira,michele}@inf.ufpr.br

Abstract. *Volumetric Distributed Denial of Service (DDoS) attack is a serious threat to network security. This attack aims at making services unavailable by flooding servers and networks with malicious traffic. This work analyzes the application of the Causal Graph Process (CGP), a technique prominent from signal processing on graphs, to detect the network of infected hosts (botnet) that generate volumetric DDoS attacks. Different from other botnet detection techniques, CGP uses fewer features from the network to detect possible malicious activities and it is independent of a training phase. CGP analyzes casual relations in a network from a dataset. Each data is treated as signal. Then, causal graphs are extracted from these signals, and a matrix of influences between the nodes is calculated. The magnitude of the values in the adjacency matrix is used to identify the level of coordination and causality between nodes and to assist in the botnet identification. We have applied CGP over three datasets containing volumetric DDoS attacks. Results demonstrate the botnet identification by CGP.*

Resumo. *O ataque volumétrico de negação de serviço distribuído (Distributed Denial of Service – DDoS) é uma séria ameaça de segurança podendo tornar os serviços indisponíveis pela geração de grandes volumes de tráfego malicioso. Este trabalho investiga a viabilidade de aplicação do Causal Graph Process (CGP), uma das técnicas proeminentes de processamento de sinais em grafos, para identificar redes de dispositivos infectados (botnets). Diferente de outras técnicas de detecção de botnets, o CGP necessita de poucas características da rede para detectar possíveis atividades maliciosas e independe de uma fase de treinamento para identificação dos bots. O CGP é aplicado para analisar dados sobre as atividades da rede, os quais são tratados como sinais. Posteriormente, os grafos de causalidade são extraídos e os graus de correlação entre os nós são calculados. Esses graus são utilizados para identificar o nível de coordenação entre os nós e auxiliar na identificação da botnet. O CGP é aplicado sobre três bases de dados diferentes contendo registros de ataques DDoS. Os resultados demonstram a identificação das botnets a partir do CGP.*

1. Introdução

Uma *botnet* é uma rede de *hosts* infectados (*bots*) coordenada por um *botmaster* para gerar diferentes tipos de ataque [Miller and Busby-Earle 2016]. Um exemplo de ataque gerado por uma *botnet* é o ataque Distribuído de Negação de Serviço (do inglês, *Distributed Denial of Service* - DDoS), um dos problemas mais desafiadores na Internet [Harris 2010]. O desafio consiste na natureza do ataque, que visa interromper o funcionamento de servidores e redes tornando seus serviços indisponíveis. A estratégia mais comum para os ataques

DDoS versa na utilização de uma grande quantidade de *bots* para sobrecarregar uma determinada rede ou servidor através da geração de tráfego ilegítimo. Essa sobrecarga visa consumir todos os recursos do servidor alvo, negando os serviços e causando grandes prejuízos em termos financeiros e de reputação das empresas [Mirkovic et al. 2002].

A maioria das abordagens de detecção de *botnets* na literatura foca principalmente na detecção das mensagens de Comando & Controle (C&C) e suas arquiteturas de comunicação. Contudo essas abordagens podem se tornar ineficazes caso as *botnets* mudem suas arquiteturas ou protocolos de comunicação [Zeidanloo et al. 2010]. Para realizar um ataque utilizando uma *botnet*, o atacante envia através de protocolos de comunicação da Internet (como IRC e HTTP) instruções de ataque para o *botmaster* contendo dados como endereço e porta de destino, protocolo a ser utilizado, quantidade de *bots* e duração do ataque [Miller and Busby-Earle 2016]. Essas instruções são repassadas aos *bots* que executam as atividades instruídas dando início ao ataque.

Estudos como os de [Chowdhury et al. 2017, Joshi et al. 2017, Wang and Paschalidis 2017, Su et al. 2017, Sapello et al. 2017] utilizam o agrupamento de características comuns de rede associadas a algoritmos de aprendizado de máquina para detectar *botnets* na rede. Esses estudos empregam o agrupamento de pares de caraterísticas comuns encontradas nos tráfegos de redes, como endereço e porta de origem e destino, protocolo, e similaridade de fluxos (requisições e respostas). Essas características auxiliam na geração de grafos que representam a rede avaliada. Esses grafos são processados por algoritmos que implementam métodos como SOM (*Self-organizing Map*) [Chowdhury et al. 2017], DFF (*Double Fast-Flux*) [Sapello et al. 2017] e SCG (*Social Correlation Graph*) [Wang and Paschalidis 2017] cujos resultados são utilizados como dado de treinamento para algoritmos de aprendizado de máquina como SVM (*Support Vector Machines*) [Sapello et al. 2017] e DT (*Decision Tree*) para detectar assinaturas ou comportamento de *botnet* na rede.

Neste artigo, investiga-se a viabilidade de aplicação do *Causal Graph Process* (CGP) [Mei and Moura 2015], um método proeminente da área de Processamento de Sinais em Grafos (SPG) [Sandryhaila and Moura 2012], para a detecção de *botnets*. O CGP realiza cálculos autorregressivos sobre uma série temporal onde os dados registrados são sinais gerados por elementos, que no contexto deste trabalho são os *hosts* ou nós da rede, durante um intervalo de tempo. A partir desses cálculos, o CGP gera uma matriz de influências indexada pelos nós, onde os valores indicam a magnitude de coordenação entre eles, isto é, o quanto os nós estão coordenados no intervalo de tempo avaliado. Um dos fatores importantes para avaliar a viabilidade do CGP na detecção de *botnets* é a quantidade de características necessárias para se extrair a coordenação entre os nós. O CGP utiliza apenas uma característica para calcular as influências entre os nós. Com o CGP, identificamos a coordenação e casualidade entre os *bots* da quantidade de emissão de mensagens em comum entre dispositivos na rede. Cada emissão dos dispositivos ajuda na criação de uma rede de interações, que quando analisadas evidenciam uma *botnet*.

Este artigo está organizado como segue. A Seção 2 apresenta uma visão geral dos trabalhos relacionados a detecção de *botnets*. A Seção 3 apresenta os fundamentos de um ataque DDoS assim como a descrição e aplicação do CGP na detecção de *botnets*. A Seção 4 apresenta a metodologia utilizada neste trabalho para avaliar a viabilidade da aplicação do CGP na detecção de *botnets*, evidenciando os métodos que utilizamos

para analisar os *datasets*. A Seção 5 apresenta os resultados e a discussão a respeito dos mesmos. Por último, a Seção 6 apresenta as conclusões e trabalhos futuros.

2. Trabalhos Relacionados

Vários trabalhos na literatura abordam ataques DDoS e detecção *botnets* [Feily et al. 2009]. Em sua maioria estes trabalhos propõem métodos que detectam anomalias na rede através do uso de aprendizado de máquina, cálculo de entropia e agrupamento [Lagraa et al. 2017, Sapello et al. 2017, Masud et al. 2008, Chowdhury et al. 2017]. Nos métodos que utilizam aprendizado de máquina, de maneira geral, um algoritmo de decisão é treinado com o que é considerado o estado normal da rede e utiliza essas informações para detectar a presença de *botnets*. Nos métodos que utilizam o cálculo de entropia, o estado considerado normal da rede é calculado e um valor é atribuído ao comportamento da rede. Quando há alguma variação nesse valor, as contramedidas estabelecidas pelo administrador da rede são tomadas. Nos métodos que utilizam agrupamento, os *hosts* da rede são agrupados em classes com base na similaridade de seu comportamento e em caso de interrupção de serviços, o administrador de rede ou algum algoritmo de contramedida é acionado de forma em que os *hosts* agrupados que apresentem comportamento similares aos de uma *botnet* sejam tratados.

Na literatura, as técnicas para a detecção de *botnets* podem ser classificadas de acordo com as abordagens utilizadas. Estas abordagens se dividem em: (i) baseadas em assinaturas; (ii) detecção de anomalias; (iii) tráfego de DNS; e (iv) mineração de dados [Khajuria and Srivastava 2013]. As abordagens baseadas em reconhecimento de assinaturas utilizam uma base de conhecimento que contém dados correspondentes às características de *botnets* conhecidas para serem comparadas com os fluxos da rede a fim de detectar atividade de *botnet*. As abordagens baseadas em detecção de anomalias apresentam técnicas em que o comportamento considerado normal é calculado seja através de seu valor de entropia ou aprendizado de máquina. Qualquer variação nesse comportamento é gatilho para acionar contramedidas. Nas abordagens baseadas em tráfego DNS, o comportamento do fluxo de dados de DNS é avaliado. Desta forma, caso algum tipo de mensagem que não corresponda à finalidade do protocolo DNS (tradução endereços de fácil memorização em endereços IP), esta seja detectada, isolada e avaliada por possivelmente conter mensagens que possam evidenciar uma *botnet*. Já as abordagens baseadas em mineração de dados analisam registros da rede como um todo, utilizando técnicas de mineração de dados como associação, agrupamento, classificação e categorização para identificar atividades de uma *botnet*. A avaliação do método CGP neste trabalho se enquadra nas abordagens de mineração de dados. Isso se dá devido ao fato de avaliarmos o CGP como um classificador que distingue os *hosts* legítimos de *bots* na rede utilizando registros tráfego de rede.

Em relação à detecção de *botnets*, em 2008, [Masud et al. 2008] propuseram uma técnica para estimar o número de *bots* geradores de ataques DDoS baseado-se em *pace regression model*. Este modelo utiliza cálculos de regressão linear no valor de entropia da rede a fim de estimar a quantidade de *bots* durante um ataque. Contudo, os autores não abordam a identificação dos *bots*, ou seja, a origem dos ataques não é exposta. Ainda em 2008, [Gu et al. 2008] propuseram o BotMiner, um *framework* que utiliza técnicas de mineração de dados para detectar tráfego C&C na rede. Este *framework* agrupa fluxos

de rede similares e fluxos maliciosos de rede similares, depois ele compara os fluxos agrupados com os fluxos de rede maliciosas para identificar os dispositivos infectados através da similaridade dos fluxos agrupados com os fluxos de maliciosos conhecidos.

Em 2016, [Kong et al. 2016] propuseram uma estrutura de agrupamento para detectar *botnets* usando uma otimização do algoritmo de agrupamento HEMST (do inglês, *Hierarchical Euclidean Minimum Spanning Tree*). Essa técnica utiliza uma série de características genéricas de rede, como tamanho de pacote, tipo de protocolo e espaçamento de segmento. O valor desses atributos são extraídos da captura de rede como parâmetros para criar agrupamentos de *hosts* baseado na similaridade do fluxo de dados. Em 2016, [Kalaivani and Vijaya 2016] fizeram um estudo comparativo de diferentes técnicas de aprendizado supervisionado de máquina utilizadas na detecção e *botnets* como SVM (*Support Vector Machines)*, C-NN (*Neural Network*), Árvores de Decisão (do inglês, *Decision Tree - DT*) e *Naive Bayes*. Nesse estudo, foi ressaltada a dependência de um estágio de treinamento dos algoritmos de aprendizagem para poder diferenciar apropriadamente uma rede legítima e uma *botnet.* Caso o comportamento da rede ou mesmo da estrutura da rede ser alterado, todo o estágio de treinamento dos algoritmos devem ser executados novamente para que os sistemas propostos possam classificar uma *botnet*. Em 2017, [Lagraa et al. 2017] propuseram o BotGM, uma técnica de mineração de dados em grafos não supervisionado que detecta e identifica a origem de *botnets* utilizando os fluxos de rede considerando os endereços e portas de origem e destino dentro de uma janela de tempo (chamados eventos) para estabelecer correlação entre essas características. Na ocorrência de eventos atípicos na rede, estes são classificados como eventos de *bot* e portanto, quando todo um conjunto de eventos atípicos são detectados, uma *botnet* é exposta.

Neste artigo, diferente dos citados, verificamos a viabilidade do uso da técnica *Causal Graph Process* (CGP) na detecção *botnets* na rede. CGP calcula a estimativa de causalidade entre os nós da rede durante um ataque sem a necessidade de um estágio de treinamento de e/ou aprendizado sobre a rede legítima ou a *botnet*. Isso é possível pois o CGP é um modelo que permite a modelagem das relações entre elementos através de cálculos autorregresivos sobre os sinais gerados por estes elementos na forma de série temporal [Mei and Moura 2015]. Esta série temporal é composta por amostras de sinal destes elementos em um intervalo de tempo pré-determinado. O fato do CGP ser baseado em um *framework* determinístico que estima a causalidade no comportamento dos nós, torna-o diferente dos métodos comuns de classificação utilizado nos algoritmos de aprendizado de máquina que utilizam métodos autorregresivos estatísticos, fazendo com que o CGP ofereça mais informações agregadas que os demais.

3. Fundamentos

A detecção de *botnets* é assunto de grande relevância na comunidade de segurança em redes. Isso se deve aos impactos negativos causados pelos ataques gerados por uma *botnet* que podem causar perdas financeiras e até mesmo físicas na rede. Por isso a detecção de *botnets* é crucial também no gerenciamento de redes, uma vez que esta tem como objetivo garantir o funcionamento seguro da rede, além da disponibilidade de recursos e serviços. Para isso, este artigo avalia o CGP como uma forma de detectar e notificar o gerente de redes sobre os principais *hosts* (ou nós) envolvidos durante um ataque DDoS Volumétrico. Nesta seção são apresentados os principais conceitos que descrevem o ataque DDoS e o modelo avaliado na detecção de uma *botnet* geradora de ataque DDoS volumétricos.

Ataques DDoS

O Ataque Distribuído de Negação de Serviço (do inglês, *Distributed Denial of Service* - DDoS) é um tipo de ataque que visa negar serviços a usuários legítimos em uma determinada rede. Para isso, um atacante se utiliza de diversos *hosts* (ou nós) infectados distribuídos na Internet para realizar o ataque. Um ataque DDoS segue geralmente três fases principais:

1. **Infecção:** o *BotMaster* é um computador com código malicioso e rotinas para disseminação do código malicioso controlado por um atacante. Ele procura *hosts* vulneráveis que possibilitem a instalação de código malicioso (*malware*). Uma vez instalado o *malware* se comunica com o *BotMaster* e aguarda ordens.
2. **Manutenção:** os *hosts* infectados (*bots*) estão aguardando ordens do *BotMaster* e, dependendo do comportamento do *malware*, tentam infectar outros dispositivos a fim de aumentar o tamanho da *botnet.* Esse é um dos estágios mais críticos, porque alguns dispositivos que possuem sistemas de segurança mais elevados podem conseguir detectar e remover o *malware* no sistema.
3. **Ataque:** o Atacante escolhe uma vítima, normalmente um servidor na Internet, e envia um comando para a *botnet* executar um ataque. O tipo mais comum de ataque é a inundação da vítima com requisições ilegítimas a fim de esgotar os recursos da rede alvo, como largura de banda, ou recursos de um servidor, como processamento e memória.

A Figura 1 ilustra o diagrama de um ataque DDoS. Na figura, o atacante envia um comando ao *BotMaster*. Este utiliza todos os *bots* infectados sob seu controle para iniciar o ataque contra uma rede ou servidor alvo definido pelo atacante. As mensagens enviadas pelo acatante ao *BotMaster* e do *BotMaster* para os *bots* são chamadas de mensagens de Comando e Controle (C&C). Essas mensagens são transmitidas através de protocolos como o IRC (do inglês, *Internet Relay Chat*), e HTTP (do inglês, *HyperText Transfer Protocol*), que são amplamente utilizados na Internet para troca de mensagens entre aplicações de conversação, navegadores e neste caso, *bots*. A infraestrutura da *botnet* ilustrada na Figura 1 representa um modelo hierárquico da *botnet*, onde os *bots* se comunicam exclusivamente com o *BotMaster*. Quando o atacante utiliza múltiplos *hosts* infectados para inundar uma rede ou servidor com tráfego ilegítimo a fim de torná-los indisponíveis, chamamos o ataque de DDoS Volumétrico, que é um dos tipos de DDoS mais comuns e que causam maior dano, devido à dificuldade de se identificar os *hosts* que estão de fato atacando a rede ou servidor.

Causal Graph Process

O DSPg (*Discrete Signal Processing on Graphs*) é um framework determinístico proposto por Sandryhaila e Moura em 2013 utilizado em diversas áreas de processamento de dados. Dentre elas, destacam-se a Predição Linear, a Compressão de Sinal e a Classificação de Dados. Esta última é seguida neste trabalho através do modelo *Causal Graph Process (CGP)*. O CGP é um modelo que pode ser descrito como um processo autorregressivo sobre uma série temporal, no qual seus coeficientes são filtros de grafos [Sandryhaila and Moura 2012]. Com isso, o CGP evidencia as relações entre nós a partir de uma única entrada no formato de série temporal. Assume-se que o estado da entrada (sinal) no tempo k é influenciado de alguma forma pela entrada no tempo $k-1$, sendo k uma janela de tempo pré-definida no início do processo [Mei and Moura 2015].

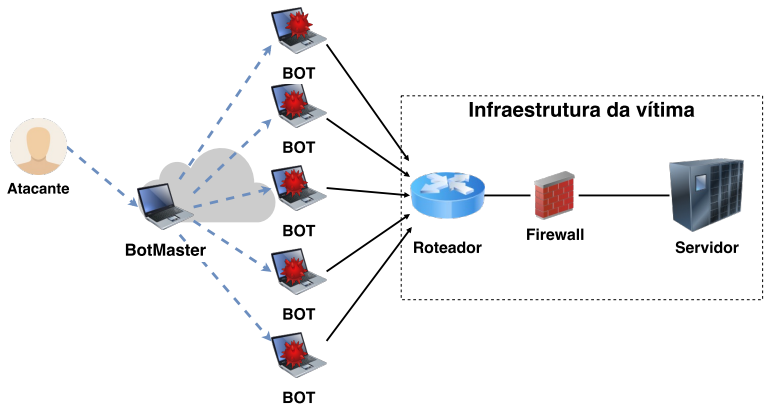

Figura 1. Arquitetura de um ataque DDoS

No contexto de redes, significa que o CGP é capaz calcular a magnitude de influências e/ou causalidade entre o comportamento dos nós da rede a partir de uma série temporal. Essa série temporal é composta pelo conjunto de sinais emitidos (dados coletados) pelos nós ao longo de uma janela de tempo. O sinal neste contexto é alguma característica da rede, por exemplo, a quantidade de pacotes ou a soma do tamanho de pacotes. A amostra temporal representa a granularidade de tempo que será utilizada para os cálculos do CGP e é configurável pelo administrador da rede assim como a janela de tempo. O tamanho da amostra temporal e da janela de tempo avaliada deve ser o mesmo para todos os *hosts* para que o CGP funcione adequadamente. Isso significa que todas as amostras temporais representam o sinal emitido por cada *host* durante o mesmo intervalo. Com isso, a série temporal apresenta a mesma quantidade de amostras de sinal para todos os nós durante a janela de tempo avaliada. Esta série temporal é a entrada do CGP, que retorna uma matriz de adjacência NxN, onde N são os nós da rede e o valor das células é a magnitude de influência de cada nó com todos os outros da rede. A seguir está um detalhamento matemático do CGP e o detalhamento de seus passos que levam ao cálculo da matriz de adjacência.

Considere $x[k]$ o resultado do CGP na seguinte forma,

$$\begin{aligned} \mathrm{x}[k] &= \mathrm{w}[k] + \sum_{i=1}^{M} Pi(\mathbf{A},\mathbf{c})\mathrm{x}[k-i] \\ &= \mathrm{w}[k] + \sum_{i=1}^{M}(\sum_{j=0}^{i} c_{ij}\mathbf{A}^j)\mathrm{x}[k-i] \\ &= \mathrm{w}[k] + (c_{10}\mathbf{I} + c_{11}\mathbf{A})\mathrm{x}[k-1] \\ &+ (c_{20}\mathbf{I} + c_{21}\mathbf{A} + c_{21}\mathbf{A}^2)\mathrm{x}[k-2] + ... \\ &+ (c_{M0}\mathbf{I} + \ ... \ + c_{MM}\mathbf{A}^M)\mathrm{x}[k-M] \end{aligned} \tag{1}$$

Onde k é uma amostra temporal, $Pi(\mathbf{A},\mathbf{c})$ é um polinômio da matriz em $\mathbf{A}$ de ordem i (isto é, $P_i(\mathbf{A},\mathbf{c})$ são filtros de grafo [Sandryhaila and Moura 2012]); $\mathrm{w}[k]$ é ruído estatístico, utilizado na avaliação precisão da auto regressão utilizada pelo CGP; c_{ij} são coeficientes polinomiais escalares, onde $\mathbf{c} = (c_{10}\ c_{11}\ ...\ c_{ij}\ ...\ c_{MM})^{(T)}$ é um vetor de todos os c_{ij}, e M é a ordem da auto regressão [Mei and Moura 2015].

Dado um grafo $G(V,\mathbf{A})$ que representa a rede com um $\mathbf{A}$ *desconhecido* (isto é, dada uma rede com fluxo de dados capturados e relação entre estes dados desconheci-

dos), para estimar a estrutura da matriz de adjacência $\mathbf{A}$, ou seja, o grau de influência entre os nós a partir do fluxo de dados capturado, e estabelecer a matriz de adjacência correspondente a essas influências), o CGP (Eq. 1) emprega três passos:

1. Resolver para $R_i = P_i(\mathbf{A}, c)$ (Aplica os filtros de grafos tendo como entrada a série temporal);
2. Recuperar a estrutura de $\mathbf{A}$ com uma das duas abordagens: usando $\hat{\mathbf{A}} = \hat{\mathbf{R}}$ como em

$$\begin{aligned}\hat{\mathbf{R}}_i = \underset{\mathbf{R}_i}{argmin}\frac{1}{2}\sum_{k=M}^{K-1}\left\|\mathrm{x}[k] - \textstyle\sum_{i=1}^{M}\mathbf{R}_i\mathrm{x}[k-j]\right\|_2^2 \\ +\lambda_1\left\|\mathrm{vec}(\mathbf{R}_1)\right\|_1 + \lambda_3\sum_{j\neq i}\left\|\left[\mathbf{R}_i, \mathbf{R}_j\right]\right\|_F^2\end{aligned} \tag{2}$$

   ou usando todos os $\hat{\mathbf{R}}_\mathbf{i}$ em conjunto para encontrar $\mathbf{A}$,

$$\begin{aligned}\hat{\mathbf{A}} = \underset{\mathbf{A}}{argmin}\left\|\hat{\mathbf{R}}_\mathbf{1} - \mathbf{A}\right\|_2^2 + \lambda_1\left\|vec(\mathbf{A})\right\|_1 \\ +\lambda_3\sum_{i=2}^{M}\left\|(\mathbf{A}, \hat{\mathbf{R}}_\mathbf{i})\right\|_F^2.\end{aligned} \tag{3}$$

3. Estimar c_{ij} em uma de duas maneiras, estimando $\mathbf{c}$ ou de $\hat{\mathbf{A}}$ e $\hat{\mathbf{R}}_\mathbf{i}$, como em

$$\hat{\mathbf{c}}_i = \underset{\mathbf{c}_i}{argmin}\frac{1}{2}\left\|\mathrm{vec}(\hat{\mathbf{R}}_i) - \mathbf{Q}_i\mathbf{c}_i\right\|_2^2 + \lambda_2\left\|\mathbf{c}_i\right\|_i \tag{4}$$

   onde

$$\mathbf{Q}_i = (\mathrm{vec}(\mathbf{I})\ \mathrm{vec}(\hat{\mathbf{A}})\ ...\ \mathrm{vec}(\hat{\mathbf{A}}^i)), \mathbf{c}_i = (c_{i0}c_{i1}\ ...\ c_{ii}). \tag{5}$$

   ou de $\hat{\mathbf{A}}$ e os dados $\mathbf{X}$ como em

$$\hat{\mathbf{c}}_i = \underset{\mathbf{c}}{argmin}\frac{1}{2}\left\|\mathbf{Y}(\hat{\mathbf{A}}) - \mathbf{B}(\hat{\mathbf{A}})\mathbf{c}\right\|_F^2 + \lambda_2\left\|\mathbf{c}\right\|_1 \tag{6}$$

O CGP se diferencia dos demais métodos de classificação por ser determinístico, isto é, o CGP encontra uma função que represente a coordenação entre os nós da rede, enquanto os demais métodos de classificação utilizados para detecção de *botnets* usam otimização gradiente, método que visa encontrar uma estimativa do valor de coordenação entre os nós da rede de modo progressivo, podendo nunca chegar no valor que represente a coordenação de fato. No contexto deste trabalho, para a matriz de influências entre os *hosts* da rede avaliada, quanto maior a influência entre os *hosts*, maior a coordenação. Quanto maior sua coordenação, maiores as chances de ser uma *botnet* [Mirkovic et al. 2002].

4. Metodologia

Nesta seção, são apresentados o método, as métricas e os parâmetros utilizados para a validação da aplicabilidade do CGP na detecção de *botnets* geradoras de ataques DDoS Volumétricos. Para isso, esta seção está organizada nas Subseções 4.1 e 4.2. A primeira descreve os *datasets* utilizados, suas características e o tratamento dos dados para a realização dos experimentos. A segunda apresenta as métricas utilizadas na avaliação do método e descreve o ambiente de testes no qual foram realizados os experimentos.

4.1. Datasets

A validação da aplicabilidade do CGP na detecção de *botnets* geradoras de DDoS foi realizada através da análise de dois *datasets*, um disponibilizado pela CAIDA (*Center for Applied Internet Data Analysis*)[Hick 2007] e outro pela CTU (*Czech Technical University*)[García et al. 2014]. Ambos *datasets* contém registros de ataques DDoS volumétricos, estão no formato PCAP (*Packet Capture*) e contém os registros anonimizados de ataques DDoS reais e simulados. O *dataset* disponibilizado pela CAIDA consiste em arquivos de captura de tráfego de rede contendo registros de um ataque DDoS volumétrico real ocorrido em agosto de 2007. Estes registros estão divididos em tipo de fluxo, sendo *to-victim* e *from-victim* (tráfego para a vítima e da vítima para a rede, respectivamente), em intervalos de cinco minutos dentro de cada tipo de fluxo. Neste *dataset* as capturas somam 21GB de registro de tráfego de rede e o tipo de *bot* utilizado na realização dos ataques está documentado. O segundo *dataset* disponibilizado pela CTU, consiste em onze capturas de distintos cenários contendo simulações de ataques variados sobre um ambiente controlado dentro da Universidade Técnica Tcheca, dentre eles três cenários apresentam simulações de ataques DDoS volumétricos. As capturas dos cenários que apresentam ataques DDoS somam 199.2GB de registro de tráfego de rede. Nesses cenários, foi utilizado o *RBot* modificado pelos autores do *dataset* a fim de evitar a propagação indevida do *bot*. O *RBot* é um *bot* escrito em Ruby que utiliza o protocolo IRC (*Internet Relay Chat*) para realizar ações diversas, dentre elas o TCP/ICMP/UDP flooding. A Tabela 1 apresenta dados complementares sobre os *datasets*.

Dataset	Duração (Horas)	Tamanho	Bot	# Bots
CAIDA	˜1	21GB	Desconhecido	Desconhecido
CTU 4	4,21	53GB	RBot	1
CTU 10	4,75	73GB	RBot	10
CTU 11	0,26	5.2GB	RBot	3

Tabela 1. Datasets Avaliados

Como mencionado, o *dataset* da CAIDA apresenta apenas registro de tráfego de rede em que a vítima é participante, ou seja, somente os registros de tráfego em que ela recebe ou envia pacotes na rede. Para padronizar estes registros nos experimentos, os cenários do *dataset* da CTU foram tratados de forma que apresentem os mesmos tipos de registro de tráfego do *dataset* da CAIDA, uma vez que estes apresentam tráfego da *botnet*, tráfego injetado e tráfego legítimo da rede onde as simulações de ataque foram realizadas. Para isso, foi utilizada a ferramenta EDITCAP tendo como entrada cada um dos cenários do *dataset* da CTU, como parâmetro o filtro de tráfego contendo o endereço da vítima que está descrito no arquivo de descrição dos cenários. A saída obtida após o uso desta ferramenta com os parâmetros mencionados é um arquivo pcap contendo apenas somente o tráfego de rede no qual a vítima participa, seja enviando ou recebendo pacotes.

4.2. Escolha dos dados e ambiente de testes

Os tratamentos e testes dos dados foram realizados em um computador com processador Intel i7-3632QM com 2MB de cache L2 e 8 GB de memória RAM. Após a padronização dos *datasets* observa-se que há uma redução no tamanho dos dados a serem analisados. Isso permite que os arquivos padronizados representem o tráfego de rede capturado por um roteador de borda ou núcleo por exemplo, indicando também os locais onde o método

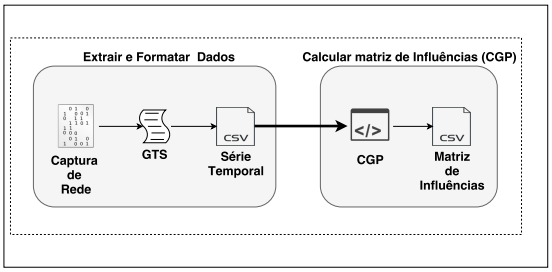

Figura 2. Processo 1 (Modelagem)

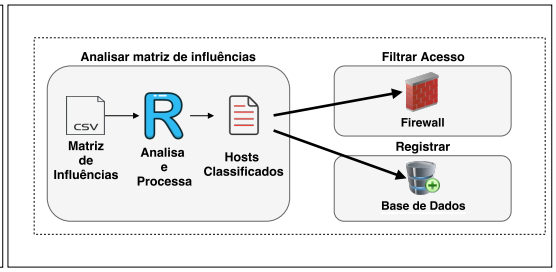

Figura 3. Processo 2 (Classificação)

pode ser implementado. Tendo em vista o objetivo deste trabalho que é avaliar o CGP como método de detecção de *botnets* geradoras de ataque DDoS volumétrico, foram selecionados apenas os cenários que apresentam mais de um *bot* para que uma *botnet* seja propriamente detectada. Com isso, o cenário 4 do *dataset* da CTU foi removido das análises por apresentar apenas um *bot*. Logo, os registros da CAIDA, CTU (cenários 10 e 11) foram selecionados para avaliação. Para melhor avaliação do CGP e seu comportamento, os *datasets* foram tratados a fim de criar uma categoria adicional. Esta categoria contém somente o tráfego recebido pela vítima para avaliar o comportamento do CGP em um cenário onde cada *host* na rede se comporta como uma engrenagem no mecanismo de defesa. Com isso, o cenário inicial que contém todo o registro de tráfego de cada *dataset* representa o comportamento de um roteador de borda ou núcleo durante um ataque.

Após o tratamento inicial dos dados, todos os cenários passam por dois processos, chamados de Modelagem e Classificação, respectivamente. O Processo 1 (Modelagem) é responsável pela formatação os dados de forma que possam ser processados pela implementação do algoritmo CGP. Este processo é composto por dois passos que consistem na extração e formatação dos dados e no calculo da matriz de influências pelo CGP. Na extração e formatação de dados, o arquivo de captura do cenário é processado por um script em shell que gera três arquivos. O primeiro arquivo consiste em três séries temporais concatenadas onde as informações registradas são a quantidade e soma de pacotes por endereço de IP de origem por unidade de tempo. A unidade de tempo escolhida é configurável via script e será detalhada na seção de avaliação. Esta série temporal é utilizada para auxiliar na escolha do intervalo de tempo a ser avaliado pelo método. Outros dois arquivos são gerados a partir desta série temporal, estes porém no formato de matriz contendo apenas a soma de pacotes ou a quantidade de pacotes respectivamente. Estes dois arquivos são utilizados como entrada da implementação do algoritmo CGP que por sua vez retorna uma matriz de influências. O Processo 2 (Classificação) é referente a análise da matriz de influências resultante do Processo 1 e possíveis medidas a serem tomadas para garantir o funcionamento da rede. A Figura 2 ilustra o Processo 1.

Após o tratamento e organização inicial descritos anteriormente, foram gerados gráficos a partir das séries temporais correspondentes a quantidade de pacotes transmitidos ao longo do tempo a fim de identificar uma janela de tempo que corresponda a ao menos um minuto antes do ataque e um minuto durante o ataque. Com isso, o estima-se identificar a eficiência do CGP na detecção de uma *botnet* em uma janela máxima de dois minutos, para avaliar a possibilidade de que este modelo possa ser empregado como uma solução *online* na detecção de *botnets* geradores de DDoS Volumétricos. Após a identificação desta janela de tempo, cada janela de dois minutos selecionada foi subdi-

vida em janelas de 30 segundos cada, a fim de simular uma redução no tempo avaliado para a detecção da *botnet*. Ao final da identificação dos janelas de tempo de interesse, foram gerados arquivos de captura individuais utilizando a ferramenta EDITCAP para cada janela de tempo de cada cenário a fim de facilitar a avaliação do CGP na detecção de *botnets*. A Figura 4 ilustra as janelas selecionadas para avaliação em cada cenário.

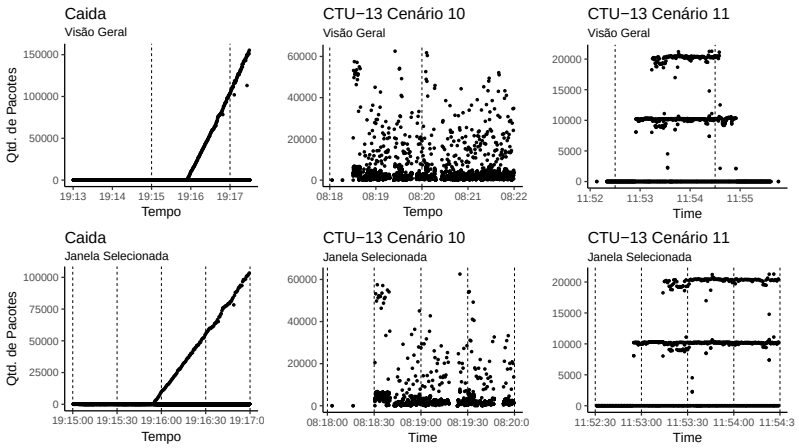

Figura 4. Janelas de Tempo Avaliadas

5. Avaliação e Resultados

Todos os arquivos de captura tratados de todos os cenários foram processados pelos processos ilustrados nas Figuras 2 e 3. A unidade de tempo escolhida como amostra dentro das séries temporais dadas como entrada no CGP é 200ms, uma vez que o tempo de ida e volta de um pacote (do inglês, *Round Trip Time - RTT*) na Internet é de 140ms [Gibson 2006]. Os resultados de cada cenário estão descritos da seguinte forma. A Subseção 5.1 apresenta os resultados obtidos após o processamento do *dataset* da CAIDA. A Subseção 5.2 apresenta os resultados obtidos após o processamento de cada um dos dois cenários avaliados correspondentes ao *dataset* da CTU-13. Para todos os cenários foram gerados gráficos que auxiliam na análise preliminar dos resultados, os gráficos foram gerados utilizando a biblioteca *igraph* disponibilizada pelo R.

5.1. Análise e resultados do dataset da CAIDA

Neste cenário foram detectadas coordenações entre 3747 *hosts*, incluindo a vítima e possíveis *bots*. Nos arquivos de descrição do *dataset* da CAIDA só existem informações concretas de quem é a vítima. Contudo, com o limiar aplicado, o método identificou 2960 *bots*. Não foi possível ilustrar o o resultado deste cenário devido a grande quantidade de *hosts* classificados como *bot*.

5.2. Análise do dataset da CTU

Em ambos os cenários da CTU, o Rbot realiza os ataques. A principal diferença entre os dois cenários da CTU avaliados é a quantidade de *bots*, sendo que o cenário 10 contém 10 *bots*, e o cenário 11, 3 bots.

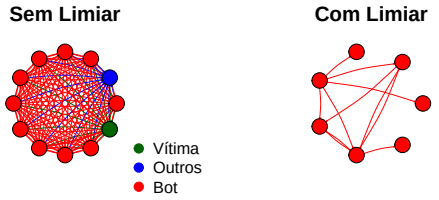

Figura 5. Resultado cenário 10

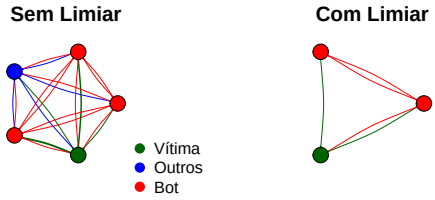

Figura 6. Resultado cenário 11

5.2.1. Cenário 10

Neste cenário os dez *hosts* infectados (*bots*) fazem parte da mesma sub-rede. Após a infeção manual desses *hosts* às 12:18:15 CEST (08:18:15 BRST) é dado início um ataque DDoS contra a vítima. Por volta de 12:31:31 CEST (08:31:31 BRST) o ataque foi interrompido manualmente pelos autores do *dataset*. Os autores mencionam que o ataque foi bem sucedido dado o fato deles não conseguirem resposta do servidor para serviços como "ssh" ou "ping". Os testes nesse cenário foram realizados utilizando a janela de tempo das 12:18:00 CEST (08:18:00 BRST) até as 12:21:59 CEST (08:21:59 BRST) como descrito em 4.2 e ilustrado na Figura 4. A Figura 5 mostra os resultados para a janela de tempo mencionada, considerando a matriz de influências com e sem o tratamento.

Neste cenário, doze dispositivos apresentaram algum tipo de correlação incluindo, um dispositivo não infectado, e dez bots. Quando aplicado o limiar citado em 4.2, somente sete *hosts* apresentam correlação, todos eles são *bots* de acordo com o arquivo de descrição do cenário, o que representa 0% de falso positivo. Após análise mais aprofundada nos arquivos de captura, constatou-se que somente esses sete *bots* estavam de fato ativos durante o ataque, os demais não apresentaram nenhuma atividade durante a captura apesar de estarem infectados.

5.2.2. Cenário 11

Neste cenário os três *hosts* infectados (*bots*) fazem parte da mesma sub-rede. Após a infeção manual desses *hosts* às 15:52:39 CEST (11:52:39 BRST) é dado inicio um ataque DDoS contra a vítima com um *bot* e por volta de 15:52:58 CEST (11:52:58 BRST) outro *bot* é adicionado ao ataque. Logo após o segundo *bot* iniciar o ataque, os autores do dataset registraram que o ataque foi bem sucedido. O ataque então foi interrompido em 15:54:44 CEST (11:54:44 BRST). Os testes nesse cenário foram realizados utilizando a janela de tempo das 15:52:30 CEST (11:52:30 BRST) até as 15:54:30 CEST (11:54:30 BRST). A Figura 6 mostra os resultados para a janela de tempo mencionada, considerando a matriz de influencias com e sem o tratamento.

Neste cenário, cinco *hosts* apresentaram algum tipo de correlação incluindo, um dispositivo não infectado, e três bots. Quando aplicado o limiar citado em 4.2, somente três *hosts* apresentam correlação, dois deles são *bots* de acordo com o arquivo de descrição do cenário e o terceiro é a vítima, o que representa 33,3% de falso positivo. Após maior análise, constatou-se que somente esses dois *bots* estavam de fato ativos durante o ataque, o outro *bot* não apresentou qualquer atividade durante a captura apesar de estar infectado.

5.3. Resultado da análise do dataset da CTU

Nos dois cenários a aplicação do CGP com foi suficiente para identificar a *botnet* nos primeiros um minto e meio de ataque. No cenário 10 foi possível identificar sete dos dez *bots* da *botnet* e no cenário 11 foi possível identificar dois dos três *bots* presentes no cenário em questão. Para o cenário 10, a taxa de falso positivo foi de 0% e de falso negativo de 30%. Para o cenário 11, a taxa de falso positivo foi de 33% e de falso negativo 33%. Ou seja, para os dados da CTU, o CGP obteve uma taxa de precisão de 68.35% na identificação dos *bots* geradores de DDoS.

5.4. Discussão

Nos três cenários avaliados, foi aplicado o CGP em cinco janelas de tempo. O primeiro compreende o intervalo de dois minutos correspondentes aos 30 segundos antes do início do ataque e aos um minuto e meio após este período. As outras quatro janelas de tempo avaliadas correspondem ao mesmo intervalo de tempo de dois minutos divididos por intervalos de 30 em 30 segundos, respectivamente. A Tabela 2 apresenta os resultados das análises realizadas nos três cenários distintos e nos cinco janelas de tempo mencionados.

Cenário	*hosts* detectados	2min	30s[1]	30s[2]	30s[3]	30s[4]	# *bots*	#FP	FN
CAIDA	3747	2960	12	187	973	?	?	?	?
CTU-13 10	12	4	4	9	0	4	10	0	6
CTU-13 11	6	3	0	2	3	3	3	1	0

Tabela 2. Bots detectados nos janelas de tempo avaliados

Na Tabela 2, quando avaliados dos três cenários os dois minutos que compreendem dos trinta segundos anteriores ao ataque até um minuto e meio após o início deles, a quantidade de *hosts* classificados como *bots* é inferior à quantidade informada nos arquivos de descrição. Contudo, quando analisados manualmente os arquivos de captura correspondentes, evidenciou-se que apesar do número inferior de *bots* detectados, estes são são de fato os únicos *bots* ativos durante o período avaliado. Isto é, são os únicos *bots* que apresentam atividade na rede durante período de tempo avaliado.

Quando avaliados nos três cenários somente os primeiros trinta segundos que correspondem aos trinta segundos antes do início do ataque, nos cenários da CAIDA e CTU-13 10 *hosts* classificados como *bots* foram detectados em quantidade inferior ao descrito nos arquivos de descrição, isso considerando que somente o cenário CTU-13 10 apresenta informações sobre a quantidade de *bots* existentes. Nos primeiros trinta segundos correspondentes ao inicio dos ataques, a quantidade de *bots* detectados foi próximo ao descrito nos cenários selecionados. Neste intervalo, quando feitas análises aprofundadas nos três cenários, evidenciou-se que todos os dispositivos classificados como *bots* eram de fato *bots* de acordo com seus respectivos arquivos de descrição.

Nos outros dois trinta segundos subsequentes para o dataset da CAIDA, o número de *bots* detectados teve um aumento de 420%. Como não se tem informações de quem é de fato *bot* neste cenário, assumimos que estes são também *bots* reais. Para os outros dois cenários, a quantidade de *bots* detectados variou de entre zero e quatro, onde no cenário CTU-13 10, a quantidade de *bots* detectados foi de zero e quatro respectivamente. No cenário CTU-13 11 foi de três em ambos os intervalos. Ao analisar com maior profundidade estes dois intervalos subsequentes nos dois cenários, constatou-se que para o cenário

CTU-13 10 os quatro *bots* detectados são de fato *bots* da botnet, já no cenário CTU-13 11, somente dois dos três *hosts* classificados como *bots*, o são de fato.

Utilizando a quantidade de falsos positivos (FP) e falsos negativos (FN), obteve-se para o cenário CTU-13 10 uma taxa de 0% de falso positivo e 60% de falso negativo; para o cenário CTU-13 11, uma taxa de 33% de falso positivo e 0% de falso negativo. Para o dataset da CAIDA, por não apresentar informações sobre a quantidade e identificação dos *bots*, não foi possível estabelecer as taxas de falso positivo e falso negativo. Contudo, assumindo uma média de 16% de falso positivo, considerando uma média dos outros cenários, pode-se considerar que pelo menos 2487 *hosts* deste cenário são verdadeiramente *bots*. Isto representa 66% de todos os *hosts* detectados nessa janela de tempo.

6. Conclusão

Este trabalho teve como objetivo a verificação da aplicabilidade do uso do CGP na identificação de *botnets* geradoras de DDoS volumétrico. Para isso foram utilizados dois *datasets* distintos contendo cenários diferentes de ataques DDoS volumétricos. Após os tratamentos necessários para a avaliação dos *datasets* por uma implementação do CGP, constatou-se que é possível utilizar o CGP na detecção de *botnets* geradoras de ataques DDoS volumétrico em um intervalo de até dois minutos. Foi observado também que quando não utilizado o limiar calculado pelo método dos quartis, a estrutura da rede é estimada. Isso indica que o CGP pode ser utilizado também como um método passivo de varredura de redes. Trabalhos futuros vão focar nos ajustes finos do CGP na detecção de *botnets* e em um comparativo entre o CGP e outros métodos de detecção de *botnets* existentes na literatura.

Referências

Chowdhury, S., Khanzadeh, M., Akula, R., Zhang, F., Zhang, S., Medal, H., Marufuzzaman, M., and Bian, L. (2017). Botnet detection using graph-based feature clustering. *Journal of Big Data*, Vol. 4:14.

Feily, M., Shahrestani, A., and Ramadass, S. (2009). A survey of botnet and botnet detection. In *Intern. Conf. on Emerging Security Inform., Systems and Technologies*, pages 268–273.

García, S., Grill, M., Stiborek, J., and Zunino, A. (2014). An empirical comparison of botnet detection methods. *Computers & Security*, 45:100 – 123.

Gibson, B. (2006). TCP limitations on file transfer hamper the global Internet. `http://www.niwotnetworks.com/gbx/TCPLimitsFastFileTransfer.htm`. último acesso: 19/03/2018.

Gu, G., Perdisci, R., Zhang, J., and Lee, W. (2008). Botminer: Clustering analysis of network traffic for protocol- and structure-independent botnet detection. In *Conf. on Security Symposium*, pages 139–154.

Harris, L. (2010). The botnet challenge. `http://corporate.comcast.com/comcast-voices/the-botnet-challenge`. último acesso: 19/03/2018.

Hick, P. (2007). The CAIDA DDoS attack 2007 Dataset. `http://www.caida.org/data/passive/ddos-20070804\_dataset.xml`. último acesso: 19/03/2018.

Joshi, H. P., Bennison, M., and Dutta, R. (2017). Collaborative botnet detection with partial communication graph information. In *IEEE Sarnoff Symposium*, pages 1–6.

Kalaivani, P. and Vijaya, M. (2016). Mining based detection of botnet traffic in network flow. In *International Journal of computer Science and information Technology & Security.*

Khajuria, A. and Srivastava, R. (2013). Analysis of the DDoS defence stratigies in Cloud Computing. *Intern. Journal of Enhanced Research in Management & Computer Applications*, Vol. 2.

Kong, X., Chen, Y., Tian, H., Wang, T., Cai, Y., and Chen, X. (2016). A novel botnet detection method based on preprocessing data packet by graph structure clustering. In *Intern. Conf. on Cyber-Enabled Distributed Comp. and Knowledge Discovery*, pages 42–45.

Lagraa, S., Francois, J., Lahmadi, A., Miner, M., Hammerschmidt, C., and State, R. (2017). BotGM: Unsupervised Graph Mining to Detect Botnets in Traffic Flows. In *1st Cyber Security in Networking Conference*, Rio de Janeiro, Brazil.

Masud, M. M., Al-khateeb, T., Khan, L., Thuraisingham, B., and Hamlen, K. W. (2008). Flow-based identification of botnet traffic by mining multiple log files. In *Intern. Conf. on Distributed Framework and Applications*, pages 200–206.

Mei, J. and Moura, J. M. F. (2015). Signal processing on graphs: Modeling (causal) relations in big data. *CoRR*, abs/1503.00173.

Miller, S. and Busby-Earle, C. (2016). The role of machine learning in botnet detection. In *11th International Conference for Internet Technology and Secured Transactions (ICITST)*, pages 359–364.

Mirkovic, J., Prier, G., and Reiher, P. (2002). Attacking ddos at the source. In *IEEE Intern. Conf. on Network Protocols*, pages 312–321.

Sandryhaila, A. and Moura, J. M. F. (2012). Discrete signal processing on graphs. *CoRR*, abs/1210.4752.

Sapello, A., Serban, C., Chadha, R., and Izmailov, R. (2017). Application of learning using privileged information(lupi): Botnet detection. In *Intern. Conf. on Computer Comm. and Networks*, pages 1–8.

Su, Y. H., Rezapour, A., and Tzeng, W. G. (2017). The forward-backward string: A new robust feature for botnet detection. In *IEEE Conference on Dependable and Secure Computing*, pages 485–492.

Wang, J. and Paschalidis, I. C. (2017). Botnet detection based on anomaly and community detection. *IEEE Transactions on Control of Network Systems*, 4(2):392–404.

Zeidanloo, H. R., Shooshtari, M. J. Z., Amoli, P. V., Safari, M., and Zamani, M. (2010). A taxonomy of botnet detection techniques. In *Intern. Conf. on Computer Science and Information Technology*, volume 2, pages 158–162.