UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DACOM - DEPARTAMENTO ACADÊMICO DE COMPUTAÇÃO
ENGENHARIA DE COMPUTAÇÃO

BRUNO ANKEN MOROMIZATO ZANINELLO

ANOMALIAS EM REDES DE COMPUTADORES: EXTRAÇÃO E ANÁLISE DE DADOS E INJEÇÃO DE ANOMALIAS

TRABALHO DE CONCLUSÃO DE CURSO 2

CORNÉLIO PROCÓPIO
2019

BRUNO ANKEN MOROMIZATO ZANINELLO

ANOMALIAS EM REDES DE COMPUTADORES: EXTRAÇÃO E ANÁLISE DE DADOS E INJEÇÃO DE ANOMALIAS

Trabalho de Conclusão de Curso 2 apresentado ao curso de Engenharia de Computação da Universidade Tecnológica Federal do Paraná, como requisito parcial para a obtenção do título de Bacharel.

Orientador: Prof. Dr. Lucas Dias Hiera Sampaio
Universidade Tecnológica Federal do Paraná

CORNÉLIO PROCÓPIO
2019

Dedico este trabalho à minha família, que sempre me apoiou de diversas formas em toda a minha trajetória vida.

AGRADECIMENTOS

Agradeço aos meus amigos, sem os quais minha caminhada até este momento certamente teria sido mais árdua, ao meu professor orientador, que me ajudou, apoiou e orientou de maneira impecável neste trabalho e, principalmente, à minha família, sem a qual teria sido impossível chegar até este momento.

Navigare necesse; vivere non est necesse.
(MAGNUS, Cnaeus Pompeius, I. A.C.).

RESUMO

ZANINELLO, Bruno Ankem Moromizato. Anomalias em Redes de Computadores: Extração e Análise de Dados e Injeção de Anomalias. 2019. 34 f. Trabalho de Conclusão de Curso 2 – Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2019.

A segurança em redes de computadores é uma área que movimenta grande quantidade de capital no mercado, desde gastos com reparos de danos causados por ataques efetuados a investimentos em proteções para que os ataques não venham a se concretizar. Os estudos acerca de sistemas de detecção de anomalias iniciaram-se em meados da década de 1980 e foram avançando juntamente às novas tendências tecnológicas na área. Atualmente existem sistemas de detecção de anomalias com arquiteturas distribuídas e atuando na nuvem, por exemplo. Este trabalho propõe métodos para a extração de dados de interesse de uso de uma rede de computadores e cálculo do perfil de uso destes dados para, posteriormente, simular um ataque através da injeção de anomalias nos dados extraídos e comparar estes dados com os dados de perfil da rede.
Palavras-chave: Injeção de anomalias. Redes de computadores. Segurança.

ABSTRACT

ZANINELLO, Bruno Ankem Moromizato. Anomalies in Computer Networks: Data Analysis and Extraction and Anomalies Injection. 2019. 34 f. Trabalho de Conclusão de Curso 2 – Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2019.

Computer network security is an area that involves a great amount of money, from damage repairing costs of effective attacks to investments on protection against attacks to come. Studies about anomaly detection systems began in the mid 1980s and advanced alongside the new technological trends in the area. There are, nowadays, anomaly detection systems with distributed architecture and cloud acting ones, for example. This work's purpose is to create methods to extract relevant use data from a computer network and trace its use profile to afterwards simulate an attack by injecting anomalies into the extracted data and compare this new data to the network profile.

Keywords: Anomaly injection. Computers network. Security.

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

O número de dispositivos conectados à internet vem crescendo fortemente. Desde tecnologias conhecidas que comumente são conectadas a uma rede, como computadores, notebooks, celulares e tablets até as mais novas invenções e tendências da Internet das Coisas, que adiciona televisões, geladeiras e sensores à rede mundial. A quantidade de dispositivos conectados à redes vai ser mais de três vezes a população global até 2022 (CISCO, 2019).

Com o aumento de dispositivos sendo conectados à internet e a crescente disponibilização de novos serviços multimídia o tráfego das redes de computadores conectadas à internet vai aumentar em três vezes entre 2017 e 2022 (CISCO, 2019). Junto a isso também aumentam as vulnerabilidades e possibilidade de ataques às redes, sejam elas de uso doméstico ou redes cooporativas, visto que a segurança de um sistema é inversamente proporcional à sua comodidade de uso (PRIYAMBODO; PRAYUDI, 2015).

De acordo com BOUÇAS (2016) o gasto com produtos e serviços de segurança da informação seria de 81,6 bilhões de dólares em 2016. Já em 2017, de acordo com (BRADLEY, 2017) este mesmo gasto seria de 86,4 bilhões de dólares e chegaria a 93 bilhões de dólares em 2018.

Estima-se que, em 2017, 445 bilhões de dólares serão gastos anualmente por conta de cibercrimes no mundo, com expectativas de que este valor aumente para 2,1 trilhões de dólares até 2019 (COMPUTERWORLD, 2017). Acredita-se que, só no Brasil, o gasto com crimes virtuais foi de R$35 bilhões em 2016 (WALTRICK, 2016).

Na última década diferentes ataques estamparam a capa dos principais jornais do mundo. Dentre eles podemos citar o caso do *ransomware Wannacry* o qual afetou o atendimento do Instituto Nacional de Segurança Social (INSS) e atingiu empresas e órgãos públicos no Distrito Federal e mais quatorze estados brasileiros (G1, 2017), além de ter se alastrado por cerca de 150 países (CEBRIÁN, 2017), causando diversos danos a governos e empresas por todo o mundo.

Tendo em vista o grande impacto econômico da área de segurança redes de computadores no mundo percebe-se que o estudo e implementação de técnicas, métodos e sistemas de defesa e prevenção de ataques e infecções tornam-se uma necessidade.

Este trabalho apresenta uma metodologia para a extração de dados de fluxo de rede no formato do padrão NetFlow, o cálculo das entropias destes dados em diferentes intervalos de tempo, geração de um baseline para os dados utilizando a heurística PSO para minimizar os custos da função do baseline, criação de um software para injeção de anomalias nos dados extraídos da rede e a comparação das entropias dos dados com anomalias com os baselines da rede.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a fundamentação teórica do trabalho e nele serão discutidos o histórico dos sistemas de detecção de anomalias, os trabalhos relacionado, os conceitos matemáticos e os tipos de ataque abordados neste trabalho.

2.1 Histórico

As pesquisas na área de detecção de anomalias se iniciaram em meados da década de 1980. É evidente que a internet como é conhecida hoje não existia neste período, e os meios de comunicação, protocolos e tecnologias que a originariam ainda estavam sendo propostos.

Um dos primeiros trabalhos nesta área foi de James P. Anderson. Em seu trabalho, Anderson diz que as informações colhidas por auditorias de sistemas na época não eram suficientes para que os profissionais que lidavam com a segurança dos sistemas conseguissem agir da maneira adequada às ameaças presentes. Anderson então propõe aumentar a quantidade de informações coletadas utilizando-se de ferramentas e programas para realizar tal coleta (ANDERSON, 1980).

Estes dados a serem recolhidos seriam frutos de um estudo de como os dados de técnicas de auditoria se manifestam e indicam a ocorrência de ataques ou ameaças.

Até então, os meios mais comuns de se detectar algum tipo de intrusão eram através da leitura de logs de auditorias (KEMMERER; VIGNA, 2002). Esta técnica era pouco efetiva, visto que todos os logs eram analisados um a um pelos administradores de sistemas em busca de indícios de violações no sistema.

Em 1986 Dorothy E. Denning publicou um artigo no qual descreve um modelo para detecção de intrusões em tempo real, tanto de tentativas de ataques externos quanto internos. Dorothy baseou seu modelo na premissa de que o uso anormal de um sistema pode indicar a exploração de vulnerabilidades (DENNING, 1987).

Como o modelo proposto por Denning é independente da plataforma, ambiente ou qualquer sistema ou ameaças específicos, o mesmo tornou-se um framework de propósito geral que, de acordo com McHugh, Christie e Allen (2000), inspirou diversos pesquisadores e criou a base para os sistemas de detecção de intrusão (do inglês, *Intrusion Detection System*, IDS) que viriam a ser desenvolvidos nos anos seguintes.

Ao longo do final da década de 1980 e no decorrer da década de 1990, uma máxima continuou verdadeira: nenhum computador ou sistema é livre de vulnerabilidades. Os sistemas podem sofrer tanto ataques internos, como usuários abusando de seus privilégios dentro do sistema, quanto ataques externos, onde usuários não autorizados tentam penetrar no sistema.

Em meados da década de 1990 existiam 5 tipos comuns de IDS: threshold detection, anomaly detection, rule-based penetration identification, model-based intrusion detection e

intrusion prevention (ILGUN; KEMMERER; PORRAS, 1995).

A técnica de detecção por limiares (do inglês, *threshold detection*) é a mais rudimentar das cinco, uma vez que cada ocorrência de um determinado evento é gravada e é realizada a análise da quantidade de suas ocorrências dentro de um determinado período de tempo. Se estas ocorrências ultrapassarem um certo limite no tempo estabelecido, isto pode indicar a ocorrência de uma intrusão no sistema (ILGUN; KEMMERER; PORRAS, 1995).

Já a técnica de detecção e anomalia (do inglês, *anomaly detection*) estabelece padrões de uso para cada usuário do sistema. Se o resultado de alguma auditoria de uso do sistema apontar resultados diferentes do padrão esperado de algum usuário, é possível que tenha ocorrido alguma intrusão no sistema (ILGUN; KEMMERER; PORRAS, 1995).

Um sistema de identificação de penetração baseado em regras (do inglês, *Rule-Based Penetration Identification* é capaz de identificar alguma ameaça ao sistema a partir de uma única auditoria de um evento. Ele também é capaze de encontrar indicativos de penetração a partir de uma sequência de eventos suspeitos (ILGUN; KEMMERER; PORRAS, 1995).

A detecção de intrusão baseada em modelo (do inglês, *Model-Based Intrusion Detection*) tem como objetivo modelar cenários que apresentem comportamentos característicos de uma intrusão. Desta maneira os administradores criam cenaŕios de penetração de maneira abstrata e entregam toda a responsabilidade de determinar quais resultados de auditorias são suspeitos para um software capaz de realizar as verificações necessárias de forma automatizada (ILGUN; KEMMERER; PORRAS, 1995).

Já um sistema que implementa a técnica de prevenção de intrusão (do inglês, *Intrusion-Prevention*) traz utilidades para o administrador tais como um conjunto de ferramentas que auxilia na busca por vulnerabilidades comumente exploradas por atacantes presentes nas configurações do sistema ou uma abordagem que evite a execução de vírus de computadores ou Cavalos de Tróia dentro do sistema (ILGUN; KEMMERER; PORRAS, 1995).

No começo dos anos 2000 não existiam sistemas de detecção de intrusão em tempo real robustos o suficiente para detectar ataques avançados de atacantes bem treinados (BASS, 2000). Um dos defeitos dos mesmos era a grande taxa de falso positivos, que acarretava em grandes perdas financeiras para as empresas que implementavam tais sistemas. Este problema persiste até hoje, conforme relatado por Wagner Rodrigues em sua palestra "Desconstruindo Casos de (in)Segurança da Informação: três décadas... ainda uma jornada".

Outro problema era o gerenciamento de redes, que muitas vezes falhava em prover informações úteis ou relevantes aos profissionais envolvidos na administração ou segurança de redes e sistemas. O gerenciamento da rede e os sistemas de detecção devem trabalhar em conjunto para que os dados possam ser transformados em informações úteis de tal forma que o estado da rede possa ser claramente definido e ações corretas e objetivas possam ser tomadas de acordo com cada cenário (BASS, 2000).

De maneira geral, todos os IDS podem ser classificados de acordo com sua premissa em duas categorias: baseado em anomalia (do inglês, *anomaly-based*) e baseado em assinatura

(do inglês, *signature-based*).

Um IDS baseado em anomalia coleta grande quantidade de dados de logs de uso do sistema para traçar perfis de comportamento normais de usuários e atividades do sistema. Baseado nestes perfis de comportamento, o IDS monitora o sistema em busca de desvios dos padrões comportamentais entre os usuários (BOUGHACI et al., 2006).

Já um IDS baseado em assinatura utiliza-se de uma base de dados de ataques já conhecidos e estudados para comparar o comportamento do sistema com o desta base de dados em busca de possíveis anomalias (YANG et al., 2010).

Os IDS também podem ser classificados quanto ao ambiente no qual detectam intrusões: sistema de detecção de intrusão baseado em um hospedeiro (do inglês, *host-based intrusion detection system*, HIDS) ou sistema de detecção de intrusão baseado em rede (do inglês, *network-based intrusion detection system*, NIDS).

O HIDS atua exclusivamente na máquina em que está instalado, trabalhando com grande proximidade ao sistema operacional da máquina e coletando informações, tais como dados de auditorias ou logs de atividades, utilizadas para identificar possíveis intrusões na máquina de acordo com o uso da mesma (DURST et al., 1999).

Sistemas HIDS conseguem monitorar aplicações específicas nas máquinas em que atuam, algo difícil ou até mesmo impossível em sistemas NIDS. Porém, existe um custo de desempenho na máquina afetada pelo HIDS, já que seus recursos devem ser utilizados para rodar o HIDS além de realizar sua carga de trabalho (ZHANG et al., 2002).

Já os sistemas NIDS são responsáveis por monitorar atividades de um segmento de rede ou até mesmo da rede inteira, apesar de serem instalados em um único host, assim com o HIDS. Eles analisam o tráfego de pacotes entre os hosts procurando identificar comportamento anormal no formato e dados dos pacotes (DURST et al., 1999).

Sistemas NIDS conseguem monitorar diversos hosts simultaneamente, porém nenhum de maneira aprofundada, e tendem a sofrer problemas de performance, principalmente com grandes velocidades de comunicação na rede. No entanto, a instalação e manutenção de um NIDS é, geralmente, simples e acarreta em custos computacionais quase nulos sobre as máquinas em que atuam (MCHUGH; CHRISTIE; ALLEN, 2000).

Com o avanço e popularização de novas tecnologias e arquiteturas de redes surgiu um novo modelo de sistema de detecção de intrusões: o sistema de detecção de intrusão distribuído (do inglês, *distributed intrusion detection system*, DIDS). Este sistema analisa as atividades de diversos hosts na rede, sejam estas atividades específicas de cada host ou de segmentos de redes, e faz uma agregação dos mesmos (KANNADIGA; ZULKERNINE, 2005).

A análise dos dados isolados de um único host pode não ser suspeita o suficiente para gerar um alerta dos sistemas HIDS ou NIDS. Porém, analisadas em conjunto, as atividades podem ser suficientemente anômalas para disparar um alerta de um sistema DIDS

2.2 Trabalhos Relacionados

Com regularidade ocorre o surgimento de novas tecnologias e as já existentes estão sempre em um processo de avanços e melhorias. Com o surgimento e evolução de diversas arquiteturas de sistemas, paradigmas computacionais e técnicas de ataques diversas, o mesmo deve ocorrer com as tecnologias de proteção de sistemas.

Em 2005 foi proposta uma ferramenta que se utiliza de aprendizado de máquina para a criação de uma base de dados que não necessita da intervenção humana necessária em um IDS baseados em assinatura (SHON et al., 2005).

Essa ferramenta utiliza a técnica de Algoritmo Genético para escolher os campos mais apropriados do pacote para utilizar nas análises. Estes campos são então refinados e passam por um filtro para aumentar a performance da próxima etapa, na qual os dados são enviados para uma versão aprimorada de Support Vector Machine (SVM), um algoritmo que utiliza dois métodos de aprendizado de máquina, um supervisionado e outro não-supervisionado, para classificar os pacotes recebidos como anômalos ou normais.

Em (HWANG et al., 2007) é proposto um NIDS híbrido, baseado tanto em assinatura quanto em anomalia. Desta maneira os autores conseguiram detecções com maior taxa de acurácia e menos alarmes falsos, combinando a baixa taxa de falsos-positivos de um IDS baseado em assinatura com a habilidade de um sistema de detecção de anomalias em detectar novos tipos de ataques.

Já a proposta do trabalho de (YANG et al., 2010) também é de um sistema híbrido, baseado em assinatura e com detecção de anomalia, visando a descoberta de novos ataques e mantendo uma boa taxa de detecção, e adiciona uma Árvore de Decisão, que é uma tabela de predição comumente usada na área de mineração de dados.

O modelo baseado em assinatura identifica o tipo de protocolo da instância de dados e então escolhe o algoritmo de árvore de decisão mais eficiente para realizar a rotina de detecção. No modelo baseado em detecção de anomalia, as três árvores de decisão que o estudo utiliza são testadas nas instâncias de dados e a mais eficiente é escolhida. Uma instância só é considerada como intrusa apenas quando ambos os modelos a identificarem como uma intrusão.

O trabalho (GUL; HUSSAIN, 2011) propõe um modelo de sistema multi-thread distribuído aplicado à nuvem, alegando que os IDS tradicionais não se adequam de maneira eficiente a um ambiente em nuvem, o qual está sujeito a diversas ameaças de segurança e vulnerabilidades por conta, entre outros fatores, de sua arquitetura distribuída.

Outro fator que impede a implantação na nuvem de IDS tradicionais é que os mesmos não conseguem manipular de maneira eficaz a quantidade massiva de tráfego presente na nuvem, visto que a maior parte dos IDS funciona em uma única thread.

O sistema proposto é distribuído, pois atua tanto como um NIDS quanto como um HIDS simultaneamente, transparente com o usuário e otimizado, visto que envia alertas para os usuáros e entrega informações específicas ao provedor do serviço em nuvem.

Em (MITCHELL; CHEN, 2015) é proposto um sistema de detecção de intrusão

baseado em especificação que se utiliza de regras de comportamento esperado de aparelhos em um sistema ciber-físico médico (do inglês, *medical cyber-physical system*, MCPS), onde a segurança do paciente é de extrema importância.

Um IDS baseado em especificação cria uma base através de especificações do programa que descrevem qual o comportamento esperado do programa. O sistema então monitora os programas em execução em busca de desvios de comportamento das especificações. Desta maneira, ataques podem ser detectados mesmo se não houverem detecções dos mesmos anteriormente.

Uma grande diferença entre a modelagem de um IDS aplicado a um MCPS é a relação bem próxima que existe entre as detecções de intrusões e os componentes físicos do sistema médico.

Portanto, ao invés de investigar as rotas dos pacotes ou perdas dos mesmos procurando comportamento anômalo de comunicação, deve-se testar os dados colhidos pelos sensores médicos à procura de manifestações físicas de comportamento anômalo. As regras de comportamento são transformadas em máquina de estados para que uma máquina sob monitoramento possa ter suas transformações de estado facilmente comparadas ao comportamento esperado de tal máquina.

Em (HAMAMOTO et al., 2018) é proposto um sistema que combina Algoritmo Genético à Lógica Fuzzy para detecção de anomalias em redes. O Algoritmo Genético é utilizado para gerar uma assinatura digital de segmento de rede (do inglês, *Digital Signature of Network Segment*, DSNS), que estima o comportamento esperado da rede. Um sistema que se utiliza de Lógica Fuzzy é, então, aplicado ao DSNS para determinar se uma instância representa uma anomalia ou não.

As técnicas de Algoritmo Genético e Lógica Fuzzy são adequadas para lidar com problemas que incluem incertezas, como é o caso de redes de computadores. O sistema proposto funciona de maneira autônoma, aplicando um método padrão aos dados coletados da rede, sem rotulá-los, o que implica em uma técnica de treinamento não-supervisionada.

Em (ASSIS; JR., 2015) propuseram uma ferramenta de simulação de ataques e anomalias em uma rede de computadores através da injeção de dados anômalos em uma base de dados coletados da rede. Desta maneira a ferramente não atinge a performance da rede e proporciona grande flexibilidade de análise para os responsáveis pela rede.

2.3 Conceitos e Definições

Esta seção apresenta as motivações e explicações dos conceitos e métodos matemáticos utilizados pelo trabalho.

2.3.1 Baseline (DSNS)

A análise dos dados isolados de um único host pode não ser suspeita o suficiente para gerar um alerta dos sistemas HIDS ou NIDS. Porém, analisadas em conjunto, as atividades podem ser suficientemente anômalas para disparar um alerta de um sistema DIDS. A caracterização de tráfego de uma rede permite a modelagem de padrões de comportamento em um segmento de rede em relação ao tempo (Jr; ZARPELÃO; MENDES, 2005). Este padrão é conhecido como baseline ou assinatura digital de segmento de rede (do inglês, *digital signature of network segment*, DSNS).

Pode-se definir o DSNS como um conjunto de informações que mostram o perfil de tráfego em um segmento de rede por meio de uma análise quantitativa de características da rede coomo média de bits por segundo, pacotes por segundo, entropia do IP de destino, entropia do IP de origem, entropia da porta de destino e entropia da porta de origem, por exemplo.

Um DSNS pode ser criado a partir da utilização de diversos protocolos, como em Garg et al. (1998), onde foi utilizado o Simple Network Management Protocol (SNMP), que oferece serviços de gerenciamento de redes, como quantidade de dados trafegados, ou, mais usual atualmente, através de protocolos como o Netflow (CISCO, 2011), um serviço disponível em diversos produtos da companhia Cisco, o IP Flow Information Export (IPFIX) (CLAISE; TRAMMELL; AITKEN, 2013), um protocolo da IETF utilizado para transmitir informações de fluxo de tráfego pela rede e o sFlow (SFLOW.ORG, 2017), uma tecnologia padrão da indústria utilizada para medir o tráfego da rede e coletar, armazenar e analisar os dados advindos desta mensuração.

2.3.2 Heurística

De acordo com (PEARL, 1984), heurísticas são critérios, métodos ou princípios para decidir qual dentre diversas alternativas tende a ser a mais eficiente para alcançar um determinado objetivo, como o Algoritmo Genético, por exemplo. Atualmente, diversas técnicas de heurísticas são aplicadas nos campos de mineração de dados, aprendizado de máquina e inteligência artificia. Neste trabalho a heurística Particle Swarm Optimization (PSO) foi utilizada para a criação do baseline.

2.3.2.1 Otimização

O termo otimização se refere ao processo de identificar o melhor elemento dentre um conjunto de alternativas a partir de critérios pré-definidos. Em termos matemáticos podemos descrever como encontrar os parâmetros da função parametrizável f que a maximizem ou minimizem (MARINI; WALCZAK, 2015). Realizar a minimização de f nada mais é que maximizar $-f$.

Um problema de otimização tem a forma de minimizar $f_0(x)$ sujeita a

$$f_i(x) \leq b_i, i = 1,...,m. \tag{1}$$

Aqui temos o vetor $x = (x_1,...,x_n)$ como a variável de otimização do problema, a função $f_0 : R^n \rightarrow R$, $i = 1,...,m$ são as funções de restrição e as constantes $b_1,...,b_m$ são os limites das restrições. Um vetor $x*$ é dito ótimo e a solução do problema se ele possuir o menor valor objetivo dentre todos os vetores que satisfaça as restrições: para qualquer z com $f_1(z) \leq b_1,...,f_m(z) \leq b_m$, temos que $f_0 \geq f_0(x*)$ (BOYD; VANDENBERGHE, 2009).

2.3.3 Particle Swarm Optimization (PSO)

O PSO é um algoritmo de otimização onde cada candidato a solução é chamado de partícula e representa um ponto num espaço multidimensional (KENNEDY; EBERHART, 1995). Seja N o número de partículas na heurística (chamamos isso de população) com N candidatos a solução pode ser descrita como o conjunto C onde:

$$c = \{\mathbf{C_1}, \mathbf{C_2}, \mathbf{C_3},..., \mathbf{C_N}\} \tag{2}$$

Na busca pela solução ótima do problema as partículas se movimentam pelo espaço. Com t e $t+1$ indicando duas iterações sucessivas do algoritmo e sendo v_i o vetor que armazena os componentes de velocidade da $i$ésima partícula ao longo das D dimensões, a equação da trajetória de cada partícula pode ser descrita como:

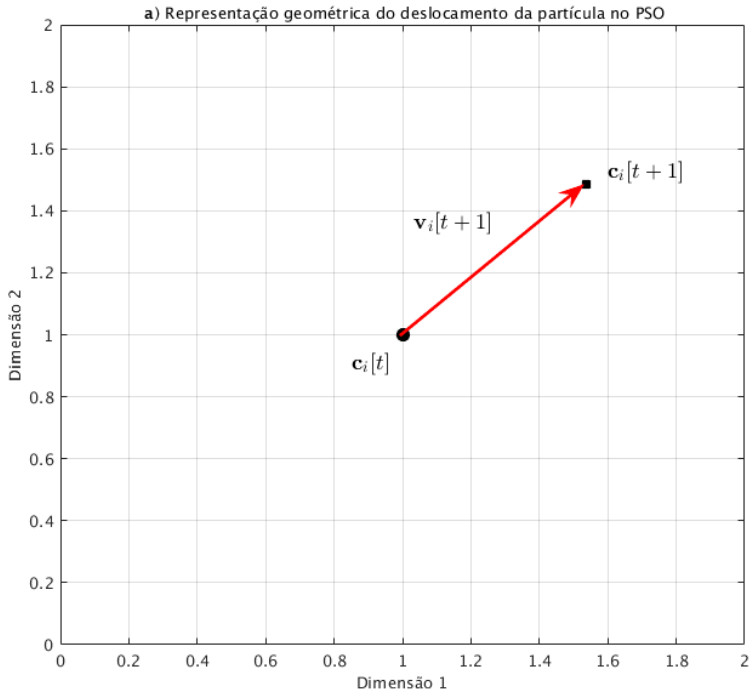
$$\mathbf{C_i}(t+1) = \mathbf{C_i}(t) + \mathbf{V_i}(t+1) \tag{3}$$

O vetor de velocidades v é composto por três diferentes componentes, sendo eles: a inércia, que impede a partícula de mudar de direção drasticamente ao manter o histórico dos fluxos de direção anteriores, o componente cognitivo, que representa a tendência das partículas de retornarem às suas melhores posições encontradas até então e, por último, o componente social, que identifica o quantoa partícula está propensa a se mover em direção à melhor posição global. Portanto podemos definir a velocidade da $i$ésima partícula como:

$$\mathbf{V_i}(t+1) = \omega \mathbf{V_i}(t) + A \times f_1 \times (\mathbf{C_i} - \mathbf{C_i^*}) + B \times f_2 \times (\mathbf{C_i} - \mathbf{C^*}) \tag{4}$$

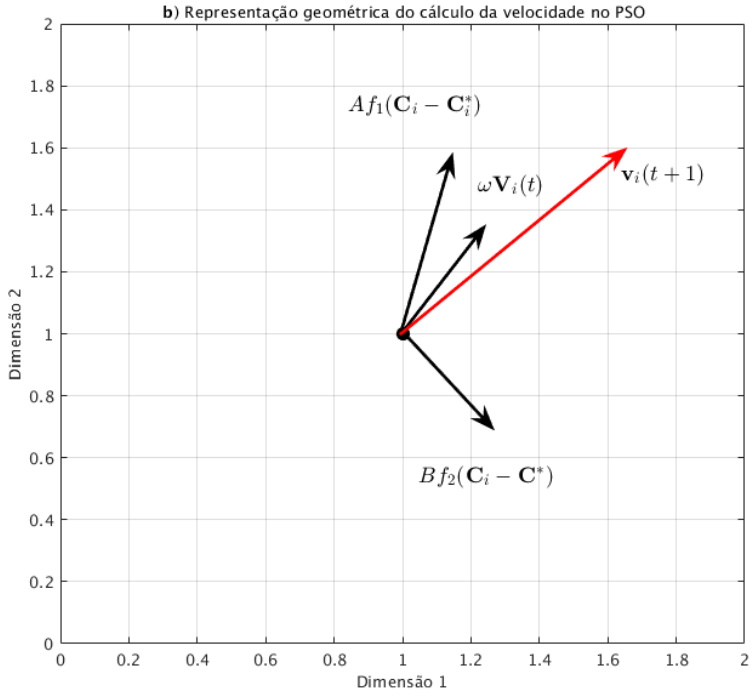
onde c_i^* é a melhor solução encontrada para a partícula i até o momento e c^* é o melhor global, ou seja, a melhor solução encontrada dentre todas as partículas. As constantes de aceleração f_1 e f_2 são valores tais que $0 \leq f_1, f_2 \leq 1$ e são chamados de coeficiente cognitivo e coeficiente social, respectivamente. Eles servem para controlar a aproximaçãoda partícula em direção ao seu melhor pessoal e seu melhor global. A e B são variáveis aleatórias a serem descobertas e ω é a constante de inércia da velocidade.

Figura 1 – Representação gráfica da posição de uma partícula quando aplicado o PSO



Fonte: Autoria própria

Figura 2 – Representação gráfica da velocidade de uma partícula quando aplicado o PSO



Fonte: Autoria própria

2.3.4 Entropia de Shannon

A Entropia de Shannon, também conhecida como entropia da informação, é utilizada para medir o grau de desordem de algum dado. Quanto maior o valor da entropia maior a desordem ou heterogeneidade presente no conjunto de dados utilizado para o cálculo da entropia. Em contrapartida, quanto menor seu valor maior a ordem e a homogeneidade do conjunto de dados (SHANNON, 1948). A entropia H é dada pela seguinte fórmula:

$$H = \sum_{i=1}^{n} p_i \, log \, p_i \tag{5}$$

onde p_i é a probabilidade de ocorrência de i e é dada por

$$p_i = M(i)/n \tag{6}$$

e $M(i)$ é a quantidade de ocorrências de i dentro do espaço amostral.

2.4 Tipos de Ataques

Esta seção apresenta os dois tipos diferentes de ataques abordados neste trabalho bem como as características utilizadas para realizar a simulação dos mesmos nos dados coletados da rede.

2.4.1 Denial of Service (DoS)

Um ataque do tipo negação de serviço (do inglês, *Denial of Service*) consiste em uma fonte única de ataque que dispara uma grande quantidade de requisições ao servidor alvo do ataque afim de prejudicar a disponibilidade de seus serviços a ponto de cessá-la totalmente através da sobrecarga do sistema.

Desta forma, em uma rede onde fluxos IP são coletados é possível observar uma concentração do número de pacotes oriundos do mesmo endereço de IP. Por outro lado, o número de requisições é limitado uma vez que apenas um dispositivo é utilizado para realizar o ataque (ASSIS; JR., 2015).

2.4.2 Distributed Denial of Service (DDOS)

Um ataque do tipo negação de serviço distribuída (do inglês, *Distributed Denial of Service*, DDoS) envolve um servidor ou máquina central que possui controle de diversas outras máquinas, sejam estas máquinas pertencentes aos atacantes ou máquinas dominadas pelos mesmos através de outros ataques de invasão. Todas estas máquinas são, então, ordenadas a dispararem requisições contra o alvo do ataque, daí o nome distribuído já que são muitas as origens das requisições neste ataque.

No ataque DDoS a quantidade de IPs e portas de origem são muito mais elevados devido à natureza distribuída do ataque. A intensidade e número das requisições também crescem de maneira proporcional visto que a quantidade de fontes envolvidas nas requisições é muito maior (ASSIS; JR., 2015).

3 METODOLOGIA

Este capítulo abordará toda a metodologia utilizada para realizar a extração de dados, cálculo e geração do baseline e a injeção de anomalias.

3.1 Extração de Dados

Esta seção explica como foi feita a extração de dados de um dataset no padrão NetFlow obtido por meio da coleta de fluxos IP na rede da UTFPR campus de Toledo.

Os dados foram cedidos pelo Professor Me. Alexandro Marcelo Zacaron da UTFPR campus Toledo e são referentes ao uso da rede no período de 1 a 31 de março de 2013, iniciando-se em uma sexta-feira e findando em um domingo.

Os dados encontram-se no formato do NetFlow, separados em 31 pastas, uma para cada dia do mês de março. Dentro de cada pasta existem 288 arquivos que correspondem aos dados trafegados pela rede durante o dia em intervalos de 5 minutos.

O formato NetFlow é um padrão criado pela empresa CISCO que realiza a coleta de dados dos fluxos de IP que trafegam pelos equipamentos da empresa (SYSTEMS, 2012). Os dados no formato NetFlow seguem o padrão mostrado abaixo:

Figura 3 – Exemplo do padrão de dados NetFlow

Date flow start	Duration	Proto	Src Ip Addr:Port		Dst IP Addr:Port	Packets	Bytes	Flows
2010-09-01 00:00:00.459	0	UDP	127.0.0.1:24920	->	192.168.0.1:22126	1	46	1
2010-09-01 00:00:00.557	0	TCP	2001:0db8:85a3:08d3:1319:8a2e:0370:7344:27840	->	192.168.0.1:22126	2	46	1

Fonte: Autoria própria

Onde cada coluna do cabeçalho (que corresponde à primeira linha da tabela) se refere aos seguintes dados:

- Date flow start: data e horário de início do fluxo de dados
- Duration: duração em milisegundos do fluxo
- Proto: protocolo de rede utilizado no fluxo
- Src Ip Addr:Port: endereço de IP de origem:porta de origem
- Dst IP Addr:Port: endereço de IP de destino:porta de destino
- Packets: número de pacotes
- Bytes: número de bytes
- Flows: número de fluxos

O cabeçalho é único e sempre está no topo de todos os arquivos NetFlow. Todas as linhas seguintes seguem o mesmo padrão e correspondem aos dados indicados pelo cabeçalho. Na figura de exemplo temos a segunda linha com um endereço de origem no padrão IPV4 enquanto na terceira linha o IPV6 é o padrão de endereço do IP de origem.

Para facilidade de análise um script na linguagem Python foi criado para realizar a leitura destes dados e gravá-los em novos arquivos únicos para cada dia no formato CSV o qual foi o formato escolhido pois, como será mostrado mais a frente no trabalho, o MatLab, uma das plataformas utilizadas para análises e cálculos pelo trabalho, possui bibliotecas nativas para lidar com arquivos no formato CSV e a linguagem Python possui bibliotecas que facilitam a leitura e uso das informações neste formato.

Estes arquivos gerados para cada dia contém os seguintes dados: horário, IP de origem, porta de origem, IP de destino, porta de destino, quantidade de pacotes e quantidade de bytes. Estes dados correspondem a todos os fluxos presentes nos duzentos e oitenta e oito arquivos referentes ao uso da rede durante um dia, cada um correspondendo a cinco minutos de uso. O padrão final corresponde à figura abaixo:

Figura 4 – Exemplo do padrão dos dados extraídos do NetFlow para o padrão CSV

```
index,horario,ip_origem,porta_origem,ip_destino,porta_destino,pacotes,bytes
0,00:00:00,10.1.1.10,12298,200.19.73.231,12223,1,75
1,00:00:00,10.1.1.10,7146,177.21.240.10,443,8,805
```

Fonte: Autoria própria

O índice é gerado automaticamente de maneira gradual para facilitar a referência a cada linha do arquivo nas análises posteriores. Todos os outros dados são os mesmos presentes nos arquivos NetFlow.

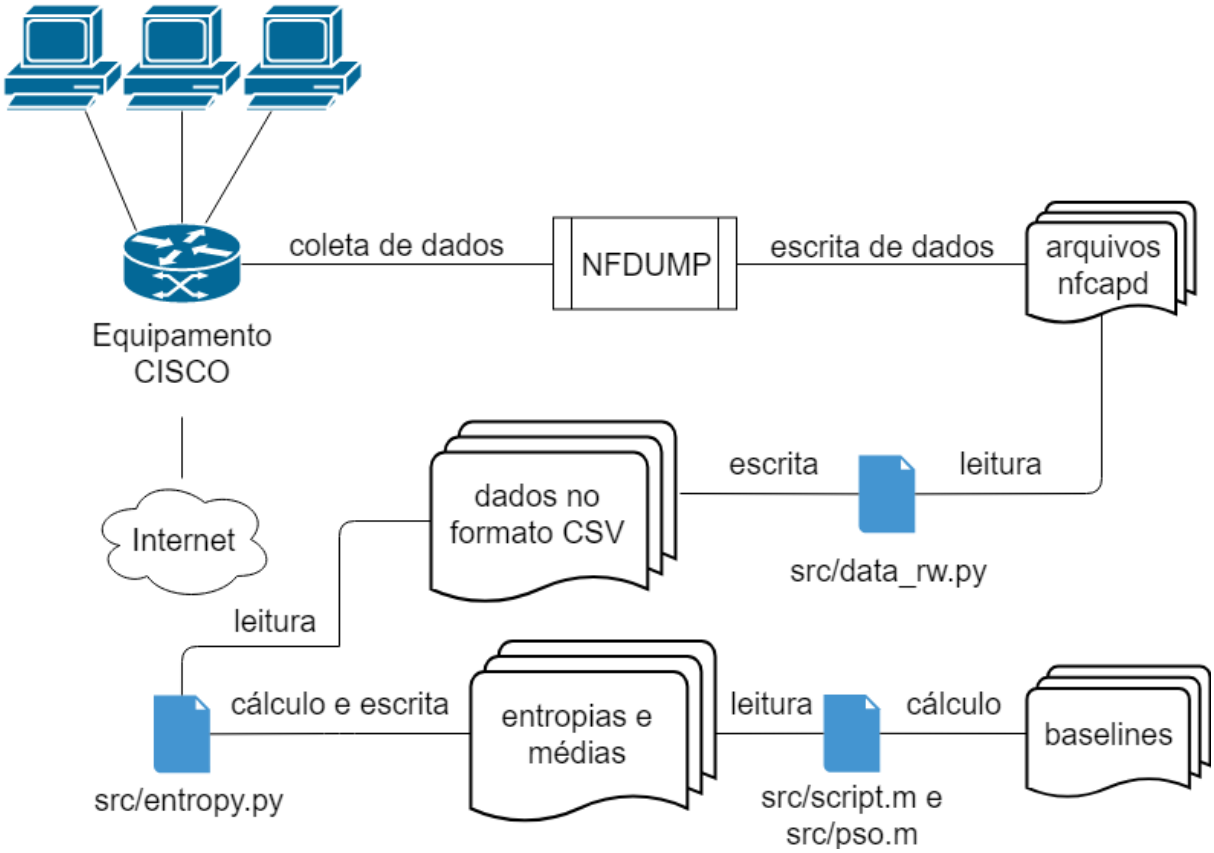
O script criado para realizar esta extração e estruturação dos dados chama-se *data_rw.py* e está presente na raíz da pasta script do projeto presente no website GitHub. O script realiza a leitura de todas as linhas, exceto o cabeçalho, dos duzentos e oitenta e oito arquivos de cada dia de uso da rede e realiza as operações necessárias para a correta extração dos dados relevantes. Esta leitura é realizada utilizando-se a interface de linha de comando *nfdump*, uma ferramenta de coleta e processamento de dados no formato NetFlow. Estes dados são, então, salvos em um único arquivo CSV para cada dia.

O script foi criado utilizando-se o sistema operacional Manjaro. Por ser um sistema operacional padrão Unix o script pode não funcionar corretamente em sistemas operacionais Windows.

3.2 Geração do Baseline

Esta seção apresentará como os dados qualitativos coletados a partir do Netflow foram transformados em dados quantitativos e, posteriormente, utilizados para a criação da assinatura digital do segmento de rede bem como apresentará os conceitos matemáticos envolvidos em tais cálculos.

Figura 5 – Fluxograma dos dados da rede desde a coleta até o cálculo do baseline



Fonte: Autoria própria

3.2.1 Estatística e Entropia

A fim de transformar os dados referentes a portas de destino e origem assim como endereço de IP de destino e origem utilizou-se a entropia de Shannon (SHANNON, 1948). A escolha foi baseada pelo fato da mesma ser utilizada por diferentes autores na literatura como Wei (2014), Yoshida (2003) e Chen e Chen (2008).

Podemos definir a probabilidade dentro de um espaço amostral como

$$p_i = o_i/n \tag{7}$$

onde p_i é a probabilidade de ocorrência do elemento i, o_i é a quantidade de ocorrências de i dentro do espaço amostral que possui um total de n ocorrências.

Desta forna, foi criado um script na linguagem Python que realiza a leitura dos arquivos no formato CSV gerados anteriormente e calcula a entropia da informação dos seguintes dados: IP de origem, porta de origem, IP de destino e porta de destino. Este arquivo chama-se *probability.py* e encontra-se na raíz da pasta *script* do projeto. Uma vez que a entropia leva em consideração a probabilidade de ocorrência e esta pode ser calculada considerando diferentes intervalos de tempo, optou-se por parametrizar tais valores. Todas as operações relatadas abaixo são realizadas para cada um dos trinta e um arquivos CSV contendo os dados da rede.

As colunas referentes ao índice (tratado como *index* nos arquivos de dados e de script criados para os fins deste trabalho) e ao horário são utilizadas para criar um dicionário que possui como número de elementos a quantidade de intervalos de minutos escolhidos presentes no intervalo de vinte e quatro horas de um dia. Cada item deste dicionário possui como chave o índice referente ao início de cada um destes intervalos e como valor o horário correspondente ao índice.

Então para cada uma das características: IP de origem, porta de origem, IP de destino e porta de destino é criado um histograma que contém o número de ocorrências de cada valor contido na coluna dentro dos intervalos estabelecidos. Este cálculo é feito utilizando o dicionário de índexes e horários criado anteriormente.

Utilizando-se deste histograma o script, então, realiza o cálculo da probabilidade de ocorrência de cada entrada única no conjunto de entradas do dado sendo analisado. Estas probabilidades são salvas em um novo dicionário, o qual tem suas probabilidades utilizadas para o cálculo da entropia em cada instante de intervalo escolhido para cada um dos dados seguindo a Equação 5.

Em seguida são calculadas as médias de pacotes por segundo e bytes por segundo. Considere que o intervalo possua n fluxos coletados e que para cada um deles exista um valor x_i de bytes por segundo e y_i de pacotes por segundo. A média aritmética de bytes por segundo é dada por:

$$X = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{8}$$

Já a média de pacotes por segundo é dada por:

$$Y = \frac{1}{n} \sum_{i=1}^{n} y_i \tag{9}$$

Após o cálculo das frequências, probabilidades, entropias e médias os dados são armazenados em um novo arquivo CSV. Cada linha deste arquivo traz as informações relativas ao intervalo ΔT escolhido como parâmetro. O cabeçalho segue o padrão dos arquivos CSV anteriormente gerados porém agora cada linha posterior corresponde à entropia ou média do dado referente a cada coluna a cada intervalo de tempo determinado no cálculo.

3.2.2 Uma proposta de baseline

O baseline representa o perfil de comportamento dos dados da rede em relação ao tempo, portanto o mesmo deve ser calculado a partir de intervalos de tempo e mostra-se importante analisar qual ou quais destes intervalos mostram-se os mais propícios para análises, por isto o cálculo das entropias e médias em diferentes intervalos de tempo.

Vale notar que o baseline deve ser calculado para cada dia da semana uma vez que os padrões de tráfego estão diretamente ligados ao cotidiano das pessoas e este tende a ser

rotineiro quando comparamos o mesmo dia da semana em diferentes semanas.

Portanto o cálculo do baseline deve ser feito utilizando-se diferentes intervalos de semanas de ao menos duas semanas. Estes diferentes intervalos serão comparados para que verifique-se qual deles apresenta menor taxa de erro e maior semelhança em relação ao histórico da rede.

Sendo assim, conforme a literatura indica prudente um importante componente deve ser decidido no cálculo do baseline: a métrica utilizada entre diferentes dados para se chegar ao valor que representa o padrão esperado em cada intervalo de tempo. Entre opções como médias, mediana, moda, média móvel, etc, optou-se pelo uso da distância euclidiana uma vez que esta apresenta bons resultados na literatura (GAO; WANG, 2014) (LI; GUO, 2007) (QIN; XU; WANG, 2015).

3.2.3 Cálculo do baseline

Esta seção abordará como a heurística PSO é utilizada para realizar a minimização do custo da equação de escolha dos valores do baseline.

A criação do baseline foi parametrizada considerando os intervalos de tempo das medições diárias, i.e. como está dividido o tempo ao longo de um dia completo, e quantas semanas são considerados para o cálculo do baseline. O objetivo desta parametrização é indicar para o dataset utilizado qual a melhor combinação de intervalo de tempo e semanas que garante o melhor desempenho em termos de erro quadrático médio.

As únicas exceções ficam por conta do intervalo de 4 semanas para segunda-feira e terça-feira visto que os arquivos NetFlow dos dias 25 e 26 apresentam informações de apenas algums períodos de horas do dia. Por conta desta diferença de amostra de dados a análise no intervalo de 4 semanas seria prejudicada pois os dados destes dias não refletem o comportamento real da rede.

A partir da leitura dos arquivos CSV contendo o resultado do cálculo das entropias e médias das dimensões analisadas anteriormente é realizada a minimização da função de distância Euclidiana através do algoritmo do PSO. A equação da distância Euclidiana d entre pontos de dois vetores Y_1 e Y_2 ambos com m elementos é dada por (LI; GUO, 2007):

$$d(Y_1, Y_2) = \sqrt{\sum_{j=1}^{m} (Y_{1j} - Y_{2j})^2} \tag{10}$$

A minimização do custo foi realizada para diferentes intervalos de minutos, como segue a tabela abaixo.

Intervalo em minutos	**Total de pontos analisados**
1	1440
2	720
3	480
4	360
5	288

Para cada intervalo em minutos a análise foi feita com amostras de intervalos de 2, 3 e 4 semanas. Os parâmetros utilizados pelo algoritmo do PSO são:

Parâmetro	**Valor**	**Função**
A	1	coeficiente de inércia local
B	1	coeficiente de inércia global
c_1	1	peso individual
c_2	2	peso global
K	100	tamanho da população
N	1000	número máximo de iterações
c	C_1,..., C_n	conjunto de possíveis n soluções

A função custo aplicada ao PSO é dada por:

$$f(\mathbf{c}) = \sum_{s=1}^{S} \sum_{t=1}^{T} \sqrt{\sum_{d=1}^{D} (c_t - x_{t,s,d})^2} \tag{11}$$

onde $\mathbf{c}$ é o vetor com o baseline para os T instantes de tempo do dia, $x_{t,s,d}$ é o histórico do dataset no instante t da semana s na dimensão analisada d.

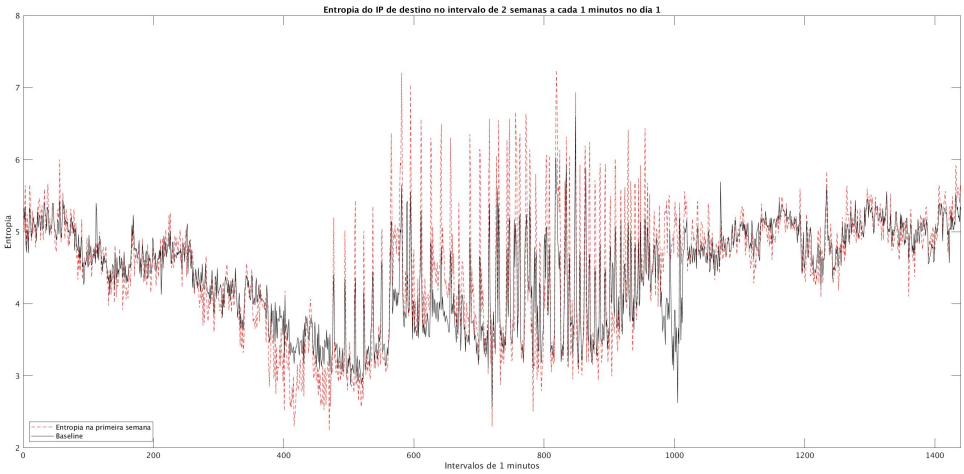
Sendo assim, são gerados, para cada dia da semana e para cada uma das seis dimensões analisadas, i.e. bytes por segundo, pacotes por segundo, entropia da porta de origem, entropia da porta de destino, entropia do IP de origem e entropia do IP de destino, vetores de dimensão T. Na Figura 6 é apresentado de forma gráfica o resultado de um caso do cálculo do baseline para uma sexta-feira e de forma simultânea comparando o resultado com uma leitura instantânea referente ao dia 1 do mês de março de 2013.

O valor do baseline encontra-se no eixo das ordenadas e os intervalos de tempo se encontram no eixo das abscissas. O gráfico em vermelho representa o comportamento da dimensão analisada na primeira semana e o gráfico em preto representa o baseline. Arquivos do formato *.fig* gerados para a criação destas figuras também são salvos junto a elas.

3.3 Injeção de Anomalias

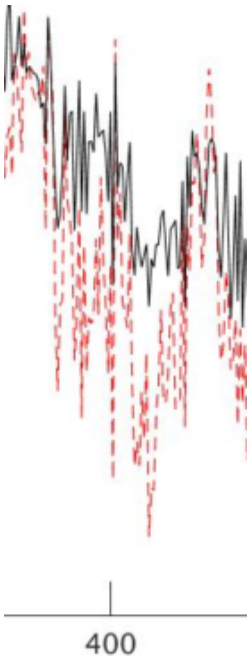
Esta seção apresentará quais critérios foram definidos para realizar a injeção de anomalias simulando ataques DoS e DDoS diretamente nos dados extraídos da rede do campus de Toledo e como ocorreu a implementação da simulação de tais ataques.

Figura 6 – Exemplo de gráfico utilizando dados de entropia do IP de origem e seu baseline resultante no intervalo de 1 minuto com 2 semanas de amostra



Fonte: Autoria própria

Figura 7 – Detalhe do gráfico envidenciando as diferenças entre a leitura e o baseline gerado



Fonte: Autoria própria

3.3.1 IP de origem

Como num ataque DoS um único host é responsável pelo envio de requisições apenas um endereço será utilizado para a injeção do campo Ip de origem em todos os fluxos em um ataque DoS. Já para um ataque DDoS um novo endereço de IP de origem será gerado para cada injeção.

3.3.2 Porta de origem

Pelo mesmo motivo do campo IP de origem uma mesma porta será utilizada na injeção de todos os fluxos e analogamente uma nova porta de origem será gerada para cada injeção em um ataque DDoS. Esta porta elatória respeita os limites de 0 a 65535 que são as portas disponíveis em uma máquina.

3.3.3 IP de destino

Para ambos os tipos de ataque será gerado um endereço de IP de destino presente nos dados originais no período em que os fluxos estão sendo injetados.

3.3.4 Porta de destino

Uma porta de destino aleatória será gerada para cada requisição tanto de um ataque DoS quanto de um ataque DDoS, respeitando o limite de ser uma porta entre os valores de 0 a 65535.

3.3.5 Bytes por segundo

Em ambos os ataques será injetada um valor aleatório de bytes por segundo que respeite os valores máximos e mínimos presentes nos dados originais.

3.3.6 Pacotes por segundo

Da mesma maneira que o valor de bytes por segundo um valor aleatório entre o mínimo e o máximo presente nos dados originais será injetado em cada tipo de ataque.

3.3.7 Implementação das simulações

A injeção de anomalias ocorre dentro de um intervalo de horas parametrizado para todos os dias de uma semana escolhida. A partir de uma matriz que representa todos os dias do mês de março de 2013 separados por semanas para que seja possível escolher em qual semana o ataque deve ser aplicado.

O próximo passo é realizar o fluxo de injeção de anomalias em cada dia da semana. O fluxo se inicia lendo o arquivo que contém os dados correspondentes ao dia sendo iterado e salvando o conteúdo de cada uma de suas colunas em uma variável lista diferente. Após isto

estão as variáveis que definem a quantidade de anomalias a serem injetadas, o horário de início, horário de fim (os horários estão no formato 24h) e o tipo de ataque a ser realizado. Nos dados e gráficos gerados por este trabalho foram usadas as seguintes configurações:

Ataque DoS:

- quantidade: 2000
- horário de início: 8 horas
- horário de fim: 10 horas
- primeira semana do mês

Ataque DDoS:

- quantidade: 2500
- horário de início: 8 horas
- horário de fim: 10 horas
- primeira semana do mês

A seguir verifica-se quais foram os valores máximo e mínimo de pacotes e de bytes nos fluxos e armazena os mesmos em memória. É feita, então, a verificação de quais índices de fluxos dos dados correspondem ao horário em que deve-se iniciar as injeções e ao horário em que as mesmas devem ser finalizadas para que ele trabalhe apenas com os dados e índices entre estes dois limites. Ele, então, gera um endereço de IP padrão e uma porta padrão para serem utilizados caso o ataque seja do tipo DoS.

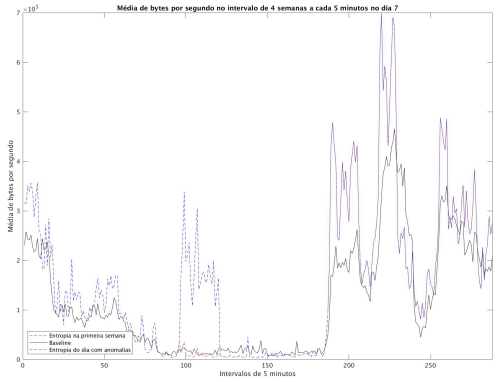
O próximo passo é realizar um laço de iterações que irá realizar a geração e injeção dos dados anômalos. Este laço irá iterar um número de vezes igual à quantidade de anomalias escolhidas para serem injetadas. O laço inicia gerando um número inteiro aleatório entre os limites dos índexes presentes na porção dos dados da rede com o qual ele está lidando. Este número inteiro será utilizado como índice para adicionar os dados anômalos na posição deste índice, deslocando todos os dados originais de acordo com esta nova inserção, inclusive o dado que encontra-se no dado índice.

Como cada coluna se encontra em uma variável as injeções são realizadas de forma sequencial em cada uma das variáveis no índex escolhido. A primeira inserção que ocorre é a do horário. O mesmo horário do dado encontrado originalmente no índex de inserção é utilizado. A próxima inserção é a do IP de origem, seguida da porta de origem, IP de destino, porta de destino, pacotes e bytes. As inserções destes dados são realizadas seguindo o definido na seção anterior.

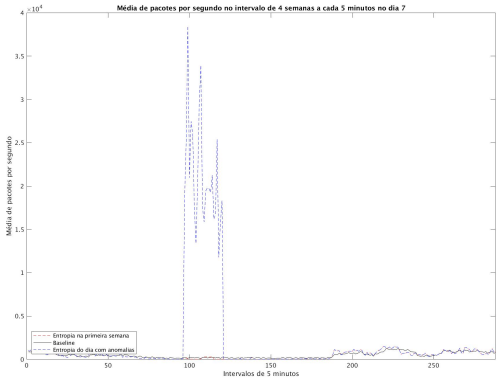
A Figura 8 apresenta os gráficos resultantes das 6 dimensões analisadas contendo os dados da medição do dia 7 de março de 2013, os dados resultantes do mesmo dia com a simulação de ataque DDoS que injetou 2500 novos fluxos entre as 8 e às 10 horas da manhã e seu baseline utilizando amostras de 4 semanas no intervalo de 5 minutos para cálculo do baseline e das entropias e médias.

A Figura 9 é o gráfico do IP de origem resultante das simulações supracitadas evidenciando a diferença da entropia entre o baseline, as medições instantâneas e as medições

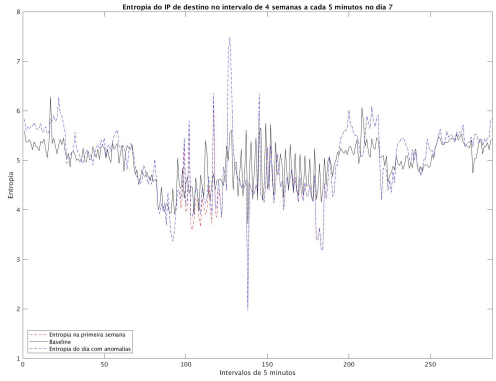
com anomalias, cada uma delas representada pelas curvas de cores preto, vermelho e azul, respectivamente. Essas configurações de cores também se aplicam à Figura 8.



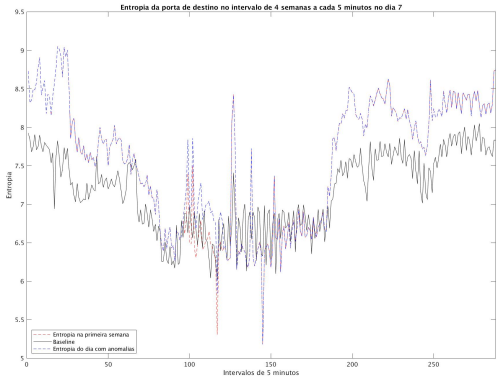
(a) Bytes por segundo



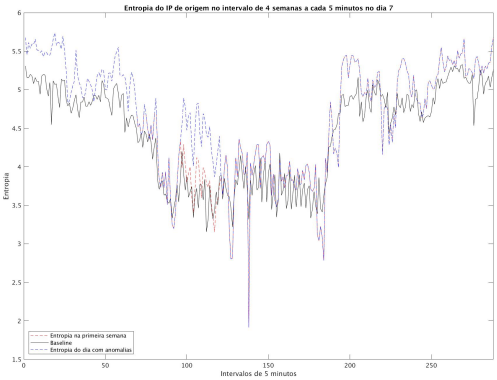
(b) Pacotes por segundo



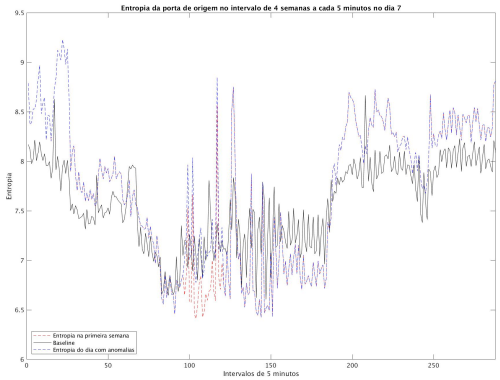
(c) IP de destino



(d) Porta de destino



(e) IP de origem

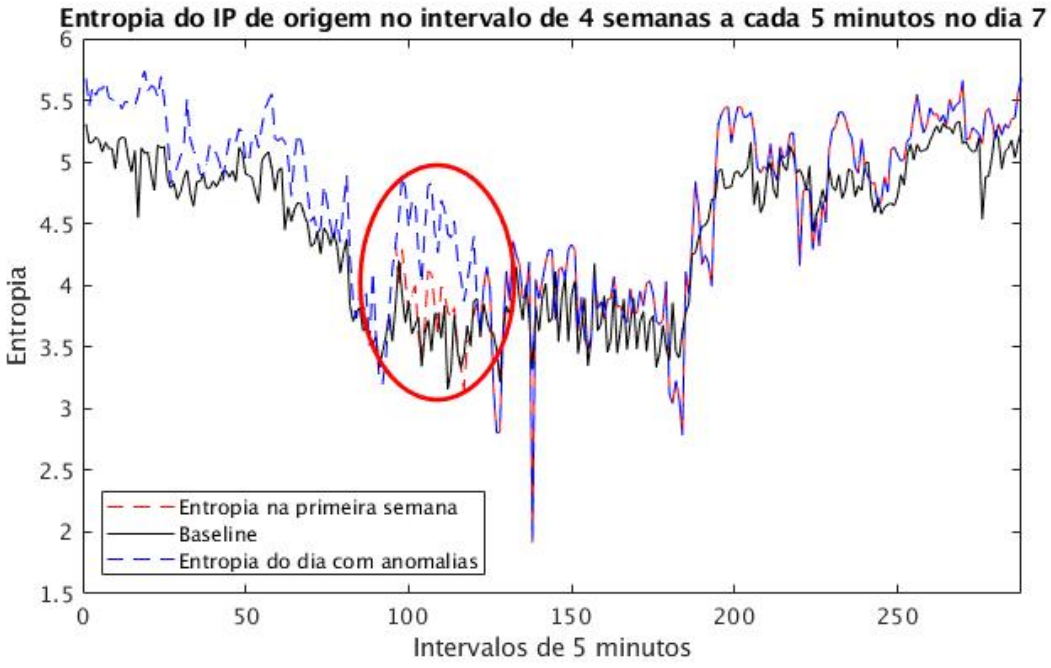


(f) Porta de origem

Figura 8 – Gráficos das 6 dimensões com as medições instantâneas do dia 7, o mesmo dia com anomalias injetadas e o baseline

Fonte: Autoria própria

Figura 9 – Gráfico do IP de origem numa quinta-feira com simulação de um ataque DDoS usando intervalos de 5 minutos e 4 semanas de amostras para cálculo das entropias e baseline



Fonte: Autoria própria

4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Este capítulo explicará a função de cada script criado para a realização do trabalho e também fará a análise dos dados extraídos a partir dos arquivos NetFlow originais, dos dados de entropia e médias calculados a partir deles, dos gráficos de baselines gerados a partir destes cálculos, dos arquivos com dados anômalos injetados e seus resultados gráficos em comparação com o baseline da rede.

4.1 Repositórios de arquivos criados e gerados pelo trabalho

Todos os scripts utilizados para a extração e análise de todos os dados se encontram no seguinte repositório do website GitHub: https://github.com/brunoanken/TCC2. Na pasta *script* do repositório encontram-se todos os arquivos de código utilizados para a extração, geração, injeção e análise de dados deste trabalho. O arquivo *data_rw.py* realiza a leitura dos arquivos no formato NetFlow e grava os dados pertinentes no formato em umnovo arquivo no formatoCSV. O arquivo *probability.py* da raíz da pasta *script* realiza o cálculo das entropias e das médias dos dados extraídos pelo arquivo *data_rw.py* e gera um novo arquivo CSV contendo os resultados.

O arquivo *script.m* realiza o cálculo das entropias utilizando-se do algoritmo do PSO presente no arquivo *pso.m*. Após o cálculo é gerado um gráfico contendo o valor computado ao longo do tempo mais os dados de entropia ou média, a depender do dado sendo analisado, referentes ao uso na primeira semana dos dados da rede. Este gráfico é salvo no formato de imagem *jpg*. O arquivo de figura utilizado pelo software MatLab para criar o gráfico também é salvo para ser posteriormente utilizado pelo arquivo *script_anomaly.m* que insere nestes arquivos os dados de entropia ou média dos arquivos com anomalias injetadas.

Dentro da pasta *injection*, que fica na raíz da pasta *script* temos os arquivos responsáveis por executarem os métodos relacionados às injeções de anomalia. O arquivo *main.py* realiza a geração e inserção de dados anômalos dentro dos arquivos com dados extraídos da rede e gera novos arquivos CSV contendo os novos dados com anomalias. O arquivo *probability.py* presente na pasta *injection* realiza o cálculo das entropias e médias destes novos arquivos com anomalias.

Todos os dados em formato CSV e imagens de gráficos gerados a partir destes arquivos encontram-se no seguinte repositório do website Google Drive: https://bit.ly/2XKV30R. Na pasta *dados_rede* encontram-se todos os dados extraídos da rede e também os dados de médias e entropias gerados a partir deles.

Todos os dados e imagens relacionados aos arquivos anômalos foram gerados simulando-se tanto um ataque DoS quanto um ataque DDoS e estão presentes em suas respectivas pastas nomeadas *dos* e *ddos*, respectivamente, tanto nas pastas *dados_anomalos*, onde estão presentes

os arquivos com dados anômalos injetados e os arquivos com os resultados das entropias e médias destes dados anômalos, quanto na pasta *imagens*, onde estão presentes os gráficos gerados por este trabalho.

No caso do ataque DoS foram injetados dois mil novos fluxos de dados. Para o ataque DDoS foram dois mil e quinhentos novos fluxos. Para ambos os casos os novos dados foram injetados entre as oito horas da manhã e as dez horas da manhã nos sete primeiros dias do mês. Portanto as comparações feitas nas próximas seções levarão em consideração tanto o baseline quanto a utilização da rede apenas na primeira semana.

4.2 Dados extraídos

Os arquivos CSV contendo os dados extraídos dos arquivos NetFlow originais contam com centenas de milhares de linhas (onde cada linha representa um fluxo na rede) no caso dos dias que correspondem a finais de semana e cerca de dois milhões de linhas no caso dos dias de semana.

As únicas exceções são os dias 25 e 26, que correspondem a uma segunda-feira e uma terça-feira, respectivamente. Os arquivos NetFlow referentes a estes dias apresentaram-se incompletos e seus dados possuem períodos de horas sem dados disponíveis. Todos os dados gerados a partir destes dias estão incompletos em relação aos outros. Por conta desta ausência de dados os arquivos dos dias 25 e 26 não puderam ser utilizados para os cálculos do baseline utilizando-se quatro semanas de intervalo.

4.3 Dados do cálculo das entropias e médias

Os arquivos CSV contendo as entropias dos dados de IP de origem, IP de destino, porta de origem, porta de destino e as médias de pacotes por segundo e bytes por segundo apresentam os devidos tamanhos levando em conta os intervalos de minutos a partir dos quais foram executados. Temos que um dia de 24 horas possui 1440 minutos. Portanto para intervalos de 1, 2, 3, 4 e 5 minutos temos, respectivamente, 1440, 720, 480, 360 e 288 linhas, excluindo-se da contagem a primeira, que corresponde ao cabeçalho.

4.4 Gráficos do baseline da rede

Os gráficos gerados contendo o baseline indicam que o perfil da rede nunca se comporta de maneira idêntica a seu uso isolado em alguma semana, sempre havendo alguns trechos do gráfico onde existe uma grande diferença entre o perfil da rede e seu uso em alguma semana. Estas diferenças ficam mais visíveis quanto maior o intervalo em minutos analisados. Devido a menor quantidade de intervalos presentes no eixo das abscissas os espaços entre os gráficos ficam mais evidentes. Apesar disto é visível que o baseline, em geral, se comporta de maneira semelhante ao uso isolado da rede.

4.5 Dados da injeção de anomalias

Os arquivos com os dados da rede mais dados anômalos injetados apresentam o tamanho esperado (a quantidade de linhas do arquivo original mais a quantidade de fluxos anômalos gerados), tanto para a simulação do ataque DoS quanto do ataque DDoS.

O mesmo pode ser dito dos arquivos gerados a partir do cálculo das entropias e médias destes arquivos. Assim como os dados descritos na Seção 7.2., os tamanhos dos arquivos mostram-se corretos.

4.6 Baselines comparados à entropia e médias de um ataque DoS

Todas as análises feitas nas nesta seção aplicam-se a todos os gráficos gerados a partir de quaisquer intervalos de minutos e de semanas.

4.6.1 IP de origem

Podemos identificar que comparado ao uso da primeira semana a entropia do IP de origem aumentou. Isto indica aumento da heterogeneidade dos dados apesar de um ataque DoS injetar dados com o mesmo IP de origem em todas as iterações. O resultado esperado era o oposto visto que mais fluxos com o mesmo IP de origem deveria acarretar em um aumento da homogeneidade e consequentemente a diminuição da entropia.

Já em comparação com o baseline podemos observar que as linhas do gráfico tornaram-se mais próximas o que indica que a injeção de anomalias deixou o uso do dia utilizado como base para o ataque mais próximo do que foi definido como o perfil de uso padrão da rede.

4.6.2 Porta de origem

Podemos verificar uma leve queda da entropia quando comparada ao seu dia utilizado como base, indicando um aumento da homogeneidade dos dados. Isto condiz com o tipo de ataque visto que a mesma porta é utilizada em cada injeção de fluxo. Assim como com o IP de origem esta variação fez com que o gráfico do dia com anomalias se aproximasse do gráfico do baseline.

4.6.3 IP de destino

A entropia sofreu uma leve elevação, tanto em relação ao seu dia usado como base quanto em relação ao baseline, indicando o aumento da heterogeneidade dos dados. Isto condiz com o comportamento da simulação visto que apesar de utilizar IPs de destino existentes nos dados originais eles são escolhidos de maneira aleatória. Desta vez os dados anômalos acabaram por se distanciar do perfil da rede.

4.6.4 Porta de destino

A entropia da porta de destino comportou-se de maneira semelhante à entropia do IP de destino. Sua heterogeneidade aumentou e a distância entre a entropia dos dados anômalos e o baseline também.

4.6.5 Pacotes por segundo

A média de pacotes por segundo sofreu uma acentuada elevação durante o período onde o ataque foi simulado. Uma possível explicação é a de que como o processo de injeção de anomalias utiliza aleatoriamente dados entre os mínimos e máximos de pacotes presentes no dia vários dados com alto valor de pacotes por segundo foram injetados e isto acabou por aumentar de maneira exagerada a média dos dados.

4.6.6 Bytes por segundo

O comportamento da média de bytes por segundo é semelhante ao de pacotes por segundo, porém sua elevação foi menos acentuada. Durante o período de ataque a média de pacotes por segundo elevou em grande magnitude os limites do gráfico enquanto a média de bytes por segundo, apesar de um grande aumento em relação ao baseline e ao uso original, não chegou ao limite original do gráfico. Esta elevação deixou o gráfico dos dados anômalos distante tanto do gráfico do dia original quanto do gráfico do baseline.

4.7 Baselines comparados à entropia e médias de um ataque DDoS

4.7.1 IP de origem

Houve um tímido aumento da entropia do IP de origem se comparada aos dados originais. Isto levou a uma maior aproximação do baseline. Resultados semelhantes aos comportamento do IP de origem na simulação de um ataque DoS.

4.7.2 Porta de origem

Houve um aumento visível da entropia da porta de origem comparada à entropia dos dados originais. Este resultado condiz bastante com o esperado visto que a simulação de um ataque DDoS injetou diversos dados com várias portas de origem diferentes visto que elas são geradas de maneira aleatória pelo processo, portanto o aumento de heterogeneidade dos dados. Isto também levou o gráfico dos dados anômalos a se distanciar do gráfico do baseline.

4.7.3 IP de destino

Houve um tímido aumento da entropia do IP de destino. Visto que o processo trabalha com valores já existentes do IP de destino dos dados originais uma leve variação que mantivesse

o gráfico semelhante ao dos dados originais era um resultado esperado.

4.7.4 Porta de destino

O aumento da entropia da porta de destino é perceptível e indica maior heterogeneidade em relação aos dados originais e também em relação ao baseline. Como a porta de destino injetada pelas simulações tanto de um ataque DoS quanto de um ataque DDoS é gerada de maneira aleatória este aumento da entropia já era esperado.

4.7.5 Pacotes por segundo

O mesmo comportamento apresentado pela simulação de um ataque DoS foi percebido em relação à simulação de um ataque DDoS. Houve um exagerado aumento da média de pacotes por segundo em relação aos dados originais e ao baseline, elevando em grande magnitude os limites do gráfico original.

4.7.6 Bytes por segundo

Assim como ocorreu com o gráfico da média de bytes por segundo da simulação de um ataque DoS, percebeu-se um relevante aumento na média de bytes por segundo tanto em relação aos dados originais quanto ao baseline da rede. Mas ao contrário do ocorrido com a média de pacotes por segundo os limites do gráfico original mantiveram-se os mesmos.

5 CONCLUSÃO

É possível concluir que a simulação de ataques não funcionou da maneira esperada tendo em vista que em alguns casos, como no IP de origem na simulação de um ataque DoS, o resultado obtido foi o oposto ao esperado e acabou deixando o comportamento da rede mais próximo do que era esperado apesar de ser o comportamento com anomalias injetadas. Além disto houve o exagerado aumento da média de pacotes por segundo nas simulações.

Já a criação do baseline mostrou-se proveitosa pois foi possível identificar algumas características acerca do comportamento da rede que vão além do puro volume de dados trafegados. É possível perceber os períodos em que a rede fica mais movimentada porém com o ganho de ser possível analisar o comportamento de individualmente de seus dados.

Como é possível a ocorrência de diferentes ataques que podem tanto aumentar quanto diminuir a entropia de um dado específico (como um ataque DoS que, em teoria, deve diminuir a entropia do IP de origem e um DDoS deveria aumentar esta entropia) ao mesmo tempo em que estes mesmos ataques aumentam a quantidade de dados trafegados pela rede uma análise mais rasa levando em consideração apenas a quantidade de dados sendo trafegado não seria capaz de identificar qual tipo de ataque está sendo efetuado ou até mesmo se trata-se de um ataque ou algum outro tipo de anomalia.

Desta maneira a análise da entropia dos dados da rede mostra-se como uma possibilidade que enriquece a análise da rede e expande as possibilidades de novos estudos acerca do comportamento de uma rede de computadores.

Também é notável a facilidade que uma ferramenta de simulação de ataques traria para as futuras análises visto que este tipo de simulação não atinge o usuário final da rede pois não é necessário simular um ataque diretamente na rede, prejudicando o funcionamento da mesma. Também vale notar que, com os devidos ajustes, diversos tipos de ataques poderiam ser simulados com eficiência.

5.1 TRABALHOS FUTUROS

A metodologia de injeção de anomalias precisa ser melhorada, tornando-se mais realista ao, por exemplo, levar em consideração a quantidade de bytes por segundo e pacotes por segundo a ser injetada, verificando se a maioria das requisições na rede contam ou não com grandes quantias destes dados e também se os ataques sendo simulados, no geral, realizam envio ou requisição de grande quantidade destes dados visto que isto depende do tipo de serviço disponibilizado pelo servidor que está sendo atacado.

A periodicidade de injeção pode ser melhorada também tendo em vista que um ataque DoS ou DDoS realiza várias requisições sequenciais de maneira ininterrupta durante um período de tempo e não de maneira espaçada como simulado pelo trabalho.

Também é possível adicionar a simulação de mais tipos de ataques. Outra melhoria notável seria a criação de uma interface gráfica onde seria possível escolher o tipo de ataque e configurar seus parâmetros sem precisar mexer diretamente no código.

Há a possibilidade de melhorar a performance dos algoritmos que fazem a extração dos dados originais, que calculam as entropias e médias de cada dia ao realizar algumas operações em paralelo. Tendo em vista que a extração dos dados e o cálculo das entropias e médias independem de um dia para o outro estes poderiam ser processados ao mesmo tempo sem a preocupação de ocorrência de inconsistência de dados.

A proposta de um método de análise que indique a ocorrência ou não de anomalias na rede, bem como um estudo com inferências para verificar a assertividade deste método proposto, seria o próximo passo a ser dado para a detecção de anomalias na rede.

Como o uso de uma rede de computadores está em constante mudanças, seja por mais ou menos usuários a acessando, por exemplo, cabe um estudo acerca da periodicidade de se definir o baseline da rede. Um exemplo seria utilizar o baseline gerado a partir de intervalos de três ou quatro semanas para verificar como a rede vem se modificando ao longo do tempo e utilizar o baseline do intervalo de duas semanas como base para verificar a ocorrência de anomalias.

**** criar um modelo baseado no PSO ou algum outro algoritmo/heurística para que seja treinado pra conseguir criar melhor o perfil da rede ****

5.2 CONSIDERAÇÕES FINAIS

Apesar do algoritmo de simulação de ataques precisar de melhorias ele mostrou-se prático e com potencial visto que sua implementação mostrou resultados quando a comparação gráfica é feita e também por conta da simplicidade que ele traz para realizar análises e estudos acerca da rede.

Também é notável o uso do baseline que pode vir a se mostrar útil em tarefas não diretamente relacionadas à detecção de anomalias e que por ser calculado através de uma heurística ao invés de um cálculo mais simples como média ou moda traz maior confiabilidade ao resultado final.

Referências

ANDERSON, J. P. **Computer security threat monitoring and surveillance.** Gaithersburg: National Institute of Standards and Technology, 1980. Citado na página 2.

ASSIS, M. V. O. de; JR., M. L. P. Scorpius: sflow network anomaly simulator. **Journal Computer Science**, Brasil, julho 2015. Citado 3 vezes nas páginas 6, 10 e 11.

BASS, T. Intrusion detection systems and multisensor data fusion. **Communications of the ACM**, v. 43, n. 4, p. 99–105, abril 2000. Citado na página 3.

BOUÇAS, C. Gastos mundiais com segurança da informação atingem US$86,1 bi no ano. **Valor Econômico**, São Paulo, out. 2016. Disponível em: <http://en.wikibooks.org/wiki/LaTeX>. Acesso em: 8 de outubro de 2017. Citado na página 1.

BOUGHACI, D. et al. Distributed intrusion detection framework based on autonomous and mobile agents. **2006 International Conference on Dependability of Computer Systems**, Poland, p. 248–255, maio 2006. Citado na página 4.

BOYD, S.; VANDENBERGHE, L. **Convex Optimization**. Cambridge University Press, Estados Unidos da América: Cambridge University Press, 2009. Citado na página 8.

BRADLEY, T. Gartner predicts information security spending to reach $93 billion in 2018. **Forbes**, Nova Iorque, Estados Unidos da América, ago. 2017. Disponível em: <https://www.forbes.com/sites/tonybradley/2017/08/17/gartner-predicts-information-security-spending-to-reach-93-billion-in-2018/#6625c0633e7f>. Acesso em: 8 de outubro de 2017. Citado na página 1.

CEBRIÁN, B. D. Cibertaque: o vírus wannacry e a ameaça de uma nova onda de infecções. **El País**, Madri, Espanha, mai. 2017. Disponível em: <https://brasil.elpais.com/brasil/2017/05/14/internacional/1494758068_707857.html>. Acesso em: 8 de outubro de 2017. Citado na página 1.

CHEN, R.-C.; CHEN, S.-P. Intrusion detection using a hybrid support vector machine based on entropy and tf-idf. **International Journal of Innovative Computing, Information and Control**, v. 4, p. 413 – 424, fevereiro 2008. Citado na página 14.

CISCO. **NetFlow Version 9 Flow-Record Format**. 2011. Disponível em: <https://www.cisco.com/en/US/technologies/tk648/tk362/technologies_white_{p}aper09186a00800a3db9.ht>. Acesso em: 9 de outubro de 2017. Citado na página 7.

CISCO. **Cisco Visual Networking Index: Forecast and Trends, 2017–2022 White Paper**. 2019. Disponível em: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html#_Toc532256790>. Acesso em: 22 de junho de 2019. Citado na página 1.

CLAISE, B.; TRAMMELL, B.; AITKEN, P. **Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information**. IETF: [s.n.], 2013. Disponível em: <https://tools.ietf.org/html/rfc7011>. Acesso em: 9 de outubro de 2017. Citado na página 7.

COMPUTERWORLD. Gastos globais com seguros contra ataques cibernéticos já somam cerca de US$ 2 bi. **COMPUTERWORLD**, São Paulo, fev. 2017. Disponível em: <http://computerworld.com.br/gastos-globais-com-seguros-contra-ataques-ciberneticos-ja-somam-cerca-de-us-2-bi>. Acesso em: 8 de outubro de 2017. Citado na página 1.

DENNING, D. E. An intrusion-detection model. **IEEE Transactions on Software Engineering**, IEEE, p. 118–131, fevereiro 1987. Citado na página 2.

DURST, R. et al. Testing and evaluating computer intrusion detection systems. **Communications of the ACM**, v. 42, n. 7, p. 53–61, julho 1999. Citado na página 4.

G1. Ciberataques em larga escala atingem empresas no mundo e afetam brasil. **G1**, Rio de Janeiro, mai. 2017. Disponível em: <https://g1.globo.com/tecnologia/noticia/hospitais-publicos-na-inglaterra-sao-alvo-cyber-ataques-em-larga-escala.ghtml>. Acesso em: 8 de outubro de 2017. Citado na página 1.

GAO, M.; WANG, N. A network intrusion detection method based on improved k-means algorithm. **Advanced Science and Technology Letters**, v. 53, p. 429 – 433, 2014. Citado na página 16.

GARG, S. et al. A methodology for detection and estimation of software aging. **Proceedings Ninth International Symposium on Software Reliability Engineering**, Alemanha, p. 283–292, novembro 1998. Citado na página 7.

GUL, I.; HUSSAIN, M. Distributed cloud intrusion detection model. **International Journal of Advanced Science and Technology**, v. 34, p. 71–82, setembro 2011. Citado na página 5.

HAMAMOTO, A. H. et al. Network anomaly detection system using genetic algorithm and fuzzy logic. **Expert Systems with Applications**, v. 92, p. 390–402, 2018. Citado na página 6.

HWANG, K. et al. Hybrid intrusion detection with weighted signature generation over anomalous internet episodes. **IEEE Transactions on Dependable and Secure Computing**, v. 4, n. 1, p. 41–55, fevereiro 2007. Citado na página 5.

ILGUN, K.; KEMMERER, R. A.; PORRAS, P. A. State transition analysis: a rule-based intrusion detection approach. **IEEE Transactions on Software Engineering**, v. 21, n. 3, p. 181–199, março 1995. Citado na página 3.

Jr, M. L. P.; ZARPELÃO, B. B.; MENDES, L. S. Anomaly detection for network servers using digital signature of network segment. **Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/E-Learning on Telecommunications Workshop (AICT/SAPIR/ELETE'05)**, Portugal, p. 290–295, julho 2005. Citado na página 7.

KANNADIGA, P.; ZULKERNINE, M. Didma: a distributed intrusion detection system using mobile agents. **Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Network**, Estados Unidos da América, p. 238–245, maio 2005. Citado na página 4.

KEMMERER, R. A.; VIGNA, G. Intrusion detection: A brief history and overview. **Security & Privacy**, Computer, University of Californa Santa Barbara, Goleta, Califórnia, Estados Unidos da América, v. 35, p. 27–30, agosto 2002. Citado na página 2.

KENNEDY, J.; EBERHART, R. Particle swarm optimization. **Proceedings of the IEEE International Conference on Neural Networks**, Washington DC, p. 1942–1948, 1995. Citado na página 8.

LI, Y.; GUO, L. An active learning based tcm-knn algorithm for supervised network intrusion detection. **Computers Security**, Elsevier, v. 26, p. 459 – 467, outubro 2007. Citado na página 16.

MARINI, F.; WALCZAK, B. Particle swarm optimization (pso). a tutorial. **Chemometrics and Intelligent Laboratory Systems**, v. 149, p. 153–165, abril 2015. Citado na página 7.

MCHUGH, J.; CHRISTIE, A.; ALLEN, J. Defending yourself: The role of intrusion detection systems. **IEEE Software**, IEEE, v. 17, n. 5, p. 42–51, outubro 2000. Citado 2 vezes nas páginas 2 e 4.

MITCHELL, R.; CHEN, I.-R. Behavior rule specification-based intrusion detection for safety critical medical cyber physical systems. **IEEE Transactions on Dependable and Secure Computing**, v. 12, n. 1, p. 16–30, fevereiro 2015. Citado na página 5.

PEARL, J. **Intelligent Search Strategies for Computer Problem Solving**. Estados Unidos da América: Addison-Wesley Publishing Company, 1984. Citado na página 7.

PRIYAMBODO, T. K.; PRAYUDI, Y. **ARPN Journal of Engineering and Applied Sciences**, Asian Research Publishing Network (ARPN), v. 10, p. 652 – 660, fevereiro 2015. Citado na página 1.

QIN, X.; XU, T.; WANG, C. Ddos attack detection using flow entropy and clustering technique. **11th International Conference on Computational Intelligence and Security**, p. 412 – 415, 2015. Citado na página 16.

SFLOW.ORG. **About sFlow**. 2017. Disponível em: <http://www.sflow.org/about/index.php>. Acesso em: 9 de outubro de 2017. Citado na página 7.

SHANNON, C. E. A mathematical theory of communication. **The Bell System Technical Journal**, v. 27, outubro 1948. Citado 2 vezes nas páginas 10 e 14.

SHON, T. et al. A machine learning framework for network anomaly detection using svm and ga. **Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop**, Estados Unidos da América, p. 176–183, agosto 2005. Citado na página 5.

SYSTEMS, C. **Introduction to Cisco IOS NetFlow - A Technical Overview**. 2012. Disponível em: <https://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ios-netflow/prod_white_paper0900aecd80406232.html>. Acesso em: 11 de junho de 2019. Citado na página 12.

WALTRICK, R. Crimes virtuais trazem prejuízo bilionário para brasileiros. **Gazeta do Povo**, Curitiba, nov. 2016. Disponível em: <http://www.gazetadopovo.com.br/economia/inteligencia-artificial/crimes-virtuais-trazem-prejuizo-bilionario-para-brasileiros-c50g4ta0o4u4dwvt7l9ippivh>. Acesso em: 8 de outubro de 2017. Citado na página 1.

WEI, L. Intrusion detection based on information entropy of multiple support vector machine. **International Journal of Computer and Information Technology**, janeiro 2014. Citado na página 14.

YANG, J. et al. Hids-dt: An effective hybrid intrusion detection system based on decision tree. **2010 International Conference on Communications and Mobile Computing**, China, p. 70–75, abril 2010. Citado 2 vezes nas páginas 4 e 5.

YOSHIDA, K. Entropy based intrusion detection. **IEEE**, p. 840 – 843, 2003. Citado na página 14.

ZHANG, X. et al. Secure coprocessor-based intrusion detection. **EW 10 Proceedings of the 10th workshop on ACM SIGOPS European workshop**, França, p. 239–242, julho 2002. Citado na página 4.