Trabalho de Conclusão de Curso

Universidade Estadual Paulista "Júlio de Mesquita Filho"

Detecção de Botnets utilizando análise completa de pacotes

Fábio Carciofi Júnior

Prof. Dr. Kelton Augusto Pontara da Costa (Orientador)

Departamento de Computação, Faculdade de Ciências, Universidade Estadual Paulista "Júlio de Mesquita Filho"
Av. Eng. Luiz Edmundo Carrijo Coube, 14-01
Vargem Limpa, CEP 17033-360 - Bauru - SP.

12/12/2025

C265d

Carciofi Junior, Fabio
Detecção de botnets utilizando análise completa de pacotes / Fabio Carciofi Junior. -- Bauru, 2025
30 f. : il., tabs., fotos

Trabalho de conclusão de curso (Bacharelado - Sistemas de Informação) - Universidade Estadual Paulista (UNESP), Faculdade de Ciências, Bauru
Orientador: Kelton Augusto Pontara da Costa

1. Ciência da computação. 2. Gestão de segurança. 3. Teoria dos grafos. 4. Processamento de linguagem natural (Computação). I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Dados fornecidos pelo autor(a).

Resumo

No passado, a internet já representava uma grande importância na economia. Segundo o relatório 1, havia 2 bilhões de pessoas online, com um crescimento de 200 milhões de usuários anualmente. A internet foi responsável por 21% do crescimento do PIB dos países: Estados Unidos da América, Brasil, Rússia, Reino Unido, França, Canadá, Suíça, Alemanha, China, Coreia do Sul, Japão e Índia. Hoje ainda se mantém a mesma importância, porém, segundo o relatório 2 de janeiro de 2023, havia 5,35 bilhões de usuários na internet, significando 66,2% da população mundial, evidenciando ainda mais a importância da internet de forma geral e econômica. Este trabalho de conclusão de curso propõe um método para detecção de *botnets* em redes de computadores, combinando técnicas de Processamento de Linguagem Natural (NLP) e Mineração de Grafos para análise de pacotes de rede. A abordagem visa superar as limitações computacionais dos sistemas tradicionais baseados em *Machine Learning*, oferecendo uma alternativa escalável para monitoramento de tráfego em tempo real. **Palavras Chaves:** Botnet, Mineração de Grafos, NLP

Abstract

In the past, the internet already represented a great importance in the economy. According to the report 1, there were 2 billion people online, with an annual growth of 200 million users. The internet was responsible for 21% of the GDP growth of the following countries: United States of America, Brazil, Russia, United Kingdom, France, Canada, Switzerland, Germany, China, South Korea, Japan, and India. Today, the same importance remains, but according to the report 2 from January 2023, there were 5.35 billion internet users, representing 66.2% of the world's population, further highlighting the overall and economic importance of the internet. This undergraduate thesis proposes a method for botnet detection in computer networks, combining techniques from *Natural Language Processing* (NLP) and *Graph Mining* for network packet analysis. The approach aims to overcome the computational limitations of traditional *Machine Learning*-based systems, offering a scalable alternative for real-time traffic monitoring. **Keywords:** Botnet. Graph Mining. NLP.

Lista de ilustrações

Sumário

0.1 Introdução

Desde a sua criação, a internet tem sido alvo de agentes maliciosos. Desde o primeiro vírus *worm*, Creeper, em 1971, até os dias atuais, esses agentes se utilizam de *botnets* e técnicas mais eficientes e potentes. Com uma crescente e dominante popularização, em 2019, cerca de 51% da população mundial tinha acesso à internet, segundo a União Internacional de Telecomunicações. O relatório *Bad Bot* de 2023, segundo o autor 2, determinou que 30% do tráfego malicioso é oriundo de *bots*, os quais, considerados avançados, têm imitado o comportamento humano, o que torna a sua detecção mais difícil. Portanto, este trabalho propõe um método inédito para lidar com a crescente ameaça representada pelas *botnets*, conforme indicado pelo relatório 3, que aponta que 47,3% do tráfego global é derivado de *bots*.

Desse total, 30,2% é tráfego malicioso, destacando-se: ataques distribuídos de negação de serviço (DDoS), exploração sistemática de vulnerabilidades e operações sofisticadas de exfiltração de dados. Tal realidade demonstra a necessidade de métodos de detecção que priorizem a minimização de recursos computacionais, elevada acurácia na identificação de anomalias e resiliência frente às constantes inovações das *botnets* e suas táticas.

0.2 Objetivos

O trabalho tem como objetivo principal cria e implementar um sistema de Detecção de Invasão que:

- Utilize técnicas de Processamento de Linguagem Natural do inglês,*Natural Processing Language* (NLP) para extração e classificação de *tokens* significativos em *payloads HTTP*;
- Integre algoritmos de mineração de grafos, com ênfase no *PageRank*, para mapeamento preciso de estruturas de comando e controle (C&C);
- Otimize o desempenho operacional em ambientes com restrições de hardware.

0.2.1 Objetivos Específicos

Para consecução do objetivo geral, estabelecem-se os seguintes objetivos específicos:

1. Realizar revisão da literatura sobre técnicas de *Machine Learning* aplicadas à detecção de intrusões;
2. Analisar criticamente as abordagens contemporâneas de detecção de *botnets* baseadas em mineração de grafos;
3. Adaptar e aprimorar a metodologia proposta por 4, eliminando a dependência de *Ensemble Learning*;
4. Implementar e validar comparativamente as técnicas de *Graph Mining* e NLP;
5. Desenvolver competência no uso da plataforma TensorFlow para análise comparativa;
6. Avaliar sistematicamente os resultados mediante métricas de desempenho;
7. Documentar conclusões e propor direções futuras para pesquisa.

O Trabalho de Conclusão de Curso está estruturado de forma a conduzir o leitor desde a contextualização inicial até a apresentação dos resultados finais. Após os elementos pré-textuais — compostos pela capa, resumo, abstract, lista de ilustrações e sumário — desenvolve-se o conteúdo principal, iniciado pela introdução, na qual são expostos o tema, a motivação e o problema investigado. Em seguida, são apresentados os objetivos geral e específicos, que orientam a condução da pesquisa. A fundamentação teórica reúne os conceitos essenciais para compreensão do método proposto, incluindo tópicos como Mineração de Grafos, Processamento de Linguagem Natural, algoritmos de clusterização e técnicas complementares. A seção de Materiais e Métodos detalha o conjunto de dados, as ferramentas utilizadas e os experimentos planejados, que são posteriormente

aprofundados na Metodologia, dividida nas Fases 1, 2 e 3. Após a descrição das fases, apresenta-se um sumário que sintetiza o processo metodológico, seguido da explicação da interface gráfica desenvolvida para facilitar o uso do método. Os resultados obtidos são então discutidos, permitindo avaliar o desempenho da proposta. Por fim, são apresentadas a conclusão e as sugestões de melhorias futuras, seguidas das referências utilizadas ao longo do trabalho.

0.3 Fundamentação teórica

A seguir serão apresentados conceitos de Aprendizado de Máquina que serão utilizados no desenvolvimento do trabalho de conclusão de curso. Em seguida, serão analisados alguns trabalhos recentes que apresentam temas semelhantes aos deste projeto proposto.

0.3.1 Mineração de Grafos

Dado que os conjuntos de dados, do inglês *Dataset*, apresentam dados estruturados, torna-se intuitivo o uso de Mineração de Grafos, pois os grafos representam facilmente as relações entre os dados, permitindo uma análise mais profunda e facilitada. No estudo 5, no qual foram feitas comparações e análises de vários métodos de Mineração de Grafos percebe-se a grande gama de possibilidade de uso dessa técnica, tanto possíveis como já realizadas.

0.3.2 Processamento de linguagem Natural

Em trabalhos semelhantes ao proposto dada a natureza da aplicação e a irregularidade dos *payload* [1] dos pacotes presentes na rede, esses têm seu conteúdo ignorado para o treinamento dos modelos. O uso dessas técnicas de extração de informação do conteúdo do pacote pode fornecer ganhos nas métricas de eficiência, assim como é evidenciado na pesquisa 6 na qual os dados foram adaptados e uma melhora dos resultados em comparação aos que apenas usam Mineração de Grafos. Essa técnica trata-se de transformar a informação dos *payloads* legíveis ao humano e extrair dados a partir das palavras presentes, originando assim a sigla do inglês, *Natural Processing Language*, NLP.

0.3.3 Markov Cluster Algorithm

O Markov Cluster Algorithm (MCA) é um algoritmo de clusterização baseado em grafos que é usado para identificar estruturas de cluster em conjuntos de dados representados como grafos. Ele opera através da simulação de um processo de difusão de Markov no grafo, onde os cliques densos de nós são tratados como estados estáveis do processo de difusão. O algoritmo itera entre duas etapas principais: expansão e inflação. Esses dois passos são executados até que haja definição e estabilização dos clusters de grafos.

Na etapa da expansão, o algoritmo simula a propagação de fluxo ou probabilidade através das arestas do grafo usando multiplicação de matriz. O objetivo é expandir o fluxo para destacar regiões densamente conectadas do grafo.

[1] Um pacote de rede tem duas partes: cabeçalho e *payload*. Cabeçalho possui campos relacionado ao trasporte do pacote enquando o *payload* é a parte responsável por carregar os dados.

Na etapa de inflação, os valores resultantes da expansão são ajustados para enfatizar ainda mais os clusters. Isso é feito aumentando exponencialmente os valores da matriz resultante da etapa de expansão.

0.3.4 Infomapa

Esse infomapa é retirado de uma pesquisa 7. A equação utilizada é retirada de 8. Nesse algoritmo há uma ênfase nas relações entre os nós do grafo, permitindo o encontro do fluxo de informação na rede via codificação ou compressão. A equação consiste na determinação de comunidades entre os nós dos grafos. O plano ou estado α_i é o primeiro e considerado a "superfície" enquanto β_j é o estado acima que representa outras informações, reduzindo visão confusa num único plano A equação é a decorrida abaixo:

$$P(x_{-m}\vec{\cdots}x_{-1}i \to x_{-m+1}\vec{\cdots}x_{-1}ij) = P_{\alpha_i\beta_j} = \frac{w_{\alpha_i\beta_j}}{w_{\alpha_i}}$$

Considerando que:

- $P_{\alpha_i\beta_j}$ é a probabilidade de transição do nó de estado α_i para β_j;
- $w_{\alpha_i\beta_j}$ são os pesos de ligação entre os nós de estado;
- w_{α_i} é o peso total dos outlinks do nó de estado α_i.

0.3.5 PageRank e HDBSCAN

A combinação de PageRank e HDBScan, começa com a etapa do primeiro algoritmo para criar grafos de comunicação e o segundo algoritmo filtra em agrupamentos os nós do grafo considerados suspeitos.

Esse algoritmo foi criado pela empresa Google para conectar páginas web com base na relevância e importância, atribuindo a cada nó um valor relativo aos outros, sendo considerado um ponto importante quando possui várias ligações e uma pontuação. Em resumo, esse se configura num grafo de comunicação com pontuação e consideração dos pesos das conexões entre nós.

O algoritmo GDBSCAN tem função de agrupamento, do inglês *clustering*, é baseado em densidade, logo o mesmo identifica regiões de maior concentração de dados e separa em *clusters* retirando ruídos como baixa densidade ou pontos isolados. Ele produz uma árvore hierárquica, permitindo a identificação de grupos de diferentes tamanhos, densidades e subgrupos.

0.3.5.1 Beautiful Soup

A biblioteca Beautiful Soup é uma ferramenta Python especializada em *Web Scraping* e análise sintática de documentos *HTML/XML*. Sua capacidade de navegação hierárquica na estrutura DOM (Document Object Model) e integração com outras bibliotecas de análise possibilitam o processamento eficaz de *payloads HTTP* em comunicações de *botnets*.

0.4 Materiais e Métodos

Nesta seção, apresentam-se os experimentos do trabalho de conclusão de curso.

0.4.1 Conjunto de Dados

Para o funcionamento do método será utilizado o *ISOT HTTP botnet Dataset*[2], uma combinação de conjuntos de dados disponíveis publicamente com o objetivo de facilitar e expandir o estudo científico na área de cibersegurança. Esse conjunto é disponibilizado pela *University of Victoria*.

0.4.2 TensorFlow

O *TensorFlow* é uma plataforma de código aberto para desenvolvimento de projetos de Aprendizado de Máquina. Esta plataforma será utilizada para comparar o método proposto com a aplicação da técnica *KNN-Means*. Ele oferece diversas ferramentas, bibliotecas e outros recursos para que desenvolvedores de sistemas de Aprendizado de Máquina possam criar seus programas 10 e aplicar técnicas de NLP.

0.4.3 Experimentos

Os experimentos deste trabalho visam utilizar técnicas de Mineração de Grafos e Processamento de Linguagem Natural para verificar se é possível obter resultados semelhantes aos da pesquisa 4, sem recorrer ao Aprendizado de Máquina na criação de um método de detecção de intrusão em redes.

A lista a seguir descreve as etapas do trabalho:

- Tratamento dos conjuntos de dados para remoção de *stopwords* e separação adequada do cabeçalho e *payload*;
- Aplicação de NLP nos *payloads* dos pacotes de rede que utilizam o protocolo *HTTP* e identificação da estrutura para isolamento dos *tokens*;
- Criação de um grafo de comunicação utilizando o um algoritmo que permite reconhecimento de *bots* na rede.

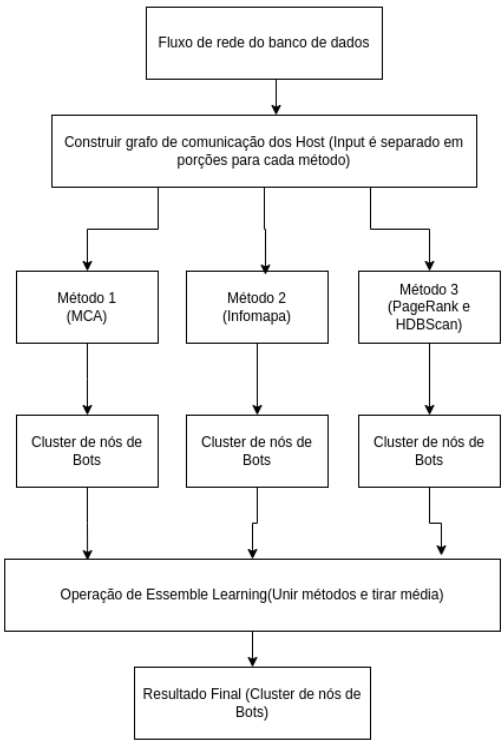
Após a separação do *payload* e cabeçalho dos pacotes de rede, será implementado um método de Detecção de Intrusão utilizando técnicas de Mineração de Grafos e NLP. A etapa seguinte consiste em processar o *payload* com técnicas de NLP e identificar padrões relevantes.

[2] 9

0.4.4 Soluções Existentes

Esta seção aborda algumas soluções existentes à proposta do trabalho de conclusão de curso.Cabe ressaltar que os trabalhos descritos foram publicados no penúltimo ano.
O principal trabalho que apoia este projeto foi desenvolvido em 2023 por 4, o método utilizado é representado na Figura 1. No artigo o estudo utiliza um modelo de Aprendizado de Conjunto e análise de grafos de três formas diferentes.Apesar de alcançar excelentes resultados nas métricas comumente utilizadas, essa abordagem não utiliza o *payload*, deixando espaço para melhorias ao introduzir o uso de NLP.

Figura 1 – Método proposto de Detection of Peer to Peer botnets using Graph Mining.



Fonte:Adaptado 4

Essa proposta utiliza os métodos de: *Markov Cluster Algorithm (MCA) Infomap*, *PageRank e HDBSCAN*, os resultados desses métodos são unificados por *Ensemble Learning*. Uma outra pesquisa 6, utiliza-se de tanto uma análise numérica, semelhante a maioria, dos pacotes e categórica, parte não expressa em números, assim como a análise do *payload* com um transformador pré-treinado o mesmo utiliza técnicas NLP. Essa pesquisa apresenta resultados superiores que outras pesquisas do mesmo campo.

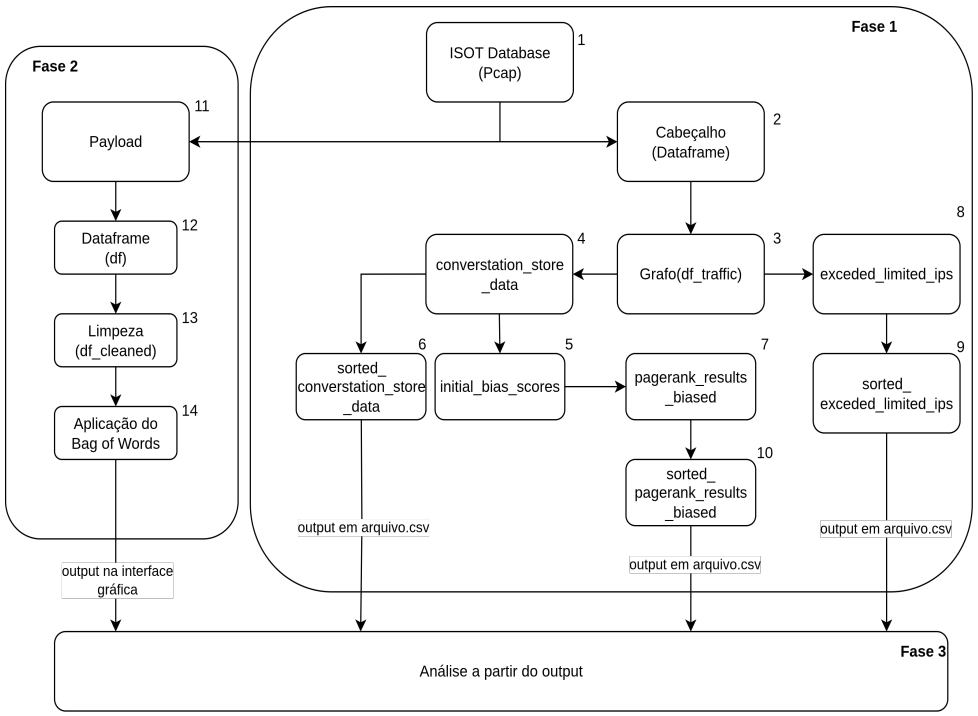
0.4.5 Metodologia

Esta seção apresenta detalhadamente a metodologia proposta, conforme ilustrado na Figura 2. A pesquisa tem como foco o uso completo do pacote de rede e de mecanismos de extração e aprendizado, oferecendo as seguintes vantagens:

- Maior aproveitamento das informações dos conjuntos de dados;
- Simplificação da análise de rede e pacotes;
- Redução do custo computacional;
- Facilidade de implementação e manutenção do método de detecção.

Essas vantagens decorrem da eliminação do Aprendizado em Conjunto e do Aprendizado de Máquina, o que dispensa a aplicação de múltiplos métodos para obtenção de métricas e reduz significativamente o custo computacional. Além disso, torna-se mais acessível a detecção de invasões de rede com resultados comparáveis ou superiores aos do estudo 4.

Figura 2 – Método proposto do Trabalho de Conclusão de Curso.



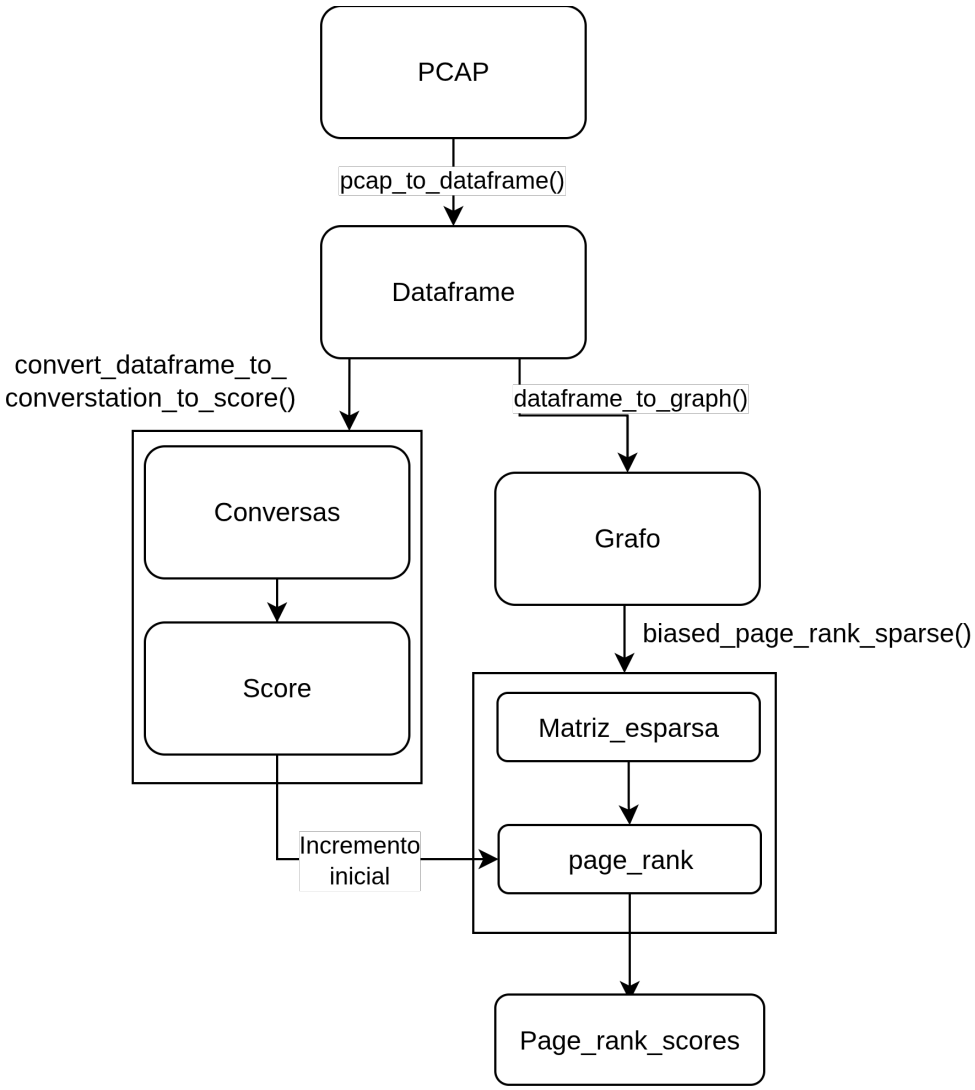
Fonte: Próprio Autor

A Figura 3 especifica em mais detalhes os fluxos de dados durante a Fase 1

0.4.6 Fase 1

A seguir, a explicação detalha o fluxo da Fase 1 e especifica o comportamento de cada função usada e ilustrada na Figura 4

Figura 3 – Diagrama de fluxo para PageRank.



Fonte: Próprio Autor

1. pcap_to_dataframe(pcap_file)

    - Essa função lê um arquivo de captura de pacotes (PCAP) e transforma em um *Dataframe*
        - Usa a biblioteca *pyshark* para ler os pacotes do arquivo;

- Extrai informações de endereço IP de origem (`src_ip`), IP de destino (`dst_ip`), portas, protocolo de transporte e tamanho do pacote (`packet_size`);
- Separa os dados em um *DataFrame* com o tráfego geral (*Dataframe*);

- Retorno: Uma tupla contendo *Dataframe* (tráfego geral).

2. dataframe_to_graph(df)

- Transforma um *DataFrame* de tráfego de rede em um grafo direcionado do *NetworkX*[3];
- O que faz:
  - Cria um objeto *nx.DiGraph()* (Grafo Direcionado);
  - Itera sobre as linhas do *DataFrame*(*df_traffic*);
  - Define os *IPs* de origem e destino como nós *(src, dst)*;
  - Cria uma aresta direcionada entre *src* e *dst*, usando o *packet_size* (tamanho do pacote) como peso da aresta.
- Retorno: O grafo direcionado (G) ponderado.

3. convert_dataframe_to_conversation_to_score

- Calcula um score de viés inicial para cada *IP* com base nos tipos de protocolos de rede utilizados em suas conversas.
- O que faz:
  - Define uma tabela de scores baseada no protocolo (*UDP* 0.05, *HTTP* 0.1, *DNS* 0.15);[4]
  - Rastreia todos os protocolos únicos usados por cada par de *IPs* em uma conversa;
  - Soma as pontuações de todos os protocolos únicos de uma conversa e adiciona esse total à pontuação de ambos os *IPs* envolvidos;
  - Limita o score de cada *IP* a um valor máximo de (0.3) e registra os *IPs* que excederam esse limite;
- Retorno: Um dicionário com as pontuações de conversa[5] (limitados) e um dicionário com as pontuações ou score originais dos *IPs* que excederam o limite (*exceeded_limit_ips*).

4. page_rank_sparse_biased(graph, score, beta, max_iter, tol)

[3] É uma biblioteca de *Pyhton* responsável por geração de grafos

[4] Os valores foram escolhidos com base de uma *botnet HTTP* e para que a soma seja até 0.30, pois valores maiores adicionados no vetor de importância do *PageRank* podem tornar muito enviesado

[5] Conversa é o termo utilizado para se tratar de um par único de *IPs*

- Calcula o *PageRank* de cada nó (*IPs*) do grafo, usando um vetor de viés inicial fornecido.
- O que faz:
  - Converte o grafo ponderado em uma matriz esparsa (**A**) para otimizar o cálculo;
  - Inicializa o vetor de *PageRank* (**r**) usando o *score* de viés fornecido (scores de conversa limitados), normalizado para somar 1;
  - Normaliza a matriz de adjacência (**A**) para criar a matriz estocástica (**M**);
  - Executa o processo iterativo do *PageRank*, onde a importância de um nó é calculada pela equação:

    $$\mathbf{r}_{novo} = \beta \cdot \mathbf{M} \cdot \mathbf{r}_{antigo} + \frac{1-\beta}{N}$$

    Onde N é o número de nós e o termo $\frac{1-\beta}{N}$ representa o teletransporte uniforme;
  - O processo para quando a diferença ($\|\mathbf{r}_{novo} - \mathbf{r}, \text{ord} = 1\|$) for menor que a tolerância (tol).
- Retorno: Um dicionário mapeando cada IP ao seu score final de *PageRank* (*pagerank_dict*).

0.4.7 Fase 2

A Fase 2 inicia após a conversão do arquivo PCAP para CSV (contendo a coluna *payload_raw*). O objetivo é perfilar o vocabulário de comunicação *HTTP* por endereço IP de origem. Essa conversão é possível pela aplicação associada ao trabalho

- Preparação e Limpeza de Dados;
  - Seleção de Colunas: O processo se concentra em duas colunas cruciais: a coluna de texto *TEXT_COLUMN*(*payload_raw*) e a coluna de IPs de origem *IP_COLUMN*(*src_ip*);
  - Limpeza: A função *df.dropna(subset=[IP_COLUMN])* garante que apenas registros com um IP de origem definido sejam mantidos, prevenindo erros na agregação;
  - Normalização: Valores nulos (*NaN*) na coluna de texto (*payload_raw*) são substituídos por ***strings* vazias**. Esta ação é essencial para que o *CountVectorizer* possa processar todas as linhas, mantendo os *IPs* que enviaram requisições ou respostas sem conteúdo no *payload*.

- Aplicação do CountVectorizer (Bag of Words); O *CountVectorizer* da biblioteca *scikit-learn* converte os *payloads* em uma matriz de *tokens*. Os parâmetros utilizados limitam a análise às **100 palavras mais frequentes** e desconsideram *stop_words*[6] do idioma inglês;
- Associação e Agregação por IP;
  - Criação do *Dataframe BOW*: A matriz esparsa gerada é convertida em um *dataframe* e associada à coluna *src_ip* original, estabelecendo a relação direta entre *IP* e os *tokens* encontrados;
  - Agrupamento: A função *df_ip_tokens.groupby(IP_COLUMN).sum()* agrega todos os pacotes de um mesmo *src_ip*, resultando na **soma total das contagens de cada *token*** para aquele endereço *IP*.
- Output de Resultados
  - Resultado Final: O *ip_token_summary* é exportado para CSV, apresentando o perfil de ocorrência de todos os 100 tokens para cada *IP* único analisado.

0.4.8 Fase 3

A Fase 3 diferentemente das Fases 1 e 2, é realizada pelo usuário do software utilizando-se dos arquivos CSV gerados e seu próprio conhecimento

[6] *Stop Words* são elementos linguísticos comuns, como 'the', 'is', 'a' e outras palavras de conexão, que não agregam valor analítico.

0.5 Sumario das fases

O trabalho tem 3 fases bem definidas, a primeira é a aplicação de um algoritmo de *PageRank* ponderado e enviesado considerando o comportamento de uma *botnet HTTP*. A segunda é focada na preparação e aplicação do *Bag of Words*, a terceira fase é a parte analítica do usuário final. A aplicação do trabalho acompanha os seguintes arquivos csv:

- *ip_token_summary.csv*;
- *biased_pagerank.csv*;
- Um arquivo csv extraído do arquivo PCAP de escolha do usúario;
- *exceeded_ips.csv*;
- *conversation_scores.csv*.

Todos esses arquivos fornecem em conjunto uma grande capacidade auxiliar ao usuário de detectar anomalias e comportamentos suspeitos na rede aplicada. Sendo respectivamente: sumário dos 100 *tokens* mais comuns, resultado da aplicação do algoritmo ponderando e enviesado de *Page Rank*, qualquer arquivo resultante da seleção do botão 'Extrair CSV', *IPs* que receberam um score de conversa acima de 0.30 e todos os scores de conversas limitados até 0.30.

0.6 Interface Gráfica

A interface gráfica é importante para a usabilidade do usuário final, a interface exemplificada na Figura 4 contém 5 botões, sendo esses:

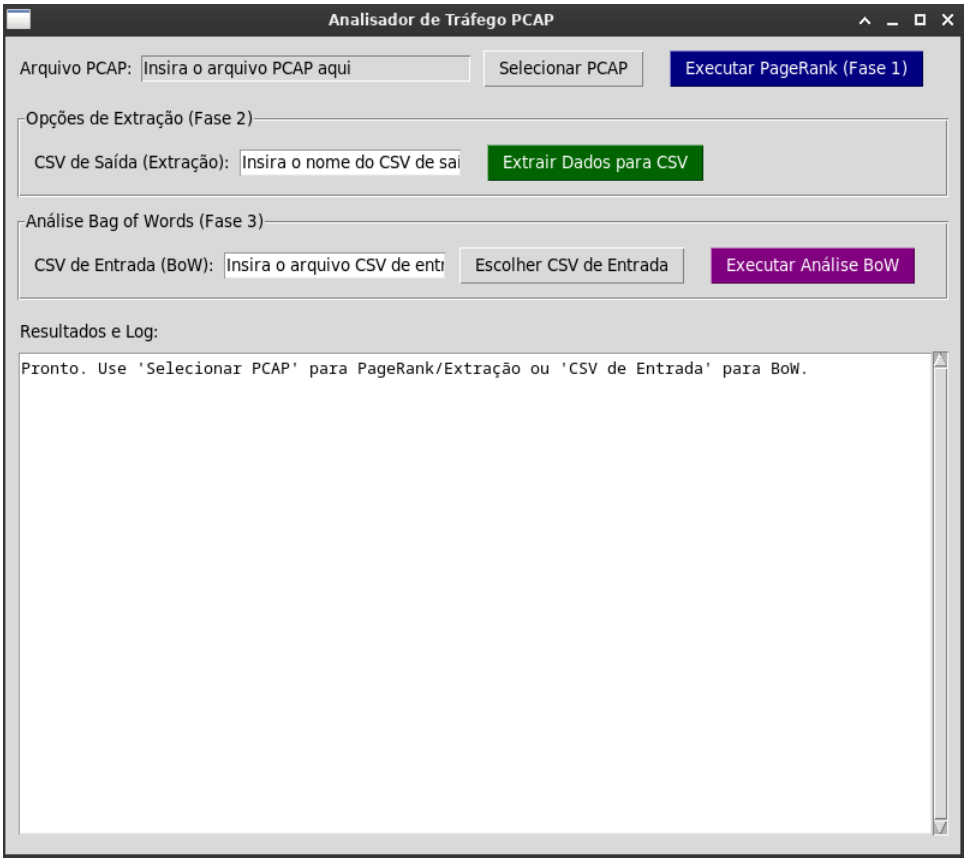
- Selecionar PCAP;
- Executar PageRank(Fase1);
- Extrair Dados para CSV;
- Escolher CSV de Entrada;
- Executar Análise Bow.

A interface também acompanha 3 caixas de entrada de texto, com os seguintes textos:

- Insira o arquivo PCAP aqui;
- Insira o nome do CSV de saída;
- Insira o arquivo CSV de entrada.

Por fim, resultados e logs são escritos e editáveis para o usuário final, assim como é possível a fácil cópia de dados diretamente da área de texto. A seguir, a Figura 4 apresenta a aplicação na visão do usuário final.

Figura 4 – Figura da Interface inicial;



Fonte: Próprio Autor

A Figura 4 apresenta os seguintes botões e suas respectivas ações:

- Selecionar PCAP;
  - Abrirá uma janela para acessar arquivos do computador;
- Executar PageRank(Fase 1);
  - Realizará o processo de *PageRank* e mostrará os 10 primeiros *IPs* com os maiores valores do algoritmo;
- Extrair Dados para CSV;
  - Realizará a extração dos dados necessários para a Fase 3;
- Escolher CSV de entrada;
  - Permite a entrada de um arquivo CSV para realização da Fase 3

- Executar Análise BoW
  - Realiza a Fase 3 e mostra todos os *IPs* detectados e os respectivos 10 *tokens* mais comuns.

0.7 Resultados

Os resultados do Trabalho de Conclusão de Curso é uma ferramenta capaz de utilizar um arquivo PCAP e gerar os seguintes itens:

- *ip_token_summary.csv*;
  - É um arquivo que relaciona os *src_ip* e os 100 *tokens* mais comuns associados aos *IPs*. Consideram-se os *IPs* de origem, pois o objetivo é identificar quem origina o conteúdo HTTP, assim, são mais evidenciados os possíveis servidores de comando e comunicação (C&C Servers), caso fosse considerado o IP de destino, seriam identificados os *bots* da rede e um considerável número de *IPs* internos. A Figura evidencia a estrutura do arquivo com apenas seis colunas e oito colunas; o arquivo possui 100 colunas.

Figura 5 – Estrutura exemplo do arquivo ip_token_summary.csv

src_ip	05a2a1d284b1f093d602cff1b16f23dd	1024	104	105	106
192.168.50.88	0	0	0	0	0
91.189.88.149	396	108	54	216	54
91.189.88.152	396	108	55	216	54
91.189.88.161	220	60	30	120	30
91.189.88.162	220	60	30	120	30
91.189.91.23	1957	452	360	263	524
91.189.91.26	2667	632	522	317	743
91.189.95.15	0	0	0	0	0

Fonte: Próprio Autor

- *biased_pagerank.csv*;
  - É um arquivo que relaciona os *src_ip* e o *PageRank* calculado de cada IP. Na interface gráfica são exibidos os 10 maiores valores. A Figura 5 exemplifica 8 *IPs* e seus respectivos valores; o arquivo possui todos os nós da rede, permitindo uma análise completa da hierarquia dos dados analisados.

Figura 6 – Estrutura exemplo do arquivo biased_pagerank.csv

IP	PageRank
186.217.119.190	0.24492929501161684
142.132.243.165	0.09333203048940493
157.240.226.60	0.0058460598307247...
35.186.224.30	0.0048183731727410...
64.233.186.108	0.00475207080770982
200.145.167.2	0.0044868613475849...
200.145.255.25	0.0042879542524913...

Fonte: Próprio Autor

- Um arquivo csv extraído do arquivo PCAP de escolha do usuário;
  - Na interface gráfica fica evidente esse output , o arquivo relaciona os campos típicos de análise para aplicações de NLP. A Figura 6 exemplifica o output e input para realizar o processo de aplicação *Bag of Words*. É importante considerar que *payload_raw* é utilizado como coluna principal na aplicação, pois os pacotes que não possuem esse valor não nulo são respostas, requisições e outras comunicações que não necessitam de um payload.

Figura 7 – Estrutura exemplo do arquivo gerado

src_ip	dst_ip	src_p...	dst_p...	proto...	packet_...	method	uri	host	user_a...	payload...
192.168.50....	91.189.88.1....	58416	80	http	74					
192.168.50....	91.189.91.26	57444	80	http	74					
91.189.91.26	192.168.50....	80	57444	http	74					
192.168.50....	91.189.91.26	57444	80	http	66					
192.168.50....	91.189.91.26	57444	80	http	276	GET				
91.189.88.1....	192.168.50....	80	58416	http	74					
192.168.50....	91.189.88.1....	58416	80	http	66					

Fonte: Próprio Autor

- *exceeded_ips.csv*;

  - Esse arquivo mostra os *IPs* que tiveram uma pontuação acima de 0,3 na Fase 1 e antes da inserção do viés inicial no vetor de importância do algoritmo de *PageRank*. Na Figura 7 é exemplificada a estrutura do arquivo. Esse output será naturalmente menor que os outros arquivos gerados pela aplicação, pois os *IPs* presentes aqui geralmente são nós da infraestrutura de rede ou nós que apresentam atividade suspeita.

Figura 8 – Estrutura exemplo do arquivo exceeded_ips.csv

IP	Score Original
224.0.0.251	2.000000000000001
186.217.119.190	1.5000000000000007
239.255.255.250	0.49999999999999...

Fonte: Próprio Autor

- *conversation_scores.csv*.

  - Esse arquivo mostra todos os *IPs* e seus correspondentes valores das conversas. Os valores presentes na segunda coluna desse arquivo são usados para gerar o viés inicial que é adicionado no vetor de importância do algoritmo de *PageRank*. Aqui é possível conferir quais *IPs* tiveram sua importância incrementada pelo uso dos protocolos *UDP*, *HTTP* e *DNS*. Na Figura 8 é exemplificada a estrutura do arquivo que contém todos os scores.

Figura 9 – Estrutura exemplo do arquivo *conversation_scores.csv*

IP	Score
186.217.119.190	0.3
224.0.0.251	0.3
239.255.255.250	0.3
186.217.173.72	0.1
186.217.114.36	0.1
186.217.113.115	0.1
200.145.219.242	0.1

Fonte: Próprio Autor

0.8 Conclusão

O Trabalho de Conclusão de Curso fornece dados suficientes para uma análise completa da rede de forma simplificada, ampliando o acesso e a segurança de redes com limitação de hardware. A otimização do desempenho se demonstrou uma dificuldade maior que o previsto dado a grande quantidade de informações processadas e analisadas, porém o desempenho atual é adequado e sua análise parcial[7] fornece informações precisas e permite uma detecção de invasão com mais facilidade.

0.8.1 Melhorias para o futuro

Considerando o cenário crescente de IoT de os resultados obtidos do Trabalho de Conclusão de Curso, é possível reconhecer melhorias para torna-lo mais abrangente em sua aplicação.

- Detecção Modular;
  - O trabalho é focado na detecção de uma *botnet HTTP*, seu uso seria expandido no cenário onde o usuário pode selecionar protocolos ou parâmetros na criação da pontuação de conversas e no processo de análise do NLP.
- Otimização;
  - A premissa do trabalho é reduzir o custo operacional, no cenário atual o software pode ser mais otimizado e ser retrabalhado para ser um processo que consiga processar mais dados.
- Implementação de análise.
  - A Fase 3 é dependente da ação do usuário final, esse pode ser leigo e não ser capaz de analisar os resultados ou compreender os protocolos e mecanismos de uma rede.

[7] Com arquivos que não contêm o cenário completo de uma rede

Referências

1 INTERNET Matters: O impacto sobre crescimento, emprego e prosperidade. 2011. Acesso em: 09 fev. 2024. Disponível em: <https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/internet-matters/pt-br>.

2 IMPERVA. *Bad Bot Report 2023*. 2023. Acesso em: 22 fev. 2024. Disponível em: <https://www.imperva.com/resources/resource-library/reports/2023-imperva-bad-bot-report/>.

3 KEMP, S. *Digital 2024 Global Overview Report*. 2024. Acesso em: 11 mar. 2024. Disponível em: <https://datareportal.com/reports/digital-2024-global-overview-report>.

4 BORAH, D.; SARMA, A. Detection of peer-to-peer botnets using graph mining. *International Journal of Computer Networks and Communications*, v. 15, p. 105–125, 03 2023.

5 REHMAN, S. U.; KHAN, A. U.; FONG, S. Graph mining: A survey of graph mining techniques. In: *Seventh International Conference on Digital Information Management (ICDIM 2012)*. [S.l.: s.n.], 2012. p. 88–92.

6 DüZGüN, B. et al. *Network intrusion detection system by learning jointly from tabular and text-based features*. 2023. Disponível em: <https://onlinelibrary.wiley.com/doi/10.1111/exsy.13518?af=R>.

7 EDLER, D.; BOHLIN, L.; ROSVALL, M. Mapping higher-order network flows in memory and multilayer networks with infomap. *Algorithms*, v. 10, n. 4, p. 112, 2017. Disponível em: <https://doi.org/10.3390/a10040112>.

8 ROSVALL, M.; AXELSSON, D.; BERGSTROM, C. T. The map equation. *The European Physical Journal Special Topics*, v. 178, n. 1, p. 13–23, 2009.

9 University of Victoria. *ISOT HTTP Botnet Dataset*. 2025. Acesso em: 04 nov. 2025. Disponível em: <https://onlineacademiccommunity.uvic.ca/isot/2022/11/27/botnet-and-ransomware-detection-datasets/>.

10 TENSORFLOW. 2024. Acesso em: 16 fev. 2024. Disponível em: <https://www.tensorflow.org/?hl=pt-br>.