Secflow: Aprendizado não supervisionado para análise e detecção de anomalias em Redes de Computadores

Felipe M. Salles[1], **Taiane C. Ramos**[1], **Luiz Claudio Schara**[2]

[1]Departamento de Ciência da Computação
Universidade Federal Fluminense (UFF) – Niterói, RJ – Brazil

[2]Departamento de Engenharia de Telecomunicações
Universidade Federal Fluminense (UFF) – Niterói, RJ – Brazil

sallesfelipe@id.uff.br, taiane_ramos@id.uff.br, schara@telecom.uff.br

Abstract. *Anomaly and intrusion detection systems arise from the growing concern with network data security. Such systems need to identify anomalies in network traffic. Characterizing the anomaly types allows us to identify vulnerabilities and propose strategies to mitigate attacks. This research proposes network traffic analysis methods for detecting anomalies through unsupervised learning. This model, applied to a sliding window, aims to classify flows in an active network to identify different types of traffic anomalies. The model developed in this work will be applied to the production network at Universidade Federal Fluminense.*

Resumo. *A preocupação com segurança de dados em redes levou ao surgimento dos chamados sistemas de detecção de anomalias e intrusão. Para isso, sistemas precisam identificar anomalias no tráfego da rede observada. Com a caracterização dos tipos de anomalia, podemos identificar vulnerabilidades e propor estratégias para mitigar ataques. Esta pesquisa tem por objetivo propor métodos de análise de tráfego de redes para detecção de anomalias por meio de aprendizado não supervisionado. Esse modelo, aplicado a uma janela deslizante, visa classificar fluxos em uma rede em funcionamento para identificar diferentes tipos de anomalias no tráfego. O modelo desenvolvido será aplicado à rede da Universidade Federal Fluminense.*

1. Introdução

O crescente aumento de sistemas finais nas redes contribuiu para que as trocas de informações entre estes dispositivos apresentassem riscos à segurança dos usuários. Sistemas de Detecção de Invasão (IDS, do inglês *Intrusion Detection System*) são utilizados para detectar possíveis ataques no ambiente de Redes de Computadores. Existem dois tipos de sistemas IDS: baseados em assinaturas e baseados em anomalias.

Apesar de sistemas IDS baseados em assinaturas serem amplamente utilizados, eles possuem algumas limitações, como exigir conhecimento prévio do ataque para criar assinaturas precisas e a possibilidade de alarmes falsos quando tráfegos que não são ataques correspondem à uma assinatura do conjunto de dados [Bay and Schwabacher 2003]. Por outro lado, um IDS baseado em anomalias cria um perfil de tráfego enquanto observa o tráfego em operação normal. Ele busca fluxos de pacotes que são estatisticamente

incomuns, como uma porcentagem irregular de pacotes ICMP ou um crescimento exponencial de análises de porta e varreduras de ping. Embora esse sistema também sofra com a questão dos alarmes falsos, uma vantagem é que ele não recorre a conhecimentos prévios de outros ataques, ou seja, tem potencial para detectar ataques que não foram documentados [Kurose and Ross 2017].

Ter um sistema de IDS é obrigatório para proteger redes críticas. No entanto, é difícil conseguir um conjunto de dados adequado para avaliar o sistema desenvolvido [Sharafaldin et al. 2019]. Em geral, os conjuntos de dados disponíveis estão desatualizados em relação às tendências atuais de ataques, além de sofrerem com a falta de diversidade de ataques no processo de classificação do tráfego [Su et al. 2018]. Nesse contexto, sistemas baseados na análise de fluxos são mais adequados para o monitoramento de uma rede, por analisarem o tráfego da rede em funcionamento [Hofstede et al. 2014].

Nesta pesquisa, optamos por inicialmente adquirir conhecimento do domínio do problema adotando uma abordagem simples para identificar anomalias [Yang et al. 2009]. Utilizamos como método o algoritmo de clusterização DBSCAN, que clusteriza os dados baseado em similaridade e identifica os pontos *outliers* como anomalias. Nossos resultados mostram que o DBSCAN é capaz de obter uma boa acurácia, porém, não identifica uma proporção satisfatória de ataques (alta taxa de falsos negativos). Nossos resultados preliminares sugerem a necessidade do uso de algoritmos de detecção de anomalias mais complexos para identificar ataques, como a detecção coletiva de anomalias [Wang et al. 2022].

Este projeto de pesquisa tem como objetivo final desenvolver um sistema de IDS para avaliar o tráfego da rede da Universidade Federal Fluminense (UFF). Além disso, pretendemos fornecer informações importantes para que os setores responsáveis possam implementar mais políticas para mitigar ataques que comumente são propagados na rede da universidade.

2. Método

Nesta pesquisa adotamos uma abordagem contínua de *Clustering* utilizando uma janela deslizante de forma a simular a análise de um tráfego contínuo de uma rede. A abordagem não supervisionada (*Clustering*) é vantajosa para detectar anomalias sem precisar previamente classificar os dados [Mirsky et al. 2017]. Optamos inicialmente por aplicar o algoritmo DBSCAN, pois é uma técnica que identifica anomalias (*outliers*) diretamente devido à sua abordagem de clusterização baseada em densidade. Acreditamos que testes com diferentes algoritmos e técnicas precisam ser feitos para avaliar quais algoritmos podem fornecer melhores resultados para identificar ataques em um tráfego de rede.

2.1. Banco de Dados

Nesta etapa da pesquisa, avaliamos a abordagem utilizando a base de dados pública UNSW-NB15 [Moustafa and Slay 2015]. Essa base de dados possui um conjunto híbrido de registros normais (2.218.761) e nove categorias de ataques, sendo elas: *Fuzzers* (24.246), *Analysis* (2.677), *Backdoors* (2.329), *DoS* (16.353), *Exploits* (44.525), *Generic* (215.481), *Reconnaissance* (13.987), *Shellcode* (1.511) e *Worms* (174) coletados de um tráfego normal [Moustafa and Slay 2016, Moustafa et al. 2019].

2.2. Pré-Processamento

A etapa de pré-processamento dos dados do conjunto UNSW-NB-15 foi iniciada extraindo 20% dos dados (aproximadamente 50.000 fluxos) para cada rótulo, incluindo os registros normais e de anomalia (ataque). Fazemos a padronização *Z-Score* para eliminar a influência da dimensão entre as *features* e para garantir o ajuste de cada *feature* a distribuição normal.

Utilizamos a Análise de Componentes Principais (PCA) para fazer uma seleção de variáveis. A seleção de variáveis reduz a redundância nas informações, além de reduzir a complexidade e o custo computacional. Alguns exemplos de *features* que selecionamos são: (protocolo de transação, número da porta, duração, bytes, contagem de pacotes, número de sequência TCP, hora de chegada do pacote e hora de início do registro). As *features* de maior relevância na Primeira Componente Principal foram selecionadas adotando um limiar de 0.05. Foram selecionadas 36 das 49 *features* presentes no UNSW-NB15.

2.3. DBSCAN com Janela Deslizante

O DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) é um algoritmo de *Clustering* que se baseia na densidade dos pontos para identificar *clusters* em conjuntos de dados. O algoritmo utiliza dois parâmetros principais: o raio de vizinhança (*epsilon*), que define a distância máxima para que um ponto seja considerado seu vizinho, e o número mínimo de pontos (minPts), que determina o número mínimo de pontos na vizinhança para que um ponto seja considerado um ponto central. Os pontos centrais, juntamente com os pontos vizinhos, formam um *cluster*.

A decisão de utilizar o DBSCAN partiu da ideia de que esse algoritmo consegue classificar os pontos que não atendem aos critérios de densidade mínima, chamando-os de anomalias. Esses pontos podem estar isolados, ou podem estar em posições de baixa densidade insuficientes para formar um *cluster*. Aplicamos o DBSCAN no nosso conjunto de dados em uma janela deslizante de tamanho 1000 e deslizamento de valor 10. Em cada janela, aplicamos o DBSCAN para identificar anomalias que possam ser caracterizadas como ataques em nosso conjunto de dados. O ajuste dos parâmetros do DBSCAN é de extrema importância para obter resultados satisfatórios e avaliamos o nosso modelo utilizando os seguintes valores: (*epsilon* = 1.5 e minPts = 2), (*epsilon* = 0.95 e minPts = 2), (*epsilon* = 1.5 e minPts = 5) e (*epsilon* = 3 e minPts = 10).

2.4. Métricas de Avaliação

Para avaliar a eficácia do algoritmo DBSCAN na detecção de anomalias em redes de computadores calculamos as métricas Verdadeiros Positivos (TP), Falsos Positivos (FP), Verdadeiros Negativos (TN) e Falsos Negativos (FN) para avaliar o desempenho do algoritmo em relação à classificação correta dos dados.

Para mensurar a eficácia da detecção, calculamos a acurácia para cada janela analisada. Essa métrica é obtida dividindo o número total de instâncias corretamente classificadas (TP e TN) pelo número total de instâncias avaliadas (TP, TN, FP e FN) em cada janela. Reportamos a acurácia média obtida em todas as janelas. Também apresentamos uma matriz de confusão reportando a média dos valores TP, TN, FP e FN obtidos em cada janela para cada combinação de parâmetros do DBSCAN.

3. Resultados Preliminares

A análise de anomalias em redes de computadores ao utilizar a abordagem do DBSCAN, nos permite avaliar a sensibilidade do algoritmo a diferentes valores de parâmetros. Nesta seção, apresentamos os resultados obtidos desta análise (Tabela 1), demonstrando os valores dos parâmetros que utilizamos e as respectivas taxas de acurácia média total das janelas e a matriz de confusão média para os respectivos parâmetros utilizados.

Tabela 1. Resultados Preliminares

Avaliação	Epsilon	MinPts	Acurácia Média	Matriz de Confusão Média
Avaliação 1	3	10	0.83	$\begin{bmatrix} 28.41(TP) & 98.10(FN) \\ 68.87(FP) & 804.60(TN) \end{bmatrix}$
Avaliação 2	1.5	2	0.72	$\begin{bmatrix} 28.55(TP) & 97.96(FN) \\ 178.84(FP) & 694.63(TN) \end{bmatrix}$
Avaliação 3	1.5	5	0.57	$\begin{bmatrix} 51.27(TP) & 75.24(FN) \\ 351.87(FP) & 521.60(TN) \end{bmatrix}$
Avaliação 4	0.95	2	0.43	$\begin{bmatrix} 53.81(TP) & 72.70(FN) \\ 492.17(FP) & 381.30(TN) \end{bmatrix}$

Analisando os resultados apresentados na Tabela 1, observamos que as configurações de parâmetros que apresentaram uma boa acurácia média não alcançaram uma boa taxa de acerto dos ataques. Essa disparidade é atribuída ao desbalanceamento das classes, já que a quantidade de amostras normais é maior do que as de ataques. Como nosso principal interesse é identificar os ataques, testamos outras configurações de parâmetros. Dentre os parâmetros explorados, a configuração que melhor identifica ataques (TP) obteve 53 TP com 72 FN por janela, o que ainda deixaria de identificar mais de 50% dos ataques. Esta configuração resulta em uma alta taxa de FP que pode prejudicar o desempenho do sistema e aumentar a sobrecarga da análise de fluxos. Valores maiores de *epsilon* aumentam a acurácia, pois os fluxos normais são corretamente agrupados em *clusters*, porém, diversos fluxos de ataques também acabam por ser agrupados e não são identificados como anomalias. Já valores pequenos de *epsilon* aumentam a detecção de ataques como anomalias, mas diminuem a acurácia por não agrupar diversos fluxos normais que serão identificados como anomalias.

4. Discussão

O objetivo deste estudo é desenvolver um método de monitoramento de rede capaz de avaliar fluxos em tempo real e identificar possíveis ataques. Inicialmente, optamos por identificar ataques utilizando um algoritmo simples de clusterização capaz de detectar anomalias. Visando o sistema final de monitoramento, escolhemos uma abordagem de janela deslizante capaz de avaliar novos fluxos juntamente com fluxos já avaliados. Como os tipos de fluxos contidos em uma janela podem variar, não sabemos previamente quantos *clusters* esperamos obter. Por isso, escolhemos o algoritmo DBSCAN que não exige

determinação prévia do número de *clusters*. Ao variar os parâmetros do DBSCAN, percebemos que um *epsilon* maior aumenta a acurácia devido a classificação correta dos TN, porém, há o aumento dos FN. Um menor valor de *epsilon* aumenta a quantidade de TP, porém diminui o TN, resultando em uma menor acurácia. Com base nos resultados preliminares obtidos, concluímos que a avaliação individual dos fluxos como anomalias dentro da janela deslizante não é uma estratégia adequada para identificar ataques em uma rede.

Na próxima fase do projeto, utilizaremos os mesmos métodos já mencionados para o pré-processamento, seleção de features e clusterização. Porém, vamos utilizar uma estratégia de detecção de anomalias coletivas [Wang et al. 2022]. Para cada conjunto analisado, verificaremos se existe algum fluxo que é um ataque, sem classificar individualmente os fluxos. Iniciamos com uma janela N contendo 990 fluxos que não apresentam anomalias após a clusterização do DBSCAN. Uma janela menor P recebe 10 fluxos novos. A janela a ser clusterizada (Figura 1) é formada por N + P (1000 fluxos). Assumimos que os fluxos em N são todos normais, portanto, se o DBSCAN identificar uma anomalia, o possível ataque está na janela P. Reportamos esses 10 fluxos de P para serem analisados pelo administrador da rede. Dessa forma, aumentamos a probabilidade de identificar um conjunto de fluxos que contenham um ataque na rede, pois capturamos padrões e comportamentos anômalos que podem não ser evidentes ao analisar cada fluxo individualmente. A ideia é reduzir a quantidade de dados que o administrador da rede precisa analisar maximizando a chance de que os ataques estejam entre os fluxos reportados.

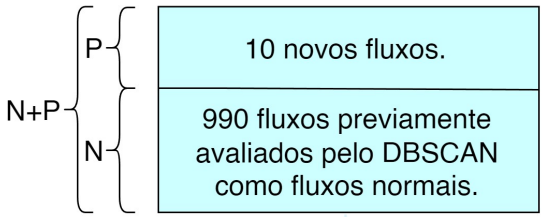

Figura 1. Janela Deslizante

Temos como objetivo aplicar a nossa abordagem no tráfego da rede da Universidade Federal Fluminense (UFF) para obter a caracterização dos ataques sofridos e mitigar ataques que são comumente propagados. Também pretendemos contribuir com novas melhorias nas abordagens já existentes de detecção de anomalias. Embora ainda não tenhamos todas as respostas, essa análise proporciona uma base sólida para investigações futuras, levantando questões que serão exploradas nas próximas etapas da pesquisa.

5. Conclusão

Este artigo, se propôs a fazer um estudo preliminar de monitoramento de rede através de identificação de anomalias aplicando o algoritmo DBSCAN. Os resultados mostram que essa abordagem não alcançou o nível de desempenho desejado na identificação de ataques. Diante disso, existe a necessidade de buscar metodologias mais eficazes para solucionar o problema. Assim, na próxima etapa deste projeto pretendemos desenvolver a abordagem de identificação de anomalias coletivas descrita na seção 4.

Exploraremos novas metodologias na detecção de ataques, buscando identificar de

forma mais eficiente os ataques (TP) e apresentar resultados melhores do que os conquistados com a abordagem inicial adotada. Este artigo representa apenas o primeiro passo em direção à detecção eficiente de ataques no tráfego de redes de computadores.

Referências

Bay, S. D. and Schwabacher, M. (2003). Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 29–38, New York, NY, USA. Association for Computing Machinery.

Hofstede, R., Čeleda, P., Trammell, B., Drago, I., Sadre, R., Sperotto, A., and Pras, A. (2014). Flow Monitoring Explained: From Packet Capture to Data Analysis With NetFlow and IPFIX. *IEEE Communications Surveys & Tutorials*, 16(4):2037–2064.

Kurose, J. F. and Ross, K. W. (2017). *Computer Networking: A Top-down Approach*. Pearson.

Mirsky, Y., Shabtai, A., Shapira, B., Elovici, Y., and Rokach, L. (2017). Anomaly detection for smartphone data streams. *Pervasive and Mobile Computing*, 35:83–107.

Moustafa, N. and Slay, J. (2015). UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *2015 Military Communications and Information Systems Conference (MilCIS)*, pages 1–6.

Moustafa, N. and Slay, J. (2016). The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Information Security Journal: A Global Perspective*, 25(1-3):18–31.

Moustafa, N., Slay, J., and Creech, G. (2019). Novel Geometric Area Analysis Technique for Anomaly Detection Using Trapezoidal Area Estimation on Large-Scale Networks. *IEEE Transactions on Big Data*, 5(4):481–494.

Sharafaldin, I., Lashkari, A. H., Hakak, S., and Ghorbani, A. A. (2019). Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy. In *2019 International Carnahan Conference on Security Technology (ICCST)*, pages 1–8. ISSN: 2153-0742.

Su, L., Yao, Y., Li, N., Liu, J., Lu, Z., and Liu, B. (2018). Hierarchical Clustering Based Network Traffic Data Reduction for Improving Suspicious Flow Detection. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 744–753. ISSN: 2324-9013.

Wang, C., Zhou, H., Hao, Z., Hu, S., Li, J., Zhang, X., Jiang, B., and Chen, X. (2022). Network traffic analysis over clustering-based collective anomaly detection. *Computer Networks*, 205:108760.

Yang, D., Rundensteiner, E. A., and Ward, M. O. (2009). Neighbor-based pattern detection for windows over streaming data. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '09, pages 529–540, New York, NY, USA. Association for Computing Machinery.