Detecção de ARP Spoofing utilizando aprendizado de máquina

EDUARDO H. F. DE SOUZA[1], ANDRÉ L. OLIVETE[2]

[1] Graduando em Ciência da Computação, IFSP, Campus Presidente Epitácio, eduardo.faustino@aluno.ifsp.edu.br.
[2] Docente Área de Informática, IFSP, Campus Presidente Epitácio, olivete@ifsp.edu.br.
Área de conhecimento (Tabela CNPq): 1.03.04.03-7 Software Básico

RESUMO:

O presente trabalho tem como objetivo o estudo e desenvolvimento de uma técnica para a detecção de ataques ARP *Spoofing* em redes de computadores, utilizando técnicas de aprendizado de máquina. O estudo iniciou pela análise das bases de dados existentes e a criação uma base de dados própria, resultante da captura de dados de um ambiente de utilização real. As redes neurais recorrentes como LSTM e GRU foram selecionadas para os testes, dada sua capacidade de lidar com sequências temporais. A principal contribuição deste estudo é a base de dados desenvolvida, os experimentos conduzidos, que serão avaliados por meio de métricas como precisão, recall e F1-score, e um sistema de monitoramento capaz de identificar em tempo real esse tipo de ataque.

PALAVRAS-CHAVE: arp; redes neurais; redes de computadores.

ARP Spoofing detection using Machine Learning

ABSTRACT: The aim of this work is to study and develop a technique for detecting ARP Spoofing attacks in computer networks using machine learning techniques. The study began with the analysis of existing databases and the creation of a proprietary dataset, resulting from data capture in a real usage environment. Recurrent neural networks such as LSTM and GRU were selected for testing due to their ability to handle temporal sequences. The main contribution of this study is the developed dataset, the experiments conducted—which will be evaluated using metrics such as precision, recall, and F1-score—and a monitoring system capable of identifying this type of attack in real time.

KEYWORDS: arp; neural networks; computer networks.

INTRODUÇÃO

A tecnologia está cada vez mais presente na sociedade, impulsionada pelos avanços em Inteligência Artificial (IA), que capacitam computadores a realizarem tarefas complexas, expandindo o uso da tecnologia para novas áreas. Com esses avanços, as redes de computadores, especialmente em contextos como a Internet das Coisas (IoT) e grandes data centers, enfrentam desafios de escalabilidade e segurança. O aprendizado de máquina surge como uma abordagem promissora para auxiliar na mitigação desses problemas, permitindo a identificação de padrões e anomalias que indicam atividades maliciosas, tanto em algoritmos supervisionados quanto não supervisionados (Müller & Guido, 2016).

O protocolo ARP (*Address Resolution Protocol*), essencial para o mapeamento de endereços IP na cama de rede para endereços MAC na cama de acesso, é vulnerável a ataques como o ARP *Spoofing*, que envolve a falsificação de endereços, resultando em ataques de negação de serviço e *man-in-the-middle* (Omar, Pinto & Saide, 2013).

Estudos recentes, como o de Camargo et al. (2021), demonstram a eficácia do aprendizado de máquina na detecção de anomalias em redes, justificando a escolha dessa abordagem para o presente trabalho, que busca desenvolver uma técnica para detectar ARP *Spoofing* utilizando aprendizado de máquina.

Diante das vulnerabilidades associadas ao ARP *Spoofing*, torna-se premente adotar medidas de detecção e mitigação para fortalecer a segurança nas redes. Observa-se, ainda, em trabalhos relacionados, que a IA tem sido amplamente empregada como uma alternativa eficaz para detectar e classificar intrusões em redes.

MATERIAL E MÉTODOS

O desenvolvimento do projeto iniciou com a coleta de dados, que envolveu tanto a análise de bases de dados previamente existentes quanto a criação de uma base de dados própria, obtida por meio de coleta de dados em um ambiente controlado no laboratório do Instituto Federal de São Paulo – Campus Presidente Epitácio.

Para as simulações, foram utilizadas 12 máquinas, cada uma com funções específicas, como interceptação de pacotes, geração de tráfego comum e execução de ataques de ARP *Spoofing*, utilizando ferramentas como Wireshark, Ettercap e arpspoof. O resultado foi uma base de dados robusta, composta por milhares de pacotes de rede, dos quais foram extraídos os pacotes ARP, considerados essenciais para a análise do ARP *Spoofing* (Vojtko, 2021). A figura 1 apresenta uma ilustração do ambiente em que a base de dados foi criada, onde as máquinas foram divididas nas três seguintes categorias: realização de ataques, captura de pacotes e tráfego comum na rede.

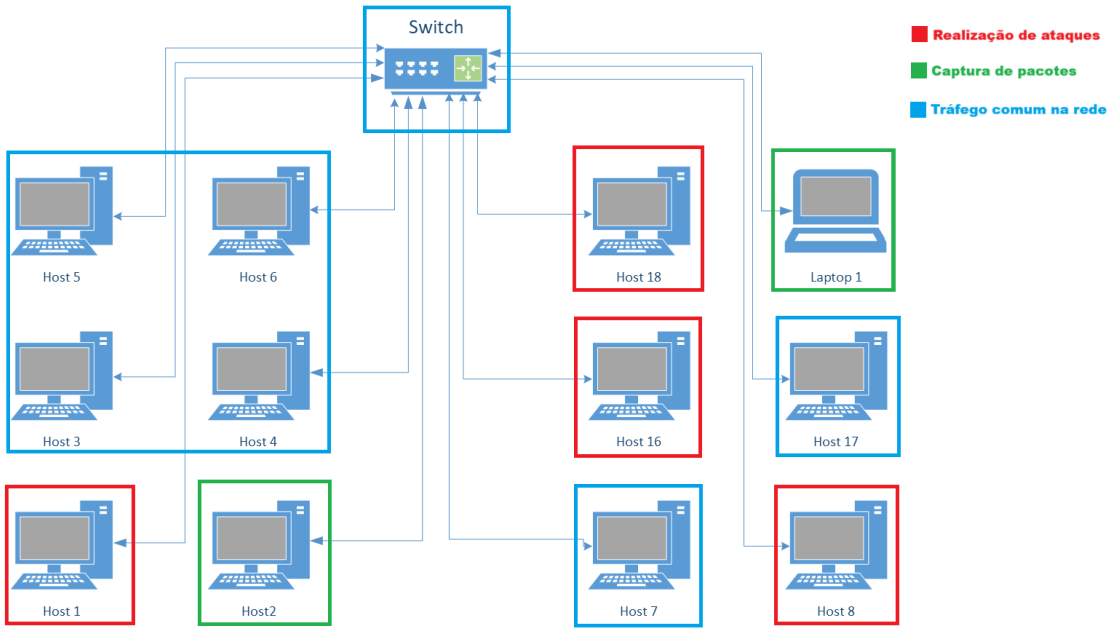

FIGURA 1. Ilustração do ambiente utilizado para coleta da base de dados.
FONTE: Autores.

Após a coleta dos dados, foi realizada uma etapa de pré-processamento para garantir a qualidade e adequação dos dados para os modelos de aprendizado de máquina. O pré-processamento envolveu a filtragem dos pacotes focando na detecção de ARP *Spoofing*, por ser um ataque simples onde o atacante falsifica a tabela ARP da vítima através do envio de respostas ARP falsas, onde somente os pacotes ARP são necessários para a detecção do ataque. Além disso, foram aplicadas técnicas de limpeza e transformação, como a generalização de valores específicos, para evitar que informações exclusivas da rede de origem tivessem um impacto desproporcional no treinamento, evitando o sobreajuste.

As variáveis categóricas também passaram por um processo de codificação utilizando a técnica de *one-hot encoding*. A tabela 1 apresenta o formato da base de dados utilizada para alimentar os modelos durante os testes, sendo que estes valores foram decididos com base nas informações de um pacote ARP.

TABELA 1. Dados coletados no formato definido no pré-processamento.

Time	Source MAC	Source IP	Destination MAC	Destination IP	Request	Reply	IsAttack
0.0	1	1	2	2	1	0	0
1.268927 e-08	2	2	1	1	0	1	0
0.5746517	7	1	2	2	0	1	1
1.0	7	243	2	2	0	1	1

FONTE: Autores.

Após essas etapas, foi implementada a criação de uma base de dados sintética para expandir o volume dos dados coletados. Este processo foi realizado após a separação dos pacotes ARP e a generalização dos endereços, sendo que os endereços de IPs foram alterados de forma randômica, exceto pelos endereços de ***gateway*** e ***broadcast***, que foram mantidos intactos para simular o comportamento de uma rede real. Essa abordagem foi crucial para aumentar a diversidade dos dados sem comprometer a autenticidade das interações de rede. Além disso, pacotes comuns de comunicação de rede, como pacotes TCP, UDP e ICMP, foram inseridos aleatoriamente entre os pacotes ARP, no intuito de simular o tráfego normal de uma rede.

Com esses processos de expansão e simulação do tráfego de rede, a base de dados resultante se tornou mais diversa e representativa de um ambiente de rede real, possibilitando ao modelo de detecção de *ARP Spoofing* um aprendizado mais efetivo e com menor risco de sobreajuste.

A fase de treinamento dos modelos foi realizada utilizando redes neurais, sendo realizados alguns testes com as redes MLP (*MultiLayer Perceptron*), porém testes também foram realizados com as redes recorrentes LSTM (*Long Short-Term Memory*) pois esta arquitetura se destaca por sua capacidade de processar dados sequenciais e de longo prazo, uma característica importante para a identificação de padrões temporais associados a ataques de ARP *Spoofing*. O treinamento dos modelos envolve a divisão da base de dados em conjuntos de treinamento e teste, permitindo que os algoritmos aprendam a identificar as características distintivas dos ataques.

A avaliação dos modelos foi realizada por meio de diferentes métricas, visando uma análise abrangente e rigorosa do desempenho dos algoritmos, onde a acurácia foi utilizada para medir o desempenho geral do modelo, enquanto as métricas de precisão e *recall* serão aplicadas para avaliar a capacidade dos modelos em identificar corretamente os ataques de ARP *Spoofing* e o *F1-Score* será utilizado para balancear a precisão e o *recall*, especialmente em contextos onde há um *trade-off* entre essas métricas. Adicionalmente, a matriz de confusão proporcionará uma análise detalhada das taxas de verdadeiros e falsos positivos e negativos, oferecendo uma compreensão ampliada do comportamento dos modelos.

Com a definição dos parâmetros da rede, será implementada uma aplicação de monitoramento da rede, que implemente um algoritmo de detecção de intrusões eficiente, capaz de identificar ataques de ARP *Spoofing* com precisão, contribuindo assim para a segurança das redes e a mitigação dos riscos associados a essas ameaças. A figura 3 apresenta um fluxograma do funcionamento do algoritmo de detecção em tempo real.

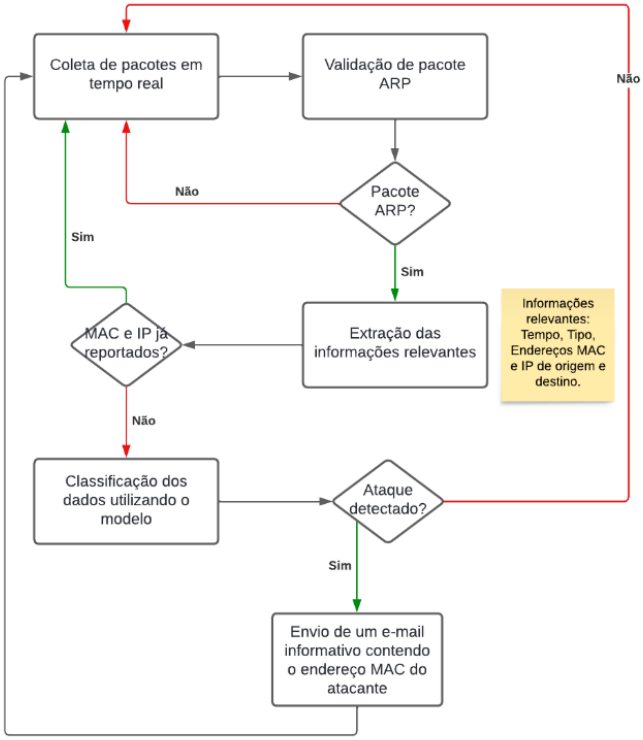

FIGURA 3. Fluxograma do funcionamento do algoritmo de detecção de ARP *Spoofing*.
FONTE: Autores.

RESULTADOS E DISCUSSÃO

Os resultados deste estudo apontam que a criação de uma base de dados própria, a partir de simulações controladas, é um fator determinante para o sucesso na detecção de ARP *Spoofing*. A base de dados capturada no laboratório, composta por 75.180 pacotes, possui uma diversidade superior em comparação com bases de dados existentes, como a "*poisoning_test*" (Ikken9, 2024) e a "*ARP_MitM_pcap*" (Mirsky, 2020), que apresentaram limitações em termos de variedade e representatividade dos ataques.

A base criada incluiu várias instâncias de ataques ARP *Spoofing*, abrangendo uma gama maior de condições e configurações de rede, o que permitirá um treinamento mais robusto dos modelos de aprendizado de máquina. A tabela 2 apresenta uma comparação entre as bases de dados encontradas e a base de dados criada por esse trabalho, que apesar de ter uma quantidade maior de ataques ainda não é suficiente, e será realizada uma nova captura com maior quantidade de ataques.

TABELA 2. Comparação das bases de dados escolhidas com a gerada.

Base de Dados	Total de Pacotes	Pacotes Maliciosos
poisoning_test	15675	456
ARP_MitM_pcap	2504267	1145272
Base Gerada	75180	2534

A literatura indica que muitas bases de dados públicas são limitadas em termos de variedade de cenários de ataque, o que pode comprometer a capacidade dos modelos de generalizar para diferentes situações de rede (Usmani et al., 2022). Enquanto bases de dados disponíveis foram eficazes em cenários específicos, elas não foram testadas de maneira efetiva em ambientes de rede mais diversificados. A base de dados própria, por sua vez, tem o intuito mitigar esse problema, permitindo uma detecção mais precisa em ambientes mais diversos.

Foram treinados dois modelos de redes neurais com a base de dados gerada: inicialmente, uma rede neural MLP, que apresentou ótimos resultados no conjunto de teste, com apenas 4 falsos positivos

e nenhum falso negativo, o que significa que nenhum ataque foi classificado incorretamente. Após essa etapa, uma rede neural recorrente LSTM foi treinada. Devido à sua arquitetura, a LSTM é mais adequada para dados sequenciais de longo prazo, resultando, em sua maioria, em um desempenho superior ao da MLP. A Tabela 3 compara os resultados obtidos pelas principais métricas de aprendizado de máquina.

TABELA 3. Comparação das redes neurais treinadas através de diferentes métricas.

Métricas	MLP	LSTM
Acurácia	99,347%	99,51%
Precisão	98,63%	99,65%
Recall	100%	99,3%
F1 Score	99,31%	99,478%
Falsos positivos	4	1
Falsos negativos	0	2

FONTE: Autores

Para o treinamento, a MLP foi configurada com 64 camadas ocultas, cada uma contendo 32 neurônios, enquanto a LSTM utilizou 50 camadas ocultas. Essas configurações foram escolhidas por representarem valores intermediários, evitando redes muito profundas ou superficiais, e os resultados satisfatórios justificaram a manutenção dessas configurações. Em ambos os casos, o conjunto de teste representou 20% da base de dados, e foi validado para garantir uma proporção equilibrada entre ataques e pacotes ARP legítimos. A Figura 2 ilustra a evolução da perda ao longo das 20 épocas de treinamento configuradas para as redes.

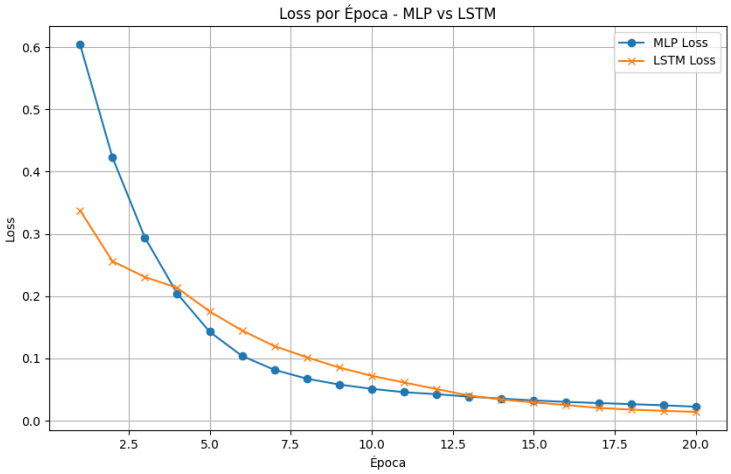

FIGURA 2. Gráfico de perda das redes neurais MLP e LSTM através das épocas.
FONTE: Autores.

CONCLUSÕES

A criação da base de dados própria possibilitou a obtenção de um conjunto variado e adequado para o desenvolvimento de modelos de detecção de intrusões. A base criada não só facilita o desenvolvimento de algoritmos capazes de identificar padrões característicos de ataques, mas também oferece um recurso para a comunidade científica em futuras pesquisas sobre detecção de intrusões.

A próxima fase do projeto envolverá o treinamento de uma rede neural recorrente GRU (*Gated Recurrent Unit)*, que compartilha características semelhantes à LSTM, mas que, em determinadas situações, pode oferecer melhor desempenho em termos de performance. Além disso, será essencial testar as redes neurais em diferentes bases de dados para verificar se o número limitado de pacotes não provocou um sobreajuste, o que poderia gerar uma falsa impressão da capacidade da rede de detectar ataques de ARP *Spoofing* em ambientes reais.

CONTRIBUIÇÕES DOS AUTORES

Eduardo H. F. de Souza: concepção, curadoria de dados, análise de dados, pesquisa, redação do manuscrito.

André L. Olivete: concepção, orientação, definição da metodologia e atividades, análise e validação dos resultados e correção dos textos.

AGRADECIMENTOS

A todos que participaram, direta ou indiretamente do desenvolvimento deste trabalho de pesquisa, enriquecendo o meu processo de aprendizado.

REFERÊNCIAS

Camargo, Luiz Felipe de; Reis, Carlos; Paiola, Pedro Henrique; Papa, João Paulo; Brega, José Remo F.; Costa, Kelton A. P. da. Métodos de Aprendizado de Máquina Adversariais na Detecção de Anomalias em Redes de Computadores. In: SIMPÓSIO BRASILEIRO DE SEGURANÇA DA INFORMAÇÃO E DE SISTEMAS COMPUTACIONAIS (SBSEG), 21., 2021, Belém. **Anais [...].** Porto Alegre: Sociedade Brasileira de Computação, 2021. p. 169-182. Disponível em: https://sol.sbc.org.br/index.php/sbseg/article/view/17314. Acesso em: 6 mai. 2024.

Ikken9. **ARP Poisoning Machine Learning**. GitHub. Recuperado em 13 abr. 2024, de https://github.com/Ikken9/arp_poisoning_machine_learning.

Mirsky, Yisroel. **Kitsune Network Attack Dataset**, version 1. Adquirido em: 10/04/2024 de https://www.kaggle.com/datasets/ymirsky/network-attack-dataset-kitsune/data.

Müller, Andreas C.; Guido, Sarah. **Introduction to Machine Learning with Python**: A Guide for Data Scientists. O'Reilly Media, Inc., 2016.

Omar, Leila Aquima Agy; Pinto, Celso Mahomed; Saide, Nacir Amir. **Criptografia e segurança de dados**. Maputo, Moçambique: Universidade Eduardo Mondlane, 2013. Disponível em: https://googlegroups.com/group/enginformaticadiurno2010/attach/49527ab9f5faaf14/Criptografia_ModeloOSI%20(DOPS).pdf?part=0.1. Acesso em: 8 abr. 2024.

Usmani, Mehak; Ahmed, Ghufran; Anwar, Misbah; Farooq, Komal; Siddiqui, Shahbaz. Predicting ARP Spoofing with Machine Learning. In: 2022 INTERNATIONAL CONFERENCE ON EMERGING TRENDS IN SMART TECHNOLOGIES (ICETST), 2022, Karachi, Pakistan. **Conferência [...].** [S. l.: s. n.], 2022. DOI 10.1109/ICETST55735.2022.9922925. Disponível em: https://ieeexplore-ieee-org.ez338.periodicos.capes.gov.br/document/9922925. Acesso em: 6 jan. 2024.

Vojtko, Mark. **Everything you need to know about ARP spoofing**. 2021. Disponível em: https://www.thesslstore.com/blog/everything-you-need-to-know-about-arp-spoofing. Acesso em: 3 jun. 2024.