

ARP Poisoning Detection Through Deep learning

AI Applications



Mohammad Gharbieh 202010009

GHAITH KHARASHGAH 202010470

# Introduction

The rising usage of interconnected devices demands robust security measures to safeguard against a ton of different kinds of digital threats. Among these, ARP poisoning stands out as a particular attack. By injecting fraudulent ARP messages into a network, attackers can reroute traffic towards themselves, enabling them to intercept confidential information, inject malware, or disrupt communication channels. This project, driven by the pursuit of enhanced network security, aims to develop a deep learning-based system capable of efficiently and accurately detecting ARP poisoning attempts in real-time.

# Dataset Design and Acquisition

The base of any successful machine learning project lies in its underlying data. For this project, ARP\_Dataset.csv will be used. This comprehensive dataset, carefully collected from a real-world network environment, encompasses a diverse range of features related to ARP packets, including:

* **Source and Destination MAC Addresses:** Identifying the physical devices involved in communication attempts.
* **IP Addresses:** Mapping devices to their logical network addresses.
* **ARP Opcodes:** Differentiating between ARP requests, replies, and various error messages.
* **Packet Rates:** Analyzing potential anomalies in network traffic patterns.
* **Switch IDs and Network Ports:** Pinpointing the location and origin of network traffic within the infrastructure.

This rich dataset, encompassing both normal and attack-ridden network behavior, provides the critical foundation for training and evaluating the deep learning model at the heart of this project.

# Deep Learning Architecture

The project leverages a hybrid convolutional neural network (CNN) and long short-term memory (LSTM) architecture to achieve exceptional accuracy in ARP poisoning detection. This powerful combination harnesses the strengths of both architectures:

* **Convolutional Neural Networks (CNNs):** adept at extracting intricate features from sequential data, like ARP packet streams. The CNNs within the model identify subtle patterns and anomalies within the raw data, highlighting potential malicious activity.
* **Long Short-Term Memory (LSTM) Networks:** capable of capturing long-term dependencies within data sequences. LSTMs within the model analyze the temporal dynamics of ARP traffic, identifying deviations from normal network behavior indicative of ARP poisoning attempts.

By combining the feature extraction capabilities of CNNs with the temporal sensitivity of LSTMs, the model achieves exceptional accuracy in efficiently identifying even the most discreet ARP poisoning attacks.

# Code Implementation

The project's code follows the most common pipeline for all neural networks mentioned below:

## Data Preprocessing:

* + The code meticulously loads the ARP\_Dataset.csv, handling missing values and outliers.
  + Categorical features are transformed into numerical representations for compatibility with the deep learning model.
  + The data is then split into training, validation, and test sets for robust model evaluation.
  + Finally, feature scaling techniques like StandardScaler ensure consistent data representation and optimal model performance.

## Model Architecture:

* + The code defines the hybrid CNN-LSTM architecture, specifying the number and configuration of layers.
  + Convolutional layers with varying kernel sizes extract diverse features from the data.
  + Max pooling layers reduce data dimensionality while preserving key features.
  + LSTM layers capture temporal dependencies within the data sequences.
  + Dense layers with appropriate activation functions map the extracted features to the final prediction of "normal" or "ARP poisoning."

## Model Training:

* + The code trains the model on the training set, utilizing efficient optimization algorithms like Adam.
  + Validation accuracy is continuously monitored to prevent overfitting and ensure generalizability.
  + The best-performing model weights are saved for subsequent evaluation and deployment.

## Model Evaluation:

* + The code rigorously evaluates the trained model on the unseen test set, calculating key metrics like accuracy, precision, recall, and F1-score.
  + Comprehensive analysis of these metrics provides insights into the model's strengths and weaknesses.

## Real-Time Detection:

* + The code leverages libraries like Scapy to capture live network traffic, focusing on ARP packets.
  + Captured packets are preprocessed and fed into the trained model for real-time prediction.
  + Suspicious activity exceeding a predefined threshold triggers an alert, notifying network administrators of potential ARP poisoning attempts.

## Logging and Analysis:

* + The code logs detected attacks along with relevant packet information, facilitating post-event

# Limitations and Challenges

While the deep learning model demonstrates considerable promise in detecting ARP poisoning attacks, it's important to acknowledge its limitations and potential challenges:

* **Data Dependence:** The model's accuracy relies heavily on the quality and diversity of the training data. Limited or biased datasets could lead to false positives or negatives.
* **Resource Requirements:** Training and deploying complex deep learning models can be computationally expensive, requiring specialized hardware and expertise.
* **Real-Time Performance:** Maintaining high accuracy while processing live network traffic at high speeds might pose challenges in resource-constrained environments.
* **Evolving Attack Techniques:** Attackers continually refine their methods, requiring the model to be regularly updated and adapted to stay ahead of the curve.
* **False Positives and Negatives:** The model might occasionally misclassify normal network behavior as an attack (false positive) or miss a genuine attack (false negative), requiring careful threshold tuning and analysis.

# Conclusion

This project presents a powerful deep learning-based system for real-time ARP poisoning detection. Despite its limitations, the model demonstrates a high degree of accuracy and paves the way for further exploration of deep learning in network security. Continuous monitoring, data augmentation, and integration with other security measures can enhance the system's robustness and effectiveness in safeguarding network environments. Overall, this projected demonstrated the power of combining cybersecurity with deep learning.