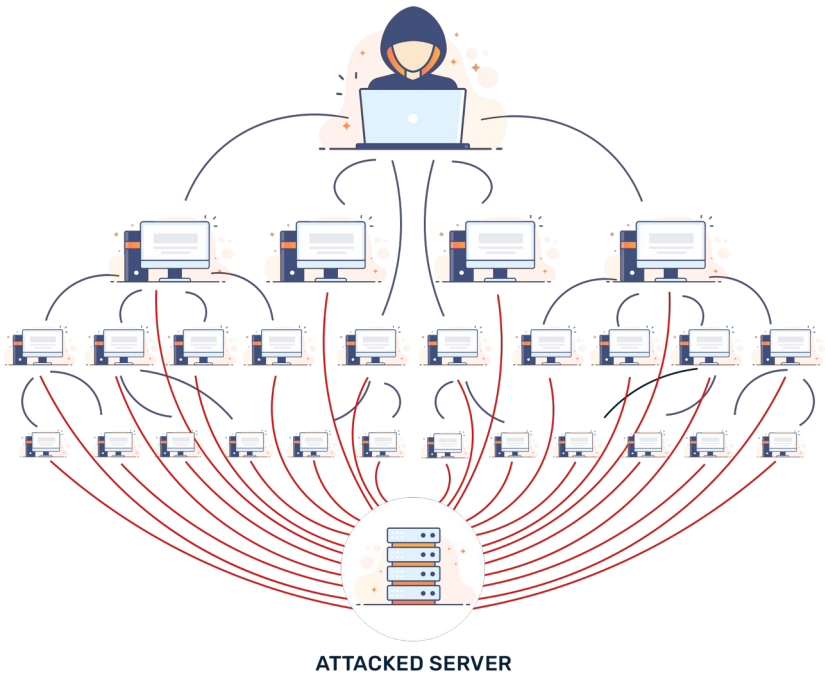
ML4N - Group Project 5

DDoS attacks detection and characterization

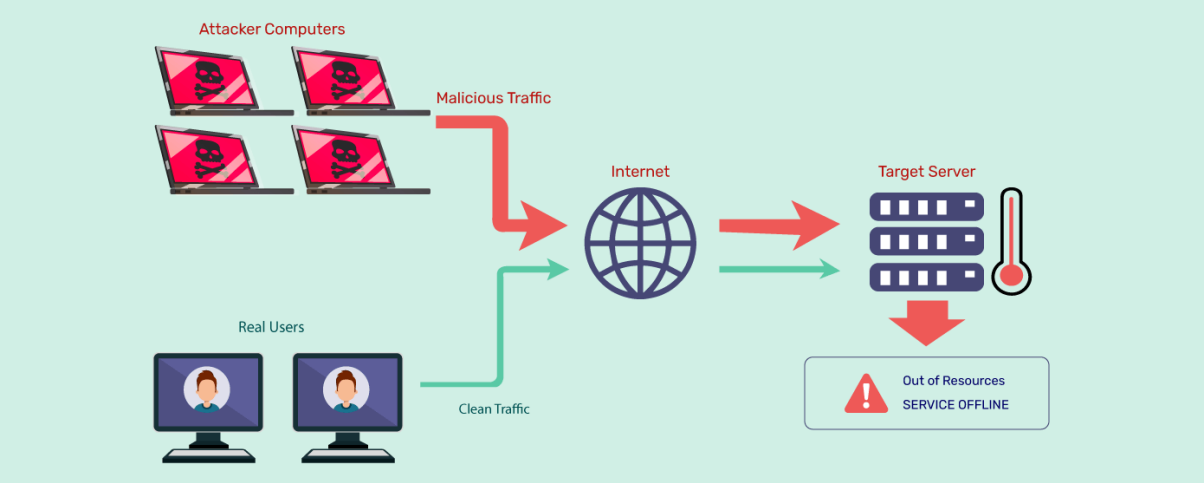
Clarifications for this project can be asked to **Zhihao Wang**: zhihao.wang@polito.it



Internet security is one of the most important challenges, especially when the demand for IT services is increasing every day. Among the many existing threats, DDoS (Distributed Denial of Service) attack is a relatively simple but very effective technique to attack intranet and Internet resources. Typically, this attack uses a large number of compromised machines to prevent legitimate users from using web-based services. DDoS attacks can be carried out at the network, transport and application layers using various protocols such as TCP, UDP, ICMP, and HTTP.
By assuming that different DDoS attacks exhibit different traffic patterns, researchers focused on the application of ML algorithms to detect and characterized such patters. Indeed, the automatic detection of DDoS attacks can ease the network monitoring activity of network administrators and allow to quickly take countermeasures.
The goal of this project is to develop a complete Machine Learning pipeline to automatize the detection and analysis of flows generated during DDoS attacks solving two tasks: (i) one supervised task, i.e., classification and (ii) one unsupervised task i.e., clustering. Through the analysis of the results coming from the two tasks you should be able to identify and characterize flows generated during a specific DDoS attack understanding, when possible, the attack behaviors.

The provided dataset contains benign and the most recent common DDoS attacks that resemble the real world data. It also includes the results of network traffic analysis using CICFlowMeter-V3 (https://www.unb.ca/cic/research/applications.html#CICFlowMeter) with labeled flows-based on timestamps, source and destination IPs, source and destination ports, protocols, and attacks. The dataset has been built replicating the behavior of 25 users based on the HTTP, HTTPS, FTP, SSH and email protocols.
In this dataset, you have different modern reflective DDoS attacks such as NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, SYN, NTP, DNS and SNMP. Attacks were subsequently executed during the data acquisition period.
The ground truth (GT) is made of the name of the attack a considered flow is referred to, i.e., the **label** column of the dataset.

For more details regarding the features check the file readme.md.



Section 1 – Data exploration and pre-processing

Data characterization and features engineering: explore the dataset and learn about the behavior of features at different levels, e.g., flow, ip, ports, etc. Then, select or generate the features to accomplish other ML tasks.

The first task of the project is to present the dataset through various data visualization techniques and statistical analysis.

1. **Investigate the provided DDoS attack dataset.**
    a) Produce different visualizations and statistical analysis both at the generic traffic level (e.g., number of flows, etc.) and GT level. (e.g., distribution of features, GT class characterization, ECDF of ports, flows, etc.) Can you observe the difference between normal traffic and various types of attack traffic from the visualizations?

2. **Generate additional features** e.g., quantifying the traffic related to each flow based on the previous analysis (e.g., avg, min, max, quantiles, etc.)
3. **Data Pre-processing.**
    a) Perform correlation analysis and visualization through PCA. If you think it could improve the tasks solution, you can do dimensionality reduction for generating the features used in the next tasks.
    b) Evaluate if you need to scale or standardize data, and if you need perform label encoding (e.g., one-hot) on some features. If you choose to do so, explain why.
4. **Characterize the new final feature matrix**, by producing plots regarding distributions of features (EPDF or ECDF), and correlation analysis.

Section 2 – Supervised learning – classification

The second task consists of classifying the flows to different types of attacks according to the flow features – supervised classification. You are provided by a ground truth containing the label of the attacks.

1. **Train Test Split.**
    a) Perform a split to segment the dataset into training and test dataset, in a stratified way with respect to the labels.
2. **ML model Training and Evaluation.**
    a) Choose at least 3 ML methods, and perform the model training, with default parameter configuration, evaluating the performance on both training and test set. Output the confusion matrix and classification report. Which model generates the best performance? Why does the performance vary among different attacks?
    b) Do you observe overfitting or under-fitting? If so, why does this happen and how to optimize it?
    c) Tune the hyper-parameters of the models through cross-validation. How does performance vary? What is your choice of the best model and hyper-parameter according to training cost (basically time) and performance?
3. **Result Investigation.**
    a) Investigate the False Positive and False Negative. Can you draw considerations about the misclassification in terms of features? Report your analysis and findings for the ones you consider the most notable samples.
    b) Can you explain the False Positive, False Negative, Recall, Precision in a real DDoS detection context? For example, which metric indicates the possibility to generate false alarms for normal traffic or miss some subtle attacks?

Section 3 – Unsupervised learning – clustering

In this task you will group flows that produce similar/correlated/coordinated patterns. The clustering will be done in an unsupervised fashion, independent of the labels (i.e., the attack

label) used in Section 2. The goal is to understand if there exist similar “families/groups” of attacks.

Choose at least 2 Clustering Algorithms, and for each of them:

1. **Determine the number of clusters.**
   This can be done using methods like the elbow method or silhouette analysis. Explain your reasoning.
2. **Performance Tuning.**
   a) Find the best hyper-parameters, if any.
   b) Evaluate the clusters through clustering metrics and performance indicators.
   c) Use visualization method to show how the data samples are clustered.
3. **Report a coarse analysis of the detected clusters** (e.g., ECDF of number flows per cluster, silhouette, etc.).

Section 4 – Clusters explainability and analysis

The fourth and final task is to examine the detected clusters and try to find new patterns. Characterize the found clusters in terms of features distribution and activity patters. Try to use GT labels to explain the patterns and behaviors of the clusters you generated in the previous section. Draw considerations about DDoS attack traffic analysis.

1. **Cluster Distribution of Traffic Types.**
   a) Ideally, we except the same type of traffic to be clustered into the same group. However, there are many uncertainties in the real traffic, so one cluster may include multiple traffic types and one type of flows may also distribute in multiple clusters. Analyze how the traffic types are distributed among clusters. What is ECDF of number of clusters assigned to each class?
   b) Are there clusters where all elements belong to a single class or very few number of classes? Remember that clustering is unsupervised, hence the GT should not be used in the clustering algorithm.
   c) Based on step b, can you determine whether clusters reflect the GT labels? Is there benign traffic with similar characteristics to malicious one?
2. **Feature Importance Analysis.**
   a) What are the most important features in the obtained clusters? You can use a method that provide feature importance or use an explainability technique, like permutation importance.
3. **Attack Pattern Investigation.**
   a) Can you identify sub-attacks which belongs to the same class but have different patterns? Which feature contributes to your identification?
   b) Which attacks are more similar? According to which feature? Can you explain why according to their definition or attack manner?

Project acknowledgment: Luca Gioacchini luca.gioacchini@polito.it and Giordano Paoletti: giordano.paoletti@polito.it