Identifying Brute Force Attacks with Network Packet Capturing and Support Vector Machines

Gabriel Hideki Stanzani Onishi

Computer and Information Science, Northumbria University
gabriel.onishi@norhtumbria.ac.uk

***Abstract*— This document shares an approach to detecting brute force attacks over a network using the CSE-CIC-IDS2018 dataset by analysing the packet count and average connection time over time and feeding the data to a Support Vector Machine (SVM) model. The final model was not able to identify the malicious connections, being biased to label most time frames as benign. Despite the subpar results, exploratory analysis and final model still reveal interesting findings about networking and cyber security.**

***Keywords*— Cybersecurity, Computer Networks, Brute Force Attack, Machine Learning, Support Vector Machines.**

I. Introduction

Cyber security has been a concern since the very inception of computer networks and protocols (Dave et. al, 2023). As technology advanced, so did threats of data breaches that could culminate in losses of millions of dollars for countries and companies alike (Venkatachary et. al, 2024). Therefore, understanding how these attacks work and learning to prevent them is crucial to keep a network safe.

One of the most common attacks, being responsible for over 19% of cyber attacks (Gauri & Ingole, 2018) is also one of the simplest: the brute-force attack. This threat consists of an attacker trying to connect to an authenticated server by simply trying out a large combination of keys.

The following study uses the CSE-CIC-IDS2018 dataset, created by the Canadian Institute of Cybersecurity (CIC) with the Communication Security Establishment (CSE). The dataset contains packets of a simulated network under different types of attacks. For the purposes of this study, only attacks to the Secure Shell (SSH) protocol and to the File Transfer Protocol (FTP) were considered.

SSH is a service that provides authenticated remote sessions to server terminals, allowing control of the machine by the creation of an encrypted tunnel (IETF, 2006). Similarly, FTP is a service for the transfer of data typically protected by a layer of authentication. Although the two serve different purposes, the authentication stage is the same, which means that a brute force algorithm can be used for both.

On a packet level, brute forcing can be recognized by a number of features. The attacker attempts to log into the network multiple times in a short period of time. That means that the amount of packet exchange is big - the attacker keeps trying keys to get in and the server keeps giving a negative response, whilst in a normal login the exchange is minimal, as the person already knows the key.

This paper aims to use such characteristics to build a Machine Learning Model based on Support Vector Machines (SVMs), a model that has been proven to work with time series (Sapankevych & Sankar, 2018).

II. Methodology

The project was made using Pyhton 3.12.5, whilst Poetry 1.8.3 was used for dependency management. Git and GitHub were used for version control, with the entire codebase available publicly at https://github.com/gabrielonishi/data-science-for-cyberattack-detection.

The exploratory analysis was made used Pandas 2.2.3 and Scikit-Learn 1.6.1 was used for Machine Learning models.

A. Dataset Selection

The first step of the project was to select a high quality dataset. One of the issues of the intersection of Machine Learning and Cyber Security relies on data privacy (CIC and CSE). As in order to create them, packets with every bit of information sent via the network are captured, it is clear why real world examples are not made publicly available. Therefore, finding a dataset that is both ethically created and faithful to how a cyberattack occurs on a regular network becomes a challenge.

Thankfully, academic centers all around the world try to replicate network topologies for benchmark datasets. One such authority is the Canadian Institute of Cybersecurity (CIC), that in partnership with Communication Security Establishment (CEC) created a source for training models against multiple network threats, making it one of the benchmarks for this purpose (Ghurab et. al 2021). The dataset is built upon a realistic network simulation with 50 attacking machines and 420 victim machines spread across five different organization profiles, containing a comprehensive size and with over 80 features for analysis.

B. Data Preprocessing and Feature Extraction

In order to avoid uploading big files to the cloud and avoid redundancy, the dataset download was made in-script. To run it, it is necessary to have installed the AWS CLI (as the data is being hosted on a AWS Bucket).

The dataset is organized in multiple days. Therefore, only the day concerning Brute Force attempts was selected (Wednesday, February 14th).

Once downloaded, some sanity checks were made to make sure that the download was successful and that there are no consistency problems.

The features selected were as following:

- Dst Port: Network port used;
- Protocol: Network protocol used;
- Flow Duration: Time between first and last packet exchange (no measurement unit specified);
- Timestamp: Date and time of the start of packet exchange;
- Tot Fwd Pkts: Total forward packets sent;
- Tot Bwd Pkts: Total backward packets sent;
- Label: Indicates if the connection was Benign or an attack, specifying the attack.

Data processing revealed there were faulty logs, with unmatching dates and negative flow duration. These lines were removed from the dataframe. For model training, the Label column was transformed using dummy variables, creating new columns columns 'Malicious', 'FTP-BrtueForce' and 'SSH-BruteForce'. Some data type correction also had to be made. The Pickle module was used to store dataframes and make them available across notebooks.

C. Exploratory Analysis and Additional Data Processing

Before analysing the data, some additional data processing took place. As both FTP and SSH services use the TCP transport protocol (IANA, 2024), only packets with this protocol were considered.

Three different dataframes were created for analysis from the filtered one. One is a direct copy of the original, one contains exclusively packets destined to the SSH port and one contains exclusively packets destined to the FTP port.

Another modification needed was to group the dataset by minute so it could be treated as a time series instead of a group of singular connections. The dataframes were reindexed to consider the duration of the whole capture. Packet count was summed, whilst the average of the flow time was considered (both were considered 0 when no packets were exchanged during the time frame). The label for the time series was considered 'Malicious' if at least one of the packets sent on the time frame was malicious.

With the datasets grouped by minute it is possible to paint a picture of what the packet exchange looked like on that day. A graph showing the amount of traffic considering all ports can be seen on Fig. 1.

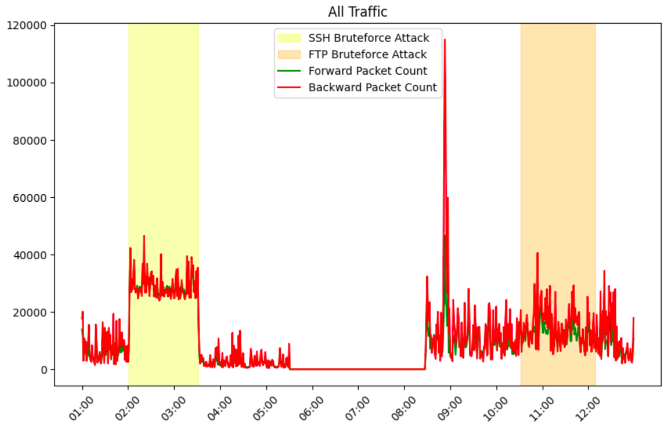

Fig. 1 A line graph showing Forward and Backward Packet Count on the day of the experiment. Areas with the times of attack are highlighted.

Although the time of the SSH attack is clearly marked by a spike in traffic, the same doesn't happen with the FTP attack. This is probably due to noise from other connections. Additionally, it is possible to see a clear spike around the 09:00 mark, although the dataset description specifies that no attack took place around that time.

Another feature that was examined was flow duration. This graph can be seen on Fig. 2.

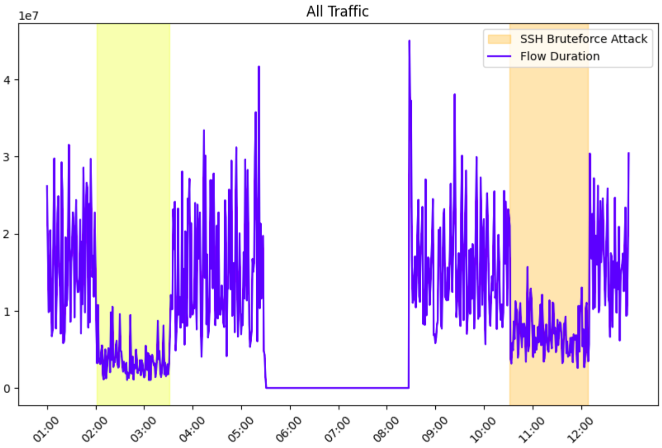

Fig. 2 A line graph showing the flow duration of packets on the day of the experiment. Areas with the times of attack are highlighted.

Differently from the last graph, a clear valley of average flow time can be found for both of the attacks.

Additional graphs made from the dataframes filtered by port can be seen on the appendix.

D. Model Training

When dealing with time series, it would not be interesting to split data like it is usually done in machine learning (that is, shuffling it). As these were the only examples of brute force attacks on the dataset, the only possibility was to use the SSH attack to predict the FTP attack. As there is a hiatus around the 6:30 and the 8:30 mark, where absolutely no packets were exchanged, the traffic until the last packet before the hiatus was chosen for the training set and the traffic after the first packet when connection returns was used for predictions.

III. Results

The results yielded from the sci-kit learn classification report can be seen in the Table 1 below.

TABLE I
Classification Report

Class	**Metric**		
	Precision (%)	**Recall (%)**	**f1-score**
0 (Benign)	65	97	78
1 (Malicious)	55	6	11
Accuracy	-	-	64
Macro Average	60	52	44
Weighted Average	61	64	54

The model was able to partially classify non-malicious traffic. About 65% of the instances predicted as non-malicious are correct, while the high Recall indicates that the model was able to identify almost all of the non-malicious traffic (98%).

However the results for the malicious traffic were poor. The model was able to identify 55% of the predicted malicious traffic correctly, but leaving behind 94% of the actual attack. The confusion matrix seen on Fig. 3 also helps to understand the results.

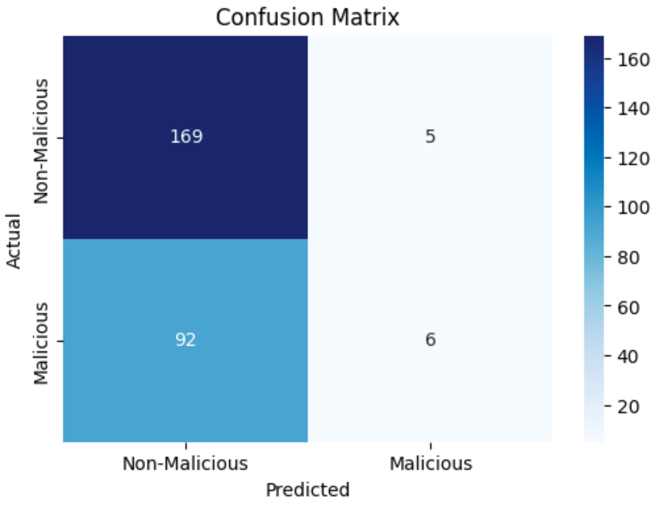

Fig. 3 Confusion Matrix for the SVM model

As previously noted, benign traffic identification was high, whilst malicious traffic identification was not. The model labeled only 5 minutes of the capture as being under attack, incorrectly labeling 6 minutes as under attack. The graph on Fig. 4 helps visualize those minutes.

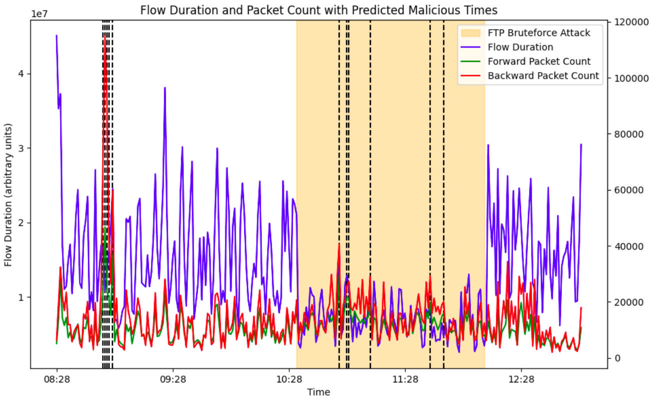

Fig. 4 Flow Duration and Packet Count across time after traffic hiatus. Areas of actual and predicted attack are highlighted.

IV. Discussion and Conclusions

The model was unable to label times of attack precisely. This is due to a number of factors, namely the features selected, how the data was compiled and dataset limitations.

Firstly, the exploratory analysis of packet exchange shows a clear difference of behaviour between the SSH attack and FTP attack. Whilst the former showed a clear spike of traffic, the latter didn’t, as pointed out on Section III. This indicates that the model has correctly taken significant variance on traffic as a distinctive feature. This is expected because in fact brute force attacks rely on multiple connections being made in a short moment of time. But that alone does not seem to be able to identify an attack, as the FTP scenario indicates. This is probably due to how the agents of the network behave. For instance, a script that makes multiple connections to the network might take place at a given time every day, creating a benign traffic like the one seen on the 09:00 mark.

Another issue might be related to how the data was grouped. A time frame of 1 minute was chosen arbitrarily, in an attempt to save up on memory and time training the model. This might not have been ideal, since the Brute Force attack relies on multiple connections made in a very short period of time.

Finally, the dataset, although a benchmark for the area of cybersecurity, might still have its faults. As there is only one day available for training, the understanding of how the network operates normally or when under attack is limited. A more comprehensive dataset might increase the model's ability to accurately identify attacks.

References

Apache.org. (2020). Introduction to the FTP Protocol - Apache HTTP Server Version 2.4. [online] Available at: https://httpd.apache.org/mod_ftp/ftp/ftp_intro.html.

CIC and CSE (2018). A Realistic Cyber Defense Dataset (CSE-CIC-IDS2018). Available at: https://www.unb.ca/cic/datasets/ids-2018.html.

Dave, D., Sawhney, G., Aggarwal, P., Silswal, N. and Khut, D. (2023). The New Frontier of Cybersecurity: Emerging Threats and Innovations. arXiv (Cornell University).

Gauri, M. & Ingole, R.Y (2018). A Review on Maintaining Web Applications and Brute Force Attack. International Research Journal Of Multidisciplinary Studies, Vol. 4.

Ghurab, M. et al. (2021) ‘A detailed analysis of benchmark datasets for network Intrusion Detection System’, Asian Journal of Research in Computer Science, pp. 14–33.

IANA - Internet Assigned Numbers Authority. (2025, January 16). Service Name and Transport Protocol Port Number Registry. https://www.iana.org/assignments/service-names-port-numbers

IETF - International Engineering Task Force (2006). The Secure Shell (SSH) Connection Protocol. Available at: https://www.ietf.org/rfc/rfc4254.txt [Accessed 16 Jan. 2025].

Sapankevych, N. & Sankar, R. (2009). Time Series Prediction Using Support Vector Machines: A Survey. IEEE Computational Intelligence Magazine, 4(2), pp.24–38.

Venkatachary, S. et al (2024). Cybersecurity and Cyber-terrorism Challenges to Energy-Related Infrastructures - Cybersecurity Frameworks and Economics – Comprehensive review. International journal of critical infrastructure protection.