Analysis and detection of malicious events in Network traffic

Tim Coleman colemat@colorado.edu
Anusha Basavaraja anba9017@colorado.edu
Christopher Bisbee chbi4702@colorado.edu

ABSTRACT

As IoT and network related devices keep increasing at a rapid rate, the attack surface also increases allowing malicious actors to gain access to systems and cause damage and/or extort money for personal gain.

This paper aims to understand a commonly known type of malware that is associated with botnet type of cybercrime. This type of malware normally operates as a command and control on the network of the infected where the infected device connects to the central control agent to receive instructions such as DDoS. This paper looks closely at the corresponding network traffic and uses data mining and machine learning techniques to try and predict whether a network pattern is likely malicious or benign.

PROBLEM STATEMENT / MOTIVATION

Cybersecurity is a large and growing area of research that spans a broad range of IT disciplines from securing and defending large corporate networks, computers, IoT devices, securing sensitive data, encryption best practices, access controls, user responsibility, etc., all of which are important to consider when implementing cybersecurity policy for an entity, such as corporation. However, even when following a comprehensive cybersecurity policy such as the NIST cybersecurity framework (https://www.nist.gov/cyberframework), mistakes happen and malicious events can occur which can be very costly in terms of money, time, and reputation. According to Douglas Thomas [3], it was estimated that US losses in 2016 ranged from $167.9 to $770.0 billion and have been growing ever since.

This paper investigates a specific type of cybercrime that is closely associated with malware that have participated in distributed types of attacks such as DDoS. These types of malware usually involve a command and control connection across the network in which they are connected to and are often referred to as botnets which consists of a collection of infected hosts that are controlled by a central agent.

Using techniques from data mining and machine learning, the goal is to try and analyze the network traffic and understand to see if there are network features that can be extracted that correlate to known malicious patterns such as botnets. And then answer the question, can these features be used to train a machine learning model to try and predict whether a network flow is malicious or benign?

LITERATURE SURVEY

The paper *Cyber Attack Detection thanks to Machine Learning Algorithms* [1] mentions using NetFlow data that does not contain sensitive information and is widely used by network operators. With proper analysis methods, NetFlow data can be an abundant source of information for anomaly detection. One major drawback to NetFlow has to do with the volume of data generated, particularly for links with high load. This could have an effect on the accuracy of the results and the amount of processing to be done. A lot of relevant work has already been performed to evaluate and process NetFlow data to address this. One method is by looking at NetFlow sampling (Wagner, Francois, Engel, et al. 2011). It further explains about incorporating Machine Learning techniques in malware detection. Machine learning capabilities are utilized by many applications to solve network and security-related issues. It can help project traffic trends and spot anomalies in network behaviour. Large providers have embraced machine learning and incorporated tools and cloud-based intelligence systems to identify malicious and legitimate content, isolate infected hosts, and provide an overall view of the network's health. Stevanovic and Pedersen 2014 developed a flow-based botnet detection system using supervised machine learning. Santana, Suthaharan, and Mohanty 2018 explored a couple of Machine Learning models to characterize their capabilities, performance and limitations for botnet attacks.

Machine learning has also been seen as a solution for evaluating NetFlows or IP-related data, where the main issue would be selecting parameters that could achieve high quality of results. There are multiple ML approaches mentioned in this paper such as Random Forest, Gradient-Boosted Trees, Classification voting based on Decision trees and NaiveBayes, Logistic regression.

Random Forest (RF) is a supervised machine learning algorithm that involves the use of multiple decision trees in order to perform classification and regression tasks (Ho 1995). The Random Forest algorithm is considered to be an ensemble machine learning algorithm as it involves the concept of majority voting of multiple trees. *Gradient boosting* is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It combines the elements of a loss function, weak learner, and an additive model. *Support Vector Machine* (SVM) is a supervised learning model used for regression and classification analysis. It is highly preferred for its high accuracy with less computation power and complexity. SVM is also used in computer security, where they are used for intrusion detection. *Logistic regression* is a supervised learning model that is used as a method for binary classification. The term itself is borrowed from Statistics. At the core of the method, it uses logistic functions, a sigmoid curve that is useful for a range of fields including neural networks. Logistic regression models the probability for classification problems with two possible outcomes. Logistic regression can be used to identify network traffic as malicious or not.

The paper also mentions using the CTU-13 dataset, one of the datasets we are also using for our project. The paper provides more ideas on *Feature extraction* and *Feature selection*, the major steps in our project. According to the paper, a popular idea is to summarize data inside a time window. This is justified by the fact that botnets "*tend to have a temporal locality behavior*" (Garcia et al. 2014). Moreover, it enabled them to reduce the amount of data and to deliver a botnet detector which gives live results after each time window (in exchange for input data details). The extracted features try to represent the NetFlow communications inside a time window. To do so, the time window data is gathered by source addresses. There are 2 features extracted from each categorical NetFlow characteristics: the number of unique occurrences in the subgroup and the normalized subgroup entropy. As for the numerical NetFlow characteristics, 5 features are extracted: sum, mean, standard deviation, maximum and median. These extracted features enable training of different ML models. They have used *Filter* and *Wrapper* methods for Feature selection. The *Filter method* involves filtering features in order to select the best subset of features for training the model. The best features are selected based on scores of Pearson Correlation statistical test since the pre-processed data involves continuous data. The *wrapper method* involves testing different subsets of features on a training model in an iterative fashion in order to determine the best features based on the inferences made by the training model. During each iteration, features are either added to or removed from the subset which will then be used to train the model. This is repeated until no more improvement is observed.

The paper *Machine Learning-Based Early Detection of IoT Botnets Using Network-Edge Traffic* [2] presents a lightweight IoT botnet detection solution, EDIMA, which is designed to be deployed at the edge gateway installed in home networks and targets early detection of botnets prior to the launch of an attack. EDIMA includes a novel two-stage Machine Learning (ML)- based detector developed specifically for IoT bot detection at the edge gateway. The ML-based bot detector first employs ML algorithms for aggregate traffic classification and subsequently Autocorrelation Function (ACF)-based tests to detect individual bots. The EDIMA architecture also comprises a malware traffic database, a policy engine, a feature extractor and a traffic parser. The EDIMA is designed to have a modular architecture with following components: *Feature Extractor*, *ML-based Bot Detector*, *Traffic Parser/ Sub-sampler*, *Malware PCAP Database*, *ML Model Constructor* and *Policy Engine*. Each of these components play their roles in botnet detection. The Feature extraction ML-based bot detector and ML Model Constructor are of main points of interest to us. In a traffic session, they have extracted features from TCP packet headers only and not the payloads. This method works on unencrypted as well as encrypted traffic. Once the features are extracted, the ML model will be trained on it for the Classification task to identify whether the network traffic is *benign* or *malicious*. Once the aggregate traffic at an edge gateway has been classified as malicious, the second fine-grained stage of the ML-based bot detector attempts to detect the underlying bots by checking the ingress/egress traffic from each IoT device for the presence of bot-CnC communication patterns. The set of IoT devices is sorted according to their IP addresses. Performance evaluation results show that EDIMA achieves high bot scanning and bot-CnC traffic detection accuracies with very low false positive rates. The detection performance is also shown to be robust to an increase in the number of IoT devices connected to the edge gateway where EDIMA is deployed.

The paper *Intelligent Mirai Malware Detection for IoT Nodes* [6] develops and compares the performance of Random Forest (RF) and Artificial Neural Network (ANN) models trained using a merged dataset of benign and malicious network traffic data across seven IoT devices, including commonly used devices such as indoor and outdoor security cameras, a Phillips Monitor, and a Samsung Webcam. The malicious traffic data contains Mirai malware which can be used to turn networked devices running Linux into bots for DDOS attacks. The features of the dataset used to train, cross-validate, and test each model are gathered from network traffic moving between source and destination hosts. This paper relies on another paper [7] for feature extraction to identify the most revealing of the 115 features in the dataset (N-BaIoT, which captures normal network traffic patterns) across five different time windows and uses Matlab to create the ANN model and Python to create the RF model. When cleaning the data, the authors note that the most difficult problem was aligning time windows across datasets during the merger to ensure that a proper comparison can be made between benign and malicious network traffic occurring concurrently. Also noteworthy was the authors' use of Microsoft Excel to import the dataset into Matlab and also "for modification of data and for initializing the features of the data." Excel is a fine tool for most purposes but is prone to introducing errors during data analysis so analysts and researchers must be especially careful to avoid or cross-check data that they work on in Excel to ensure that no errors or inconsistencies have been introduced. When comparing the RF and ANN models, the authors found that the ANN model with two hidden layers performed better than the RF model with an accuracy of 92.8%, a False Negative percentage of only 0.3%, and an F-1 score of 0.99.

BACKGROUND

Malware is defined as any software that can be installed and can run on a host that is designed to inflict some form of damage such as disabling, deleting, encrypting, stealing, crypto mining, controlling, etc.. Malware comes in different forms and usually are targeted for a specific purpose. Common types of malware include viruses, worms, trojan horses, and blended malware that combine several forms into one.

This paper focuses on a type of malware associated with a botnet cybercrime, which essentially refers to a collection of network devices that can be controlled from a command control center and/or peer-to-peer and carry out specific actions such as engaging in a DDoS attack. The malware installed on the device itself can stem from a virus, worm or trojan horse, but they all perform a similar function in that it allows the attacker to control the device and turn these devices into what is commonly known as "zombies."

The goal of the paper is to better understand the type of network connections commonly associated with botnet malware, and determine if there are features that can be filtered and trained to determine and predict if the traffic pattern is malicious or benign.

PROPOSED WORK

The goal of this project is to develop a machine learning model to try and predict traffic patterns that may be malicious and associated with botnet type malware. Data sets provided by Stratosphere Laboratory such as the CTU-13 and IoT23 contain classified data sets that contain different network captures where each scenario represents a start and end capture time.

The first step will be to transform any network capture data from .pcap file format into NetFlow v5 using python based networking libraries. Once the data is all in the same format, data exploration such as generating statistical features such as max, mean, min, standard deviations for numerical attributes to understand any interesting patterns. In the case of NetFlow v5, these could be the total number of packets and total bytes in the payload. Other data exploration techniques such as producing histograms and illustrating the distribution of total flow connection time, destination ip addresses can also be revealing and open up new questions to answer.

The next step would be to filter down the number of features using techniques such as using chi-squared and Pearson's correlation coefficient to determine the best features to extract from the NetFlow data set.

Once the data is reduced to the feature selected, ⅔ part of the data will be used to train a machine learning model and the other ⅓ will be used to test the model. Other training methods such as k-fold cross validation can also be used to train and test the machine learning models.

DATASET

The Stratosphere Laboratory is the provider of the following datasets that we have selected for our project: https://www.stratosphereips.org/team,

Dataset 1: *CTU-13*
https://www.stratosphereips.org/datasets-ctu13
Dataset 2: *IoT23*
https://www.stratosphereips.org/datasets-iot23

NetFlow data is a type of network protocol that was designed by Cisco in the mid 1990s to collect data from network traffic as it entered and exited from network interfaces. This type of network data is used by many IT professionals vs raw network captures as it's reduced into manageable data chunks and easier for analysis. Below is a reference format of NetFlow v5, which is the format used for this paper. The IoT23 data set is in raw NetFlow capture format that will need to be converted to NetFlow v5.

NetFlow v5 format:

Bytes	Contents	Description
0-3	srcaddr	Source IP address
4-7	dstaddr	Destination IP address
8-11	nexthop	IP address of next hop router
12-13	input	SNMP index of input interface
14-15	output	SNMP index of output interface
16-19	dPkts	Packets in the flow
20-23	dOctets	Total number of Layer 3 bytes in the packets of the flow
24-27	First	SysUptime at start of flow
28-31	Last	SysUptime at the time the last packet of the flow was received
32-33	srcport	TCP/UDP source port number or equivalent
34-35	dstport	TCP/UDP destination port number or equivalent
36	pad1	Unused (zero) bytes
37	tcp_flags	Cumulative OR of TCP flags
38	prot	IP protocol type (for example, TCP = 6; UDP = 17)
39	tos	IP type of service (ToS)
40-41	src_as	Autonomous system number of the source, either origin or peer
42-43	dst_as	Autonomous system number of the destination, either origin or peer
44	src_mask	Source address prefix mask bits
45	dst_mask	Destination address prefix mask bits
46-47	pad2	Unused (zero) bytes

https://www.cisco.com/c/en/us/td/docs/net_mgmt/NetFlow_collection_engine/3-6/user/guide/format.html

EVALUATION METHODS

To understand how well the machine learning model predicted the traffic as malicious or benign, the three common metrics Precision, Recall and Accuracy will be used. However, since the vast majority of the traffic is classified as benign, the data will have imbalances and be biased towards benign, so accuracy metrics will not be as useful in this report.

PRELIMINARY RESULTS

Data Preprocesssing:

The CTU-13 data set is formatted using netflow version 5, however, another column "Label" was added by the CTU-13 data maintainers as a way to distinguish if that particular netflow originated from a botnet. Since this label is not a boolean value and represented many different combinations of names, code was developed to include another data column to represent a binary classification to easily distinguish between malicious or benign as represented as boolean value.

In addition to adding another column, data cleaning was required for particular columns of data as non-numeric data existed in some columns such as NaN and/or corrupted data that had alpha characters when numeric was expected causing run-time errors when manipulating and casting to integers or floats. This was remedied by using built-in pandas regex to replace occurrences in columns with suitable replacement values.

Data Visualization of CTU-13 dataset:

Out of 13 scenarios in CTU-13 dataset, each of them consists of over millions records (netflow data objects) and the presence of percentage of botnet in each scenario is listed below:

Scenarios	***Percentage of Botnet***
Scenario - 1	1.451
Scenario - 2	1.158
Scenario - 3	0.569
Scenario - 4	0.231
Scenario - 5	0.694
Scenario - 6	0.828
Scenario - 7	0.056
Scenario - 8	0.207
Scenario - 9	8.862
Scenario - 10	8.120
Scenario - 11	7.612
Scenario - 12	0.666
Scenario - 13	2.078

Table: The presence of percentage of Botnet traffic in each scenario in CTU-13 Dataset

The above table implies that the scenarios 9, 10 and 11 contain maximum Botnet traffic in provided netflow data.

Out of 17 attributes belonging to different attribute types, we have selected a few of them for plotting different graphs for better understanding and visualizing the CTU-13 dataset. The attributes of CTU-13 dataset are as follows:

Dur : Duration of the communication in seconds. Where min = 0sec, max = 3600sec

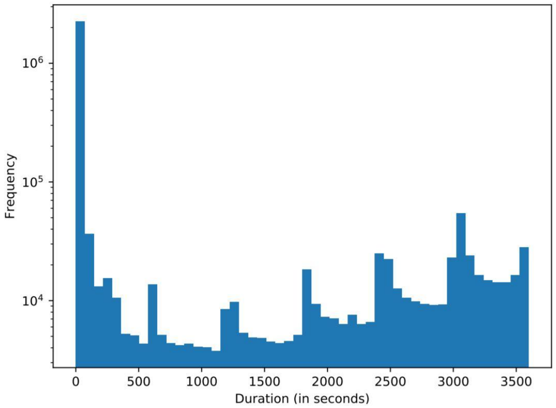

Figure: Duration of Communications [1]

Proto: Transport Protocol Used. There are 15 protocols used in total.

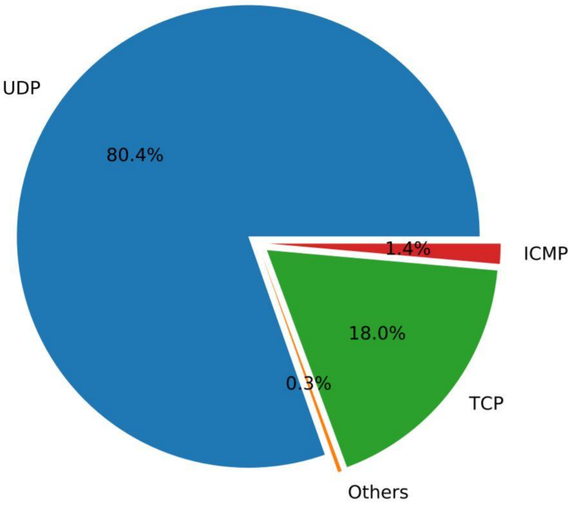

Figure: Distribution of Transport Protocols [1]

Dport: Destination port. There are a total of 73,786 ports. It is a range.

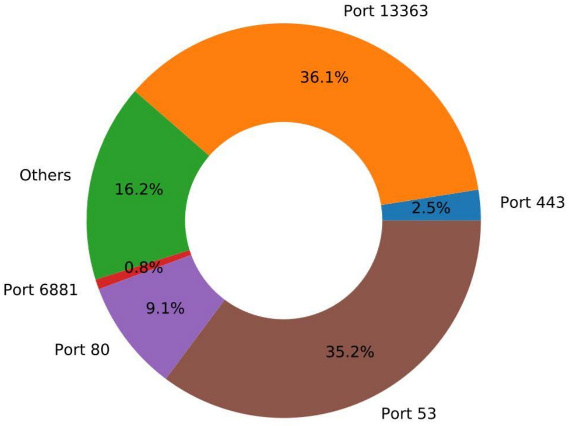

Figure: Distribution of Destination Ports [1]

TotPkts: Total number of transaction packets.

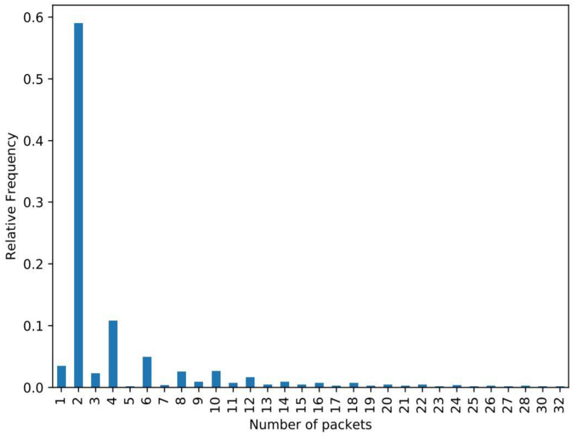

Figure: Most frequent number of packets [1]

Other attributes in the dataset are,
StartTime: Initiation of packet transfer
SrcAddr: IP address of the source
Sport: Source port
Dir: Direction of the flow
DstAddr: IP address of the destination
State: Transaction state
sTos: Source TOS byte value
dTos: Destination TOS byte value
TotBytes: Total number of transaction Bytes
SrcBytes: Total number of transaction Bytes from the Source
Label: Label made of "flow=" followed by a short description of netflow type
Botnet: Indicates the presence or absence of botnet traffic in the netflow (absence = 0 and present = 1). We have considered 'Botnet' as our class label for training our model.

Feature Selection:

A preliminary look at some of the feature correlations for this project is underway. Currently, using the python based scipy packages that contain correlation functions such as *pointbseirialr* and *chi-squared*, which are used to correlate a continuous variable vs a categorical variable and categorical vs categorical, respectively. These tools have been useful to determine if a feature is independent or correlated to botnet traffic, and a good indicator if a feature is worth considering as input for building a machine learning model.

For example, determining the correlation between the total number of packets vs whether it's a botnet yields the following results:

Total Packets vs Botnet:
PointbiserialrResult(correlation=-0.0011626170736417556, pvalue=0.0930014441334596)

Using the results above, we can determine that with a pvalue > 0.05, it is determined to not be statistically significant so this may not be a good candidate for feature selection.

Another example is comparing the netflow duration to the botnet, with the idea that botnets are typically associated with quick connections. Using the same formula and the same CTU data set as above, we can determine the following:

Duration vs Botnet:
PointbiserialrResult(correlation=-0.10172736084672689, pvalue=0.0)

As this has a p value < 0.05, this is significant and may be further explored and perhaps used as a feature input for building a machine learning model.

Below are the correlation results when comparing the continuous variables vs the binary botnet categorical values for CTU scenario 9 data set.

CTU Scenario 9		
Total flows	2087508	
Botnet flows	184987	
Percentage Botnet	8.86%	
	Botnet	
PointbiserialrResult	p value	correlation
Duration (time sec)	*0.000*	*-0.102*
Total Bytes	*0.093*	*-0.001*
Total Packets	*0.004*	*-0.002*
Source Bytes	*0.084*	*-0.001*

Further data exploration needs to be done to select the final features for extraction, but from preliminary results, duration and total packets should be considered as potential feature candidates.

Another correlation technique is the chi-squared test which can be used to measure correlation between two categorical values. For example, it can be used to correlate between destination port and botnet as destination port is a category and contains a range of ports from 1-65535 . If a particular port is often associated with a botnet, there should be an expected correlation between these features. Using pandas we can construct a contingency table known as crosstable and then use this as input to the scipy.stats.chi2_contingency method to compute chi-squared and p value.

To compute the correlation between destination port vs botnet, we arrive at the following.

Full range of ports 1-65535
(chi-square=772594.3099241449, pvalue=0.0, df=41695)

From the example above, the number of degrees of freedom equals 41695, and with an expected critical value based on significance level of .05, we can compute the critical value, which yields 42171.124. Since our chi-squared value (*772594.30)* is higher, we can reject the null hypothesis and this may not be a good candidate for feature selection based on this outcome, or further data reduction and binning the data may be required to yield a most positive result.

TOOLS

Many programming and related tools will be used for this project, which are listed below for reference. The main one to highlight is the Python based machine learning framework called Scikit-learn.

Preprocessing/Formating

- PyPCAPKit for extracting PCAP file: https://pypi.org/project/pypcapkit/
- Netflow: https://pypi.org/project/netflow/
- Scapy - https://scapy.net/

Machine Learning/Data Analysis

- Scikit-learn for implementing Machine Learning in Python: https://scikit-learn.org/stable/
- scipy - https://www.scipy.org/

Database Storage

- CSV flat files
- hdf5 library: https://www.hdfgroup.org/solutions/hdf5/

Visualization

- Plotly: https://plotly.com/python/

- Matlab https: https://matplotlib.org/

MILESTONES

Milestones Completed:
1)Data Preprocessing and Data Transformation
2)Visualization of our Dataset after Data preprocessing
3)Feature selection

Milestones Todo:
1)Feature extraction
2)Training/testing our model
3)Results and Data visualization

Our github repo: https://github.com/anushab97/Data-Mining-CSCI-5502-Project-Group-04.git

REFERENCES

Research Papers:
[1] Cyber Attack Detection thanks to Machine Learning Algorithms: https://arxiv.org/pdf/2001.06309.pdf

[2] Machine Learning-Based Early Detection of IoT Botnets Using Network-Edge Traffic: https://arxiv.org/pdf/2010.11453.pd

[3] D. Thomas. *Cybercrime Losses An Examination of U.S. Manufacturing and the Total Economy* NIST, 2020. https://doi.org/10.6028/NIST.AMS.100-32

[4] Sebastian Garcia, Martin Grill, Jan Stiborek and Alejandro Zunino *An empirical comparison of botnet detection methods*. Computers and Security Journal, Elsevier. 2014. Vol 45, pp 100-123. http://dx.doi.org/10.1016/j.cose.2014.05.011

[5] Sebastian Garcia, Agustin Parmisano, & Maria Jose Erquiaga. *IoT-23: A labeled dataset with malicious and benign IoT network traffic* 2020 (Version 1.0.0) [Data set]. Zenodo. http://doi.org/10.5281/zenodo.4743746

[6] Tarun Ganesh Palla & Shahab Tayeb. *Intelligent Mirai Malware Detection for IoT Nodes.* MDPI AI/ML Techniques for Intelligent IoT Systems, 2021. https://www.mdpi.com/2079-9292/10/11/1241/htm

[7] Davis, A.; Gill, S.; Wong, R.; Tayeb, S. *Feature Selection for Deep Neural Networks in Cyber Security Applications*. In Proceedings of the 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Vancouver, BC, Canada, 21–24 April 2020; pp. 1–7.