Botnet Detection based using traffic flow analysis

S.Shreyas (16IT135), Sanjay Ashok Shetty (16IT136), Srinag Rao (16IT141)
Information Technology Department
National Institute Of Technology, Karnataka, Surathkal.

***Abstract*—Detection of Botnets is one of the challenging issues in network security. Botnet detection is difficult due to the efforts made by them to evade detection by blending in with the genuine traffic flow. We propose a method for Botnet detection based on traffic flow on the network. This is independent of the different protocol and commands used by different Botnets and hence is more generalized. Experimental evaluation is done using ISOT dataset, which is a combination of public genuine traffic as well as Botnet traffic, computer generated traffic.**

***Index Terms*—Botnet Detection, Traffic flow, PCA, F-test, Information Gain**

I. Introduction

A botnet consists of compromised machines called bots which are remotely controlled by a malicious actor. A Command and Control(C&C) channel is setup between the bot-master, the malicious actor, and the pool of compromised machines. Botnets have been known to grow to large sizes and are a serious threat to cyber security. They are used in DDoS attacks, spam distribution, data theft, blackmail etc.

The CC channel is an integral part of a botnet system. Based on the C&C architecture and organisational structure of bots, botnets can be classified as centralized, decentralized or hybrid. The C&C channels use transports like HTTP, P2P, IRC and IM to synchronize and deliver commands to the bots. Centralized IRC-based botnets are the most common as they are simple to setup and very responsive.

ISOT is a combination of several publicly available network traffic dataset. The individual datasets are represented using subnets. The dataset contains traffic data from 22 non-malicious subnets and 1 malicious subnet. The non-malicious traffic includes traffic from a variety of applications like HTTP web browsing behavior, World of Warcraft gaming packets, a medium-sized enterprise network and packets from popular bit-torrent clients such as Azureus. The malicious subnet includes both infected and uninfected machines and the dataset documentation labels IP and MAC address as malicious and non-malicious. The MAC address and IP addresses are anonymized.

We propose and discuss the merits and performance of a botnet detection technique based off traffic flow analysis. A traffic flow is the time series data of packet length of packets of a given protocol between two ports on two machines. We derive several summarising features from the traffic flow and use them classify them as malicious or non-malicious. We test the relevance of features using tests like F-test, Chi Squared test, etc.

A. Challenges

Web-based centralized botnets use HTTP/S as the transport mechanism in their CC channel. They communicate with the bot-master using HTTP/S requests which are similar to regular internet traffic. This makes them difficult to detect and isolate. The bot traffic blends into the regular web traffic.

The growing use of encryption of all web traffic makes payload content analysis impossible. Data like source-destination identifiers, protocol and payload length are available but the payload itself is cannot be parsed. This adds to the complexity of botnet traffic detection.

It is difficult to have information about every transaction in a large network. To use topology or graph based detection method a large amount of traffic data is needed which is impractical to obtain due to the decentralized nature of the internet. Only large ISP have access to such data.

B. Motivation

The ability to classify traffic as malicious or non-malicious is important to cyber security. Even for un-encrypted data, detecting traffic using packet traffic characteristics is more efficient than reconstructing requests using packets and then comparing them against known strains. The only data available in packets of an encrypted channel is source-destination identifiers and payload length. By generating traffic flow characteristics a higher level of understanding can be obtained from the limited given data.

Section II summarises related work in this field and discusses some of the shortcomings in them. Section III defines the problem statement and objectives. In Section IV we propose our approach to classifying traffic flows as malicious or non-malicious. We report and analyse the experimental results of our approach in Section V. Finally section VI concludes the paper.

II. Literature survey

Signature based anamoly detection[6] where known signatures can be mathced easily but encrypted. Network anamoly detection which is discussed in [4] works even if packet is encrypted. Honeypot based intrusion detection[3] protects network through deception, during the course of attack it understands the attackers motives works with a narrow field of view. DNS traffic analysis[1] easily detect botnets effectively while bots are connecting to their server but they are not useful to find new Botnets. DNS traffic similarity[7] using Bayesian is effective and robust. Unique communication patterns from overlay topologies[5] has the ability to detect modern botnets

that use P2P channel but real-world implementation requires access to traffic data usually only present with ISPs. Statistical features of a session like connection level, req-res bytes, duration time[8] can be used to detect web-based bots that blend into regular HTTP traffic, this can be bypassed by introducing randomness to break periodicity.

A. Outcome of literature survey

Most of the current botnet detection techniques are designed only for specific botnet CC communication protocols and structures. When botnets change their CC architecture, these methods will not be effective in detecting them. But, some of the detection approaches based on DNS and data mining can detect botnets irrespective of the CC architecture.

III. Problem Statement

Design and develop a botnet detection technique based on network traffic flow analysis.

A. Objectives

1) Obtain traffic flow data from the ISOT dataset
2) Label data using the datasheet of the dataset
3) Visualize traffic flow
4) Derive features from traffic flow characteristics
5) Derive statistical features from traffic flow
6) Test the relevance and importance of features
7) Classify processed features

IV. Methodology

A Botnet detection model is proposed, in which the traffic flow characteristic of the network is analyzed for detection of Botnets in the network. A traffic flow consists of all the packets that are communicated to and fro by machines in the network. A single traffic flow can be characterized as the set of packets with the same following values:

1) Source IP address
2) Destination IP address
3) Source port
4) Destination port
5) Protocol

Flow is also given a direction. We assume that flows that packets that packets ingress to *172.16.0.0/16* as incoming. This nomenclature has been adopted since most of the traffic present in our dataset is generated from that subnet.

Most Botnet detection systems take into account information such as source and destination IP address, MAC address etc., which are characteristics of the machine. These information can be subject to spoofing as is the case with in case of IP spoofing and MAC spoofing. Another problem with such a detection system is the fact that most datasets have one to one mapping between an IP and target label (in this case whether given IP is malicious or not) which forces the model to heavily rely on such a characteristic, which as mentioned above can be spoofed. Due to this reason, instead of considering machine characteristics, traffic flow characteristics are considered.

Botnets are known to shows some uniformity in their behaviour, which distinguishes them from genuine traffic. These behaviours are exploited by forming features that capture the traffic flow such as,

1) **Number of Small Packets**: Botnets usually generate traffic that contains large number of small packets (packets within the range of 40 - 320 bytes) in order to avoid considerable impact on the network. These small packets are sent out from the Botmaster to the check for susceptible hosts. On the other hand genuine traffic operates above the boundary of small packets, such as HTTP responses.
2) **Packet Ratio**: Botnets traffic consists of limited commands and protocols, hence their traffic does not fluctuate much as compared to genuine traffic. Due to this, packet ratio of a flow in a specified interval of time can be considered as a characteristic having a good correlation with the target.

$$\text{Packet Ratio} = \frac{\text{Number of ingress packets}}{\text{Number of egress packets}}$$

3) **Length of Initial Packet**: Once a client is infected and joins the network of botnets, there is a clear communication protocol is followed during the exchange of packets inititally.
4) **Response ratio**: It has been observed that a common response time exists between request to a particular IP, port and response from the same. This response time is a constant and is used as an important feature with high correlation to the label. To capture this we use

$$\text{Response Ratio} = \frac{\text{Response Packets}}{\text{Total Packets in a flow for a time interval}}$$

Traffic flow will first be generated based on packets with similar values that characterize a flow as mentioned above. The traffic flow is then used to generate the above mentioned features, which are then used for prediction. Correlation between features and labels will be analysed, and features that have the best correlation with the labels will be used.

Along with features that represented the characteristics of botnets, features that represent the traffic flow are also extracted, which are as follows:

1) **Packet average length**: Average traffic is maintained low by the the botnet to avoid suspicion. But the average packet length varies in normal traffic.
2) **Packet length variance**: Normal traffic has mostly work at extreme packet sizes so attributes like outgoing packet variance and incoming packet variance will be less.

The flow as described above is summarized in figure 1.

A. Work Done

The following steps were performed

1) Selecting all the packets in *172.16.0.0/16* subnet from ISOT pcap file and making csv with relevant features
2) Construct flows which consists of consecutive packets having the same five-tuple i.e source IP, destination IP, source port, destination port, protocol.

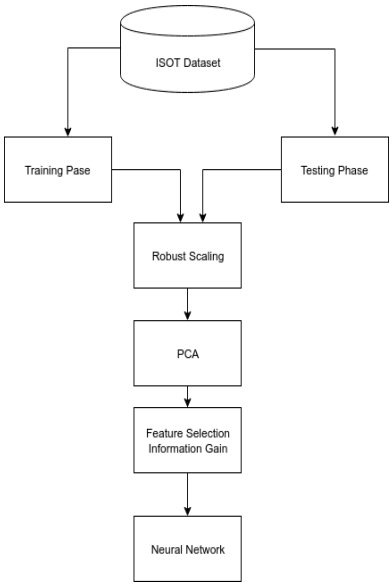

Fig. 1. Flow

3) From each of the flows we plot different graphs for windows size of thirty seconds for each flow to extract features. The following are the information extracted from the graphs
   a) Traffic flow graph
   The malicious traffic are generated by BOT so they show repeating patterns when plotted as shown in 2 but the non malicious traffic does not show any uniformity as shown in Figure 4. UDP storm bot which just storm the network with small UDP packets and doesn't wait for reply is shown in Figure 3

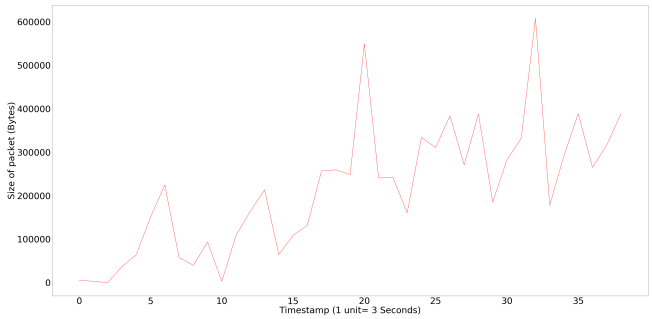

Fig. 2. Malicious flow traffic

   b) Packet size
   Malicious bots communicate with packet size in the range 40 to 320 bytes as shown in the Figure 5. Non malicious nodes communicate with either small sized packets as shown in Figure 6 or large packets when the node is streaming data.
   c) First five seconds traffic analysis

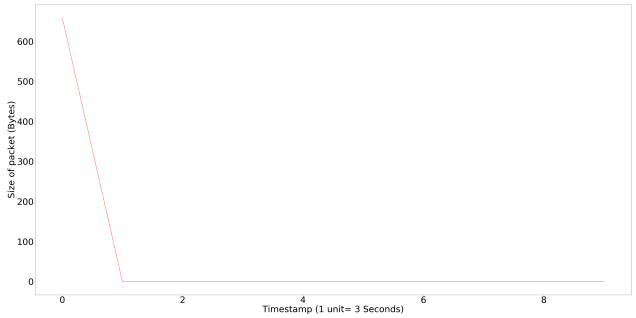

Fig. 3. Malicious UDP storm flow graph

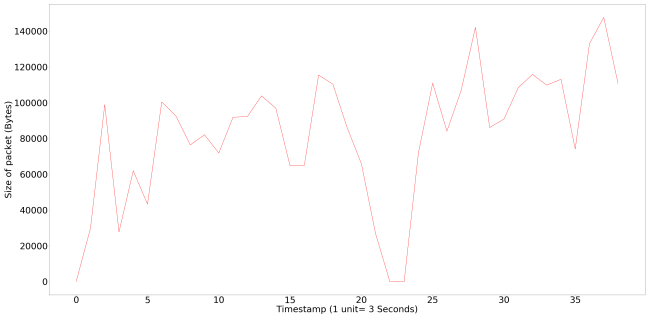

Fig. 4. Nonmalicios flow graph

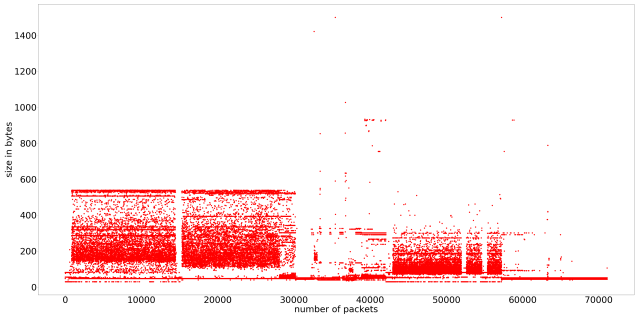

Fig. 5. Malicious flow packet sizes

   When we plot the average packet size for first five seconds we can observe that most of the bot net communication falls in lower bins of the plots as shown in Figure 7 but it is distributed in more bins in 8.
4) We create the dataset with all the attributes as mentioned in methodology
5) The rows with same feature values were dropped to avoid mutiple entries of UDP storm bots.
6) The Robust scaler is applied to avoid giving more weightage to specific attributes

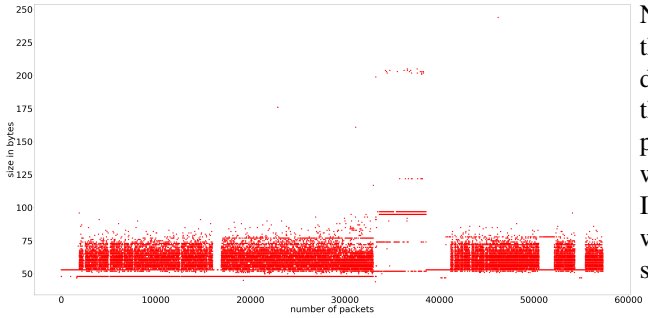

Fig. 6. Non malicious flow packet sizes

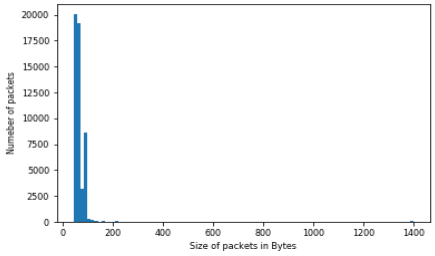

Fig. 7. Malicious average packet size for first five seconds of each flow

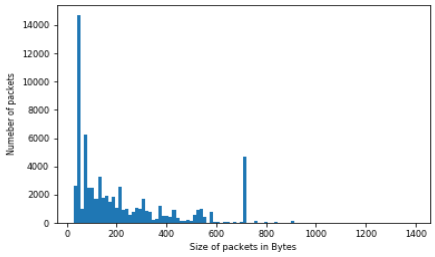

Fig. 8. Nonmalicious average packet size for first five seconds of each flow

7) PCA is applied to transform the attributes which also helps in dimentionality reduction
8) Information gain is used for feature selection
9) The transformed attributes are used to train multilayer perceptron.

V. Results and Analysis

Series of experiments were made in order to evaluate the capability of the model. A Neural Network was used to predict the target label. Precision, Recall, F1-score were used as metrics to analyze the results.

Two different sets of features were analyzed for evaluation, one that captures the Botnet characteristics, and one that captures the network flow characteristics.

Table I shows the results obtained from botnet features. We first used all the 4 botnet features as mentioned in section IV. A good F1-score was obtained which meant that the model had predicted very less False Positives and False Negatives. Duplicates were then removed from this data, and the model was analysed again. This lead to a considerable decrease in the F1-score. The reason for this could be due to the presence of UDP storm, which added duplicate packets present in both train as well as test data. The four features were subject to feature selection tests, such as F-test Annova, Information Gain, from which the best features were taken which were **Packet ratio**, **Response time** which resulted a slight improvement in F1-score.

Table II shows the results obtained by adding traffic flow features to the previous features. This data was evaluated after transforming it by using PCA, and gave the highest F1-score, because it converts co-related variables into a set of linearly non co-related variables . Duplicates were then removed which caused reduction in F1-score. PCA transformation gives the transformed features in order of variance, we then analyse by taking first 3 and first 6 features, which lead to a decrease in F1-score. Feature selection using information gain was then used to on top of PCA to select 4 best features, which increased the F1-score slightly from before.

Finally in Table III we summarize and compare our results with the results of the base paper [2]. Their F1-score was calculated by averaging scores from different models used.

TABLE I
Results on using the Botnet features

Description	precision	Recall	F1-Score
Using the 4 botnet features	0.94	0.94	0.94
Using 4 features with removing duplicates	0.68	0.71	0.60
Only Packet ratio, Response time (no duplicates)	0.71	0.73	0.68

TABLE II
Results on adding the traffic flow features

Description	precision	Recall	F1-Score
Using all features, PCA	0.99	0.99	0.99
Using PCA, removing duplicates	0.89	0.90	0.89
Using PCA, select first 3(no duplicates)	0.76	0.78	0.75
Using PCA, select first 6(no duplicates)	0.87	0.87	0.87
Using PCA, select 4 best(using Information gain)	0.81	0.81	0.80

TABLE III
Benchmarking with base paper [2]

Time Window	Our F1-score	Base paper's F1-score
30s	0.99	-
60s	0.96	0.95
120s	0.97	0.96
180s	0.99	0.96

VI. Conclusion

Botnets are used usually to perform attacks such as DDos, in which the infected machine is told to ping a specific site, thereby denying service for genuine users. They also can be used for spamming, phishing, click fraud etc..Even

though Botnet detection techniques continue to improve, there is a need for techniques that identify botnets in the early stage itself, irrespective of Botnet commands, protocols. This method has an advantage over others on the fact that only packet, traffic flow characteristics have been analysed, hence this method can be used even in the case where the packet is encrpyted. The time window used for detection of Botnets is only 30 seconds, which is very less compared to the existing approaches. We also analyse the traffic flow data to find out the best features, that have high correlation with the label. It is shown that PCA, feature selection have helped improve the overall performance of the model.

REFERENCES

1 Hyunsang Choi, Hanwoo Lee, Heejo Lee, and Hyogon Kim. Botnet detection by monitoring group activities in dns traffic. In *7th IEEE International Conference on Computer and Information Technology (CIT 2007)*, pages 715–720. IEEE, 2007.

2 G Kirubavathi and R Anitha. Botnet detection via mining of traffic flow characteristics. *Computers & Electrical Engineering*, 50:91–101, 2016.

3 Janardhan Reddy Kondra, Santosh Kumar Bharti, Sambit Kumar Mishra, and Korra Sathya Babu. Honeypot-based intrusion detection system: A performance analysis. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 2347–2351. IEEE, 2016.

4 Wei Lu, Goaletsa Rammidi, and Ali A Ghorbani. Clustering botnet communication traffic based on n-gram feature selection. *Computer Communications*, 34(3):502–514, 2011.

5 Shishir Nagaraja, Prateek Mittal, Chi-Yao Hong, Matthew Caesar, and Nikita Borisov. Botgrep: Finding p2p bots with structured graph analysis. In *USENIX Security Symposium*, volume 10, pages 95–110, 2010.

6 Yong Tang and Shigang Chen. Defending against internet worms: A signature-based approach. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 2, pages 1384–1394. IEEE, 2005.

7 Ricardo Villamarín-Salomón and José Carlos Brustoloni. Bayesian bot detection based on dns traffic similarity. In *Proceedings of the 2009 ACM symposium on Applied Computing*, pages 2035–2041. ACM, 2009.

8 Binbin Wang, Zhitang Li, Dong Li, Feng Liu, and Hao Chen. Modeling connections behavior for web-based bots detection. In *2010 2nd International Conference on E-business and Information System Security*, pages 1–4. IEEE, 2010.