IJOCIT
ISBN: 2345-3877
www.ijocit.org
Volume 6, Issue 1

Original Research

P2P Botnet Detection Based on Traffic Behavior Analysis and Classification

Hojjat Beiknejad [1, *]
Hamed Vahdat-Nejad [1]
Hossein Moodi [2]

Received: 04 Feb 2018
Accepted: 10 Mar 2018

http://dx.doi.org/10.14196/IJOCIT.2018.001

Abstract

Nowadays, botnets are being considered as the most important security threats in the internet and it is important to find new ways for their detection. Peer to Peer (P2P) botnets are the most important kinds of botnets that use P2P communication protocols to control their bots remotely. Therefore, their detection is more difficult than other botnets. In this paper, we propose a new approach to detect P2P botnets in the command and control (C&C) phase of life cycle based on the analysis of traffic behavior. The proposed approach is able to detect C&C traffic of P2P botnets by using flow-based features and classification methods. The performance of the proposed approach is evaluated based on different parameters. The results of the evaluation show that the proposed approach is able to distinguish P2P botnet from normal network traffic with high detection rate.

Keywords: Network Security, P2P botnet detection, Network flows, Classification.

Citation: Beiknejad, H., Vahdat-Nejad, H., Moodi, H., (2018). P2P Botnet Detection Based on Traffic Behavior Analysis and Classification, *Int. J. of Comp. & Info. Tech. (IJOCIT)*, 6(1): 01-12.

1 Faculty of Electrical and Computer Engineering, University of Birjand, Birjand, Iran
2 Department of Computer Engineering, Birjand University of Technology, Birjand, Iran
* Corresponding Author: hojjat.beiknezhad@birjand.ac.ir

1. Introduction

Nowadays, with the development of Information and Communication Technology (ICT) and spread of computer networks, the number of internet users has increased rapidly, and the web has become an essential part of our lives. Every day, a huge amount of data is transferred through the internet. Besides, there are some invaders or people who want to cause damages to the internet users. Therefore, in line with the development of Information and communication Technology, it is necessary to provide information security for the users and counteract security threats. Among the security threats, we can refer to malware, which puts a large number of users in danger. Malware is a program with malicious purpose, which is designed to destroy the computer or the network that runs it [1]. Malware consists of different categories of programs such as viruses, Worms, Trojans, Bots, and etc [2].

Todays, botnets have become the main source of attacks in the internet. A botnet is a network of compromised machines connected to the internet that is infected by malicious software (bot) and is remotely controlled by botmaster [3-5]. P2P botnets are the newest type of botnets that use P2P networks to remotely control their bots [6]. P2P botnets have many malicious purposes such as Spreading spam, Distributed Denial of Service (DDoS) attacks, malware distribution and stealing important information [7,8]. Therefore, it is important to find prevention ways to detect them in the initial stage. The problem of current research is about the detection of P2P botnets in C&C phase of life cycle.

Although some approaches have been proposed for the detection of P2P botnets, their detection is challenging because of the following reasons [4]: Botnet traffic is similar to normal network traffic. Besides, sometimes botnets use encrypted communication channels in order to prevent detection. Therefore, approaches that perform the detection based on the analysis of packet content [9], are unable to detect them. Furthermore, some approaches need to analyze a large amount of data, which is hardly possible to be performed in real time for a large-scale network. Finally, P2P botnets detection is more challenging in comparison with other botnets.

In this study, a new approach to P2P botnet detection in C&C phase of life cycle and before their attack is proposed. To this end, we analyze traffic behavior in order to detect P2P botnets by flow-based features. Afterward, we make use of classification methods in data mining to distinguish the normal from bot's traffic. In particular, we evaluate the performance of several data mining algorithms for P2P botnet detection. Because detection is performed based on flow-based features, the proposed approach is independent of content or packet payload and is able to detect P2P botnets that use encrypted traffic. Experimental results show that the proposed approach can detect the traffic flows of P2P botnets with a higher detection rate.

After this introduction, the rest of this paper is structured as follows. Section 2, describes the research background. Section 3, analyze the botnet detection methods and reviews related works on P2P botnet detection. Section 4, describes the proposed approach in detail. Section 5, presents the results of experiments and evaluates the performance of the proposed approach. Finally section 6, concludes the paper and outlines future work.

2. Research Background

In this section, we describe background information related to botnet. For this, we introduce botnet, their components, life cycle, structure and protocols.

2.1 Botnet

A botnet is a set of compromised connected computers that are infected by a bot and are remotely controlled by botmaster under a common C&C infrastructure [10]. Bot is a software program, which is installed on vulnerable hosts and is able to perform malicious activities. After installation of the bot program, the computer becomes a bot or zombie. The set of these bots, make a network called botnets. The botmaster sends commands to the bots by using C&C channel, and controls them [11]. Nowadays, botnets are considered to be the most important source of attacks that follow various malicious purposes. Among these purposes are Distributed Denial of Service (DDoS), Spreading Spam, gathering important and personal information of users, click-fraud, malware distribution, and network service disruption [12].

2.2 Components of a botnet

Botnets are composed of three components (figure 1): bot, C&C server and botmaster. Botnet threats are organized by these components. Botmaster is a malicious user that controls botnet by sending commands to the bots to do malicious activities. C&C server receives commands from the botmaster and send them to other bots [11,13]. The main difference between botnet and other malwares is the presence of C&C infrastructure. This infrastructure allows bots to receive malicious

commands from botmaster and provides the ability for the botmaster to control and guide bots activities within a botnet [14]. C&C Infrastructure interconnects the components of a botnet in order to transfer data between them. It is necessary to keep this connection stable for the botnet to operate efficiently [11]. The components of a botnet are shown in fig. 1.

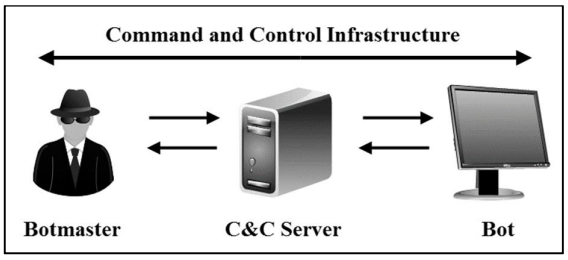

Figure 1: Components of a Botnet

Each command issued by the bot is replied by a server in the botnet or other peer bots [15]. After being installed on a vulnerable host, a bot needs to be connected to the C&C server or other bots available in the network and keep the communication active for a long period of time. For this purpose, bots send packets for C&C servers or other peers on the internet in order to connect to them. Bots connections to C&C servers or peer bots cause an increase in the number of bots in a botnet; Therefore, the botnet is able to live longer in order to perform malicious activities.

2.3 Life cycle of a botnet

The life cycle of a botnet includes four phases: formation, command and control (C&C), attack, and post-attack. The formation is a phase in which the botmaster exploits a specific vulnerability in the target system and infects it. Afterward, botmaster uses the acquired access in order to install malicious program on the target system. After running malicious program, the target system becomes a bot. In the C&C phase, the bot tries to make a connection to its C&C server and join the botnet by the help of this connection. Afterward, the attack phase starts. In this phase, bots receive the commands of botmaster through C&C channel, and perform the malicious activities based on these commands. The last phase of bot's life cycle is the post-attack phase. In this phase, botmaster performs such acts as updating bot's program to improve operation and defend against detection methods [16].

2.4 Botnet structure and protocols

The most important component of a botnet is its communication infrastructure, which is the command and control channel. Botnets are categorized into three groups based on the protocol used in their C&C channel including IRC-based, HTTP-based, and P2P. Botnets based on IRC and HTTP have Centralized C&C Structure and P2P botnets have decentralized C&C Structure [11,17,18].

Centralized C&C Structure is similar to client-server architecture [9]. In this structure, bots communicate with one or multiple C&C servers in order to receive commands [19]. The main shortcoming is the centralized design of the C&C server that is a central point of failure, hence if detected, the botnet will stop working [20]. This weak point leads the botnet developers to move toward the development of decentralized structures, which results to the introduction of P2P botnets with the use of P2P communication protocols.

In P2P botnets the C&C server is concealed [18]. Each bot can act as a client or server and the botmaster can perform its attacks from each computer [9]. The main feature of P2P botnet is that all peers can play the role of C&C server [17]. Because P2P botnets have decentralized C&C structure, they do not have the problem of centralized structure. In other words, if one bot is taken down, its effect on the whole botnet will be less and the botnet will remain under the control of other bots. On the other hand, management and maintenance of P2P botnets in comparison with centralized botnets is more complex [19]. Fig. 2 shows, structure of the decentralized C&C.

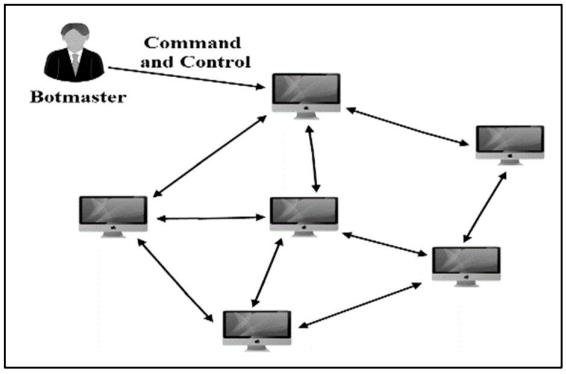

Figure 2: Decentralized C&C structure

3. Related Works

In general, the methods for the detection of botnet are divided into two categories of payload-based analysis and behavior-based analysis [21]. In the following, we investigate these approaches.

3.1 Detection based on payload analysis

In payload analysis, content of exchanged packets in the network is analyzed in order to detect botnet based on the obtained features. Although these methods show high detection accuracy, they have some limitations that restrict their performance in botnet detection. Because of some legal and privacy issues, sometimes it is impossible to have access to the content of traffic packets and payload information; therefore, the operation fails. Furthermore, it is necessary to analyze a large amount of traffic packets, which makes these methods to be time-consuming. Finally, these methods cannot detect botnets with encrypted communication channels, because traffic packets are encrypted, and hence, the payload information is not available [22].

Tarng et al. [9] propose a six-step mechanism for detecting the traffic flows of P2P botnets at the C&C phase of life cycle. In their mechanism, the traffic flows are analyzed based on ASCII distribution in the flow packets. They use IP addresses, port numbers and other host features to reach the goals of each step. ASCII distribution is the payload characteristics of packets, hence their mechanism is unable to detect botnets with encrypted packets. Moreover, it requires high computing demands. They use J48 decision tree model for classifying P2P applications and utilize K-Means clustering algorithm to categorize the traffic flows and detect the abnormal flows.

3.2 Detection based on traffic behavior analysis

The Limitations of payload based detection led to the introduction of detection methods based on traffic behavior analysis. Traffic behavior analysis follows this principle that bots within a botnet usually have uniform traffic behavior and show unique communication patterns. By the use of various features that could be extracted from network traffic such as packet size, flow duration, and number of packets in flow, these methods determine the botnet's traffic behaviors and patterns, and therefore, distinguish botnets from network's normal traffic [23]. Detection methods based on traffic behavior analysis are not dependent on the content or payload of packets, and therefore, can identify botnets with encrypted packets. In addition, the information related to the network traffic can be easily obtained from network devices [22].

Traffic behavior analysis method use combination of various features to detect the type of network traffic. These features can be extracted based on the characteristics of hosts and/or network flows. There features are generally divided into the following categories [21,23]:

1. Host-based features: features that describe the communications between hosts. They can be extracted from communication patterns of the hosts. Host-based feature extraction requires an analysis of each packet belonging to a host, and hence, this is time consuming with the flow-based feature extraction.
2. Flow-based features: features that can be extracted from network flows. Flow is the set of packets that have five similar characteristics (5-tuple). These characteristics are as follows: source IP address, destination IP address, source port, destination port and protocol [24]. Flow-based features are used to assign flows to certain class of network traffic such as bot or non-bot traffic. Extracting these features takes less time compared with the host-based features, because the number of flows is much less than the number of packets.

Chen et al. [25] investigate the detection of P2P bots and propose an approach, which targets the detection of malicious behaviors and P2P communication together. They analyze API function calls generated by the bot program on the host and utilize host behavior features including IP address, port number and type of protocol. Their approach is applicable after performing malicious activity by bot, and is not appropriate to detect bots in C&C phase. The evaluation results show the detection rate of 95.7% and false positive rate of 3.5%.

Stevanovic et al. [26] propose a flow-based detection system, which consists of two components: preprocessing entity and classifier entity. In the preprocessing stage, by analyzing the network traffic at the flow level, flow features are extracted. In the classifier stage, traffic flows are classified into malicious and non-malicious, by using supervised machine learning algorithm. In this system, 39 features are extracted from traffic flows, and the performance of 8 classifiers has been evaluated. Three classifiers, namely C4.5, Random Forest, and Random decision tree are better among others. The system achieves the rate between 95.5% to 96.5% for P2P botnet detection.

Saad et al. [21] have investigated P2P botnet detection in C&C phase using analysis of network traffic behavior. They utilize host-based as well as flow-based features to detect P2P botnet traffic. They define 17 features, which are extracted from network flows and communication patterns of host. The performance of 5 machine learning techniques for network traffic classification has been evaluated. The results show the detection rate higher than 90% and error rate less than 7% for Support Vector Machine, Artificial Neural Network, and Nearest Neighbors classifiers.

Table 1, summarizes the works related to the detection of P2P botnets.

Table 1: P2P botnet detection projects

Approach	Detection Method		
	Payload Based	Behavior Based	
		Host-based features	Flow-based features
Tarng et al. [9]	√		
Chen et al. [25]		√	
Stevanovic et al. [26]			√
Saad et al. [21]		√	√

The evaluation results of the related studies reveal the following challenges and limitations:

- Some approaches (such as Chen & et al. approach) detect botnets based on their malicious action. Such approaches are only able to detect botnets in attack phase of life cycle and they are unable to detect them in C&C phase. An effective approach is the one that is able to detect botnets before their attacks.
- Some approaches (such as Chen & et al. approach) detect botnets based on host-based features. Extracting these features in comparison with flow-based features is time-consuming and takes much more computational time.
- Some approaches (such as Tarng & et al. approach) detect botnets based on analyzing payload of the traffic packets. These approaches are not able to detect botnets with encrypted traffic.
- In approaches that use flow-based features (such as Stevanovic & et al. approach), the important point is how to choose the most effective, and at the same time, the least number of features in order to achieve the best detection rate with the minimum computational requirements.

4. Proposed Model for P2P Botnet Detection

Bots are preprogrammed to response to the commands that are received. Therefore, each botnet has its own special set of commands and C&C interactions. In a P2P botnet, bots need to be connected to other bots available in the network and keep their communication active with them. For this propose, they establish very small sessions. In each session, the data flow between a pair of P2P bots occurs. Usually, the sessions are created for small durations in order to avoid detection. Moreover, a small amount of data is transferred between P2P bots in each session [15]. These observations indicate that bots of a botnet have uniform traffic behavior and show specific traffic patterns for communication.

Based on these observations, a new detection approach is proposed that is able to detect P2P botnets in C&C phase of life cycle. The aim of the proposed approach is to detect the traffic of P2P botnets based on the flow-based features. In order to reach this purpose, we use classification methods in data mining as an effective tool for detection. After selecting and extracting the effective features from the traffic flows, we classify traffic flows as malicious (botnet) or non-malicious (normal network traffic). In what will follow, we explain the characteristics and phases of the proposed approach.

4.1 Characteristics of the proposed approach

Payload-based detection have some problems including: (a) not being able to detect botnets with encrypted traffic, (b) need for analysis of a large volume of traffic packets, which makes it time-consuming, and (c) lack of access to payload information because of legal and privacy issues. These problems lead to the suggestion of detection approach based on the analysis of traffic behavior. Because extraction of flow-based features needs less time compared with host-based features, we utilize the flow-based features to detect P2P botnets traffic. The characteristics of the proposed approach include the followings:

- Focusing on the detection of P2P botnets in the C&C phase of life cycle in order to detect bots before their attack.
- Selecting an effective set of flow-based features in order to detect P2P botnet traffic with the highest detection rate.
- Independency of any of IP address, port number, packet payload and other host

features; while being able to detect P2P botnet with encrypted traffic.

4.2 Phases of the proposed approach

Fig. 3 shows the model as well as the phases of the proposed approach.

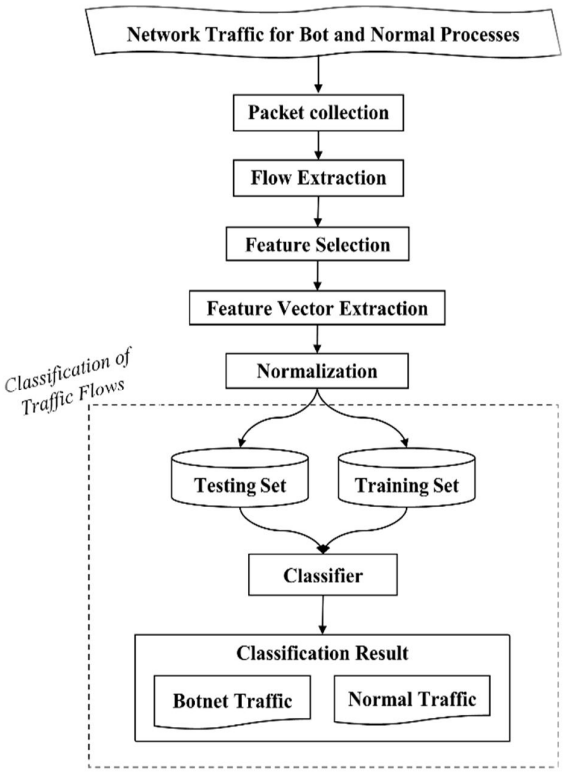

Figure 3: Model and stages of the proposed approach

Based on the model, the phases of the proposed approach are as follows:

1. Packet collection: After being installed on a host, a P2P bot tries to connect to other bots of the network in order to receive commands. For this purpose, it sends some packets to other bots, which subsequently, try to reply via sending packets. As a result, the P2P bot becomes connected to other peer bots of the network and packet transfer continues among them in order to receive the commands. In this phase, all transferred packets, which include both botnet and normal traffic, are captured and saved. Each packet has several characteristics such as source IP address, source port, destination IP address, destination port, protocol, size and sending time. These pieces of information can be easily retrieved from network hosts. Table 2 shows the characteristics of a packet in network traffic.

Table 2: Characteristics of a packet in network traffic

Row	Packet characteristic
1	Source IP address
2	Source port
3	Destination IP address
4	Destination port
5	Protocol
6	Packet size
7	Sending time

2. Flow extraction: After collecting all network packets, the flows of traffic are extracted. Flow is defined as the set of packets that have five common characteristic (5-tuple) as follows: source IP address, destination IP address, source port, destination port and protocol.
3. Feature selection: Choosing and defining the right set of features that distinguish between P2P botnet traffic and normal network traffic is very important. Opposite to normal network traffic, botnets traffic shows more uniform behavior in C&C operation. Bots behavior affects some packet-related parameters such as size, number of packets and duration of a flow. In this research, features are selected based on three parameters including: size of packets, number of packets, and arrival time of packets in a flow.

Packets that belong to the C&C phase of botnets are usually smaller than packets of normal network traffic. The number of packets in a typical bot flow is usually fewer in comparison with the normal flows, because bots keep their communication alive by transferring a few number of packets to keep themselves hidden by consuming a trivial amount of bandwidth. On the other hand, botnet traffic is made of fewer packets with distinct sizes compared with normal network traffic. In other words, most of the packets that belong to botnet traffic have the same sizes in comparison with the packets of normal network traffic.

Moreover, Duration of botnets traffic flows is usually less than normal network flows. In C&C phase, P2P bots send packets to other peer bots through short-duration flows, and then, change their communication port on the host in order to connect to other bots in the network. Bots continuously search their C&C channels to receive commands. Therefore, a typical P2P bot generates a lot of flows

with short-time duration compared with the most of normal network flows.

Responding flows are flows that transfer at least one packet in each of the two directions. In a responding flow, protocol, source IP address and source port of one flow is equal to the protocol, destination IP address, and destination port of the other flow, respectively. For responding flows, we consider two extra features including "difference in the number of packets" and "time difference between arrival-time of the first packets" in each direction. The other selected features are "size of the first packet", "average" as well as "variance" of packets size", flow duration, ratio of the largest packet, and average number of packets sent per time unit. According to this discussion, we utilize eight features, which are summarized in table 3.

Table 3. Selected features of traffic flows

Row	Feature
1	Size of the first packet in a flow
2	Average of packets size in a flow
3	Variance of packets size in a flow
4	Flow Duration
5	Ratio of the largest packet to all packets in a flow
6	Average number of packets sent per time unit in a flow
7	Difference in the number of packets in responding flows
8	Time difference between receiving the first packet in responding flows

4. Feature vector extraction: In this phase, selected features are extracted from flows. For each flow, a feature vector is produced, which consists of the value of selected features. Feature vectors together, produce the dataset for classification. Since eight is the number of features, assuming m to be the total number of flows, the dataset consists of the following feature vectors:

$$F_i = < f_1, f_2, \dots, f_8 > \quad i = 1, \dots, m$$

5. Normalization: In order to improve the performance of classification and increase the detection rate, extracted values for each feature in the entire dataset, are normalized to the range between 0 and 1. By normalizing each feature, feature vectors are also normalized to the range of 0 to 1. To normalize a set of numbers to the range of 0 to 1, we compute the maximum and minimum of the numbers. Then, a constant number (ratio) is computed as the inverse of the difference between the maximum and the minimum numbers. Afterward, each value is normalized by subtracting the minimum number from it, and multiplying the obtained value by the constant ratio. At the end, the numbers in the set are normalized to 0 and 1 area. Pseudo-code of normalization is depicted in fig. 4.

Input

Set: Set of Numbers
Min: Minimum Number of Set
Max: Maximum Number of Set

Output

Normal-Set: Set of Normal Numbers

```
Constant-Ratio = 1 / ( | Max – Min | );
For All Numbers in Set
Normalized-value= ( Number – Min ) * Constant-Ratio;
Add Normalized-value to the Normal-Set
End
```

Figure 4: Pseudo-code of normalization

6. Classification: This phase is responsible for distinguishing P2P botnet traffic from normal network traffic. For this, we use a classifier that is able to categorize traffic flows as either malicious (P2P botnet traffic) or non-malicious (normal network traffic). Classification has generally two phases: training and testing. In the training phase, classifier is trained to detect the two classes of data by using training data. In the testing phase, the trained classifier is evaluated by using testing data.

5. Experimental results

We conduct an experiment to collect data and evaluate the proposed scheme. To this end, at first some botnet traffic as well as normal traffic is generated and then, flows are identified. Afterward, features are extracted from the flows and the dataset is prepared for further analysis and evaluation. We utilize several well-known classifiers to obtain the best classifier for the proposed model, and then, compare the results with those obtained in previous research.

5.1 Packet collection and flow extraction

To produce a dataset, it is necessary to collect packets related to both P2P botnet and normal network traffic. In order to make botnet traffic, Waledac P2P botnet is used. The Waledac bots use TCP packets to communicate with each other. Normal network traffic is also generated by a combination of different programs including P2P application (BitTorrent), chat application (Skype), and web traffic (Web browsing).

Testing environment is a small local-area network that is connected to the internet. On some computers, Waledac P2P botnet, and on the others BitTorrent, skype and Web browsing are executed. We only collect those packets that are exchanged by processes of these programs. For this purpose, we used Microsoft Network Monitor software to capture traffic packets. It is installed on all computers and is responsible for capturing exchanged packets. In order to be able to make a comprehensive dataset, Waledac P2P botnet has been executed on hosts in different times and its traffic packets have been collected for various time intervals.

After collecting packets, flows are extracted from them. Flow extraction is performed by Matlab. In total, 9930 packets from P2P botnet traffic and 14680 packets from normal network traffic have been collected and 3296 flows from P2P botnet traffic and 1233 flows from normal network traffic have been extracted. Observations show that P2P bots generate more flows than non-malicious programs, because the number of packets in flows generated by bots is less compared with normal network flows. Table 4 shows the number of collected packets and extracted flows from botnet traffic and normal network traffic.

Table 4. Number of collected packets and extracted flows

Traffic Class	Traffic Trace	Number of Packets	Number of Flows
P2P botnet traffic	Waledac	9930	3296
Normal network traffic	BitTorrent Skype Web Browsing	14680	1233

5.2 Feature vector extraction and dataset creation

After extracting traffic flows from collected packets, feature vectors should be extracted from them. We used Matlab software to extract feature vectors. Each feature vector shows traffic behavior of a specific flow. A feature vector is labeled into two classes, “P2P botnet” and “normal”.

Feature vectors together constitute the dataset. Many of extracted feature vectors are similar to feature vector of another flow. Therefore, we delete the replicated feature vectors in order to have only one instance of each of them in the dataset. As a result, 432 vectors for botnet and 918 vectors for normal network traffic remain. In other words, the dataset consists of 1350 traffic flows, which include 432 P2P botnet and 918 normal flows. Table 5 shows the number of distinct feature vectors.

Table 5. Number of distinct feature vectors

Traffic Class	Number of feature vectors
P2P botnet traffic	432
Normal network traffic	918
Total traffic	1350

5.3 Classification and evaluation parameters

By the help of different classifiers, we classify traffic flows into two classes: malicious and non-malicious. We use Weka as an appropriate tool for classification. Weka is an environment in which different algorithms of data mining and machine learning are implemented [27]. In this phase, we utilize the mostly used classifiers in botnet detection field. These classifiers are as follow:

1. Bayesian Network Classifier (BNet)
2. Naive Bayes Classifier (NB)
3. Support Vector Machine Classifier (SVM)
4. J48 Decision Tree Classifier (J48)
5. Random Forest Classifier (RF)

For training and testing each classifier, we use the widely exploited 10-fold cross validation technique. It divides the dataset into 10 random subsets. In each iteration, one subset is used for testing and the other 9 subsets for training. This process is repeated until each of the 10 subsets is used once as the testing set.

Afterward, we evaluate the results of the utilized classifiers. The evaluation parameters include Recall, Precision, F-Measure, and Accuracy. Table 6, shows the evaluation parameters.

We make use of the confusion matrix [28] to evaluate the classifiers. The confusion matrix is shown in table 7 and its parameters are obtained as follow:

1. True Positive (TP): the number of bot flows detected as bot flows
2. True Negative (TN):the number of normal flows detected as normal flows
3. False Positive (FP): the number of normal flows detected as bot flows
4. False Negative (FN): the number of bot flows detected as normal flows

Table 6: Evaluation parameters for the classifiers

Traffic class	Recall	Precision	F-Measure	Accuracy
P2P botnet traffic	$\text{Recall}_{\text{Bot}}$	$\text{Precision}_{\text{Bot}}$	$\text{F\_Measure}_{\text{Bot}}$	Accuracy
Normal network traffic	$\text{Recall}_{\text{Normal}}$	$\text{Precision}_{\text{Normal}}$	$\text{F\_Measure}_{\text{Normal}}$	

Table 7. Confusion Matrix

	Predicted Bot	Predicted Normal
Actually Bot	True Positive	False Negative
Actually Normal	False Positive	True Negative

Based on the confusion matrix, evaluation parameters are defined as follow:

$\text{Recall}_{\text{Bot}}$: It shows how much percent of bot flows are detected by classifier. This parameter is obtained by the ratio of the number of bot flows that were correctly detected by classifier to the total number of bot flows as follow:

$$\text{Recall}_{\text{Bot}} = \frac{TP}{TP + FN}$$

$\text{Precision}_{\text{Bot}}$: It shows what percentage of the flows that were detected as bots by classifier are really bot. This parameter is obtained by the ratio of the number of bot flows that were correctly detected to the total number of flows that were correctly or wrongly detected by classifier as bot:

$$\text{Precision}_{\text{Bot}} = \frac{TP}{TP + FP}$$

$\text{Recall}_{\text{Normal}}$: It shows what percentage of normal flows are detected by classifier. This parameter is obtained by the ratio of number of normal flows that were correctly detected by classifier to the total number of available normal flows and is calculated as follow:

$$\text{Recall}_{\text{Normal}} = \frac{TN}{TN + FP}$$

$\text{Precision}_{\text{Normal}}$: It shows what percentage of the flows that were detected as normal flows by classifier are really normal. This parameter is obtained by the ratio of number of normal flows that were correctly detected by classifier to the total number of flows that were detected correctly or wrongly by classifier as normal flow. It is calculated as follow:

$$\text{Precision}_{\text{Normal}} = \frac{TN}{TN + FN}$$

F-measure: This parameter is a combination of two parameters of precision and recall, and is defined for botnet traffic and normal network traffic as follow:

$$\text{F} - \text{Measure}_{\text{Bot}} = 2 * \frac{\text{Recall}_{\text{Bot}} * \text{Precision}_{\text{Bot}}}{\text{Recall}_{\text{Bot}} + \text{Precision}_{\text{Bot}}}$$

$$\text{F} - \text{Measure}_{\text{Normal}} = 2 * \frac{\text{Recall}_{\text{Normal}} * \text{Precision}_{\text{Normal}}}{\text{Recall}_{\text{Normal}} + \text{Precision}_{\text{Normal}}}$$

Accuracy: It shows the total accuracy of the classifier. This parameter is obtained by the ratio of number of flows that were correctly detected by the classifier to the total number of flows as follow:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

5.4 Evaluation of the classifiers

We compute the introduced parameters for all 5 classifiers by using 10 fold cross validation method. Table 8 shows confusion matrix values for each classifier. Based on TP factor, Random Forest Classifier has the best performance (TP=429), while SVM has the worst (TP=409). In other words, Random Forest classifier detects the most number and SVM detects the least number of bot flows. On the other hand, based on TN factor, SVM has the best performance (TN=914), while Naive Bayes Classifier has the worst (TN=889). In other words, SVM detects the most number, while Naive Bayes classifier detects the least number of normal flows. As we have used 10 fold cross validation for training and testing classifiers, for each classifiers, TP+FN=432 (the number of all distinct bot flows) and TN+FP=918 (the number of all distinct normal flows).

Table 8. Confusion Matrix values for all classifier

Classifier	TP	FN	TN	FP
BNet	426	6	911	7
NB	418	14	889	29
SVM	409	23	914	4
J48	425	7	912	6
RF	429	3	911	7

The main purpose is to detect P2P botnet traffic flows in comparison with normal flows. Table 9 shows the results of classifiers in detecting P2P botnet based on the defined parameters. It is summarized as follows:

1. Based on Recall parameter, Random Forest classifier has the best performance (99.3%) and SVM has the worst performance (94.7%).
2. Based on Precision parameter, SVM classifier has the best performance (99%) and Naive Bayes classifier has the worst performance (93.5%).
3. Based on F-Measure parameter, Random Forest classifier has the best performance (98.8%), and Naive Bayes classifier has the worst performance (95.1%).

Table 9. Results of classifiers in detecting P2P botnet

Classifier	Recall (%)	Precision (%)	F-Measure (%)
BNet	98.6	98.4	98.5
NB	96.8	93.5	95.1
SVM	94.7	99	96.8
J48	98.4	98.6	98.5
RF	99.3	98.4	98.8

In botnet detection, Recall is much more important than Precision, because it represents what percentage of bot flows are detected by classifier. If the Recall is small, it means that the classifier has been able to detect a small percent of bot flows. In other words, many of bot flows are detected as normal flows, which threatens the system security. On the other hand, if classifier has detected a significant number of normal flows as bot, the precision drops, but it is not regarded as a security risk. In fact, it just bothers the user.

Based on this fact, Random Forest with botnet recall of 99.3 percent is the best classifier in detecting P2P botnet flows. Moreover, for F-measure parameter, this classifier has shown the best result.

Similarly, table 10, shows the results of classifiers in detecting normal network traffic flows based on the defined parameters. It is summarized as follows:

1. Based on Recall parameter, SVM has the best performance (99.6%), while Naive Bayes classifier has the worst performance (96.8%).
2. Based on Precision parameter, Random Forest has the best performance (99.7%), while SVM has the worst performance (97.5%).
3. Based on F-Measure parameter, Random Forest classifier has the best performance (99.5%), while Naive Bayes classifier has the worst performance (97.6%).

Table 10. Results of classifiers in detecting normal traffic flows

Classifier	Recall (%)	Precision (%)	F-Measure (%)
BNet	99.2	99.3	99.3
NB	96.8	98.4	97.6
SVM	99.6	97.5	98.5
J48	99.3	99.2	99.3
RF	99.2	99.7	99.5

As table 9 and 10 show, based on F-measure parameter, Random Forest classifier has totally the best performance, while Naive Bayes classifier has totally the worst performance in detecting all flows. Besides, table 11, shows the result of evaluating classifiers based on the accuracy parameter. Accuracy shows the total accuracy of classifier in that what percentage of P2P botnet as well as normal network flows are detected correctly by the classifier. Based on the accuracy parameter, Random Forest classifier has the best performance (99.26%), and the Naive Bayes classifier has the worst performance (96.81%).

Table 11. Result of classifiers based on accuracy parameter

Classifier	BNet	NB	SVM	J48	RF
Accuracy (%)	99.04	96.81	98	99.04	99.26

Finally, fig. 5 visually summarizes the performance of classifiers based on Recall(Bot), Precision(Bot), F-measure(Bot) and Accuracy parameters. As it is clear from this figure, Random forest classifier has the best performance compared to other classifiers based on these parameters.

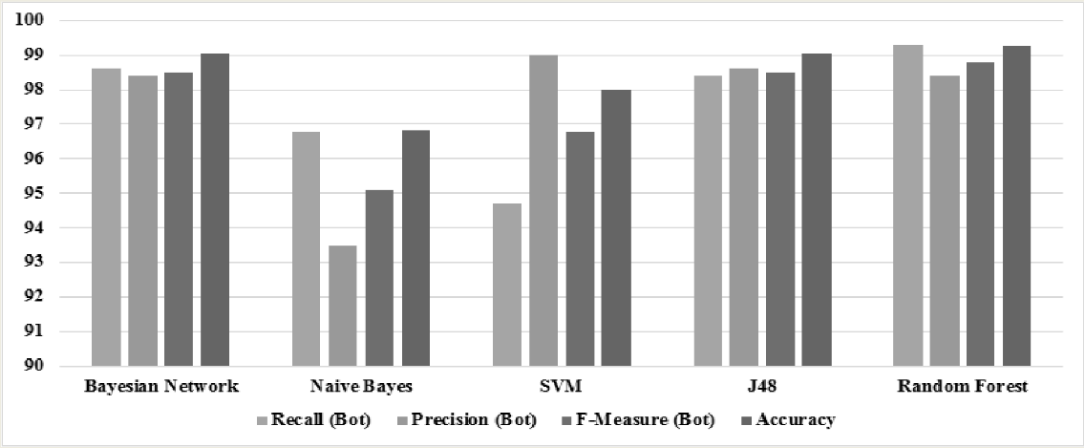

Figure 5: Performance of classifiers based on evaluation parameters

5.5 Comparison with other works

In this part, we compare the results of our proposed approach with two approaches that are based on analysis of traffic behavior.

Stevanovic et al. [26] propose a system based on traffic behavior analysis and flow-based features. In their system, 39 features are extracted from traffic flows and the performance of 8 classifiers has been evaluated including Naive Bayesian, Bayesian Network, Logistic Regression, Artificial Neural Networks, Support Vector Machine with linear kernel, C4.5 decision tree, Random Tree, and Random Forest classifier. In their system, the Random Forest Classifier shows better results than others classifiers.

Saad et al. [21] propose an approach based on network traffic behavior and using host-based as well as flow-based features. In their approach, 17 features are extracted from network flows and host communication patterns, and performance of 8 classifiers has been evaluated including Nearest Neighbors Classifier, Linear Support Vector Machine, Artificial Neural Network, Gaussian Based Classifier, and Naive Bayes classifier. In their approach, the SVM Classifier shows better results than other classifiers.

Table 12 shows the results of comparing our proposed approach with these systems. The best case, which is the best classifier for each system, has been considered for all approaches. The evaluation results show better performance for the proposed approach in all criteria.

Table 12. Comparison of proposed approach results with other published works

Approach	Best Classifier	Recall (%)	Precision (%)	F-Measure (%)
Stevanovic et al. [26]	Random Forest	95.7	96.2	96
Saad et al. [21]	SVM	98	------	------
Proposed Approach	Random Forest	99.3	98.4	98.8

Conclusion and Future Works

In this paper, we have proposed a new approach to detect P2P botnets by analyzing traffic behavior. The proposed approach is able to detect P2P botnets by using flow-based features and performing classification. The research has focused on detecting P2P botnets in C&C phase of life cycle in order to detect bots before their attack. Within a botnet, bots show unique communication patterns, which are different from the behavior of normal traffic flows. Therefore, we have selected a small particular set of flow-based features to help us distinguish P2P botnet from normal traffic flows. These features do not need to analyze the content or payload of traffic packets. As a result, the proposed approach is able to detect P2P botnets with encrypted traffic. The evaluation has been conducted by using five classifiers. Evaluation results have shown that the proposed approach has better performance in comparison with other systems.

As the botnet detection rate of the proposed system is not 100 percent, it is necessary to complete the approach by investigating P2P botnet detection in attack phase of life cycle. In future, we intend to

extend the system by addressing the attack phase of life cycle.

Based on the widespread use of mobile devices by people, they increasingly become the interested targets of new botnets. These botnets operate maliciously in mobile environment in order to reach their purposes like stealing important and critical information. Because of this issue, it is necessary to address the detection of these botnets. Therefore, another line of future research is to investigate on detecting botnets in mobile environment.

References

[1] M. D. Preda, M. Christodorescu, S. Jha, and S. Debray, (2008), A semantics-based approach to malware detection, "*ACM Transactions on Programming Languages and Systems" (TOPLAS)*, Vol. 30.

[2] M. A. Siddiqui, (2008), "*Data mining methods for malware detection"*, Ph.D. Thesis, College of Sciences at the University of Central Florida Orlando, Florida.

[3] M. Abu Rajab, J. Zarfoss, F. Monrose, and A. Terzis, (2006), A multifaceted approach to understanding the botnet phenomenon, SIGCOMM conference on Internet measurement, ACM, pp. 41-52.

[4] H. Choi, H. Lee, and H. Kim, (2009), "BotGAD: detecting botnets by capturing group activities in network traffic", International ICST Conference on Communication System software and middleware, ACM, p. 21-28.

[5] G. Gu, V. Yegneswaran, P. Porras, J. Stoll, and W. Lee, (2009), "Active botnet probing to identify obscure command and control channels", Annual Computer Security Applications Conference, IEEE, pp. 241-253.

[6] M. J. Elhalabi, S. Manickam, L. B. Melhim, M. Anbar, and H. Alhalabi, (2014), "A Review Of Peer-To-Peer Botnet Detection Techniques", *Journal of Computer Science, Science Publications*, Vol. 10, pp. 169-177.

[7] M. Feily, A. Shahrestani, and S. Ramadass, "A survey of botnet and botnet detection", International Conference on Emerging Security Information, Systems and Technologies, IEEE, 2009, pp. 268-273.

[8] M. Bailey, E. Cooke, F. Jahanian, Y. Xu, and M. Karir, (2009) "A survey of botnet technology and defenses", Cybersecurity Applications & Technology Conference for Homeland Security, IEEE, pp. 299-304.

[9] W. Tarng, L.-Z. Den, K.-L. Ou, and M. Chen, (2011), "The Analysis and Identification of P2P Botnet's Traffic Flows", *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 3, pp. 138.148.

[10] K. Muthumanickam and E. Ilavarasan, (2012), "P2P Botnet detection: Combined host-and network-level analysis", International Conference on Computing Communication & Networking Technologies (ICCCNT), IEEE, pp. 1-5.

[11] S. S. Silva, R. M. Silva, R. C. Pinto, and R. M. Salles, (2013), "Botnets: A survey", Computer Networks Journal, Elsevier, vol. 57, pp. 378-403.

[12] S. Khattak, N. Ramay, K. Khan, A. Syed, and S. Khayam, (2013), "A Taxonomy of Botnet Behavior, Detection and Defense", *Communications Surveys & Tutorials*, IEEE, Vol. 16, pp. 898-924.

[13] J. Govil and J. Govil, (2007), "Criminology of botnets and their detection and defense methods", International Conference on Electro/Information Technology, IEEE, pp. 215-220.

[14] H. R. Zeidanloo, M. J. Z. Shooshtari, P. V. Amoli, M. Safari, and M. Zamani, (2010), "A taxonomy of botnet detection techniques", International Conference on Computer Science and Information Technology (ICCSIT), IEEE, pp. 158-162.

[15] P. Barthakur, M. Dahal, and M. K. Ghose, (2013), "An Efficient Machine Learning Based Classification Scheme for Detecting Distributed Command & Control Traffic of P2P Botnets", *International Journal of Modern Education & Computer Science, MECS Publications*, vol. 5, pp. 9-18.

[16] J. Leonard, S. Xu, and R. Sandhu, "A framework for understanding botnets", International Conference on Availability, Reliability and Security, IEEE, 2009, pp. 917-922.

[17] K.-S. Han and E. G. Im, (2012), "A Survey on P2P Botnet Detection", *International Conference on IT Convergence and Security, Springer*, pp. 589-593.

[18] A. H. Lashkari, S. G. Ghalebandi, and M. R. Moradhaseli, (2011), "A Wide Survey on Botnet", *Digital Information and Communication Technology and Its Applications, Springer*, pp. 445-454.

[19] A. Al-Bataineh, (2012), "*Botnets analysis and detection methods based on network behavior*," Ph.D. Thesis, The University of Texas at San Antonio, College of Sciences Department of Computer Science.

[20] P. Wang, S. Sparks, and C. C. Zou, (2010), "An advanced hybrid peer-to-peer botnet," *IEEE Transactions on Dependable and Secure Computing, IEEE Computer Society*, vol. 7, pp. 113-127.

[21] S. Saad, I. Traore, A. Ghorbani, B. Sayed, D. Zhao, W. Lu, et al., (2011), "Detecting P2P botnets through network behavior analysis and machine learning", Annual International Conference on Privacy, Security and Trust (PST), IEEE, pp. 174-180.

[22] D. Zhao, I. Traore, B. Sayed, W. Lu, S. Saad, A. Ghorbani, et al., (2013), "Botnet Detection based on Traffic Behavior Analysis and Flow Intervals", *Computers & Security Journal, Elsevier*, vol. 39, pp. 2-16.

[23] S. Garg, A. K. Sarje, and S. K. Peddoju, (2014), "Improved Detection of P2P Botnets through Network Behavior Analysis", *Recent Trends in Computer Networks and Distributed Systems Security, Springer*, pp. 334-345.

[24] P. Narang, S. Ray, C. Hota, and V. Venkatakrishnan, (2014), "PeerShark: Detecting Peer-to-Peer Botnets by Tracking Conversations", IEEE Security and Privacy Workshops, IEEE, pp. 108-115.

[25] F. Chen, M. Wang, Y. Fu, and J. Zeng, (2009), "New detection of peer-to-peer controlled bots on the host", International Conference on Wireless Communications, Networking and Mobile Computing, IEEE, pp. 1-4.

[26] M. Stevanovic and J. M. Pedersen, (2014), "An efficient flow-based botnet detection using supervised machine learning", International Conference on Computing, Networking and Communications (ICNC), IEEE, pp. 797-801.

[27] I. H. Witten and E. Frank, "*Data Mining: Practical machine learning tools and techniques, Third Edition, (The Morgan Kaufmann Series in Data Management Systems)*", San Francisco, USA: Morgan Kaufmann Publishers Inc., 2011.

[28] R. Kohavi and F. Provost, "Glossary of Terms, Special Issue on Applications of Machine Learning and the Knowledge Discovery Process", *Machine Learning Journal, Kluwer Academic Publishers*, vol. 30, pp. 271-274, 1998.