*International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*
**Web Site: www.ijettcs.org Email: editor@ijettcs.org**
**Volume 5, Issue 2, March - April 2016** **ISSN 2278-6856**

# Survey of the P2P botnet detection methods

**Atef A. Obeidat[1], Mohmmad J. Bawaneh[1]**

[1]Al-Huson University College,
Al-Balqa Applied University, Al-Huson, Jordan

**Abstract:** ***Botnets are one of the important problems of the Internet. There are many proposals to detect botnets. The survey analyzes and compares significant proposals in the P2P based detection area. In the beginning, the work studies the previous surveys. Then, it classifies, compares, and discusses the proposals in the area. We conclude that P2P detection botnets ways significantly evolved. However, many open problems still exist.***

**Keywords:** Peer-to-Peer, Information security, Network behavior, Botnets detection, Survey

## 1. INTRODUCTION

A "botnet" is a network of computers (bots) running a malicious program. it is controlled by botmaster which is called the attacker, such as worms, and viruses. They are employed by botmaster to initiate different malicious activities, such as email spam, distributed denial-of-service attacks, password cracking and key logging.

In the beginning, most botnets have centralized command and control (C&C) architecture, such as Internet Relay Chat to receive instructions from a single source. later, peer-to-peer (P2P) structured botnets have developed as a new advanced form of botnets. Due to the distributive nature of P2P networks, P2P botnets have become more difficult to detection. Nugache[1] and Storm worm [2, 3] are two representatives of P2P botnets.

In this paper, first we systematically study surveys on P2P Botnet detection[4-7]. Although these surveys provided more information on botnets, these surveys didn't include an analysis of P2P botnet detection methods. Then we provide classify and compare the more significant P2P botnet detection methods.

The contributions of this work are summarized as follows: An analysis of the previous surveys. A classification of the more significant proposals. A novel comparison and discussion of the more significant proposals. We conclude that P2P botnet detection methods have gained significant advancements so far. However, many open problems still exist.

## 2. PREVIOUS SURVEYS

No complete and comprehensive surveys on P2P botnet detection methods were carried out so far. However, several surveys on botnets detection include a brief analysis of detection methods. In this section, these surveys are analyzed to describe how the detection methods are classified.

Several methods for P2P botnet detection are enumerated in a previous work [4]. These methods are not thoroughly analyzed. In addition, the detection techniques in previous studies are individually considered, and the ideas and drawbacks of these techniques are explained. A previous work [4] analysis three approaches more elaborately and one of the three approaches is proposed.

The general ideas, advantages, and shortcomings of these approaches are presented[5].

P2P botnet detection techniques based on traffic features of the P2P botnet were studied [6]. Detection techniques are classified into three categories: data mining, machine learning, network behavior and traffic analysis. The general principles of these techniques are presented. However, the shortcomings, advantages, and algorithms of these techniques were not explained.

P2P botnet detections methods are surveyed by several review papers in the field, which determined the current problem and existing solution[7]. In addition, the survey uses a table to relate the proposed techniques in the literature.

## 3. COMPARISON APPROACH

This section classifies the papers according to two dimensions: (1) the type of detection algorithms and (2) The categories of detection techniques.

a. The type of detection algorithms differentiates the types of algorithms.

The neuronal network-based algorithm uses the neuronal network to build supervised machine learning models for the detection of P2P botnets. The approaches are validated with the following multiple machines learning algorithms: decision trees[8].

a) boosted REP trees[8], Bayesian networks[8], discrete Fourier transforms and Shannon's entropy theory[9], Weka machine learning suite[15], SVM [10, 11], J48 [10, 11], C4.5 decision tree classifiers [10, 11], Apache Mahout Algorithms [18], and FURIA, which is a fuzzy rule generating algorithm [17].

b) The heuristic threshold-based algorithm uses a threshold for detection. For example, the Packet ratio[22], which is the sum of up packets divided by

the sum of down packets, is less than 0.4, and it uses Traffic patterns with three periods of more than five minutes apart and with a standard deviation of less than 150. In addition, the distance between two bot-compromised hosts that is decided by the minimum distance of their respective fingerprint clusters is employed in a previous work [24]. The distances of fingerprint clusters from botnet P2P protocols are smaller compared with those from legitimate P2P protocols.

c) The mining-based algorithm uses a data mining technique to detect botnets. For example, synergistic graph-mining is applied in a previous work [12], whereas another work [16] is based on mining the periodic patterns of traffic datasets.

b. The categories of detection techniques differentiate the main techniques used for detection.

a) Flow-based techniques examine network flows between two nodes.
b) Resource sharing behavior monitoring-based techniques model the evolution of the number of peers sharing a resource in a P2P network over time.
c) Node-based techniques examine the input and output flow for every node.
d) Conversation-based techniques aim to detect the stealthy behavior of P2P botnets.

### 3.1 The categories of detection techniques comparison

Table 1 shows a comparison of the algorithms and techniques employed in each paper. The existing solutions for P2P botnet detection can be broadly classified into the following techniques:

1. Flow-based[9-17] or flow analysis-based bot detection examines network flows between two nodes where a flow is defined as a set of packets with the same source address, source port, destination address, and destination port. The principle of these approaches is that the flow features, such as the count of the packets in the flow, order of the packet arrivals, and interval between packets, can model the botnet communication patterns more accurately than direct packet inspection. The extracted features are adopted to construct a classifier that can differentiate normal flows from malicious bot flows. Classifiers use statistical profiling. Thus, flow-based analysis can detect unknown bots that exhibit behavioral similarities to known bots. However, flow-based techniques have two key limitations. First, several flows between any two network nodes need to be analyzed. Generally, most of these flows belong to normal network processes. Second, the flow features must be extracted at runtime, which implies that flow-based analysis requires considerable computational overhead at runtime. At any given instant, a significant number of flows exists in the network, which aggravates the effect of these limitations further more.

2. Resource sharing behavior monitoring-based detection [18] is grounded on modeling the evolution of the number of peers sharing a resource in a P2P network over time. This allows the detection of abnormal behaviors associated to parasite P2P botnet resources in this kind of environment.

3. Node-based detection[19, 20][25]examines input and output flow for every node where the approaches aggregate behavioral metrics for each P2P node (host) seen in network communications and use them to distinguish benign P2P hosts from hosts infected by P2P botnets.

4. Conversation-based detection[14, 15, 21, 22]does not rely on deep packet inspection or signature-based mechanisms. This approach aims to detect the stealthy behavior of P2P botnets, that is, when they lie hidden in their rally or waiting for stages or while they perform malicious activities (spamming, password stealing, etc.) that are not observable by a network administrator.

**Table 1**:Categories of comparing detection techniques

| Category | Papers |
|---|---|
| Flow based | [9, 12, 13, 16, 17, 19, 23][25] |
| Node based | [19, 20] |
| Based on resource sharing behavior monitoring | [18] |
| Conversation-based | [10, 11, 14, 21, 22] |
| Hybrid with flow-based and conversation-based | [15] |

### 3.2 Comparison of detection algorithms

Table 2 shows a comparison of the algorithms and techniques used in each paper. Creating this table was difficult because some proposals did not describe the algorithms explicitly. All the papers employed heuristic-based rules at some point of the analysis.

**Table 2**:Comparison of detection algorithms

| Algorithm | Papers |
|---|---|
| Neuronal network based | [9-11, 14, 15, 23] |
| Heuristic threshold based | [16, 18, 21, 22][25] |
| Mining-based | [12, 16] |

In the next section, each method is described in detail.

## 4. DETAIL ANALYSIS OF PAPERS

Previous sections described a set of tables to compare the detection proposals. This section analyzis the context on which each proposal was created. It also summarizes the steps of each detection method and highlights each proposal difficulty, bias, assumption, hypothesis, dataset,

and result. Each paper is described and analyzed in the next subsections.

### 3.4 Building a scalable system for stealthy p2p-botnet detection

The paper in[13] uses unsupervised machine learning approaches to separate P2P botnet traffic from benign traffic. The approach used 'control flows' of P2P applications to extract statistical fingerprints. P2P bots were identified based on certain features like fingerprint similarity, the number of overlapping contacts, persistent communication, etc. However, the work can detect P2P bots inside a network only when there are multiple infected nodes belonging to the same botnet.

In this paper, the system identifies all hosts that are likely engaged in P2P communications. Then it derives statistical fingerprints to profile P2P traffic and also distinguish between P2P botnet traffic and legitimate P2P traffic.

The approach depends on a flow-clustering-based analysis approach to identify hosts that are mostly likely running P2P applications. The approach can detect and profile various P2P applications rather than identifying a specific P2P application (e.g., BitTorrent); and the analysis approach can estimate the active time of a P2P application which is critical for botnet detection.

The system is divided into two phases: "Phase I" aims at detecting all hosts within the monitored network engaged in P2P communications. The system analyzis raw traffic collected at the edge of the monitored network and applies a pre-filtering step to discard network flows that are unlikely to be generated by P2Papplications. The system then analyzis the remaining traffic and extracts a number of statistical features to identify flows generated by P2P clients.

In the second phase, the system analyzis the traffic generated by the P2P clients and classifies them into either legitimate P2P clients or P2P bots. Specifically, the system investigates the active time of a P2P client and identify it as a candidate P2P bot if it is persistently active on the underlying host. The system further analysis the overlap of peers contacted by two candidate P2P bots to finalize detection.

To illustrate the statistical features and motivate the related thresholds used by the system, it ran five popular P2P applications for 24 hours to collect their traffic traces.

The training dataset is composed of 200-300 Mbps from private traffic; which consists of 5 real legitimate P2P clients from DNS traffic and 5 virtual legitimate P2P clients of popular P2P applications. In addition, a 16 P2P network of a compiled bonnets, which are consisting of the Storm traces included 13 hosts and the Waledac included 3 hosts.

### 3.5 Towards Accurate Node-based Detection of P2P Botnets

In [19] proposed an approach for P2P botnet detection called node-based detection. This approach focuses on the network characteristics of individual nodes. Based on the specific model, examining node's flows and extract the useful features over a given time period.

The approach consists of four steps: P2P bot quantification, efficient flow monitoring, classification, and evaluation. In the first step the approach monitors 7 features: (1) Node: Computer address for transmitting information,(2) NP: Number of protocols used for time interval, (3) NF: Number of flows used for time interval, (4) NPS: Number of packets sent for time interval, (5) ALPS: Average length of packets sent,(6) RNP: Ratio of number of packets sent to number of packets received for time interval.(7) RLP: Ratio of average sending packets length to average receiving packets length for the time interval. The second step, the approach used the sampling approach, in which, sampling the packets in a periodic manner, thereby reducing the number of packets that need to be captured. The third and fourth steps: Machine learning classification techniques attempt to cluster and classify data based on feature sets.

The decision tree is selected as a classifier technique for evaluation. Decision tree based classifiers exhibit desirably low computational complexity with high performance. In a decision tree, interior nodes represent input features with edges extending from them that correspond to possible values of the features. These edges eventually lead to a leaf node which represents an output variable corresponding to a decision. During the detection phase, the feature set extracted from node's flow information is given to the classifier which essentially classifies this feature set into malicious or no malicious feature set.

The difficulty in the approach; it is necessary to design the system that can evaluate the performance of the detection online instead of the present off-line mechanism. It is also important to train the detection system online, instead of an off-line training process so that it is suitable for live deployment.

The approach is tested on real-life data sets and achieved detection rates of 99-100% and low false positives rates of 0–2%.

### 3.6 Resource monitoring for the detection of parasite P2P botnets

The paper in[18] introduces a detection scheme which is based on modeling the evolution of the number of peers sharing a resource in a P2P network over time. This allows detecting abnormal behaviors associated to parasite P2P botnet resources in this kind of environments.

The main intuition behind the proposal is the fact that resources shared by legitimate users in a P2P network will be accessed in a different way rather resources shared by

nodes belonging to a botnet. the approach contains two models that are building for both legitimate and botnet resources. The approach develops a detection architecture that relies on these models to detect botnet resources in P2P networks. The models are based on the evolution of the number of P2P nodes that share a specific resource over time. The approach aim at building a detection system which is able to be the number of nodes sharing a source for the different resources in a P2P network, and detect potential botnet resources patterns.

The data from the resource monitoring system is fed into the system, through three stages:

- Preprocessing: Both the training and detection processes require a previous common stage to preprocess the data given by the resources monitoring system. All shared resources of Mainline with the same 8 bits prefix for the ID: "0x8C" are monitored.

- Training: First, a normality model is built to represent the sharing evolution of legitimate resources in the monitored P2P network. In order to make the training, firstly the project identifies the legitimate resources. This is done by obtaining information published in well-known sites about the resources IDs and considering those that have a good reputation from users. Out of the 34,075 resources, the project selected 14,869 resources which are corroborated as legitimate.

- Detection: After the model is obtained, every resource shared in the P2P network is analyzed in quasi-real-time in order to determine potential deviations with respect to the expected behavior.

The training data set is divided into four parts so that a cross validation process is performed by taking three partitions for training purposes and the remaining one for testing. Additionally, 42,000 synthetic bot resources following the above model are added to the test partition in each case to the traces collected from Mainline. For discovering botnet patterns, the system uses the remaining 19,206 resources obtained in the monitoring process which is not previously used for training and testing the system.

### 3.7 PeerShark: Detecting Peer-to-Peer Botnets by Tracking Conversations

In [14], PeerShark: Detecting Peer-to-Peer Botnets by Tracking Conversations (PeerShark_TC) approach is proposed, a method to detect P2P botnet traffic and differentiate it from benign P2P traffic in a network. Instead of the traditional 5-tuple 'flow-based' detection approach, PeerShark_TC uses a 2-tuple 'conversation-based' approach which is port-oblivious, protocol-oblivious(<sourceIP,destinationIP>) and does not require Deep Packet Inspection.

Using the simple and novel conversation-based features, the approach can also correctly categorize different kinds of P2P applications with high accuracy. The present evaluation is also limited to two benign P2P applications. Another limitation that PeerShark_TC's present approach gives a bird's eye-view of the conversations happening in the network. Being flow-oblivious, many lower-level details are neglected.

PeerShark_TC consists of the following four modules: Packet Filtering Module which reads each packet and discards those which have invalid IPv4 header. From each packet, the Source IP, Destination IP, Payload length, and Timestamp are extracted and stored for future use. Conversation Creation Module which creates a list of conversations by aggregating packets received from the previous module. Each conversation is identified by the binary tuple <IP1, IP2>and an initial FLOWGAP value. Conversation Aggregation Module: The conversations created in the creation module are aggregated for a higher FLOWGAP value as desired by a network administrator. The resultant conversations are then used to train the classification model. The attributes of each conversation analyzed are Number of packets, Conversation volume, Conversation duration and the Median value of Inter-arrival time of packets in the conversation. Classification Module uses supervised machine learning algorithms for training its model and classifying the test data. To validate proposal approach, models were built using a number of algorithms, namely Bayesian networks, Decision trees and Boosted REP trees.

The training dataset consists of two P2P applications (eMule and uTorrent) and two P2P botnet applications (Waledac and Storm) while the testing dataset consists of the Benign P2P Data: 50,000 conversations each of eMule and uTorrent and 50,000 conversations each of Storm and Waledac.

### 3.8 Poster: Machine-learning approaches for P2P botnet detection using signal-processing techniques

The work in [9] proposes an approach for the detection of P2P botnets by converting the 'time-domain' network communications of P2P botnets to 'frequency-domain'. The method adopts a signal-processing-based approach by treating the traffic of each pair of nodes seen in network traffic as a 'signal'. Apart from the regular 'network behavior' based features, the work extracts features based on Discrete Fourier Transforms and Shannon's Entropy theory to build supervised machine learning models for the detection of P2P botnets.

This work attempts to detect P2P botnets in the network by identifying their hidden communication patterns by incorporating signal-processing techniques. The work aims to detect malicious communication activity of P2P botnets in a network and thus aid a network administrator wanting to detect and block such botnet activity.

For the detection of P2P botnets, the approach depends on detecting P2P bots entirely on the basis of their 'P2P' behavior and C & C communications with other bots. It extracts 2-tuple conversations from network traffic and

treats those conversations as a 'signal'. Furthermore, in order to uncover hidden patterns between the communications of bots, the approach converts the time-domain network communication of peers to the frequency-domain. Apart from extracting the regular' network behavior' based features; it extracts several 'signal-processing' based features using Discrete Fourier Transforms and Shannon's Entropy theory.

The approach consists of the following modules:

1. Packet filter module. The module takes network log files (.pcap) as an input. The approach keeps only the packets with a valid TCP/UDP header. For each packet, the SourceIP, DestinationIP, Payload-Length and Timestamp are extracted. This information is used to generate conversations and develop an elaborate feature set in next modules.

2. Conversation creation module. Conversations are created by aggregating packet-level data. Each Conversation is identified by <IP1, IP2> and an FLOWGAP parameter. FLOWGAP is defined as the maximum permissible inter-arrival time between 2 packets in a conversation. If a packet arrives which belongs to the IP pair of a conversation and its timestamp lies within FLOWGAP time from the beginning or the end of that conversation, the packet will be added to the conversation. Otherwise, a new conversation will be created for that IP pair.

3. Feature extraction module. The C & C traffic of different bot families exhibits certain regularities that can be leveraged for network-based detection of bot-infected hosts. Thus, in addition to 'network behavior' based features, the approach also extract several Discrete Fourier Transform (DFT) based features and Entropy-based features. The features extracted for this work are Duration, Payload, Inter-arrival times, Number of Packets and Compression Ratio.

The dataset used for this work includes 3 P2P botnets and multiple P2P applications. The dataset generates a representative with the ratio of a botnet to benign traffic being 20:80. A total of 78,000 conversations of benign traffic were sampled from the entire p2p applications dataset.

### 3.9 PeerShark: flow-clustering and conversation-generation for malicious peer-to-peer traffic identification

Paper [15] presents a methodology to detect P2P botnet traffic and differentiates it from benign P2P traffic in a network. The approach aims to detect the stealthy behavior of P2P botnets. That is; the aim is to detect P2P botnets when they lie dormant or while they perform malicious activities in a manner which is not observable to a network administrator. The work combines the benefits of flow-based and conversation-based approaches with two-tier architecture, and addresses the limitations of these approaches by extracting statistical features from the network traces of P2P applications and botnets, and by building supervised machine learning models which can accurately differentiate between benign P2P applications and P2P botnets

PeerShark uses a two-tier approach to differentiate P2P botnets different from benign P2P applications. The first phase clusters P2P traffic flows based on the differing behavior of different applications. In the second phase, conversations are created from flows within each cluster. Several statistical features are extracted from each conversation and are used to build supervised machine learning models for the detection of P2P botnets.

The system design of the work consists of the following phases: Flow-clustering phase. At this stage, the approach separates the flows into different clusters based on their behavior. Conversation-generation phase. In this phase, the approach creates conversations from flows within each cluster. Limiting conversation creation to the flows within each cluster. Since flows within the same cluster have similar behavior, the approach is creating conversations out of only those flows which show similar behavior. After creating conversations from flows, the approach will extract four statistical features from each conversation: the duration of the conversation, the number of packets exchanged in the conversation, the volume of data exchanged in the conversation, the median value of the inter-arrival time of packets in that conversation. These features are then used to build supervised machine learning models to differentiate between benign and malicious P2P traffic.

This work uses data of benign P2P applications and P2P botnets obtained from two different sources. The data of four benign P2P applications, namely uTorrent, eMule, Vuze, and Frostwire, and the data of three P2P botnets, namely Storm, Waledac, and Zeus, was obtained from the University of Georgia. The dataset contained 1,654,730 conversations(1,589,808 benign and 64,922 malicious). This dataset was split into training and testing datasets in a 2:1 ratio. The training dataset had 1,092,122 conversations (1,049,242 benign and 42,880 malicious), and the test split contained 558,348 conversations (540,566 benign and 22,042 malicious).

### 3.10 PeerMinor: Behavioral fine-grained detection and classification of P2P bots

Paper [10] presents PeerMinor, a fully behavioral system that detects and classifies P2P bots inside corporate networks. PeerMinor learns the behavior of known malware and benign P2P applications in order to detect P2P bots and provide security administrators with a correct diagnosis of ongoing malware infections.

PeerMinor operates in two phases, learning, and detection. In the learning phase, it processes known malware and benign P2P traffic in order to build a two-stage classifier. In the first stage, it uses supervised learning in order to build a detection model that separates malicious and

benign P2P network activity. In the second stage, it builds a one-class classifier for each known P2P malware family and uses these classifiers to associate detected P2P bots with a known malware family where possible, thus providing a better situational awareness for system administrators. During detection, PeerMinor processes network traffic using its learning-based model in order to detect P2P bots. The P2P detector applies both inspection and classification models to network traffic in order to detect P2P bots, with no need for deep packet inspection.

PeerMinor detects and classifies P2P bots using only network traffic features. The system builds a P2P malware detection model using a learning set of malware and benign P2P traffic. Then it applies this model to network traffic in order to detect P2P bots inside a given network perimeter. PeerMinor goes beyond the simple detection of P2P bots in order to associate infected nodes with a known P2P malware family where possible.

The training phase of PeerMinor includes two main components: the P2P inspector and P2P malware classifier. P2P inspector uses supervised learning to separate malicious and benign P2P traffic when observed for a given network node. On the other hand, P2P malware classifier builds a P2P footprint for very infected node and assigns this footprint to a known P2P malware family. PeerMinor also notifies the administrator of a new, yet unknown malware P2P footprint so it can be submitted to a deeper manual analysis.

The advantages for PeerMinor are the first behavioral system that goes beyond simple detection in order to provide an accurate diagnosis about ongoing malware infections. Addition Experimental results explained that PeerMinor achieves both scalability and accuracy. It uses only network features with no need of pattern-based signatures, which can be easily evaded by botnet herders.

The training dataset consists of: 794 benign P2P clusters and 1, 445 malware clusters. 80% of 1, 445 malware clusters for training and 20% of 1, 445 malware clusters for testing.

### 3.11 A P2P Botnet Detection Method Used On-line Monitoring and Off-line Detection

In paper[21], a new method is proposed to detect the P2P botnet through the analysis of the P2P botnet host's life cycle, use the method of off-line detection to find the suspected botnet hosts, and determine the P2P botnet host through online monitoring method.

The present detections of P2P botnets focus on the analysis of the botnet flow. After that, the infected hosts by botnets are recognized through off-line detection and on-line detection. The paper observes different ability behaviors of a botnet in different stages, analyzes with synthetic judgment, thereby obtains the information of P2P botnet host. The design here suggests the steady monitoring of the inactive and no-harm botnet hosts but takes measures to stop the contact between the control nodes and the attack stage botnet hosts.

The proposal approach depends on the features of different stages of lifecycle as the following:

(1) P2P botnet hosts produce many ICMP reports with low rate of successful linking during the initial stage.

(2) P2P botnet hosts are linked to many nodes with the same communication traffic during the trance stage,

(3) P2P botnet hosts produce too much SMTP contact with too much similar data package of destination address during the attack stage.

The features in (1), (2) can be extracted off-line to detect the address of P2P botnets hosts and the hosts can be monitored on-line. If behaviors like in (3) are spotted, the hosts can be stopped immediately.

The design of the approach consists of the two parts:

4. The Off-line Detection. After a certain period of collection of net flow from the network exports, if the low rate of connection succeeds and many nodes with the same communication traffic distinguished as the behavioral features in those botnet stages, the source host address can be separated as the suspected botnet host address. The off-line detection depends on connection success rate; If the rate is between 0 and 0.1, it shows the low connection between the source address and outside destination address, which can be caused by the suspected botnet hosts.

5. The On-line Monitoring. The suspected botnet hosts through the off-line detection are all in their initial stage and trance stage and cause harm to other hosts or network. There will be misreported if they are judged as botnet hosts. So the paper here suggests the on-line and continuous monitoring of the suspected hosts to precisely locate P2P botnet hosts. Once there is the harmful behavior, the suspected host will be stopped to minimize the harm.

The approach used for testing compiled virtual 5 P2P hosts, traffic each of them for 2 minutes.

### 3.12 Peer-to-Peer Botnet Detection Using NetFlow

The work in [22] aims at the detection of individual p2p bots within a network perimeter. This is done by looking at the communications with their p2p overlay network. The NetFlow protocol is used to gain insight in all traffic within the network. A detection algorithm is proposed so that it can detect p2p malware in live NetFlow data. The algorithm is based on characteristics that separate malicious from benign p2p traffic, such as: traffic volume, packet symmetry, and traffic patterns.

The approach consists of the following steps:

6. P2P filter. To reduce the scope of traffic and the chance of false positives, the P2P traffic was first filtered out from other traffic.

7. Traffic volumes. the system uses only the total amount of traffic over a certain amount of time, it's likely that false positives would be triggered when a benign p2p application downloads a small file. Therefore, it's better to look at statistics such as the average bytes per flow and the average bytes per packet.

8. Packet symmetry. Packet symmetry is the relation between the outgoing and incoming packets. The packet ratio is calculated by dividing the outgoing packets by the incoming packets.

9. Traffic pattern. The Zeus malware has a control loop that periodically wakes up the bot to contact its peers and query them for their current configuration.

The implementation of the approach is explained in the following algorithm[22]:

Detection algorithm

1. *Group flows by source IP + port*
2. *Filter p2p traffic: sources with more than 4 failed connections to different hosts are considered p2p*
3. *Detect Zeus based on either:*
   a. *Packet ratio: the sum of up packets divided by the sum of down packets is less than 0.4*
   b. *Traffic pattern: there are 3 periods of more than 5 minutes apart, with standard deviation of less than 150*

The dataset of the work includes web traffic generated by crawlers that simulate human-like browsing behavior. Web traffic also includes DNS traffic and data streams from YouTube and Internet radio. The data set also includes a lot of p2p traffic generated by different file sharing applications on different p2p networks. Three different binaries of the Zeus p2p malware were obtained via the public sandbox malwr.com.

All the traffic in the data set was replayed and the exported flows were sent to the implemented NetFlow collector in real time. The result was a 100% true positive rate and a 0% false positive rate. Of course, because of the limited data set, no statements can be made about what the results would be with real user data. The detection algorithm needs to be tested with more and different data.

### 3.13 Entelecheia: Detecting p2p botnets in their waiting stage

The work in [12] presents Entelecheia, which is an approach for detecting peer-to-peer botnets during their Waiting stage by exploiting their "social" behavior. The driving insight for the work focused on long-lived and low-intensity flows.

The approach is decomposed into two modules: (a) model the network-wide interactions of the hosts in the network as a graph, (b) focus on likely botnet activities, and(c) identify clusters of potentially infected machines.

Superflow Graph Module: it is responsible for creating the Superflow Graph in which each node represents a host in the network and each edge is a vector of attributes that summarizes all of the Superflows sent and received between a pair of source and destination IPs. This Superflow Graph represents the social communication behavior between hosts.

Filtering and Clustering Module: The work examines the Superflow Graph and assigns as edge weights the total duration of the Superflows (α d)between the two nodes then clustering nodes together by their connectivity would group infected hosts into communities with a high percentage of long-lived edges.

Datasets are traced of 24 continuous hours from a Trans-Pacific back bone line between the U.S. and Japan on 03/03/2006. 89% of all flows are TCP and the rest are UDP. The real-world malicious network traces contain observed data from 13 hosts infected with Storm and 82 hosts with Nugache during a period of 24 hours.

A graph-clustering approach may not scale as the network size grows. Further, the work evaluates the detection of P2P botnets only with regular web traffic. This is a serious limitation because P2P botnet traffic exhibits many similarities to benign P2P traffic, and distinguishing between hosts using regular P2P applications and hosts infected by a P2P botnet would be of great relevance to network administrators protecting their network.

### 3.14 BotSuer: Suing Stealthy P2P Bots in Network Traffic through Netflow Analysis

The paper in [11] presents BotSuer; it aims at detecting p2p bots based on their behavior, using NetFlow data from within a company network. The approach observes only high-level malware traffic features with no need of deep packet inspection. It uses machine learning techniques to differentiate between benign and malicious p2p trail.

The approach strategy is by replacing signatures with behavioral network models. The goals of the approach are: Extract P2P traffic, it is based on empirical facts and behavioral patterns of P2P applications, that is by extracting P2P network flows and cluster similar P2P

The approach uses multiple heuristics to discard flows unlikely to show P2P activity. It uses two filtering steps, including coarse-grained and fine-grained filtering. Clustering P2P flows by signaling activity and discarding non-P2P flows using geographical distribution and destination ports statistics.

The approach is supervised machine learning to build P2P botnet detection model. It considers three categories of features to characterize P2P flows: Time features; to describe long term malware P2P signaling activity, space features; to describe chunk rate and distribution of P2P botnets and flow-size features; to describe control operations in P2P botnets. The approach is applied set of learning algorithms for testing.

The approach is considered an initial dataset of up to 20 thousand distinct malware samples by using virus Total

API to identify P2P malware in the initial dataset. An overall number of 1,317 P2P malware samples are used to build the malware classifier, belonging to 8 different malware families.

In the learning stage, the approach uses 2,975 P2P flow clusters that is used to build the supervised P2P botnet detection model. The benign P2P learning set includes 794 benign P2P flow clusters; 415 P2P clusters using the P2P filter applied to corporate network traffic and 379 P2P clusters obtained by manually executing P2P applications .

The approach was tested by 3 hours of anonymzed NetFlow for 4,347 distinct IP addresses. The result was 793 P2P flow clusters are discovered by the P2P filter, associated with146 distinct IP addresses, with no false positives and with 3.4% false negatives. The approach also discovers 11 P2P flow clusters identified by the system as being malicious botnet communications, with 4 P2P flow clusters associated with the same IP address and 20% suspicious destination IPs according to the rbls framework. Whereas 1 true positive associated with a P2P botnet infection and 0.8% false positives rate.

### 3.15 PeerViewer

Paper [23] presents PeerViewer, a system that automatically classifies malware according to its network P2P behavior. PeerViewer builds classifiers for known P2P malware families. Then it builds a network footprint for malicious code running in a sandbox, and compares this footprint with those for known P2P malware families. It associates malicious code with a known botnet family where possible, or it notifies the security analysts of a new or unknown P2P malware family, so it can be considered for a deeper analysis.

PeerViewer includes three separate modules.

1. The P2P flow filter implements several heuristics which aim to discard malware that does not show any P2P activity during analysis. For instance, the rate of failed connection attempts is usually used as a way to detect P2P applications. Therefore, the filter discards malware whose rate of failed connection attempts does not exceed a given threshold. It also uses other heuristics such as flows initiated after successful DNS requests, number and geographical distribution of remote contacted IPs.

2. The flow clustering module, its input the flows P2P malware in the dataset. Malware that implements the same P2P protocol and belongs to the same P2P botnet topology would have the same P2P signaling activity, thus resulting in similar flows when observed at the network level. PeerViewer uses unsupervised clustering in order to group together similar malware flows that are likely to implement the same P2P activity. The flow clustering process uses high-level malware traffic features such as flow size, number of packets, bits per packet, and flow duration. The output of this process is a multiple set of clusters, each one including P2P flows triggered by multiple malware samples, but carrying the same P2P signaling activity (e.g. keep-alive, route discovery, search request, push data) and protocol.

3. The malware classifier module uses P2P flow clusters in order to build families of malware that implement the same P2P protocol. In fact, PeerViewer builds a P2P footprint of size m for each malware in the initial learning set, and which specifies the rate of malware P2P flows within each cluster. PeerViewer uses malware footprints as a training set to build P2P malware clusters, each cluster represents a new P2P malware family. Hence, malware that belongs to the same family implements the same P2P protocols and has the same P2P botnet topology.

The dataset includes thirty minutes of network traffic for malware executed in a dynamic analysis environment. The dataset at the disposal includes network traffic for almost twenty thousand distinct malware samples collected during a three months period. PeerViewer builds clusters of flows in order to group together malware flows that implement the same P2P protocol and signaling activity. It applies incremental k-means to the entire set of malware P2P flows. The flow clustering module, applied to the 450 malware samples in the dataset, provided a total number of 28 P2P flow clusters, including 22 clusters of UDP flows and 6 clusters of TCP flows.

### 3.16 Hades: a Hadoop-based framework for detection of peer-to-peer botnets

Paper [20] presents HaDeS, a Hadoop-based framework for detection of P2P botnets in an enterprise-level network, which is distributed and scalable by design.

The system architecture of HaDeS consists of the following:

1. HaDeS proposes a distributed data-collection technique wherein data collectors sit close to the peers inside the network perimeter. The implementation of HaDeS has multiple data collectors distributed inside the network perimeter. The initial deployment of the system has data collectors deployed at WiFi access points within the University campus.

2. The automated parser module which parses the network traces and extracts packet-level features. The features extracted from each packet are Time-stamp of the packet, Source IP, Destination IP, Time-to-live (TTL) value, Transport layer protocol (TCP/UDP), TCP or UDP payload length (as applicable). The extracted features are stored in a .csv file at each data collector. Then the files are periodically transferred from all data collectors to the Hadoop Distributed File System.

3. The Hadoop Distributed File System (HDFS). For the purpose of data sanitization, all packets which didn't contain a valid IPv4 header are removed. Presently HaDeS does not support IPv6.The present approach of HaDeS also disregards all packets corresponding layers below the IP layer, such as ARP broadcast messages. Packet-level

data obtained from multiple data collectors is aggregated per host for every host seen in network communication. The packet-level data is stored in Hive.

4. Hive in the form of external tables. For the task of detecting P2P botnets, the following statistical features are aggregated over a time-period T (say, one hour) for every P2P host inside the network: the number of distinct destination hosts contacted, the total volume of data sent from the source host and the average of the TTL value of the packets sent from the source host

5. The host-based features extracted above are used to train supervised machine learning models. Apache Mahout is used for this purpose. Mahout is a fairly new tool, and at present does not offer many machines learning algorithms. Further, many of Mahout's algorithms (for classification and clustering) do not run as Map Reduce jobs.

6. The results generated from HaDeS can be used to trigger rules to a Firewall to alert the network administrator for suspicious malicious activity in the network and/or log or drop suspicious botnet traffic. This way HaDeS can be used by network administrators as an assisting tool which is 'P2P-aware'.

The dataset consists of network traces of two P2P applications, and two P2P botnets.

The system could detect bot-infected hosts with a True Positive rate of 97% and 99%, and a low False Positive rate of 5% and 2% over training and testing datasets respectively.

The present approach of HaDeS is limited to detection of bots when bots and apps are not running on the same host. If a host which is running P2P applications is also infected with a bot, HaDeS will be unable to correctly classify it as an infected host.

### 3.17 Detecting P2P bots by mining the regional periodicity

The work in [16] presented a novel method for the detection of P2P botnets traffic based on the nature of P2P applications, we denoted by DMRP in our work. The work investigates some normal P2P applications (such as eMule and BitTorrent) and malicious P2P botnets (such as the Storm botnet and the improved P2P-Zeus botnet). One of the work discoveries is that periodicity is an inherent feature through the P2P communication activities, such as the periodical search for nodes, periodical finding for buddy-node, and periodical detection for firewalls. Although some non-P2P network applications also exhibit such a feature, like software updating per hour or per day, periodic behavior in P2P is more intensive. Consequently, it is possible to identify P2P botnets traffic by mining such periodic patterns.

The P2P bot detection is composed of three major components: time series extraction, regional periodic pattern mining, and evaluation of traffic.

The time series extraction phase, to reduce redundant and noisy packets, only suspicious traffic will be retained and then classified by internal hosts' IP addresses. For each IP's traffic, a time series of traffic features will be extracted. At last, the feature time series will be transformed into an event time series.

In the second phase, for an event time series, the circular autocorrelation function is first used to search for candidate periods. Then, as the most important step in the model, the work needs to recognize regional periodic patterns from current time series based on the candidate periods.

In the final phase, the mined regional patterns will be evaluated by an evaluation function.

The work contains the following datasets: The first dataset was obtained from local university. It is a combination of several publicly available malicious and non-malicious datasets. The malicious datasets contain malicious traffic involving Storm, Waledac, and Zeus botnets. The non-malicious datasets are composed of a variety of network activities spanning Web and email to software backup and streaming media.

The characteristics of the work are: Experimental evaluation based on public datasets. The time consumption in machine learning is only 16 s for using the RBI algorithm in this work. The detection based on the inherent features of P2P applications in this work has the advantage of directly detecting the P2P bots in the C&C phase.

### 3.18 Detecting stealthy P2P botnets using statistical traffic fingerprints

The paper in [24] presented a botnet detection system that is able to identify stealthy P2P botnets, even when malicious activities may not be observable. First, the system identifies all hosts that are likely engaged in P2P communications. Then, it derives statistical fingerprints to profile different types of P2P traffic, and the system weights these fingerprints to distinguish between P2P botnet traffic and other legitimate P2P traffic.

The work uses a new flow-clustering-based analysis approach to identify hosts that are most likely running P2P applications and estimate the active time of the detected P2P nodes. then it uses a proposal algorithm for P2P traffic fingerprinting, which is used to build a statistical profile of different P2P applications. In addition, the system is able to identify bot-compromised machines, even in the case in which the P2P botnet traffic is overlapped with traffic generated by legitimate P2P applications (e.g., Skype) running on the same compromised machine.

The system is divided into two phases. The first phase aims to detect all hosts within the monitored network that appear to be engaged in P2P communications. It analyzes the raw traffic collected at the edge of the network and

applies a pre-filtering step to only consider network flows that are related to P2P communications. Then, the system analysis the remaining traffic and extracts a number of statistical features, which are used to determine p2p flows, and are identified as candidate P2P clients. In the second phase, the botnet detection system analysis the traffic generated by the candidate P2P clients and classifies them into either legitimate P2P clients or P2P bots. The architecture of the botnet detection system is based on the behavior of bots. The bots are malicious programs used to perform profitable malicious activities. They represent valuable assets for the botmaster, who will intuitively try to maximize their utilization.

The popular P2P applications are run in order to increase the number and diversity of P2P nodes in the network. P2P applications in two different (virtual) hosts for several hours (e.g., 24 or 5 hours) simultaneously run each of them - Each host was represented by a WindowsXP (virtual) machine with a public IP address selected within a /24 network. The dataset contains two popular P2P botnets, Storm, and Waledac. Both botnets trace in the controlled environment and record their network behavior. The Storm contains 13 different bot-compromised hosts while the Waledac contains 3 different bot-compromised hosts.

The experimental results of the work show that the detection rate is100% and false positive rate is 0.2%.

### 3.19 Adoption of a Fuzzy-Based Classification Model for P2P Botnet Detection

Paper [17] explains that the numeric flow feature values of P2P botnet C & C traffic can be used to generate fuzzy rule-set which can then be used to develop an efficient fuzzy based classification model. A fuzzy rule-based models are generated using Fuzzy Unordered Rule Induction Algorithm (FURIA) [17] on flow attribute from C & C traffic collected from Nugache, Zeus and Waledac botnets. A flow is defined by <source IP, destination IP, protocol, source port, destination port>. Fuzzy logic often leads to the creation of small rule, where each rule is an embodiment of meaningful information.

The work in [17] is a novel approach for detecting P2P botnet traffic flows through identification of significant flow-level features of P2P botnet traffic. Flow level features are basically an aggregation of packet-level features in that flow. The core of the detection approach relies on the identification of likely botnet traffic flows through the development of efficient machine learning based models and then correlating the marked botnet flows to identify the group of flows that belong to the same botnet.

The architectural of the model contains two compounds:

1. The first component has a module to extract flows from raw data. Then the attributes in the flows are scaled and useless flows are deleted. The final task of this component is to label the retained flows.
2. The second component is taken as an input to the dataset containing refined flows prepared by the first component and generates fuzzy rules for classification.

The work performs botnet C & C traffic classification using 10-fold cross validation. In general, in n-fold cross validation, the training set is the first divided into n subsets of equal size. Sequentially one subset is tested using classifier trained on remaining n-1 subsets. Finally, when all subsets are tested, n results from folds are averaged to produce a single estimation. Detail analysis of the behavioral characteristic of botnet C & C traffic flow was conducted after which, useful features for classification were extracted from packet headers. the work was used 10 features.

The percentage accuracy value achieved using FURIA are about 99.5%. while when using C4.5 algorithm are about 99.3%.

The work prepared three datasets having 20,000 flows each, each one for flow extracted for Nugache, Zeus and Waledac traces. Datasets are prepared in such a way, that each has 15000 flows of botnet C & C traffic and 5000 flows of benign traffic. The benign traffic samples include various traffic such as HTTP, FTP, SMTP etc. the dataset also includes traffic captured from legitimate P2P application.

### 3.20 PeerDigger: Digging Stealthy P2P Hosts through Traffic Analysis in Real-Time

In paper [25], PeerDigger is proposed, a novel real-time system capable of detecting stealthy P2P bots. PeerDigger first detects all P2P hosts based on several basic properties of flow records, and then distinguishes P2P bots from benign P2P hosts by analyzing their network behavior patterns.

PeerDigger detects P2P bots in a two-step scheme. In the first step, based on several basic properties of flow records, all hosts that one engaged in P2P communications are identified. In the second step, PeerDigger differentiates P2P bots from other benign P2P hosts by analyzing their network behavior patterns.

PeerDigger is featured by four characteristics.

First, PeerDigger focuses on the behavioral characteristics of the network traffic with no access to the payload of individual packets, and, therefore, can be unaffected by encryption or obfuscation of payload contents.

Second, PeerDigger is able to detect P2P bots even if their malicious activities are stealthy and non-observable.

Third, PeerDigger does not employ any complicated statistical features or any additional sophisticated algorithms such as machine learning. In addition, the system does not need any training phase, so none labeled P2P botnet traffic trace is needed which is hard to obtain. Thus, PeerDigger is capable of real-time processing because of this well cost effective scheme and its short detection time windows.

Finally, most previous approaches based on traffic analysis would fail if there were only one single bot of a botnet within a network perimeter, or when a bot coexists with a benign P2P application on the same host. However, PeerDigger performs well in these scenarios and shows good flexibility.

The PeerDigger contains a two-phase system that consists of a P2P host detection phase and a P2P bot identification phase.

PeerDigger consists of three types of datasets: Dataset of non-P2P traffic, it is collected the from non-P2P traffic, the first monitoring the traffic of a subnet in the local network and captured all packets crossing the gateway router for a whole day. Overall, 35 active hosts are observed. A dataset of P2P traffic, it is collected in a fully controlled environment. Three of the most popular P2P applications are chosen. A dataset of P2P botnet traffic, it is also obtained from third parties. which includes a 24-hour trace of Storm which contains traffic from 13 bots, and a 24-hour trace of Waledac which contains 3 bots.

The experimental results demonstrate that the system is able to identify P2P bots with an average TPR of 98.07% and an average FPR of 1.5% within 4 minutes.

## 5. CONCLUSIONS AND FUTURE WORK

The most relevant papers and surveys in P2P-based botnet detection were reviewed. The survey and papers characteristics were compared to understand the issues in this area. No complete and comprehensive surveys on P2P botnet detection methods were carried out so far. This work compares the papers according to two dimensions: (1) the type of detection algorithms and (2) The categories of detection techniques.

Our research has the limitation that some papers did not publish the details of their algorithms and techniques.

Our findings suggest that the most relevant issues in the previous surveys are the undefined terminology, focused on different aspects of botnets, limited analysis of the papers, and the small number of papers covered. Meanwhile, the problems in the area that require attention are the use of unverified captures, lack of public datasets, small amount of botnets in the datasets, inaccurate outcomes of experiments and lack of comparison with other proposals.

The causes for these problems include the fast evolution of botnets and the difficulties in obtaining real and working botnet traffic.

In future, work will summarize the survey dimensions, and compare the approaches from a different point of view such as comparison of accuracy-based performance metrics, dataset comparison, comparison of statistical features and comparison of the detection sources.

## References

[1] Lemos, R., Bot software looks to improve peerage. Http://www.securityfocus.com/news/11390, 2006.
[2] Grizzard, J.B., et al. Peer-to-peer botnets: Overview and case study. in Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets. 2007.
[3] T. Holz, M.S., F. Dahl, E. Biersack, and F. Freiling, Measurements and mitigation of peer-to-peer-based botnets: A case study on storm worm. In Proceedings of the First USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET'08), 2008.
[4] vadivu, P.S. and K.S.Karthika, A Survey On Botnet Detection Approaches In Peer-To-Peer Network. International Journal of Advances in Computer Science and Technology, 2014. 3(5): p. 311-317.
[5] Elhalabi, M.J., et al., A Review of Peer-To-Peer Botnet Detection Techniques. Journal of Computer Science, 2013. 10(1): p. 169.
[6] Han, K.-S. and E.G. Im, A Survey on P2P Botnet Detection. Proceedings of the International Conference on IT Convergence and Security 2011, 2011.
[7] Ghalebandi, S.G., R.B.M. Noor, and A.H. Lashkari. A Survey on P2P Botnets Detection. in International Conference on Computer Engineering and Technology, 3rd (ICCET 2011). 2011: ASME Press.
[8] Narang, P., et al. Peershark: detecting peer-to-peer botnets by tracking conversations. in Security and Privacy Workshops (SPW), 2014 IEEE. 2014: IEEE.
[9] Narang, P., V. Khurana, and C. Hota. Machine-learning approaches for P2P botnet detection using signal-processing techniques. in Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems. 2014: ACM.
[10] Kheir, N., X. Han, and C. Wolley, Behavioral fine-grained detection and classification of P2P bots. Journal of Computer Virology and Hacking Techniques, 2014: p. 1-17.
[11] Kheir, N. and C. Wolley, BotSuer: Suing stealthy P2P bots in network traffic through netflow analysis, in Cryptology and Network Security. 2013, Springer. p. 162-178.
[12] Hang, H., et al. Entelecheia: Detecting p2p botnets in their waiting stage. in IFIP Networking Conference, 2013. 2013: IEEE.
[13] Zhang, J., et al., Building a scalable system for stealthy p2p-botnet detection. 2014.
[14] Narang, P., et al., PeerShark: Detecting Peer-to-Peer Botnets by Tracking Conversations. 2014.
[15] Narang, P., C. Hota, and V. Venkatakrishnan, PeerShark: flow-clustering and conversation-generation for malicious peer-to-peer traffic identification. EURASIP Journal on Information Security, 2014. 2014(1): p. 1-12.
[16] Qiao, Y., et al., Detecting P2P bots by mining the regional periodicity. Journal of Zhejiang University SCIENCE C, 2013. 14(9): p. 682-700.

[17] Barthakur, P., M. Dahal, and M.K. Ghose, Adoption of a Fuzzy Based Classification Model for P2P Botnet Detection. 2015.
[18] Rodríguez-Gómez, R.A., et al., Resource monitoring for the detection of parasite P2P botnets. Computer Networks, 2014. 70: p. 302-311.
[19] Yin, C., et al., Towards Accurate Node-based Detection of P2P Botnets. The Scientific World Journal, 2014. 2014.
[20] Narang, P., A. Thakur, and C. Hota. Hades: a Hadoop-based framework for detection of peer-to-peer botnets. in Proceedings of the 20th International Conference on Management of Data. 2014: Computer Society of India.
[21] Fan, Y. and N. Xu, A P2P Botnet Detection Method Used On-line Monitoring and Off-line Detection. International Journal of Security & Its Applications. , 2014. 8(3): p. 87-96.
[22] Dillon, C., Peer-to-Peer Botnet Detection Using NetFlow. 2014.
[23] Kheir, N. and X. Han, Peerviewer: Behavioral tracking and classification of P2P malware, in Cyberspace Safety and Security. 2013, Springer. p. 282-298.
[24] Zhang, J., et al. Detecting stealthy P2P botnets using statistical traffic fingerprints. in Dependable Systems & Networks (DSN), 2011 IEEE/IFIP 41st International Conference on. 2011: IEEE.
[25] He, J., et al. PeerDigger: Digging Stealthy P2P Hosts through Traffic Analysis in Real-Time. in Computational Science and Engineering (CSE), 2014 IEEE 17th International Conference on. 2014: IEEE.

**AUTHOR**

**Atef Ahmed Obeidat** received the B.S. degree in Computer science from Yarmuk University in 1991 and M.S. degrees in Computer science from Jordanian University in 2001. But the PhD degree in communication and network systems. He received from Novosibirsk State Technical University in 2009.