# Analysis of Network Traffic Flows for Centralized Botnet Detection

Pedram Amini[1], Reza Azmi[2], and Muhammad Amin Araghizadeh[3]

*[1]Academic Complex of Information, Communications and Security Technologies, Malek-e-Ashtar University of Technology, Tehran, Iran*

*[2]Faculty of Engineering, Al-Zahra University, Tehran, Iran*

*[3]Department of Electrical and Computer Engineering, University of Tehran, Tehran, Iran*

*amini@mut.ac.ir*

***Abstract*— At present, the Internet users are facing the most serious threats considering the malwares have become a powerful tool for attackers. Botnets are one of the most significant malwares. A Bot is an intelligent program run by worms, Trojans or other malicious codes that could perform a group of cyber-attacks on the Internet. Botnets are used for attacks such as stealing data, spam, denial-of-service, phishing etc. A variety of methods and algorithms have been proposed to detect botnets, in which each of them has an emphasis on specific data or methods. Using Netflow data is an effective and agile method compared to other methods in detecting botnets. This research focuses on centralized and HTTP botnets. In the proposed method, we used the hierarchical clustering, X-Means clustering, and rule-based classification. The methods helped to achieve fast and accurate recognition. Hierarchical clustering improved the speed and accuracy rate in the process of separating the flows. The X-Means algorithm led to the highest cohesion inside the clusters and the maximum distance between clusters by choosing optimal K. Using rule-based classification, each cluster with the similar flow is placed in a bot cluster, a semi-bot cluster or a normal cluster. By performing network traffic flow analysis for the proposed method, sets of botnets have been evaluated and the results indicated that more than 95% accuracy in detection. By a minimum overhead, this approach can provide botnet detection with high accuracy and speed.**

***Index Terms*— Botnet Detection; Centralized Botnet; Data Clustering; Netflow Protocol; Rule-Based Classification.**

## I. INTRODUCTION

Nowadays, information and communication technology is presented as a new approach, which is different from the old methods to process and exchange data. Information and communication technology refers to the study or trade of all technologies used or developed in information processing and communication improvement. One of the requirements for the success and progress in information and communication technology is the security issue. The latest research from the CenturyLink Threat Research Lab has shown that there were roughly 195,000 threats every day, affecting 104 million unique targets daily in 2017 [1]. The greatest and most important security threats that disrupt the success of information and communication technology is the malware. Kaspersky is claimed to detect more than 315,000 new malware files every day [2]. Among the different types of malwares, botnets are recognized as the newest Internet threats used in designing attacks to steal information in comparison to Distributed Denial-of-Serce and spam [3]. The term bot is taken from the word robot: Bot is an intelligent program run by worms or other malicious codes that could perform a group of cyber-attacks on the Internet. In some texts, bots are also known as the Zombies [4]. A group of bots connected to each other forms a botnet that performs malicious activities under a human remote controller, called Botmaster [5].

The concept of botnet was introduced in 1993 with the detection of the Eggdrop botnet activity [6]. There are various research challenges in botnets detection focusing on aspects such as real-time detection of attack type, deep analysis of network traffic, improvement of detection accuracy, improvement of machine learning techniques, behavior analysis-based techniques, botnet detection frameworks, fast-flux techniques for anomalous communications and many others. [7, 8].

In the last two decades, various mechanisms have been proposed to detect botnets. Each mechanism has its advantages and disadvantages. One of these mechanisms is the use of Netflow protocol data. Bot detection using Netflow protocol data has advantages, such as low data volume, easy processing, low false positive, and being online compared to other approaches. However, these advantages are derived mainly from the high level definition presented by Netflow data from Internet connections rather than the analysis of the transmitted real data sets [9]. In short, Netflow data is a technique suitable for analyzing large datasets, high true positive detection rates, and low false positive rates [10].

The perspectives of using Netflow protocol data for botnets detection have disadvantages too. The first challenge is that the majority of these approaches tend to focus on the use of high volume Netflow features, although they may have any one of the following goals for the design and use of the Netflow protocol, which are the monitoring of network traffic, troubleshooting of network, and detection of overload factors to the network with minimum data and maximum speed. The second challenge relates to the computational and time complexities as most of the botnets detection approaches propose algorithms with high and complex calculations that increase the computational time, leading to the difficulties to conduct online diagnosis for large data sets. In addition, a common problem in the Netflow architecture is the selection of solution for its three main components: flow exporter, flow collector, and flow analyzer [11]. Relating to the challenges mentioned above, we will suggest several solutions for online and offline botnet analysis modes.

## II. Definitions and Concepts

In this section, we describe the phenomenon of botnets, Netflow protocols and two main concepts, namely the hierarchical clustering and K-Means clustering.

### A. *The Phenomenon of Botnets*

Threats cause the security of computer networks to be compromised. Malware is the most common threat that could compromise the systems. The malware is a key tool to commit digital crime in modern society. Botnets are one of the most important malwares. Bot is derived from the term robot that is sometimes called zombies. The concept of botnet was introduced in 1993 with the detection of the Eggdrop botnet activity. Botnets are sets of smart and connected software that are run by worms, Trojans or other malicious codes to perform a group of cyber-attacks on their network. First, the botnets infect the computers with their malicious codes and then they use this vulnerability to allow exploiting a remote agent. In fact, botnets are networks of infected machines that act under a remote command, called Botmaster [12].

The main difference between botnets and other malwares is the existence of the factor, command and control structure (C&C). Botmasters attempt to make the botnets difficult to be detected using mechanisms and technologies. Encryptions, malicious code obfuscation, Fast-Flux, and Domain-Flux are among the many methods that make botnets difficult to be detected.

### B. *Netflow Protocol*

Netflow protocol is a network protocol that is responsible for traffic analysis. This protocol will store information about the nature of the traffic; in fact, it stores information about whom, when, and how the traffic is used. In the past, monitoring network traffic has been done by the SNMP protocol. Regarding the shortcomings of this protocol and the new requirements, the new Netflow protocol was designed to collect IP layer traffic data and cover the shortcomings of the SNMP protocol [13].

Netflow plays a vital role in troubleshooting the network, improving the performance, and the availability of users. After activating Netflow protocol on router or switch interfaces, monitoring traffic information begins. After the end of each flow, the data of each flow is set on the port as UDP protocol and is sent to Netflow. Each flow is considered as a unidirectional path of network packets between source and destination [14]. In general, seven main data are stored for each flow: input interface, source IP, destination IP, source port, destination port, IP layer protocol type, and service type.

Netflow is installed by default on Cisco routers and whatever other routers; thus, there is nothing new to install on enterprise networks. This is the big selling point of Netflow. Versions 5 and 9 of Netflow protocol are more common than the other versions. Before being sent to the collector, each flow is stored in the cache until one of the following events happen:

1) Termination of a TCP flow with FIN or RST flags;
2) The cache of flows is full;
3) The connection is inactive for a certain period of time;
4) The connection is active for a certain period of time.

### C. *Clustering*

Hierarchical clustering is a technique used in grouping data. In this technique, the data points are located in categories and subcategories based on a similarity measure. In the hierarchical clustering method, the hierarchical structure - usually as a tree - is given to the final clusters based on their generality level. The hierarchical clustering technique method is usually based on greedy algorithms and stepwise optimization. The clustering methods are usually divided into two categories based on the hierarchical structure: divisive and agglomerative.

One of the famous clustering methods is the K-Means clustering that is based on the minimum distance of each data from the center of a cluster. In fact, this clustering method makes separate sets, in which each set the data points are close to the center of the cluster. In the K-Means clustering, firstly, K should be defined as the number of clusters. The parameter K represents the number of desired clusters. Usually, the initial cluster centers are chosen randomly from the initial samples. Therefore, clusters obtained in the clustering are not unique because the initial cluster centers in two independent K-Means clustering can be different. In the K-Means algorithm, it is possible to use various distance measures and the quality of a criterion depends on the type of data to be clustered.

## III. Related Work

In this section, we compare the research background based on the detection data, the chosen mechanisms, and the proposed algorithms.

Table 1 shows a comparison based on the advantages of using Netflow protocol data compared to other proposed data in the detection of the botnets.

Table 1
Comparison between Netflow Protocol Data and Other Data of Botnet Detection Mechanisms

| Data | Quantity<br>Conversion from Gaussian and<br>CGS EMU to SI [a] |
|---|---|
| Network Packets | Netflow protocol data have insignificant volume versus the network packets. Also processing speed and processing overhead are improved. |
| Log | Log-based approaches are based on network packet analysis tools and generate logs that slow down the botnet detection process. |
| DNS Data | DNS data are more appropriate to explore Botmaster migration but Netflow protocol data have better detection speed. |
| Honeypot Data | Honeypot data are more appropriate to identify the targets and less appropriate in detecting the internal infected hosts. |

According to Table 1, the benefits of Netflow protocol have caused them to be used as the suggested data to discover the botnet. In the rest of this section, we will discuss the techniques and algorithms.

Today, Netflow is supported by most networking equipment, making it easier for the analysis. By using Netflow data, the volume of memory and processing resources are greatly reduced. Further, it is more efficient than other network management protocols, such as the SNMP. Netflow facilitates the identification of unauthorized traffic. Despite all the advantages of Netflow, there are limitations in the network traffic analysis. In the Networks, where routers and switches do not support Netflow, the Netflow generation imposes a lot of overhead on the network. The payload in the network packet is required to identify some signature-based threats that Netflow cannot provide packet details [15].

Botnets are identified using network traffic, network behavior, statistical approaches, and many others. The reference [16] has compared and introduced the sources, data, methods, and algorithms. Most botnet detection methods using Netflow protocol data apply multiple techniques and algorithms for detection.

One of the ways for detecting botnets is identifying the correlation between the flows. Two flows are correlated if they show similar features. Two flows present similar features if they are produced by similar applications; a flow has led to another flow (causal relationship) or there are a sender and several receivers (such as multicast) [17]. Vertical correlation uses for channel detection and the commands presented by the server to bots and horizontal correlation is to detect botnets based on the crowd behavior pattern in response to the commands [18].

Strayer et al. [17] have proposed a method based on network behavior to detect bots. In this method, properties of each flow is stored and then an algorithm detects data correlations. Bilge et al. [18] proposed a method called Disclosure for distinguishing server channel bots from normal network traffic channel based on three characteristics of Netflow protocol feature, features based on the flow size (the number of bytes transferred in one direction between two final point for each flow), client-based patterns' features (pattern linking the infected clients with malicious servers), and time-based features (linking the infected clients with malicious servers in different time periods).

The use of the flow correlation for detecting botnets based on graph-based features is another approach that resolves some of the limitations of statistical features of flow traffic. Chowdhury et al. [19] have proposed a graph-based botnet detection approach that can detect changing behaviors of bots. Kirubavathi and Anitha [20] discovered statistical correlation in the traffic flows in constant time to build an efficient classification system. They consider the small packet correlation information, which can significantly improve the classification accuracy.

One of the common ways to link the attacker with its botnets is the use of IRC; the infected machines are automatically connected to a specific channel on a public server or private IRC to receive instructions. Each user connected to the IRC server is given a name, called the nickname [21].

Goebel and Holz [21] proposed a method to detect botnet called Rishi. In this method, an object is created for each IRC connection and the data of suspicious connection time, IP address and source host port, IP address and destination server of IRC port, channel and nickname are stored along with an Id. The connection Id is the combination of destination IP and destination port. When a connection to a channel is created, if the object (according to Id) does not exist, it will be created; otherwise, updates will be done. There is an array of objects: When an object is created or updated, it is transferred to the front line and related object is removed from the line by cutting each connection. After extracting the data, analysis is performed and warnings associated with each Id are generated. In anomaly-based detection, DNS traffic or botnet traffic is adapted to identify anomalous network behaviors. When bots are connected to an IRC server, they query a DNS server to obtain the IP address of the IRC Server. The collective query behavior can be adapted to identify IRC-based botnets [22].

One of the most common strategies is clustering the flows based on various algorithms. Francois et al. [23] proposed an architecture called BotTrack that is based on Netflow protocol data and the PageRank algorithm. PageRank algorithm is a linear analysis algorithm used by the Google search engine to give relative importance to any web page. PageRank algorithm determines the score of each page based on the link structure on the Internet. Further references from other pages to a particular page present greater importance of that page. Significant scalability and efficiency of PageRank have made it an ideal candidate for the analysis of link structure in the host to communicate with Large-scale networks. PageRank is used to detect P2P botnets because each bot should communicate with a large number of bots and it should be the communication destination of many bots.

In this architecture, the routers monitor the network traffic and send data to the collectors. The data are then sent to BotTrack to be analyzed. In the first step, interactions between systems (dependency graph) are plotted. This graph is analyzed by the PageRank algorithm to extract the nodes that have many connections. In the third step, suspicious nodes are analyzed based on their role and connections so that the detection is made with higher accuracy. In the end, the reduction techniques are used so that the bots are detected according to the infected nodes that are already detected.

Amini et al. [24] used a hybrid approach based on clustering and correlation. They implemented hierarchical clustering on network event and Netflow and gain similar clusters using correlation. Finally, they label off abnormal behaviors. Hsu et al. [25] have proposed a traffic inspection solution, called Web-based Botnet Detector (WBD). WBD is able to detect suspicious C&C servers of HTTP botnets regardless of whether the botnet. Dollah et al. [26] have proposed to use several learning algorithms, although K-Nearest Neighbor classifier (KNN) is the best among the classification algorithms. Commands are encrypted or hidden in normalWeb pages.

Most methods that have been developed for botnet detection have used the statistical approaches. Karasaridis et al. [27] have proposed a method based on calculations and statistics. This method is used to discover the botnets in Tier 1 ISP network. The proposed methodology is offered based on four levels:

1) Dense factors to detect hosts with suspicious behavior and isolate flow records to/from the hosts;
2) Analysis of current activities to identify candidate control flow and their summarization to conversations;
3) Compression and analysis of candidate control conversations to isolate suspicious controllers and controller ports;
4) Sending reports and warnings.

In each candidate control conversation, all of the information and activities such as the source IP, the destination IP, the destination port, the number of flows, the number of packets, the number of bytes, the timestamps of the first and the last conversation flows, and the link in which the activity has occurred are discovered and saved as a record. Additionally, the detection is performed based on the calculations and botnet specifications.

The influence of botnets on mobile networks and Internet of objects causes specific approaches to be addressed to them. The use of deep autoencoders is recommended to identify IoT botnets [28]. Also, deep learning is counseled for mobile network botnets [29]. Although deep learning is growing rapidly, it will soon propose better solutions for detecting botnets, but it has yet been able to overcome its limitations in the training phase. Deep Network requires huge computing power, very time-consuming, non-interpretable results, and large amounts of data to train.

The main disadvantage of using a signature or classifier based detection method is that these systems are usually not as effective at detecting new, or updated malware due to an inherent assumption of stationary data. Clustering based systems can account for unknown behavior. In these systems, the algorithms attempt to separate different patterns of behavior [30]. Among the presented algorithms that lead to behavioral similarities with botnets in the network, the clustering algorithms are the best option to classify similar behaviors and detect botnets.

## IV. THE PROPOSED APPROACH

In this section, the proposed approach is introduced considering the new ideas and the disadvantages of the previous methods. The components of the proposed approach are described also.

The proposed approach is presented to detect centralized botnet. In this method, the Netflow collects the data generated at the routers and then it is sent to a central location. Routers generate the flows from the network traffic and send them to the database of the proposed system, known as Netflow Collector. Next, filtering is performed. The purpose of the filtering is to reduce the excess flow that can disturb the final results or increase the processing overhead. Then, the clustering process is performed on the remaining flows. Clustering involves two processes: hierarchical clustering to separate unrelated flows and X-Means clustering to identify the similar flows. Rule-based classification places the clusters in one of the classes of bot cluster, normal cluster, and network cluster by analyzing the formed clusters based on the proposed rules. Finally, a report of the conducted evaluation is presented. Figure 1 shows the flow diagram of the proposed method which will be described in detail later.

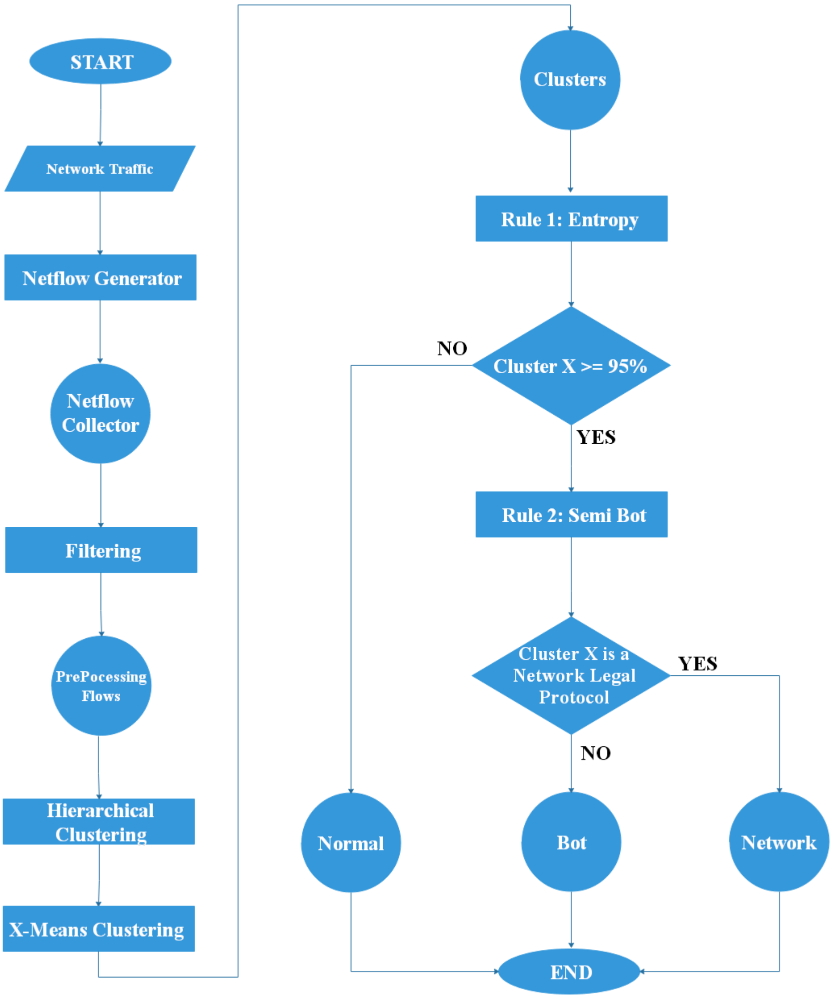


Figure 1: Flow chart of the proposed method

### A. Network Traffic

Network traffic may be available in two ways, online or offline. In the online process, the Internet or internal network traffic must be generated by the routers and sent to the proposed system. In the offline process, the network traffic is collected and stored by the analysis tools of network packets such as Wireshark so that they are converted to Netflow in an independent process.

The storage and processing are done on the Netflows rather than the network traffic because the flow volume is much lesser than the corresponding network traffic.

### *B. Netflow Collector*

Most modern routers have the capacity to produce Netflow and send it to the collector. The most widely used Netflow collectors are Cflowd, Flowd, and flow-tools [31]. However, as mentioned in the previous section that traffic is sent to the proposed system both online and offline; hence, the flow generation should also be conducted both online and offline. In this case, the Cisco routers or OSSIM Alien Vault could be used in the online process and the Argus in the offline process. If the used routers do not have the ability to generate flow, the network traffic will be sent to OSSIM host and the system using Nfdump that generates and stores the flows in real time. Then, the stored flows are evaluated by the proposed system. An alternative approach is to use the flow generation tools, such as the nProbe. In most programming languages, there are libraries to generate Netflow. It is possible to design Netflow generation tools using these libraries. In the offline process, the stored traffic should be converted into the Netflows. The most common tool to monitor the network and store network traffic is the Wireshark. This tool uses Libpcap libraries to manage network traffic. The best tool to convert stored traffic with pcap format to the flows is the Argus. This tool has server and host versions: In the host version, the network traffic is converted into a binary file before it is converted to human-readable output.

The generated flows are stored in the database of the proposed system. Each flow has the following features: unique flow Id (id), number of test samples (ns), time of the flow generation (dt), flow protocol (pr), source IP (si), source port (sp), destination IP (di), destination port (dp), the number of packets sent by the source (ss), the number of bytes sent by the source (sb), the number of packets sent by the destination (ds), the number of bytes sent by the destination (db), and the final status (fs). The equation (1) presents the flow briefly.

$$< id.ns.dt.pr.si.sp.di.dp.ss.sb.ds.db.fs > \quad (1)$$

### *C. Filtering*

In the third step, filtering is performed on the flows. Two types of filters are definable: the basic filter and the condition filter. The basic filter has simple flow properties such as protocol, port or network IP. The condition filters are sets of flow properties that should comply with all conditions so that the filtering could be performed. Broadcast and multicast IPs and ARP and ICMP protocols are the most important features of the basic filters.

### *D. Clustering*

The fourth step is the clustering of flows to identify similar flows. The proposed method is focused on the hierarchical clustering and X-Means clustering. Before describing the clustering process, data preparation is carried out. In the process of preparation, a new definition of each flow is presented. Each flow with 13 properties of Eq. 1 is transferred to 8 properties of Eq. 2.

$$< id.pr.si.sp.di.dp.sbp.dbp > \quad (2)$$

The six basic properties have the definition presented in Eq. 2. The sbp property is the byte to the sent packet ratio and dbp property is the byte to the received packet ratio. Further, the X-Means clustering process is based on these properties. Flow clustering process begins with the hierarchical clustering on the flows, followed by the X-Means, which is is done on each cluster of the hierarchical clustering process. The aim of hierarchical clustering is to separate the unrelated flows that cause the X-Means clustering process to be performed with higher speed and accuracy.

In the proposed method, the hierarchical clustering process is performed at three levels, in which the first algorithm presents its pseudo-code:

- Protocol-based clustering: At the first level, the flows are classified based on the protocol; all of the tcp, udp, and icmp flows are in the corresponding clusters;
- Source IP-based clustering: At the second level, each cluster of the first level creates a new cluster based on the number of source IP;
- Destination IP-based clustering: At the third level, each cluster of the second level is converted to a new cluster based on the number of the destination IP.

The purpose of the proposed hierarchical clustering algorithm is to separate the unrelated data to increase the speed and accuracy of cluster classification and diagnosis. The clustering algorithm is based on protocol, source, and destination IP of flows. To evaluate the proposed algorithm, a set of valid datasets is used. The assessment is based on datasets used by reference [32]. Algorithm 1 is the pseudo-code of hierarchical algorithm.

Algorithm 1: Hierarchical Clustering

```
Function Hierarchical(Flows)
{
    ProtocolList = select all protocols from Flows
    foreach any_protocol in ProtocolList do
        Level1Result = select flows from Flows where protocol=any_protocol
        SourceList = select all source IP from Level1Result
        foreach any_source in SourceList do
            Level2Result = select flows from Level1Result where sourceIP=any_source
            DestinationList = select all destination IP from Level2Result
            foreach any_destination in DestinationList do
                Level3Result = select flows from Level2Result where destinationIP=any_destination
            end foreach
        end foreach
    end foreach
}
```

In this evaluation, the time difference of clustering the flows using X-Means with and without hierarchical clustering algorithm is compared. The results are presented in Table 2.

Table 2
The Time Difference of Flow Clustering Based on X-Means is Compared Between using Hierarchical Clustering Algorithm and Without Using It

| Data | Flows | using algorithm | without using algorithm |
|---|---|---|---|
| Zeus-1 | 26279 | 1.57592 | 15.13605 |
| Zeus-2 | 1638 | 0.08446 | 0.17709 |
| Citadel | 20730 | 0.81879 | 6.95384 |

The clustering time using the proposed hierarchical algorithm is reduced between 2 to 10 times. The difference in reduction is due to the number of flows and hosts of the network. As the number of flows and hosts increases, the proposed hierarchical algorithm reduces more clustering speed. Although the clustering result is not the same, the conversion of a large data set into a smaller set followed by the clustering is faster than clustering a large set of data. However, the main purpose is to separate the unreliable data and increase the classification accuracy.

In the centralized botnets, the command-and-control channel is classified into Push-based and Pull-based categories based on the way the bots receive commands from the Botmaster. In Push-based channels, there is a stable communication between the Botmaster and bot, leading to the bot to immediately response to commands. In pull-based channels, the connection is not stable, causing the Botmaster to put the commands into the server and the bots examine the server to receive new commands. IRC botnets use push-based channels and HTTP botnets use pull-based channels [33].

The centralized botnets send two types of data to the Botmaster: control data and target data. The control data are data that are confirmed as being alive, hence the Botmaster specifies their location on the network. The target data are data such as financial information and identity of the victim. They are computational data sent to the Botmaster by a bot. The basis for the detection of this proposed method is the control data. For similarity, the control data are sent to the Botmaster at a fixed time period. Therefore, the X-Means algorithm presents the data with maximum similarity in the same cluster.

The K-Means clustering has a fundamental problem: In this clustering, it is necessary to determine the number of clusters before starting the process. Some extensions are proposed to solve this problem. The X-Means clustering algorithm repeats the K-Means cycle based on Bayesian information criterion (BIC) to calculate the best K value [34]. The basis for X-Means clustering in the proposed method is the sbp and the dbp properties. The sbp property refers to the source bot IP and the dbp refers to the destination bot IP. After the hierarchical clustering process is performed, the X-Means clustering is done on each cluster (the third level cluster) of the hierarchical clustering process. Algorithm 2 is the pseudo-code of X-Means algorithm. The timing algorithm is stopped when the number of clusters is maximized or the Bayesian information criterion is minimized for all clusters.

Equation (3) presents the Bayesian information measure equation [35]. $L_j$ calculates the log-likelihood of the dataset D. Pj is the function of the number of independent parameters. R is the number of points.

$$BIC(Mj) = L_j(D) - \frac{P_j}{2}\,Log\,R \tag{3}$$

Algorithm 2: X-Means Clustering

```
Function Xmeans(points, Kmin, Kmax)
{
    allClusters = apply k-means to create kmin clusters
    repeat
        foreach cluster in allClusters do
            split the cluster into two clusters by K-means
            evaluate two clusters compare father cluster
            if BICfather > BICchild then
                the two splits are continued
            else
                clusters are no longer divided
            end if
        end foreach
        if at least one evaluation made then
            delete bad quality clusters and keep the best splits
        else
            keep the splits having better improvement evaluated
        end if
        renumber allClusters to become unique
        if allClusters are best evaluated then
            break
        end if
    until clusters are equal to Kmax
}
```

### *E. Rule-Based Classification*

In the previous step, similar and relevant flows are put in the same cluster. In the rule-based classification, the clusters of the previous process are analyzed and based on the defined rules, they are placed in one of the bots: normal or network clusters. Two main rules are involved in this classification; in the first rule, the decision is made based on the time entropy of a cluster flows about flows (not) being a bot. If the cluster entropy is high, it means that in addition to data similarity, they are sent at fixed intervals. In the second rule, the clusters with high entropy are compared with characteristics of conventional network protocols and flows' (not) being a bot is judged. At the end of this step, each cluster receives a bot/nonbot label.

Entropy has different meanings in various scientific fields; basically, entropy is used to detect irregularity among the data. Various equations have been proposed to calculate the entropy. In this method, entropy equation is defined to detect the time order of the cluster flows. First, the time difference between any two consecutive flows of a cluster is determined in seconds. Then, using the equation (4), the time entropy of the flows of a cluster is calculated. The value of this entropy determines the sending time order of clusters [36].

$$X = \{n_1.n_2.\ldots.n_N\}$$

$$S = \sum_{i=1}^{n_c} n_i \tag{4}$$

$$H(X) = \frac{-\sum_{i=1}^{N}(\frac{n_i}{S})Log_2(\frac{n_i}{S})}{Log_2 N} * 100$$

In Equation (4), $n_i$ is the time difference between the flow i and i + 1. The time difference of the consecutive flows in a cluster is determined by the symbol X. The entropy of X presents the similarity of the cluster members. In this equation, S calculates the sum of X members. The function H(X) calculates the amount of entropy for the set X, which is a value between 0 and 100. The higher value presents the greater similarity among the data. The proper value to present the correlation between the current flows is the entropy value is more than 95. Clusters that have the entropy value less than 95% present a normal traffic. Clusters that have the entropy more than 95 percent are compared with a list of common network protocols, such as NetBIOS, DNS, etc. If there is a match with the list, it will be labeled as a cluster network (semi bot) and if there is no match, it is known as a bot. Table 3 shows the characteristics of the semi bot clusters.

It is not possible to classify the network clusters as a bot or normal. They can be located in the bot or the normal clusters based on the type of behavior. Experiences have shown that the infected hosts with semi bot feature have bot flows in most cases and they are detected. Thus, this comparison is done in the final step and the semi bot cluster of the hosts that have been detected as bot are not considered as semi bot in the final report.

Table 3
Characteristics of Semi Bot

| Property | Descriptions |
|---|---|
| Protocol=tcp | Port 139 of TCP protocol is related to printer and file sharing in the internal network of companies. This port is blocked by the firewall in a normal situation but it is left open in the internal networks due to within the enterprise confidence and ease of information exchange. This is usually the first port that hackers are trying to use it. |
| DP=139 | Internet Group Management Protocol (IGMP) that are used in one-to-many network applications such as games. |
| Protocol=igmp | Port 514 of UDP protocol for the exchange of system logs that are responsible to manage system and security analysis. |
| Protocol=udp | Real-Time Transfer Protocol (RTP) is used to transfer audio and video multimedia packets on IP network. |
| DP=514 | Netbios Datagram service uses this port. Also botnets such as Spybot and Chode have used this port. |
| Protocol=rtp | Netbios Name Service uses this port. Security threats by Spybot, Qaz, Nimda and etc. have been reported on this port. |
| Port=138 | Dropbox server usually uses port for synchronization and exchanges similar packets every 30 seconds. |
| Port=137 | This port belongs to DNS packets that are used by some bots to communicate with the server. |
| Port=17500 | This port belongs to SSH and allows the bots to communicate with the remote server |
| Protocol=UDP | Simple Service Detection Protocol (SSDP), uses port 1900 to analyze network services |

### F. *Report*

In the final phase, a report of the clusters and flows are presented. The report includes the number of flows, clusters, cluster status, report of infected hosts, and Botmaster detection. This report is a feedback from the network activities and serves to update the knowledge about botnets.

## V. Evaluation

In this section, the proposed method is evaluated, followed by a comparison analysis of the proposed method with the other methods. First, the proposed method is evaluated by a known network. Then, based on some reliable data sets, the proposed method is evaluated. Finally, the accuracy of proposed method will be calculated. The evaluations are made on a system with the following specifications: operating system 64-bit Windows 8.1, 8 GB of main memory, Intel Core i7-4702MQ CPU 2.2 GH processor and graphics card NVIDIA GeForce GT 740M. Implementation is based on C# programming language in Visual Studio 2010.

To evaluate the proposed method, a Zeus botnet network was created and the traffic was stored inside the network. Zeus botnet is a network of centralized bots controlled by Botmaster used for banking information theft. In this network, there was a computer infected with Zeus bot, a computer as Botmaster, and two non-infected computers. All network traffic were stored by Wireshark and then using Argus tools were used to convert them to network flows. The tools were designed based on the proposed method to store the information in the database. The designed tools, filtering, clustering, and classification were performed and an evaluation of all selected flows was conducted. Finally, a report on the state of network traffic has been generated.

### A. *The Initial Evaluation*

This network was designed virtually in VMware virtual machine simulator. In the designed network, the Botmaster operating system is Ubuntu and the other computers' operating system is Windows XP. The firewall was disabled to avoid the packet filtering by the operating system and no other security tools were installed. Figure 2 shows the details of the designed network.

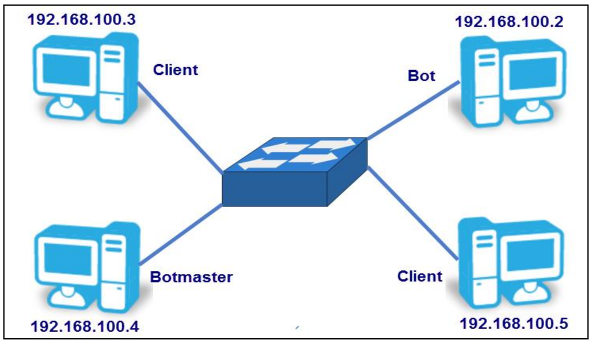


Figure 2: The topology of the designed network

Within four hours, the total number of packets taken from the network was 3088 that was converted to 466 flows using Argus. Zeus botnet briefly called ZBot collected the username, password, banking information, and other sensitive data using the technique of injection into the browser and sent them again in specific time periods or when reconnecting to the Botmaster. The evaluated Zeus botnet architecture was centralized. After setting up the network, the traffic storage and flow generation were assessed. In this assessment, filtering was based on network IP. By filtering the broadcast IPs, a considerable amount of flows was ignored. In this way, 27 clusters and 271 flows were ignored at the filtering stage. This filtering increased the accuracy and speed of detection. After filtering, hierarchical clustering was performed and clusters with less than three members were removed. Therefore, seven clusters remained for X-Means clustering. Then, the X-Means Clustering was carried out on each cluster. The seven clusters were converted to 31 clusters, after running the X-Means clustering algorithm. Clusters with less than three members were removed; thus there were only 14 clusters to be assessed.

Next, the rule-based classification process was performed; the first rule was to calculate the time difference entropy for each cluster created from the previous step. Based on the performed evaluation, six clusters have less than 95 percent entropy and they were located in a normal cluster basket. By comparing the remained clusters with Table 2, five clusters were located in network clusters basket and three clusters were considered as the bot cluster.

In Figure 3, three bot clusters are presented. The first column shows the date and time of the flow generation. As it is presented in the Figure, there is a significant time difference between any two consecutive flows. The other columns are the source IP, source port, destination IP, destination port, the byte to source packet ratio, and byte to destination packet ratio, respectively. Similar byte to source and destination packet ratios verify the idea of the existence of the control channel. Since Zeus botnet uses HTTP protocol, its destination port is the same.

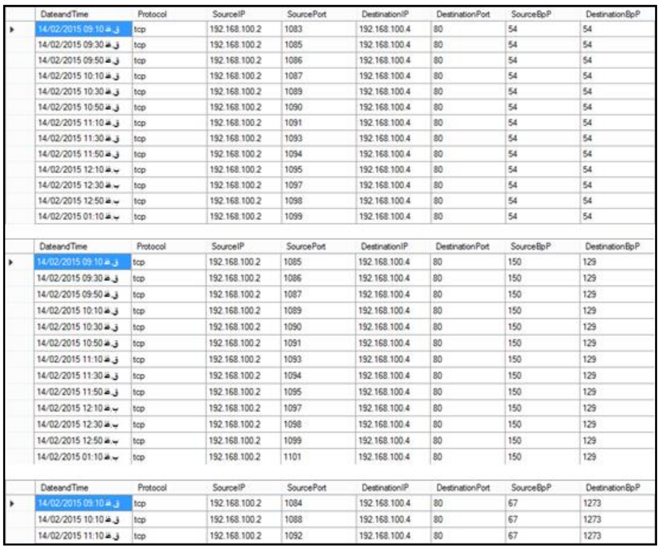
The obtained clusters indicated that each Zeus bot has two time periods to communicate with the server: the time period to confirm the being alive and the second time period to get the settings. In the period to inform being alive, the bot announced that the server is active and ready. In the time period to get the settings, the bot requested and received the latest settings from the server. The existence of similar and regular packets in Figure 3 confirms the proposed idea.

In the following, the proposed approach was compared with the other approaches and the proposed approach was evaluated in terms of recognition accuracy. In this assessment, the datasets of other sources were analyzed by the proposed method and the results were compared with the results of the reference. First, some famous data sets were evaluated and then the detection accuracy was calculated based on three data sets.

### B. Datasets

In reference [32], two Zeus and Citadel botnets were evaluated. Zeus is a well-known botnet in the banking information theft. Citadel is an improvement of Zeus that has resolved the problems and weaknesses. In a report published in 2013, the botnets have presented the most malicious activity in e-banking [37].

Ref. [38] is known as a source of malware traffic that has been cited by many authentic references. One of these datasets is the traffic stored from SDBot botnets; this botnet uses IRC protocol to exchange the control data with the server. This botnet is connected to the bot server through TCP protocol and it continuously announces the clusters that are alive and waits to receive the commands.



| | DateandTime | Protocol | SourceIP | SourcePort | DestinationIP | DestinationPort | SourceBpP | DestinationBpP |
|---|---|---|---|---|---|---|---|---|
| ▸ | 14/02/2015 09:10 ق.ظ | tcp | 192.168.100.2 | 1083 | 192.168.100.4 | 80 | 54 | 54 |
| | 14/02/2015 09:30 ق.ظ | tcp | 192.168.100.2 | 1085 | 192.168.100.4 | 80 | 54 | 54 |
| | 14/02/2015 09:50 ق.ظ | tcp | 192.168.100.2 | 1086 | 192.168.100.4 | 80 | 54 | 54 |
| | 14/02/2015 10:10 ق.ظ | tcp | 192.168.100.2 | 1087 | 192.168.100.4 | 80 | 54 | 54 |
| | 14/02/2015 10:30 ق.ظ | tcp | 192.168.100.2 | 1089 | 192.168.100.4 | 80 | 54 | 54 |
| | 14/02/2015 10:50 ق.ظ | tcp | 192.168.100.2 | 1090 | 192.168.100.4 | 80 | 54 | 54 |
| | 14/02/2015 11:10 ق.ظ | tcp | 192.168.100.2 | 1091 | 192.168.100.4 | 80 | 54 | 54 |
| | 14/02/2015 11:30 ق.ظ | tcp | 192.168.100.2 | 1093 | 192.168.100.4 | 80 | 54 | 54 |
| | 14/02/2015 11:50 ق.ظ | tcp | 192.168.100.2 | 1094 | 192.168.100.4 | 80 | 54 | 54 |
| | 14/02/2015 12:10 ب.ظ | tcp | 192.168.100.2 | 1095 | 192.168.100.4 | 80 | 54 | 54 |
| | 14/02/2015 12:30 ب.ظ | tcp | 192.168.100.2 | 1097 | 192.168.100.4 | 80 | 54 | 54 |
| | 14/02/2015 12:50 ب.ظ | tcp | 192.168.100.2 | 1098 | 192.168.100.4 | 80 | 54 | 54 |
| | 14/02/2015 01:10 ب.ظ | tcp | 192.168.100.2 | 1099 | 192.168.100.4 | 80 | 54 | 54 |

| | DateandTime | Protocol | SourceIP | SourcePort | DestinationIP | DestinationPort | SourceBpP | DestinationBpP |
|---|---|---|---|---|---|---|---|---|
| ▸ | 14/02/2015 09:10 ق.ظ | tcp | 192.168.100.2 | 1085 | 192.168.100.4 | 80 | 150 | 129 |
| | 14/02/2015 09:30 ق.ظ | tcp | 192.168.100.2 | 1086 | 192.168.100.4 | 80 | 150 | 129 |
| | 14/02/2015 09:50 ق.ظ | tcp | 192.168.100.2 | 1087 | 192.168.100.4 | 80 | 150 | 129 |
| | 14/02/2015 10:10 ق.ظ | tcp | 192.168.100.2 | 1089 | 192.168.100.4 | 80 | 150 | 129 |
| | 14/02/2015 10:30 ق.ظ | tcp | 192.168.100.2 | 1090 | 192.168.100.4 | 80 | 150 | 129 |
| | 14/02/2015 10:50 ق.ظ | tcp | 192.168.100.2 | 1091 | 192.168.100.4 | 80 | 150 | 129 |
| | 14/02/2015 11:10 ق.ظ | tcp | 192.168.100.2 | 1093 | 192.168.100.4 | 80 | 150 | 129 |
| | 14/02/2015 11:30 ق.ظ | tcp | 192.168.100.2 | 1094 | 192.168.100.4 | 80 | 150 | 129 |
| | 14/02/2015 11:50 ق.ظ | tcp | 192.168.100.2 | 1095 | 192.168.100.4 | 80 | 150 | 129 |
| | 14/02/2015 12:10 ب.ظ | tcp | 192.168.100.2 | 1097 | 192.168.100.4 | 80 | 150 | 129 |
| | 14/02/2015 12:30 ب.ظ | tcp | 192.168.100.2 | 1098 | 192.168.100.4 | 80 | 150 | 129 |
| | 14/02/2015 12:50 ب.ظ | tcp | 192.168.100.2 | 1099 | 192.168.100.4 | 80 | 150 | 129 |
| | 14/02/2015 01:10 ب.ظ | tcp | 192.168.100.2 | 1101 | 192.168.100.4 | 80 | 150 | 129 |

| | DateandTime | Protocol | SourceIP | SourcePort | DestinationIP | DestinationPort | SourceBpP | DestinationBpP |
|---|---|---|---|---|---|---|---|---|
| ▸ | 14/02/2015 09:10 ق.ظ | tcp | 192.168.100.2 | 1084 | 192.168.100.4 | 80 | 67 | 1273 |
| | 14/02/2015 10:10 ق.ظ | tcp | 192.168.100.2 | 1088 | 192.168.100.4 | 80 | 67 | 1273 |
| | 14/02/2015 11:10 ق.ظ | tcp | 192.168.100.2 | 1092 | 192.168.100.4 | 80 | 67 | 1273 |

Figure 3: Bot clusters detected

A specialized reference for botnet datasets is reference [39]. This dataset includes three types of traffic: malware, background, and normal which are used in articles, such as [40]. To evaluate the proposed algorithm, five datasets of Virut, Agobot, Rbot, Zeus, and njRAT botnets were selected. At the end of this part, the proposed idea was assessed based on the evaluation criteria to determine its accuracy and efficiency. First, the following concepts are presented [41]:

- True positive (TP): The number of flows that are clean and the algorithm has detected them as clean properly.
- False positive (FP): The number of flows that are infected and the algorithm has detected them as clean falsely.
- True negative (TN) The number of flows that are infected and the algorithm has detected them as infected properly.
- False negative (FN) The number of flows that are clean and the algorithm has detected them as infected falsely.

The overall accuracy of the proposed idea, which is called accuracy rate is calculated by Equation (5).

$$AR = \frac{TP + TN}{TP + FP + TN + FN} \tag{5}$$

The error rate (ER) of the algorithm is calculated by equation (6).

$$ER = \frac{FP + FN}{TP + FP + TN + FN} = 1 - AR \tag{6}$$

To evaluate the accuracy of detection, some data sets were selected. The first assessment (Dataset1) was conducted on a data set presented in [42]. The second assessment (Dataset2) was conducted on a Zeus data set, presented in [32]. The third assessment (Dataset3) was conducted on a data set presented in [39] and analyzed in [43].

### C. Datasets Evaluation

The mentioned datasets listed in the previous section were evaluated and the results are shown in Table 4. The information in this table includes the number of flows, the total number of hosts of the source flow, the number of bot hosts, the number of semi bot hosts, the number of normal hosts, and time to analyze the dataset. The detection time is the sum of clustering and classification time and the duration of information retrieval and filtering were ignored. Also, the hosts removed in a filtering step were not considered.

Table 4
Details of Evaluated Datasets

| Dataset | Flows | Hosts | Bot | Semi Bot | Normal |
|---|---|---|---|---|---|
| Zeus-1 | 1636 | 19 | 12 | 7 | 0 |
| Citadel | 20728 | 20 | 7 | 4 | 9 |
| SDBot | 36 | 1 | 1 | 0 | 0 |
| RBot | 35579 | 14979 | 12 | 0 | 14967 |
| Virut | 38982 | 16173 | 22 | 0 | 16151 |
| Zeus-2 | 466 | 4 | 1 | 0 | 3 |
| AgoBot | 24140 | 29 | 11 | 1 | 17 |
| njRat | 11463 | 2 | 1 | 0 | 1 |
| Dataset1 | 189443 | 3 | 1 | 0 | 2 |
| Dataset2 | 495 | 1 | 1 | 0 | 0 |

### *D. Evaluation Accuracy*

In the previous sections, we analyzed our network bot and botnets in other networks, and in this section we want to examine how many botnets can be detected with these analysis. To evaluate the accuracy of detection on three data sets, Dataset1, Dataset2 and Dataset3 were selected. Details and results of the dataset evaluation are presented in Table 5. Clustering netflows were considered normal. Results obtained from the designed tools were also compared with the other proposed methods. In this evaluation, Statefull-Sbb [32], CCDetector, and BotnetDetectorComparer [41] tools were used. Statefull-Sbb tool is based on C ++ and includes two learning and testing phases. The results of evaluating this tool on dataset2 were 98%. The results of the CCDetector and BotnetDetectorComparer tools on dataset2 and dataset3 indicated 98% of detection.

Table 5
Details and Results of Evaluated Datasets

| Dataset | Packets | Flows | Bot Flows | Correct Bot | Normal | Correct Normal |
|---|---|---|---|---|---|---|
| Dataset1 | 198818 | 56512 | 6726 | 8495 | 48886 | 48017 |
| Dataset2 | 6868 | 1636 | 232 | 244 | 1404 | 1392 |
| Dataset3 | 1599379 | 189443 | 161753 | 161572 | 27690 | 27871 |

Table 6 indicates the detection accuracy of the proposed method based on Netflows. According to the obtained results and the results reported in the reference of the dataset, Figure 4 presents the comparison of the detection accuracy for these datasets. The blue color presents the detecting percentage of the proposed method (left column) and the red color presents the detecting percentage of the compared reference (right column).

Table 6
Evaluation Accuracy

| Dataset | TP | FP | TN | FN | AR | ER |
|---|---|---|---|---|---|---|
| Dataset1 | 48017 | 1769 | 6726 | 0 | 0.96 | 0.04 |
| Dataset2 | 1392 | 12 | 232 | 0 | 0.99 | 0.01 |
| Dataset3 | 27960 | 0 | 161572 | 181 | 0.99 | 0.01 |

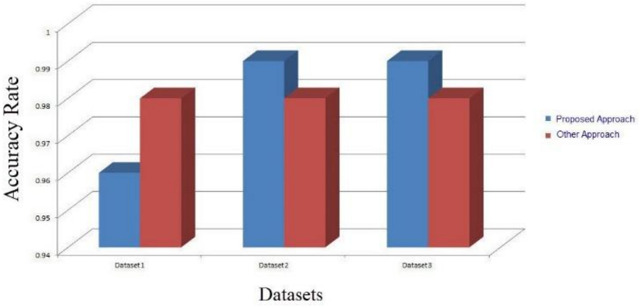


Figure 4: The comparison of the evaluation accuracy of the datasets

### *E. Evaluation of legitimate traffic*

To evaluate the proposed idea, legitimate traffic evaluation was performed on the Alexa dataset [32]. The evaluation details and the results are summarized in Table 7 and 8.

Table 7
Legitimate Traffic

| Dataset | Traffic (KB) | Packet | Flow | Bot (Error Detection) |
|---|---|---|---|---|
| Alexa | 2.186 | 21210 | 5435 | 137 |

Table 8
Evaluation Accuracy

| Dataset | TP | FP | TN | FN | AR | ER |
|---|---|---|---|---|---|---|
| Alexa | 5298 | 0 | 0 | 137 | 0.97 | 0.03 |

## VI. CONCLUSION

The bots installed on the hosts were infected and the bots then informed their status to the Botmasters. They then waited to receive the commands and executed a series of pre-defined automatic functions. Bots were connected to the Botmasters through the channels of command and control. The bots then announced their position in the network and readiness to receive commands to the Botmasters by sending information of their status. In this method, this channel and the sent data were considered as the weaknesses of the botnets. In the centralized botnets, the command-and-control channel were classified into Push and Pull based classes based on the way that bots receive commands from the Botmaster. The centralized botnets sent two types of data to the Botmaster, namely the control data and the target data.

The hierarchical, X-Means clustering algorithms, and rule–based classification are the approaches selected in this article to discover similar data in a fixed period of time. Hierarchical clustering improved the speed and accuracy rate in detecting botnets. Based on the assessments carried out, the use of hierarchical clustering could improve the speed of clustering (for example 2 to 10 times for the evaluated samples) and separate the irrelevant flows. The proposed X-Means algorithm made the highest intra cluster cohesion and created maximum distance between the clusters by choosing the optimal clustering for different Ks. By implementing the clustering, the similar flows between the source and destination IPs in the form of each specific protocol were determined. Using the rule-based classification, each group with a similar flow was placed in either the bots, semi bot, and normal baskets. Two main rules were defined in this thesis, which are the time difference entropy and the pseudo-bot properties table. Time difference entropy was defined to discover the time order of the cluster flows. Clusters with the entropy with less than 95 percent indicate normal traffic. Clusters with the entropy with more than 95 percent were compared with a list of common network protocol signatures such as NetBIOS, DNS, etc. If there were a match with the list, it would be labeled as a cluster network (semi bot), while if there were no match, it would be known as a bot.

Based on the conducted evaluations, the proposed idea was found to be capable of detecting botnets with more than 95% accuracy. In this article, the innovations in the general architecture and the details, components, and algorithms of the method include:

- New features for hierarchical clustering: The main idea of using hierarchical clustering in the proposed architecture is to separate the non-dependent flows. This separation increases detection accuracy and improves speed. Prior to this, two feature sets were discussed as the clustering features.
  1. Source IP and protocol
  2. Source IP and protocol, Destination IP and protocol

  In the proposed method, the source IP and protocol and the destination IP were used. In addition to the positive aspects of the previous ideas, they provide detection of ZeroAccess and Perlbot identification.
- Time difference entropy: The entropy was used to detect the irregularity among the data. The proposed entropy equation determines the order among the flows and helps to detect bot or normality of the cluster by calculating the entropy of the time difference both in the consecutive flow in a cluster of similar flows. Before this, the entropy was discussed to detect the botnets, but the innovation is in the feature is its usage to detect the order among the flows.

The proposed method increases the speed of network traffic analysis and improves botnet detection accuracy in most cases. Therefore, a suitable approach to detect botnets is centralized.

Today, the advances in computing and communications technology security risks are increasing. Botnets, as the most serious online threats are spread over worldwide networks and work in a distributed manner. Based on the conducted studies, issues to expand the scope of detection by improving the proposed idea should be considered. In this method, each flow cluster is classified into three categories based on two rules. It is possible to increase the accuracy of detection by defining more rules. For future work, instead of classifying the flow clusters using rule-based classification into the bot, normal, and network categories, it is suggested that each cluster includes a percentage of presence in each class based on the fuzzy logic.

## REFERENCES

[1] CenturyLink 2018 Threat Report [Online], https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwi0_8b41ZvhAhVLZ1AKHe27AXEQFjAAegQIAxAC&url=https%3A%2F%2Fwww.centurylink.com%2Fasset%2Fbusiness%2Fenterprise%2Freport%2F2018-threat-research-report.pdf&usg=AOvVaw2vXFoX1ZenSdUMDUOiukir.

[2] Number of the year: Kaspersky Lab is detecting 315,000 new malicious files every day. [Online], http://www.kaspersky.com/about/news/virus/2013/number-of-the-year.

[3] D. Plohmann, E. Gerhards-Padilla, “Case study of the miner botnet,” In Proceedings of the 4th International Conference on Cyber Conflict (CYCON), IEEE, Tallinn, Estonia, 2012, pp. 1-16.

[4] S. N. Prabhu, D. Shanthi, “A survey on anomaly detection of botnet in network,” International Journal of Advance Research in Computer Science and Management Studies, vol. 2, no. 1, 2014, pp. 552-558.

[5] R. Borgaonkar, “An analysis of the asprox botnet,” In Proceedings of the 4th International Conference on Emerging Security Information Systems and Technologies (SECURWARE), IEEE, Venice/Mestre, Italy, 2010, pp. 148-153.

[6] A. Karim, R. B. Salleh, M. Shiraz, S. A. A. Shah, I. Awan, N. B. Anuar, “Botnet detection techniques: review, future trends and issues,” Journal of Zhejiang University-SCIENCE C, vol. 15, no. 11, 2014, pp. 943-983.

[7] A. Bijalwan, V. K. Solanki, E. S. Pilli, “Botnet Forensic: Issues, Challenges and Good Practices,” Network Protocols and Algorithms, vol. 10, no. 2, 2018.

[8] R. S. Rawat, E. S. Pilli, R. C. Joshi, “Survey of Peer-to-Peer Botnets and Detection Frameworks,” International Journal of Network Security, vol. 20, no. 3, 2018, pp. 547-557.

[9] B. Li, J. Springer, G. Bebis, M. H. Gunes, “A survey of network flow application,” Journal of Network and Computer Application, vol. 36, issue 2, 2013, pp. 567-581.

[10] T. S. Hyslip, J. M. Pittman, “A survey of botnet detection techniques by command and control infrastructure,” Journal of Digital Forensics, Security and Law, vol. 10, no. 1, article 2, 2015, pp. 1-21.

[11] D. Zhou, Z. Yan, Y. Fu, and Z. Yao, “A survey on network data collection,” Journal of Network and Computer Applications, vol. 116, 2018, pp. 9-23.

[12] M. A. Rajab, J. Zarfoss, F. Monrose, A. Terzis, “A multifaceted approach to understanding the botnet phenomenon,” In Proceedings of the 6th Conference on Internet Measurement, ACM, Rio de Janeiro, Brazil, 2006, pp. 41-52.

[13] S. Yadav, A. K. K. Reddy, A. N. Reddy, S. Ranjan, “Detecting algorithmically generated domain-flux attacks with DNS traffic analysis,” IEEE/ACM Transactions on Networking, vol. 20, no. 5, 2012, pp. 1663-1677.

[14] G. Vliek, “Detecting spam machines, a Netfow-data based approach,” M.Sc. Dissertation, University of Twente, Netherlands, 2009.

[15] A. L. Buczak, E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection,” IEEE Communications Surveys & Tutorials, vol. 18, no. 2, 2016, pp. 1153-1176.

[16] P. Amini, M. A. Araghizadeh, R. Azmi, “A survey on Botnet: Classification, detection and defense,” In Proceedings of the 17th International Electronics Symposium (IES), IEEE, Surabaya, Indonesia, 2015, pp. 233-238.

[17] W. T. Strayer, D. Lapsely, R. Walsh, C. Livadas, “Botnet detection based on network behavior,” Botnet Detection, Springer US, vol. 36, 2008, pp. 1-24.

[18] L. Bilge, D. Balzarotti, W. Robertson, E. Kirda, C. Kruegel, “Disclosure: detecting botnet command and control servers through large-scale netflow analysis,” In Proceedings of the 28th Annual Computer Security Applications Conference, ACM, Florida, USA, 2012, pp. 129-138.

[19] S. Chowdhury, M. Khanzadeh, R. Akula, F. Zhang, S. Zhang, H. Medal, M. Marufuzzaman, L. Bian, “Botnet detection using graph-based feature clustering,” Journal of Big Data, vol. 4, no. 1, article 14, 2017, pp. 1-23.

[20] G. Kirubavathi, R. Anitha, “Botnet detection via mining of traffic flow characteristics,” Computers and Electrical Engineering, vol. 50, 2016, pp. 91-101.

[21] J. Goebel, T. Holz, “Rishi: Identify bot contaminated hosts by IRC nickname evaluation,” In Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets, California, USA, 2007, pp. 1-12.

[22] C. M. Chen, H. C. Lin, “Detecting botnet by anomalous traffic,” Journal of Information Security and Applications, vol. 21, 2015, pp.42-51.

[23] J. Francois, S. Wang, R. State, T. Engel, “BotTrack: Tracking Botnets using NetFlow and PageRank,” In Proceedings of the 10th International Conference on Research in Networking, Valencia, Spain, 2011, pp. 1-14.

[24] P. Amini, R., Azmi M. A. Araghizadeh, M., “Botnet detection using NetFlow and clustering,” Advances in Computer Science: An International Journal, vol. 3, no.2, 2014, pp.139-149.

[25] F. H. Hsu, C. W. Ou, Y. L. Hwang, Y. C. Chang, P. C. Lin, “Detecting Web-Based Botnets Using Bot Communication Traffic Features,” Security and Communication Networks, Volume 2017, Article ID 5960307, 2017, pp. 1-11.

[26] R. F. M. Dollah, M. A. Faizal, F. Arif, M. Z. Mas’ud, L. K. Xin, L.K., “Machine learning for HTTP botnet detection using classifier algorithms,” Journal of Telecommunication, Electronic and Computer Engineering (JTEC), vol. 10, no. 1-7, 2018, pp.27-30.

[27] A. Karasaridis, B. Rexroad, D. Hoeflin, ‘Wide-scale botnet detection and characterization,” In Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets, California, USA, 2007, pp. 1-8.

[28] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, Y. Elovici, “N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders,” IEEE Pervasive Computing, vol. 17, issue 3, 2018, pp.12-22.

[29] L. F. Maimó, A. H. Celdrán, M. G. Pérez, F. J. G. Clemente, G. M. Pérez, “Dynamic management of a deep learning-based anomaly detection system for 5G networks,” Journal of Ambient Intelligence and Humanized Computing, 2018, pp.1-15.

[30] J. Gardiner, Sh. Nagaraja, “On the security of machine learning in malware C&C detection: a survey,” ACM Computing Surveys, vol. 49, no. 3, article 59, 2016, pp. 1-39.

[31] M. W. Lucas, *Network Flow Analysis*, No Starch Press, San Francisco, USA, 2010.

[32] F. Haddadi, A. N. Zincir-Heywood, “Benchmarking the Effect of Flow Exporters and Protocol Filters on Botnet Traffic Classification,” IEEE Systems Journal, vol. 10, issue 4, 2016, pp. 1390-1401.

[33] G. Gu, J. Zhang, W. Lee, “BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic,” In Proceedings of the 15th Annual Network and Distributed System Security Symposium, California, USA, 2008.

[34] D. Pelleg, A. W. Moore, “X-means: extending K-means with efficient estimation of the number of clusters,” In Proceedings of the 17th International Conference on Machine Learning, California, USA, 2000, pp. 727-734.

[35] N. Hourdakis, “Design and evaluation of clustering approaches for large document collections, the BIC-Means method,” M.Sc. Thesis, Technical University of Crete, Greece, 2016.

[36] G. Nychis, V. Sekar, D. G. Andersen, H. Kim, H. Zhang, “An empirical evaluation of entropy-based traffic anomaly detection,” In Proceedings of the 8th SIGCOMM Conference on Internet Measurement, ACM, Vouliagmeni, Greece, 2008, pp. 151-156.

[37] Top Banking Botnets of 2013 [Online], http://www.secureworks.com/cyber-threat-intelligence/threats/top-banking-botnets-of-2013.

[38] Publicly available PCAP files [Online], http://www.netresec.com/?page=PcapFiles.

[39] Special Dataset CTU [Online], https://stratosphereips.org/category/dataset.html.

[40] S. García, V. Uhlíř, M. Rehak, “Identifying and modeling botnet C&C behaviors,” In Proceedings of the 1st International Workshop on Agents and CyberSecurity. ACM, New York, USA, 2014, pp. 1-15.

[41] S. Garcia, M. Grill, J. Stiborek, A. Zunino, “An empirical comparison of botnet detection methods,” Computers & Security, vol. 45, 2014, pp. 100-123.

[42] S. García, A. Zunino, M. Campo, “Botnet behavior detection using network synchronism,” Privacy, Intrusion Detection and Response: Technologies for Protecting Networks: 2011, pp. 1-23.

[43] Identifying Malware Traffic with Bro and the Collective Intelligence Framework (CIF) [Online], https://blog.opensecurityresearch.com/2014/03/identifying-malware-traffic-with-bro.html.