IBIMA Publishing
Communications of the IBIMA
https://ibimapublishing.com/articles/CIBIMA/2022/156851/
Vol. 2022 (2022), Article ID 156851, 20 pages, ISSN: 1943-7765
DOI: 10.5171/2022.156851

Research Article

BotTROP: Detection of a Botnet-based Threat using Novel Data Mining Algorithm

Hubert OSTAP and Ryszard ANTKIEWICZ

Military University of Technology, Warsaw, Poland

Correspondence should be addressed to: Hubert OSTAP; hubert.ostap@wat.edu.pl

Received date:19 October 2020; Accepted date: 29 August 2021; Published date: 8 February 2022

Abstract

Nowadays botnet-based threat, such as ransomwares, trojans and botnets per se, is still very dangerous for our privacy and data. Depending on their management architecture (centralized, decentralized, hybrid), they could be controlled from single or multi point servers called Command&Control (C2), what makes them very difficult to detect and mitigate before malicious action takes place. The aim of this paper is to present a method of detecting botnets based on the identification of their synchronous actions. Presented method, called BotTROP, utilizes clustering and classification methods to detect synchronous action among corporate network traffic to detect malicious activity such as a botnet of any type. Furthermore, the effectiveness of the presented method was verified in numerous experiments where simulated and real-life network traffic was used.

Keywords: botnet, security, detection, botnet detection, botnet mitigation

Introduction

Nowadays, there are a lot of different types of Internet threats such as DOS (Denial of Service), ransomware, phishing campaigns. Some of them are more dangerous, others are better organized but most of them are profit-oriented which makes them difficult to detect and mitigate. One of the biggest Internet threats are botnets (Plohmann et al. 2011) (Gralla, 2019) and RATs (Harel, 2019). They consist of infected machines controlled from a C2 (Command&Control). Rightful owners don't realize that their devices are infected and used to conduct malicious actions. The number of different types of botnets that have been implemented is truly impressive (Garcia et al. 2014). They use different infection and communication methods. The most popular are based on P2P, HTTP/HTTPS and IRC protocols. They are used for sending and receiving commands from their owner called a botmaster. Another difference between botnets is their architecture. Most of them are centralized with only one C2,

Cite this Article as: Hubert OSTAP and Ryszard ANTKIEWICZ (2022)," BotTROP: Detection of a Botnet-based Threat using Novel Data Mining Algorithm", Communications of the IBIMA, Vol. 2022 (2022), Article ID 156851, DOI: 10.5171/2022.156851

more sophisticated are decentralized with a group of C2 based on P2P protocol (Silva et al. 2013). Another type of botnets is based on hybrid architecture. This specific model is used only by few known botnets where decentralized architecture is used to spread domain addresses of the C2s and centralized is used to receive reverse-tcp connections (Chanda, 2014) (Wang et al. 2010). During the last several years, one has been able to observe an increasing number of new kind of botnets which are based on social networks.

Due to a huge number of described Internet threats, there are also a lot of methods of their detection. Unfortunately, most of them are bound by a specific protocol or architecture and are useful only against specific botnets. Others are effective only when a monitoring agent is installed on every host. This approach is also problematic especially for an organization that possesses a huge number of devices that should be secured.

This paper presents an improved and developed method for botnet detection which is based on previous research (Ostap et. al 2017). Furthermore, it includes a description of its implementation and extensive research on its effectiveness and quality. It is able to detect botnets that use any kind of architecture, protocol or social networks for communication (Ostap et al. 2018) purpose with their C2. Moreover, to mitigate botnet activity it is not necessary to install any monitoring agent on every host in the monitored network. Network traffic is analysed at the interface between the internal and the public network.

The article is organized as follows. Section 2 contains a description of the novel classification approach for botnet detection methods and an overview of some research works in this field. Section 3 contains a detailed description and the main concept of the BotTROP algorithm. In the next section we present a case study including the methodology to reach our objectives, description of the dataset and an interpretation of the results. Section 5 presents performance, evaluation and comparison of some previous solutions. The final section contains a conclusion of our paper and some future work.

An overview of botnet detection methods

Over the years an impressive number of different botnets have been discovered in the wild. It was the root cause of implementing a huge number of defense methods by researchers all over the world. So far, few authors have also proposed a classification of botnet detection methods (Feily et al. 2009) (Nallanthighal et al. 2012) (Strayer et al. 2007) (Garcia et al. 2014) (Silva et al. 2013). The first mentioned publication presented by Maryam Feily, Alireza Shahrestani and Sureswaran Ramadass is based on well-known and clearly defined classification criteria. The second classification contains a wide spectrum of botnet detection methods based on their architecture. Next publication proposed by Tim Strayer, David Lapsely, Robert Walsh and Carl Livadas contains information about detection method based on botnet network behavior. From our perspective, the most valuable classifications are presented by Garcia et al. (2014) and Silva et al. (2013). Taking those two propositions into consideration, we create a scheme that contains attributes of detecting method on which classification may be based and present it in the following subsection.

A Novel Classification Method

Figure no.1 presents a classification of the criteria/attributes based on which botnets detection methods could be described. It is necessary to point out that not all criteria have to be linked with each method but only some of them. The main reason why a new approach to classifying botnet detection methods was taken is the complexity of current methods. Today, it is not sufficient to operate only with a classification of detection methods because now they have to implement more than one technique to be effective. Furthermore, the presented classification of attributes is also helpful to describe even the first botnet detection methods which used very simple techniques. Using the presented classification approach, it is possible to compare different botnets detection methods using well known comparison algorithms.

First attribute is called *Data Extraction and Analysis* and contains two values: *Host-Based Analysis* and *Network-based Analysis*. The first one requires monitoring agent on every device in a network (at least those used by users). The main goal is to localize malicious files or actions (registry modification, file extraction etc.) After a successful identification of the malicious software, different approaches could be taken. One of the most common is to create a signature of such a file and upgrade *Host and Network Based Intrusion Prevention* and *Detection Systems*. Another one is to start the reverse-engineering (RE) process to find out more details about the implementation and techniques used by particular malware. This process is very time-consuming and not every organization is able to implement it. One of the methods based on this criterion was proposed by Stinson et al. (2007). The approach implemented by the authors is to monitor individual machines to find any suspicious and unusual behaviour such as larger CPU usage or interaction with suspicious files. Another method from this group was proposed by Liu et al. (2008). Their method called BotTracer monitors system-level activities to detect unusual actions. Mehedy Masud proposed a method where the main approach is to monitor multiple logs in the host file system. He assumes that bots respond faster than users and this feature could be spotted and compared in log time stamps (Masud et al. 2008).

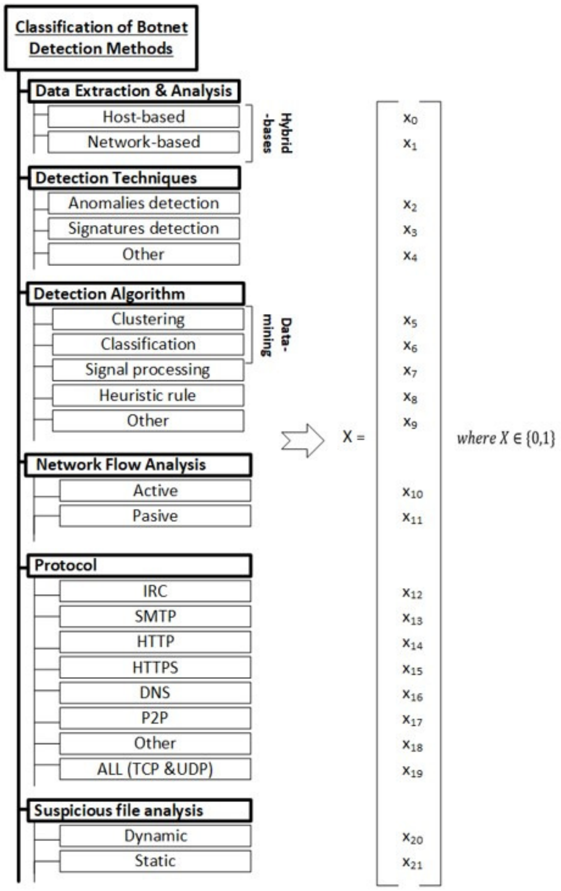


Fig. 1 :An overview of the classification criteria for botnet detection method.

DOI: 10.5171/2022.156851

There are also situations where methods with implemented *Host-based Analysis* value of *Data Extraction & Analysis* attribute are used only when a malicious action is identified by methods with implemented *Network-Based Analysis*. In this approach, a monitoring agent is not necessary, a detailed analysis of the infected machine is conducted after identifying its malicious network traffic to the C2.

The second value of the first attribute is called *Network-based Analysis* and is the most commonly implemented approach by the researchers so far. Methods with this value only need access to the analysed network without any necessity of installing a monitoring agent on every device in it. This is the main reason why they are mostly used nowadays (Silva et al. 2013). They owe their popularity to the simplicity of use. Most of them do not require any additional hardware; furthermore, no modification in the network structure is needed. *Network-based* method could be used in any part of the botnet life cycle. Their main advantage is the possibility to mitigate even unknown botnets, which have not been used for any malicious action so far.

The second attribute *Detection Techniques* contains values such as: *Anomalies detection, Signatures Detection and Others*. Methods with implemented one or more values that belong to this attribute are focused on characterization of the network flow. Next attribute is *Detection Algorithms*. Its values are used in each botnet detection method. For the needs of the presented classification, they are grouped based on their main approach. The next attribute *Network Flow Analysis* consists of two values: *Passive Network Flow Analysis* which allows only to observe network flow to localize potential bots and *Active Network Flow Analysis* which also provides methods to modify the network flow. It allows to localize infected devices by changing the content of the captured packet and send it to the suspicious destination or source IP address. The fifth attribute is linked with the protocol being analysed. The last attribute *Suspicious file analysis* represents the methods which are focused on the suspicious file identification and analysis. The values from this group are linked with the final approach of how the method makes the final decision. The first value describes a *Dynamic* method which includes opening and running potentially malicious file in a secure environment and *Static* which allows only identifying different attributes without running the files.

Nowadays most of the newly implemented methods are network based with modern data-mining algorithms to discover anomalies in the network flow (Silva et al. 2013) using passive network flow analysis algorithm against one or more protocols. According to this, we are mostly focused on the research based on methods that have implemented such values of presented attributes.

For the sake of clarity below we classify some botnets detection methods based on the scheme described above. The method proposed by Gu et al. (2008), called BotSniffer, has implemented *Network Based Analysis* to detect *Anomalies using Clustering* methods. This algorithm, due to *Passive Network Analysis,* does not require any interaction with the network or devices. Another representative with the same features was proposed by Binkley et al. (2006). Their approach is to detect bots by analysing IRC commands and messages. The only difference between those two methods is the protocol; the first one is tcp-based and the second one is effective only against botnets based on IRC communication.

Another presented group consists of attributes as follows: Network-Based Analysis, Signature Analysis, Clustering, Passive Network Flow Analysis and Multiprotocol. The method called BotMiner is one of the representatives (Gu et al. 2008). This approach combines results from two monitors; the first one utilizes Snort (Roesch, 1999) to identify malicious actions and the second one is responsible for network monitoring.

The method proposed by Gu et al. (2009) is also based on the attribute Network-Based Analysis Method which tries to identify

Anomalies using Classification with the Active Network Flow Analysis approach against only IRC based botnets. The idea behind this method is to inject modified packets that will probe the internal devices if a bot or a human is managing the other side of the communication channel.

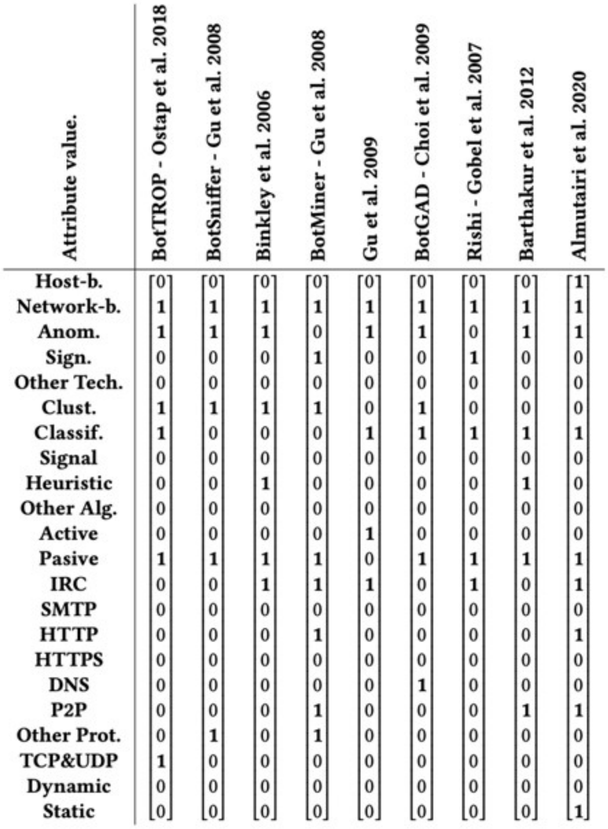
Another method strictly correlated with the clustering approach is called BotGAD (Choi et al. 2009; 2012). It is also based on Network-Based Analysis and Analysis of the Anomalies using only Passive interaction with the network flow. The main goal of BotGAD is to detect group activity which may denote the communication process between infected machines and a C2. The group activity (synchronous activity) is a widespread phenomenon widespread among bots, crawlers, and other tools developed to conduct actions simultaneously and automatically. Such tools, unlike humans, tend to communicate with the target in a highly synchronized fashion. BotGAD focuses on finding such actions by analyzing Domain Name System queries in the network flow.

The algorithm implemented by Göbel et al. (2007) is a representative of the group with attributes as follows: Network-Based Analysis, Analysis of Signature, Classification and Passive Network Flow Analysis, and is only effective against IRC network flow. This method tries to localize suspicious IRC servers, unusual ports for IRC communication and suspicious usernames by inspecting network traffic. A detection method that utilizes Heuristic approach was proposed by Barthakur et al. (2012). It is also linked with Network-Based Analysis for spotting any anomalies in P2P protocol by passively monitoring the network flow. For the network classification, Support Vector Machine (SVM) is used. The method is based on the observation that legal P2P traffic has significant differences in comparison with malicious P2P network traffic.

The representative of the Hybrid-Based Analysis was proposed by Almutairi et al. (2020). Authors utilize techniques from Host-Based Analysis such as a monitoring agent and Network-Based Analysis in one approach. Their algorithm called HANABot works at the host level and network level and focuses on HTTP, IRC, IP fluxing and P2P network traffic to detect infected machines.

Table 1 contains all the methods described above, classified using the proposed scheme of the classification criteria. It is possible to describe every botnet detection method using a previously described vector of attributes' values.

Table 1: The novel classification method.



Attribute value.	BotTROP - Ostap et al. 2018	BotSniffer - Gu et al. 2008	Binkley et al. 2006	BotMiner - Gu et al. 2008	Gu et al. 2009	BotGAD - Choi et al. 2009	Rishi - Gobel et al. 2007	Barthakur et al. 2012	Almutairi et al. 2020
Host-b.	0	0	0	0	0	0	0	0	1
Network-b.	1	1	1	1	1	1	1	1	1
Anom.	1	1	1	0	1	1	0	1	1
Sign.	0	0	0	1	0	0	1	0	0
Other Tech.	0	0	0	0	0	0	0	0	0
Clust.	1	1	1	1	0	1	0	0	0
Classif.	1	0	0	0	1	1	1	1	1
Signal	0	0	0	0	0	0	0	0	0
Heuristic	0	0	1	0	0	0	0	1	0
Other Alg.	0	0	0	0	0	0	0	0	0
Active	0	0	0	0	1	0	0	0	0
Pasive	1	1	1	1	0	1	1	1	1
IRC	0	0	1	1	1	0	1	0	1
SMTP	0	0	0	0	0	0	0	0	0
HTTP	0	0	0	1	0	0	0	0	1
HTTPS	0	0	0	0	0	0	0	0	0
DNS	0	0	0	0	0	1	0	0	0
P2P	0	0	0	1	0	0	0	1	1
Other Prot.	0	1	0	1	0	0	0	0	0
TCP&UDP	1	0	0	0	0	0	0	0	0
Dynamic	0	0	0	0	0	0	0	0	0
Static	0	0	0	0	0	0	0	0	1

Summary of botnet detection methods review

Nowadays, the biggest group of detection methods is Passive Net- work Analysis and in general data mining. They are easy to implement in a specified network. The process of analysing does not require Deep Packet Inspection (DPI) and any additional physical access to the potentially infected devices. This is the main reason why they are most frequently implemented nowadays. They owe their popularity to the simplicity of use. In most cases, they do not require any additional hardware; Furthermore, no modification in the structure of the network is needed. After suspicious communication, as well as the identification of the devise that generated that traffic, analysis commences where host-based methods are used to discover for example other C2 servers. They could also be adopted in any part of a botnet life cycle, but their main advantage is the possibility to detect even unknown botnets in creation or maintenance phases, which have not been used for any attack so far. There is a subgroup of especially effective methods which are based on detecting synchronous action. High efficiency of such methods is based on the assumption that botnets must work synchronously in order to achieve their goals. To conduct a successful SPAM campaign or a DDoS attack, it requires the Botmaster to use as many bots as possible. During the research, it was observed that synchronous activity occurs not only during the attack phase but also during the creation and management phases of the Botnet. This fact allows us to detect the threat before the first attack, which is a big improvement compared to the already known methods.

DOI: 10.5171/2022.156851

BotGAD (Choi et al. 2009; 2012) is a representative of this group. It is an interesting method but has a few disadvantages. First of all, it allows to analyse only the DNS protocol while looking for malicious activity. It is also vital to specify a proper time-window parameter, which is very problematic especially when we are dealing with an unknown botnet. This paper presents the method called BotTROP which is based on BotGAD and it is free of its disadvantages and restrictions.

BotTROP: A method for detecting botnets

The main goal of this section is to describe the BotTROP algorithm and its main concept. To improve the description, a graphical representation was also prepared.

The main concept of the method

BotTROP is able to identify infected PCs called bots in a network infrastructure of any organization. We assume that bots communicate with C2 servers outside the organization in similar moments. This feature of botnet is named synchronous activity or group activity (Choi et al. 2012). BotTROP, at first, records all public IP addresses (destination IP), which were the destination of the connections generated by nodes (source IPs) using specified protocols. Next, for a fixed destination IP and communication protocol, BotTROP uses a clustering method for grouping all the moments, when packets from source IPs were sent to the analysed destination IP. During the next step, similarity of identified groups of moments is calculated. High value of similarity factor means that the groups being analysed contain very similar source IPs and they are connecting with a destination IP in a similar moment (i.e., in a synchronous way). This fact indicates that an identified group of source IPs (internal nodes of organization) could be bots of some botnet, and destination IP is the C2 server of that botnet.

To make a decision if analysed network traffic is synchronized, it is possible to calculate average cosine similarity for every cluster, but it is very time-consuming. To increase the speed of the calculation, the Sequential Probability Ratio Test (SPRT) is used (Gu et al. 2008) (Jung et al. 2004). All the steps described above are done for all recorded public IP addresses (destination IP) and all analysed protocols separately. The exception are those destination IPs which belong to the same organization for example: Twitter, Facebook, Amazon AWS etc. They could be used by the same botmaster for the communication purposes with infected PCs. In such a situation, the destination IP is changed because those organizations use a load balancer for network optimization. This method is used by botnets based on social networks. To use BotTROP, it is necessary to collect all the public addresses of the services because they could be changed during the monitoring time. For example, clients can connect to different IPs in the Twitter range during every communication process. Missing one of them could lead to false positives. Furthermore, separate Twitter IP addresses could be used by different source IPs. Only by combining them into one group (owner of IP range like Twitter) could lead to the detection of malicious activity. It is not possible to detect synchronous activity only with one source IP, in such situation this traffic would be admitted as legal. Presented method is effective against all of the protocols used during network connectivity of any type. Details of BotTROP method are presented in the next section.

The algorithm

To simplify the presentation, it is assumed that the following form of algorithm is a study of the existence of a botnet for a fixed address or group of destination addresses and a fixed communication protocol. In order to describe the method, we define the following denotations:

- $SIP = \{1,2,\dots,I_s\}$ – the set of IP source ordered numbers.
- $T(i) = \left(t_r(i)_{r\in ST(i)}\right)$ for $i \in SIP$, where $t_r(i)$ – the moment, when computer with i-th IP source address was connected with the destination address in r-th session.
- $ST(i) = \{1,2,\dots,I_T(i)\}$ – set of the ordered numbers of moments, when i-th IP source address was connected to the destination address.
- IC_{max} – the number of connections generated by the most active source IP address.
- $\bar{C} = \{1,2,\dots,I_T(i)\}$ – set of ordered numbers of subsequent connections of the most active source IP address.
- Cl – the set of ordered numbers of IP source, which take part in synchronous action.

To increase the speed of the BotTROP algorithm, clusters with only one moment of communication are removed (cluster filtering). Furthermore, source IPs that are exhibiting low activity are also removed (source IP filtering). Test shows that cleaned data sets help to reduce the number of false positives and false negatives classifications. Cluster and source IP filtering are separate algorithms that are also included in the final BotTROP implementation.

For the needs of the Sequential Probability Ratio Testing (SPRT) method, we determine values for the parameters: *thr, θ_0, θ_1, a, b.* Detailed discussion of this problem is presented by Ostap et al. (2018). Shortly speaking, we assume that:

- thr - threshold of similarity between following clusters, where analysed source IPs were connecting to the same destination IP. From research presented by Ostap et al. (2017) results that thr could take value not less than 0.5.
- we consider the hypotheses:
    - H_0 – the analysed source IPs do not belong to the botnet,
    - H_1 – the analysed source IPs belong to the botnet.

    We are using following notations:

    - $\theta_0 = P\{Y_r = 1|H_0\}$
    - $\theta_1 = P\{Y_r = 1|H_1\}$
- α - the maximal value of false positive rate (the type I error rate or significance level)
- β - the maximal value of false negative rate (the rate of the type II error)

It means:

- $P\{rejection\ H_0|\theta_0\} \leq \alpha$
- $P\{rejection\ H_0|\theta_1\} \leq \beta$

Values of α and β are determined by the analyst. In our analysis we assume that α = 0.05 and β = 0.1. Furthermore, parameters a and b of SPRT method are determined as follows:

$$\alpha = \frac{\beta}{1-\alpha}\,;\,\beta = \frac{1-\beta}{\alpha}$$

The interpretation of the algorithm steps

The Figures below present the preparation for the clustering process of the one destination IP (organization IP range). All the source IPs which were connecting to the analysed destination IP are on the vertical axis and moments when the connection took place are on the horizontal axis. The crosses represent the moments when every source IP sent an initialization packet (SYN) to the analysed destination IP.

All information necessary for the clustering process can be gathered with any kind of network sniffer such as tcpdump or wireshark. At first, initial number of clusters is calculated (IC_{max}). It is equal to the highest number of connections from source IPs to the analyzed destination IP. The most active source IP denoted as I_{max}. Figure 2 presents the process of the calculation of the initial number of clusters.

The most active IP is denoted with an oval, and based on its connection number, the same number of clusters will be created. During the next step, all connection moments of every source IP are compared with the connection time of the IP on which clusters were created and assigned to the cluster whose connection time is most similar.

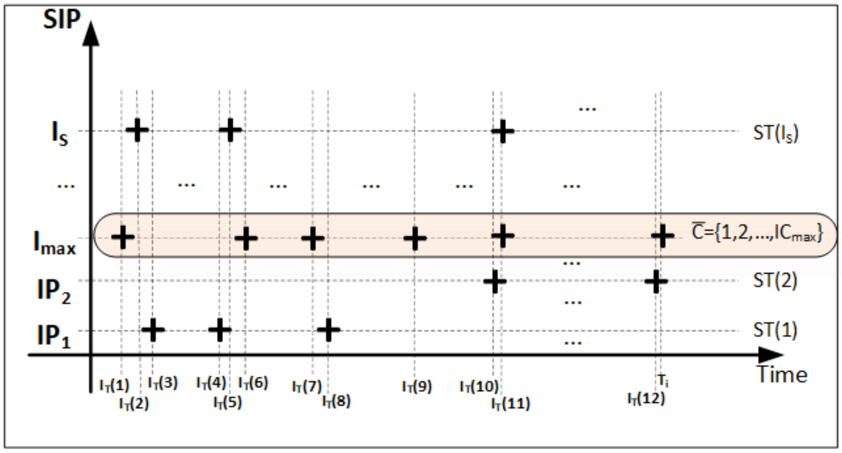


Fig. 2: The calculation process of the initial number of clusters.

To increase the speed of the algorithm, we delete low active source IPs (source IP filtering) and clusters with only one moment of communication (clusters filtering) to reduce the data bulk. They also improve results and decrease false positives and false-negatives classifications. The source IPs filtering method distinguishes legal and malicious traffic by using the K-means clustering method. Figure [3] presents the filtering process.

When clusters are prepared, Sequential Probability Ratio Test (SPRT) is used to determine if the analysed traffic is synchronous or asynchronous. Based on groups activity, presented method can detect even an unknown botnet and it is able to identify not only communication between C2 and an infected host but also working activity such as SPAM and DOS attacks.

Case Study

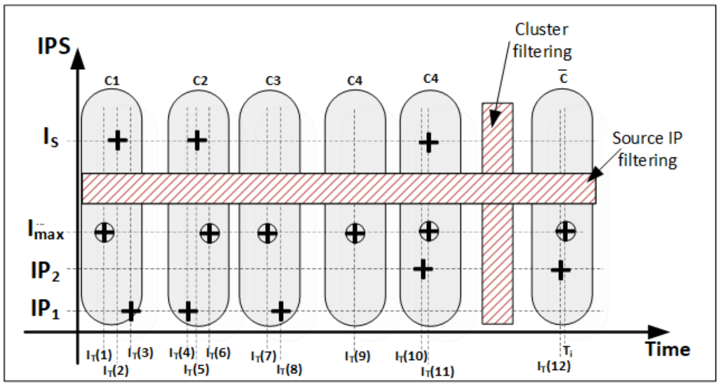


Fig. 3. Cluster filtering process.

In order to verify the properties of the BoTROP method, extensive research was conducted. Its aim was to verify the ability to detect new, previously unknown botnets and to compare the quality of the method with other methods in this category (Network-based, Passive analysis, detection of anomalies with implemented clustering/clustering techniques). The comparison also took into account the limitations of the methods due to the communication protocols used by botnets.

The measurement method

The tests were performed using two groups of data:

- Real-life network data such as:

  - Computer networks traffic of the Military University of Technology in Warsaw, Poland.
  - Traffic generated by real-life malware and registered in the Czech Technical University in Prague during the project The Malware Capture Facility Project (MCFP).
  - Network traffic generated by Powershell Empire and TwitterBot.
  - Network traffic generated by a botnet simulator and legal network traffic simulator.

- Network-traffic generated by legal and botnet traffic simulators.

The first group of data was used to verify BotTROP's ability to detect previously unknown botnets. The second group was used to verify the quality of the BotTROP method and to compare this method with the method of the same category - BotGAD. In both cases, BotTROP correctly classified mentioned traffic as malicious and was able to distinguish it from legal network flow. Figure 4 presents the simplified scheme of the experiment. The results gathered from the test with MCFP project were presented by Ostap et al. (2018) as well as the experiment with Powershell Empire and TwitterBots. Furthermore, the implementation and the environment of the TwitterBot and Powershell Empire was also included in the mentioned publication.

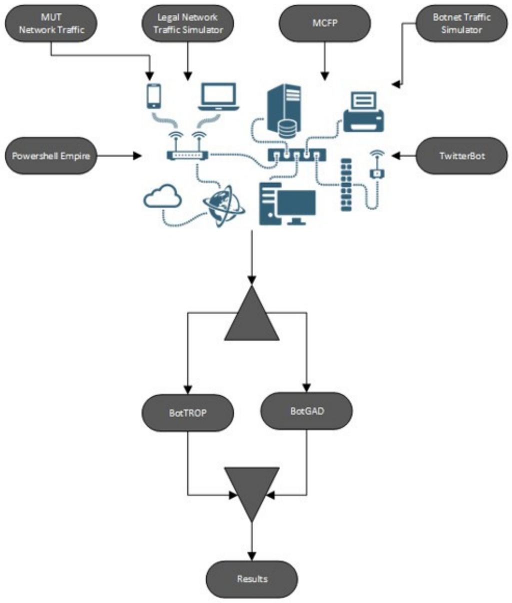


Fig. 4. The simplified scheme of the experiment.

All data sets, listed above, were mixed together and examined by BotTROP and BotGAD simultaneously as shown in the picture below. To compare the results with third party mitigation techniques, all results were verified using services listed below:

- VirusTotal - an Internet service that allows you to scan in- dividual files and present the results of finding out possible infections with malware.
- HybridAnalysis – free malware analysis services established by Payload Security. These services allow submitting potentially malicious files and analysing them using static and dynamic methods. All results are presented to the user.
- Google Safe Browsing – it is a blacklist service powered by Google that contains information about malware or phishing content. Browsers such as Firefox, Chrome, Safari use this service for checking pages against potential threats before displaying malicious content to the user.

The dissertation of Ostap (2020) presents a full description of the experiment and its results. Due to the limited number of pages, this paper contains only the most important results of the studies.

Description of the dataset

The data set used for this research was captured from the real life network used in the previously mentioned Polish university. It contains only all outgoing SYN-packets from the internal network to the destination IPs. This type of packets represent the willingness of the source IP to connect to the destination IP and according to the Mohammadi et al. (2019) and Khurum et al. (2019), SYN packets are sufficient to make a final decision if connections are made synchronously or not. By limiting the number of captured network traffic, the algorithm is able to serve final results much faster.

The packets were captured using port mirroring on the last router before the router of the ISP (Internet Service Provider). Due to security reasons, packets were also captured before the main firewall. The process of monitoring and gathering network traffic took 2 weeks. During this period of time, we also launched Powershell Empire commonly used by Red Teams and APT actors like FIN7, APT10, APT29 (Cimpanu, 2019). The C2 of this project was established on the VPS with public IP outside the private network and client software was installed on the several PCs in one of the classrooms. One of the most valuable features of PSE is the opportunity to choose jitter - the parameter that allows randomizing connection delays between the client and PSE C2. The main goal of including Powershell Empire in this research was to present that even changing the jitter parameter BotTROP is still able to detect synchronous action. The captured network traffic contains also all packets generated for the need of the Powershell Empire botnet. This open-source project was used only for 2 days due to security reasons. Table [2] describes captured network traffic in more details.

Table 2. The details about the network traffic from MUT.

Parameter	Value
Number of all packets	135 M
Number of PSE packets	58 437
File size	11 GB
No. of unique destination IP	98 117

Results of the experiment and interpretation thereof

Final results contain 49 (including PSE C2) destination IP addresses that were used by source IPs in a synchronous manner which was pointed by the SPRT method and their cosine similarity factor was greater than 0.5. Results are presented in table [3]. The thirteen destination IPs were also labeled as malicious by third-party vendors (VirusTotal, HybridAnalysis and GoogleSafeBrosing). To avoid legal consequences, only those IPs are publicly accessible, others, not detected by other malware detection system, are presented with labels.

The columns of table [3] with final results have the following meaning:

- IP - destination IP address that was used in a synchronous manner
- BotTROP ACS - contains Average Cosine Similarity. This factor is calculated as follows: cosine similarity factor is measured for each day separately, then the value is averaged after all days.
- SPRT - the number of days when the network flow was identified as synchronous by the SPRT method (network was monitored for 14 days). It is typical for universities that the number of working PCs is smaller during the weekends, moreover the number is also smaller during the nights. This is the main reason why synchronous activity is equal 0 during weekends.
- BotGAD - for comparing purposes, this column contains average cosine similarity calculated by the BotGAD method.
- Third-party vendors - the number of identified unique malicious software/campaigns linked to the analyzed destination IP. Rows where IP is labeled as "MS" means that it was used to host malicious site.

Table 3. The final results

No	IP	BotTROP (ACS)	SPRT per day	BotGAD	Third party vendors
1	Hidden-IP-1	0.98737	14/14	0.00047	0
2	192.229.220.142	0.97523	14/14	0.00510	24
3	Hidden-IP-2	0.97395	14/14	0.00597	0
4	Hidden-IP-3	0.94961	14/14	0.22230	0
5	Hidden-IP-4	0.94898	14/14	0.94583	0
6	176.9.34.43	0.89684	14/14	0.01217	9
7	Hidden-IP-5	0.81693	10/14	0.38067	0
8	Hidden-IP-6	0.78324	12/14	0.03215	0
9	Hidden-IP-7	0.78154	13/14	0.00080	0
10	Hidden-IP-8	0.73888	10/14	0.07508	0
11	Hidden-IP-9	0.72888	12/14	0.00101	0
12	Hidden-IP-10	0.72450	11/14	0.07207	0
13	Hidden-IP-11	0.71249	13/14	0	0
14	Hidden-IP-12	0.69978	10/14	0.35544	0
15	Hidden-IP-13	0.68198	13/14	0.00376	0
16	Hidden-IP-14	0.67787	14/14	0.00357	0
17	Hidden-IP-15	0.66199	10/14	0.00298	0
18	Hidden-IP-16	0.65748	11/14	0.00093	0
19	52.232.106.174	0.65499	12/14	0.00089	75
20	Hidden-IP-17	0.64972	9/14	0.25668	0
21	65.55.50.157	0.63483	11/14	0.08763	40
22	Hidden-IP-no18	0.62802	10/14	0.00010	0
23	172.217.16.46	0.62157	10/14	0.00116	185
24	Hidden-IP-no19	0.61858	10/14	0.02929	0
25	Hidden-IP-no20	0.61006	13/14	0.00700	0
26	111.111.111.111	0.60180	10/14	0.244153	9
27	188.92.40.78	0.60046	5/14	0.066036	7
28	Hidden-IP-no21	0.59850	9/14	0.00067	0
29	134.170.58.221	0.59463	11/14	0.09249	7
30	74.80.130.230	0.59033	9/14	0.25832	5
31	Hidden-IP-no22	0.58154	10/14	0.07891	0
32	Hidden-IP-no23	0.57999	10/14	0.31258	0

33	Hidden-IP-no24	0.57642	10/14	0.08646	0
34	Hidden-IP-no25	0.57104	8/14	0.96279	0
35	157.55.240.94	0.54955	9/14	0.09119	3
36	Hidden-IP-no26	0.54100	11/14	0.00092	0
37	8.18.216.151	0.54080	8/14	0.00083	10
38	Hidden-IP-no27	0.53954	12/14	0.00095	0
39	Hidden-IP-no28	0.53384	10/14	0.00013	0
40	165.227.63.195	0.53355	11/14	0.00128	MS
41	Hidden-IP-no29	0.52779	10/14	0.00740	0
42	Hidden-IP-no30	0.52238	7/14	0.51098	0
43	Hidden-IP-no31	0.51422	7/14	0.00149	0
44	Hidden-IP-no32	0.50693	9/14	0.21741	0
45	Hidden-IP-no33	0.50661	8/14	0.36822	0
46	165.227.63.195	0.5	7/14	0	MS
47	Hidden-IP-no34	0.5	7/14	0	0
48	Hidden-IP-no35	0.5	7/14	0	0
49	PowershellEmpire	0.814589	2/14	0.0925379	0

All 49 destination IPs were labeled by BotTROP due to synchronous communication pattern. PCs with source IPs sending packets to them in a synchronous manner. The protocol used for communication purpose in most cases was HTTP(port 80) and HTTPS (port 443). There were also 3 IPs (165.227.63.195, Hidden- IP-28, Hidden-IP-27) where an unknown protocol was used on port 81. HTTP(80) and HTTP(443) are commonly used by attackers due to the fact that administrators don't block this port as it is needed to connect with the Internet. Moreover, visual representation of the analyzed network traffic in Figures [7][8][9] reveals equal delays between clusters and malicious symptoms such as network flow grouping. The thirteen destination IPs addresses presented in Table [3] (rows:2,6,19,21,23,26,27,29,30,35,37,40,46) are also recognized as malicious by third party services mentioned above and two of them are labeled as phishing sites. What is worth underlining is the fact that only about 30% of all identified IPs are detected as malicious by third-party services (HybridAnalysis, VirusTotal, GoogleSafeBrowsing). The SPRT method was able to identify the whole PSE traffic for all 2 days (because of the security reason, Powershell Empire was running only for 2 days).

BotGAD was able to identify only 3 destination IPs which are used in a synchronous maner under the same circumstances as BotTROP. It is also worth underlining that BotGAD was not able to detect Powershell Empire C2. It may be caused by a wrong time-window parameter setting. Based on the presented results, we can tell that the effectiveness of the BotTROP method does not depend on the time-window parameter.

Due to the fact that BotTROP was able to identify synchronous network communication to 13 well known malicious IPs and 36 (including PSE C2) to unknown destination IPs, it could be confirmed that the presented method has the ability to detect unknown synchronous communication. It is also worth mentioning that some of the detected IPs could belong to well known vendors for example as a download center of updates for Windows or a new Antivirus signatures center. Communication for such purposes could be also synchronous. To avoid such false-positives, it is necessary to include a kind of white-list that will be fitted to the needs of a particular internal network. From our point of view, it is necessary to treat IPs detected by BotTROP as potentially malicious and require further action if they belong to well-known organizations. There are examples where a botnet was using Twitter servers as its C2. In the future, also Youtube or Instagram could be used in malicious ways. Due to this reason, all IPs that belong to services where users are able to add some content in it should be treated as potential threats, of course if network traffic generated to them is synchronous.

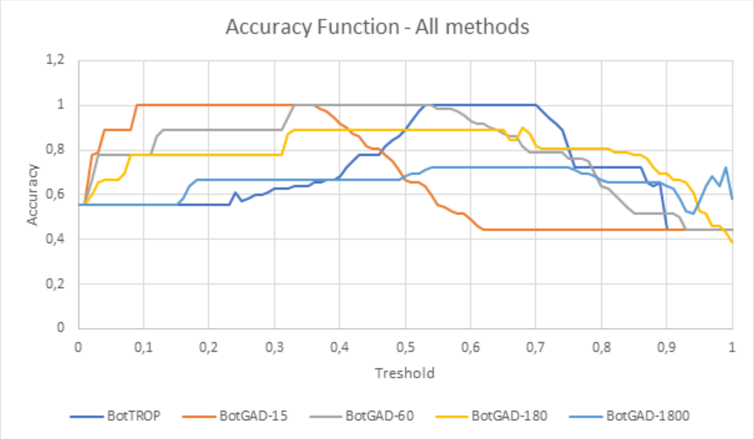


Fig. 5. Accuracy function for all the analyzed methods

To compare the functionality of BotTROP and BotGAD, accuracy function and false-positive rate function have been calculated based on the false-positives, false-negatives, true-positives and true- negatives indicators. The comparison was prepared for BotTROP and BotGAD with four different time-window parameters.

The diagram presented in Figure [5] compares accuracy for the value of threshold between 0 and 1. Threshold means a minimal value of cosine similarity factor, above which compared methods treat analyzed traffic as synchronous (BotTROP and BotGAD with different Time Window parameters).

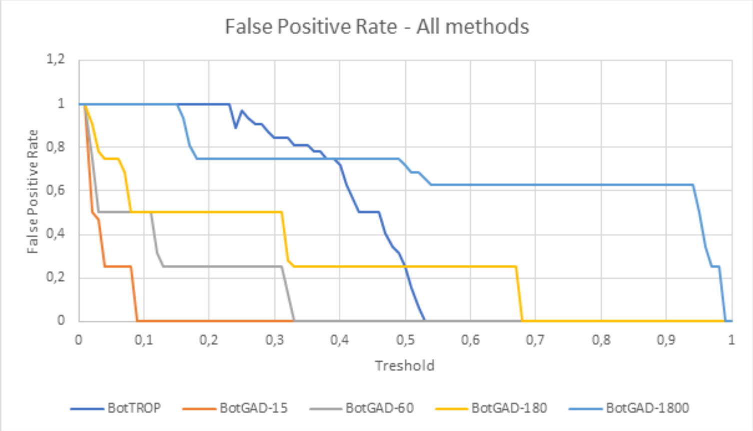


Fig. 6: The false-positive rate for all the analyzed methods.

It is shown that the BotTROP method is the most accurate method for botnet detection from all the tested methods. Its accuracy is equal to 1 for thresholds from 0.53 to 0.73. The minimal value of threshold where FPR and FNR are equal to 0 for BotTROP is also an optimal solution. BotGAD with a 60s. time window is also very accurate but only when the threshold is between 0 and 0.57. Because of this threshold, this method is unusable in real life. Based on the picture below, one can say that accuracy of the BotGAD-1800 is efficient, however its accuracy comes from the very long Time Window parameter where a lot of traffic is combined. All of the above makes the BotGAD method very vulnerable to FPR.
False-positive rate presented in Figure [6] reveals that high values for BotGAD (especially for time-window =1800) in the whole range of threshold variability reduce the prognostic capabilities of this method. Moreover, this figure also proves that the BotTROP method has very low false-positive rate when threshold is above 0.5.

Detailed view of one of the malicious IPs

The following subsection contains a detailed presentation of one of the malicious IPs identified by BotTROP. During the survey, Author- sPublications software implementation of the BotTROP algorithm was made. The implementation include the following functionalities:

- analysis of accumulated traffic using BoTROP method,
- additionally, for the comparison purpose, method BotGAD was also implemented,
- graphical representation of the results obtained,
- comparison of the results with three commercial solutions for malicious motion detection.

Presentation is made using the BotTROP software used during experiment. The public IP 172.217.16.46 was also identified by third- party software as malicious. Hybrid-Analysis lists 185 different malicious activities that were linked with this public IP. This address was also used as a malicious site and the drop zone for trojans based on JAVA. BotTROP identified synchronous activity for 11 days out of 14 days of monitoring time. Cosine similarity was quite high, it was equal 0.62157. The number of internal devices that were connecting to this C2 amounted to 240. All graphics comes from the software implemented for the needs of the BotTROP algorithm. Figure [7] presents a graphical representation of the network traffic to this domain. This graph is also helpful for an analyst in making a final decision because it is easy to spot group activity in similar moments.

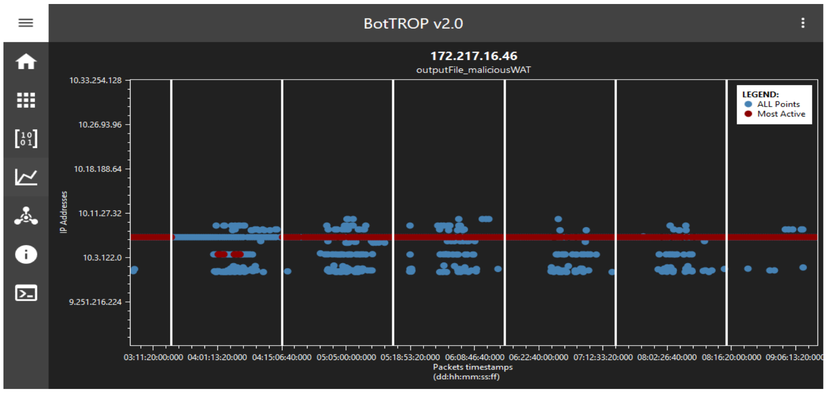


Fig. 7 The network flow directed to the IP 172.217.16.464.

The horizontal axis denotes times of connection and the vertical one contains the source IP addresses. Red dots represent the connection time that belongs to the most active source IP. Blue dots denote a connection to the selected destination IP from an IP address that belongs to the analyzed network. The white lines represent borders between subsequent days.

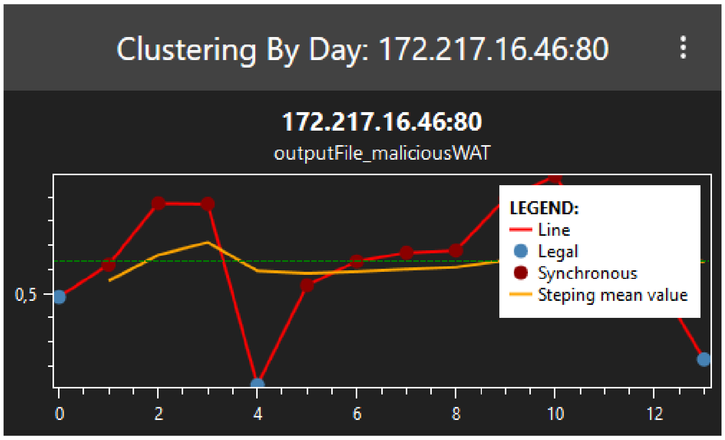


Fig. 8. The cosine Similarity factor per day for 172.217.16.46.

Cosine similarity factor for each day separately is shown in Figure [8]. The horizontal axis shows days and the vertical axis average cosine similarity. Blue dots denote that network traffic was labeled as legal by the SPRT algorithm, the red one means that the traffic was synchronous that day.

Average cosine similarity stays on a high level most of days including the SPRT results. This graph underlines that this particular destination IP was used in a synchronous manner.

A graphical representation for the cosine similarity factor between the following clusters is presented in Figure [9]. There are situations where neighboring clusters differ a lot between each other. The horizontal axis denotes subsequent clusters and vertical one value between them. It can be spotted that most of similarities' values are higher than 0.7, which may indicate malicious activity. As we can see, a high level of cosine similarity is mostly equal for the whole monitoring time. Those three graphical representations prove that this public IP was also used in a synchronous manner.

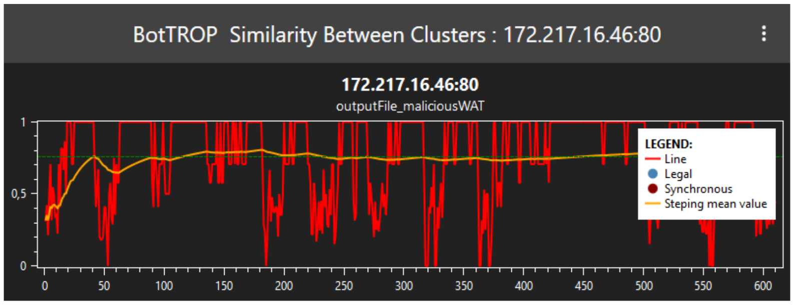


Fig. 9. The cosine Similarity for the following clusters for IP 172.217.16.46.

The summary

There are a lot of different botnet detection methods but most of them are not efficient against unknown threats, moreover most of them are able to detect a botnet based only on a specified protocol. The algorithm described in this dissertation is an attempt to change this situation and make networks more secure. BotTROP, just like BotGAD, identifies not only well-known botnets (which was proved in experiments with Powershell Empire, Neris and others), but also unknown threats (for example TwitterBot based on social networks, which was implemented for the needs of the survey [AuthorsPub- lication]). It also improves detection of synchronous activity via different protocol.

The aim of the presented experiment was to verify the ability of the BotTROP method to detect known and unknown botnets and to analyze any communication protocol. We also evaluate the quality of the BotTROP method and compare it with the quality of the BotGAD method – the results are presented in table [3]. All the results prove that BotTROP is a very efficient method for botnet detection of any kind, no matter the protocol or architecture. It is very important to underline that the role of the BotTROP method is to identify synchronous activity which in most cases takes place in malicious network flow. Implementing a white list where synchronous activity is also spotted, for example during system upgrades, can help to decrease the number of false negatives and false positives rate. Presented results also reveal that the BotGAD method is very dependent on the time window parameter, which is one of its biggest disadvantages. This makes the method unusable against unknown botnets because it is necessary to set up a relevant parameter for communication sessions with its C2. The BotTROP method is independent of Time Window parameters and can be used efficiently even against unknown - malicious or legal - network traffic, which in turn makes it possible to identify an unknown botnet. It is possible to use the BotTROP method in a production environment without any modification but there are a lot of improvements that are going to be made in the nearest future, for example on-line monitoring with graphical representation.

In order to use BotTROP in a working network, it is only necessary to provide its traffic for analysis. Currently, work is underway on the possibility of an online analysis without the need to collect the analyzed network traffic. Afterwards, all the collected network traffic is analyzed by the BotTROP algorithm, and the administrator has the possibility to verify the cosine similarity for each destination address and a preview of which hypothesis was adopted by the SPRT method.

The experiment helps us to observe that there are situations in which legal users generate synchronous traffic, for example to Facebook or Twitter servers during very important events such as football world championships or significant political events. All posts and comments written during such events are generated with high intensity, therefore this kind of network traffic is recognized as a synchronous action. Despite this, even in such situations, cosine similarity factor never exceeds the level of 0.5. During the research, even more intense legal network traffic never exceeded 0.5.

References

- Almutairi S., Mahfoudh S., Sultan Almutairi and Jalal Alowibdi. (2020) 'Hybrid botnet detection based on host and network analysis'. Journal of Computer Networks and Communications 2020 (01 2020), 1–16.
- Barthakur P., Dahal M. and Mrinal Ghose. (2012) 'A framework for P2P botnet detection using SVM'. 195–200.
- Binkley J. and Singh S. (2006) 'An algorithm for anomaly-based botnet detection'. (01 2006).
- Chanda K. (2014) 'Hybrid botnet detection mechanism'. International Journal of Computer Applications 91 (03 2014). https://doi.org/10.5120/15876-4823
- Choi H. and Lee H. (2012) 'Identifying botnets by capturing group activities in DNS traffic'. Computer Networks 56 (01 2012), 20–33.
- Choi H., Lee H. and Kim H. (2009) 'BotGAD: Detecting botnets by capturing group activities in network traffic'. Proceedings of the 4th International Conferenceon Communication System Softwareand Middleware,2.
- Cimpanu C. (2019). 'Development stops on PowerShell Empire framework after projectreachesitsgoal'. https://www.zdnet.com/article/development-stops-on- powershell- empire-framework- after- project- reaches- its-goal/
- Feily M., Shahrestani A. and Ramadass S. (2009). 'A Survey of botnet and botnet detection'. 268–273. https://doi.org/10.1109/SECURWARE.2009.48
- García S., Zunino A. and Campo M. (2014). 'Survey on network-based botnet detection methods'. Security and Communication Networks 7(052014). https://doi.org/10.1002/sec.800
- Zunino A. and Garcia S. (2011). 'CTU-Malware-Capture-Botnet-50 or Scenario 9 in the CTU-13 dataset'. (2011). https://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-50
- Göbel J. and Holz T. (2007). 'Rishi: Identify bot contaminated hosts by IRC nickname evaluation'. (04 2007).
- Gralla P. (2019) Malicious Bot Attacks: 'Why they're more dangerous than ever'. https://symantec-blogs.broadcom.com/blogs/feature-stories/malicious- bot- attacks- why-theyre- more- dangerous- ever
- Gu G., Perdisci R., Zhang J. and Wenke Lee. (2008) 'BotMiner: Clustering analysis of network traffic for protocol- and structure- independent botnet detection'. CCS'08, 139–154.
- Gu G., Yegneswaran V., Porras P., Stoll J. and Lee W. (2009) 'Active botnet probing to identify obscure command and control channels'. (122009),241–253. https://doi.org/10.1109/ACSAC.2009.30
- Gu G., Zhang J. and Lee W. (2008). 'BotSniffer: Detecting botnet command and control channels in network traffic'.
- Harel D. (2019) 'Threats of the Year. A look back at the tactics and tools of 2019'. CISCO CYBERSECURITY SERIES 2019,. CISCO.
- Jung J., Paxson V., Berger A. and Balakrishnan H. (2004) 'Fast portscan detection using sequential hypothesis testing'. 211 – 225. https://doi.org/10.1109/SECPRI.2004.1301325
- Khurum A. (2019) 'Removing SYN flooding in TCP/IP Network'. (01 2019). https://doi.org/10.31224/osf.io/nwrj7

- Liu L., Chen S., Yan G. and Zhang Z. (2008) 'BotTracer: Execution-based bot-like malware detection,'Vol.5222.97–113. https://doi.org/ 10.1007/978- 3- 540-85886- 7_7
- Masud M., Al-Khateeb T., Khan L., Thuraisingham B. and Hamlen K. (2008) 'Flow-based identification of botnet traffic by mining multiple log files'. 2008 1st International Conference on Distributed Frameworks and Application,DFmA2008(112008),200–206. https://doi.org/10.1109/ICDFMA.2008.4784437
- Mohammadi R., Conti M., Lal C. and Kulhari S. (2019) 'SYN- Guard: An effective counter for SYN flooding attack in software-defined networking'. International Journal of Communication Systems (07 2019), e4061. https://doi.org/10.1002/dac.4061
- Nallanthighal R., Sahgal D. and Chandna S. (2012) 'Classification of botnet detection based on botnet architecture'. (052012). https://doi.org/ 10.1109/CSNT.2012.128
- Ostap H. and Antkiewicz R. (2017) 'A Concept of clustering-based method for botnet detection'. https://doi.org/10.1007/978-3-319-65127-9_18.
- Ostap H. and Antkiewicz R. (2018) 'A concept of detection method for botnets based on social networks'. https://doi.org/10.21936/si2018_v39.n1.836
- Ostap H. (2020) Dissertation, 'Clustering-based method for botnet detection'.
- Plohmann D., Gerhards-Padillia E. and Leder F. (2011)'Botnets: Detection, measurement, dsinfection&Defence'. https://www.enisa.europa.eu/publications/botnets- measurement- detection- disinfection- and- defence
- Roesch M. (1999) 'Snort - Lightweight intrusion detection for networks. Proceedings of the 13th USENIX conference on System administration', 229–238.
- Silva S., Rodrigo S., Silva M., Pinto R., and Salles R. (2013) 'Botnets: A survey. Computer' Networks 57 (02 2013), 378–403. https: //doi.org/10.1016/j.comnet.2012.07.021
- Stinson E. and Mitchell J. (2007) 'Characterizing bots remote control behavior'. https://doi.org/10.1007/978-0-387-68768-1_3
- Strayer T., Lapsely D., Walsh R. and Livadas C. (2007) 'Botnet detection based on network behavior'. Vol. 36. 1–24. https://doi.org/10.1007/978-0-387- 68768- 1_1
- Ping Wang, Sherri Sparks, and Cliff Zou. 2010. 'An advanced hybrid peer-to-peer botnet'. IEEETrans.DependableSec.Comput.7(042010),113–127. https: //doi.org/10.1109/TDSC.2008.35