https://doi.org/ 10.33472/AFJBS.6.10.2024.4535-4450

AFJBS

African Journal of Biological Sciences

Journal homepage: http://www.afjbs.com

Research Paper Open Access

A MACHINE LEARNING BOTNET DETECTION FEASIBILITY ANALYSIS

1. Dr. Deevi HariKrishna
Associate Professor, DEPARTMENT OF CAI,
KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES, GUNTUR. harikrishnadeevi@gmail.com
2. Dr A.HANUMAT PRASAD
Associate Professor, Dept of CSE- AI&ML.
Kallam HaranadhaReddy Institute of Technology (AUTONOMOUS), Guntur
hanuma.alahari@gmail.com

Article History
Volume 6,Issue 10, Feb 2024
Received:28 Apr 2024
Accepted : 25 May 2024
doi: 10.33472/AFJBS.6.10.2024.4435-4450

Abstract:

The security threat is one of the most pressing challenges in today's networks, and it is regarded as the most critical in the management of multimedia data. Bots are a type of security attack that targets a person or a group of nodes in a network, turning them into bots. A botnet is a network of bots that can be controlled by a bot master. These bots are programmed to disrupt network activity. It collects very sensitive data including bank account numbers and personal information. Botnets are now identified utilising an intrusion detection system (IDS), which efficiently monitors network activity in entities and companies on a regular basis. Bots create bogus or undesirable data, which is then passed on to all nodes in the network, lowering network efficiency. The botnet's primary goal is to assault as many devices as possible while also disseminating malicious programmes as much as possible. Botnet assaults infect many types of technology, and even the most basic internet security suites, firewalls, and antivirus software provide some defense. We offered dynamic analysis in advance, looking for signs of infection in behavioral analysis, as well as network and network traffic anomalies. Individual symptoms of botnet attacks are combined with network-level attacks.

Botnets have attracted the interest of scholars all around the world in the last ten to fifteen years. A lot of work has gone into building technologies that can detect the existence of a botnet quickly and effectively. The goal of this thesis is to address the issues with standard botnet detection methods and come up with more effective solutions. The research presented in this thesis looks at the botnet at various phases in order to develop effective detection mechanisms and overcome the limitations of standard botnet detection methods on both the Windows and Android platforms. To address this one-of-a-kind challenge, researchers used machine learning (ML). We present a quick introduction of the various machine learning (ML) approaches and their role in botnet identification in this study. The primary goal of this work is to clearly clarify the function of various machine learning algorithms in Botnet identification. The development of effective and efficient real-time online detection algorithms and more robust models requires a comprehensive knowledge of these responsibilities.

1.Introduction:

The Bots, machines infected with bot-malware, compromised to report back to their command and control (referred to as “c2c” hereafter) servers and form a network controlled by the botmaster. This network of bots empowers the master to execute malicious activities remotely and anonymously. The bot machines are not only used as a computing resource to launch an attack on other targets rather these machines are also directly affected by malicious activities like data theft and loss. The ransomware is the recent example of such attacks where individual victims should pay to get back one’s data. Botnets are composed of three main components including bots, c2c servers, and bot-master. The c2c servers operate as an intermediate layer between bots and their master. The bots registered with their c2c servers to establish a communication channel. This communication channel is used for heartbeat and commands exchange. The botmaster connects with c2c servers to update instruction sets and get visibility of the bot network. Botnet generally has three phases including infection, c2c communication, and attack phase in their life cycle. The infection and attack phase of the botnet is the same as in other malware where the infection phase is used to infect machines and then these machines are used to launch specific attacks. The c2c communication phase classifies a malware into bot-malware. This phase is used to register newly infected machines with their c2c The Bots, machines infected with bot-malware, compromised to report back to their command and control (referred to as “c2c” hereafter) servers and form a network controlled by the botmaster. This network of bots empowers the master to execute malicious activities remotely and anonymously. The bot machines are not only used as a computing resource to launch an attack on other targets rather these machines are also directly affected by malicious activities like data theft and loss. The ransomware is the recent example of such attacks where individual victims should pay to get back one’s data. Botnets are composed of three main components including bots, c2c servers, and bot-master. The c2c servers operate as an intermediate layer between bots and their master. The bots registered with their c2c servers to establish a communication channel. This communication channel is used for heartbeat and commands exchange. The botmaster connects with c2c servers to update the instruction set and get visibility of the bot network. Botnet generally has three phases including infection, c2c communication, and attack phase in their life cycle. The infection and attack phase of the botnet is the same as in other malware where the infection phase is used to infect machines and then these machines are used to launch specific attacks. The c2c communication phase classifies a malware into bot-malware. This phase is used to register newly infected machines with their c2c The increasing number of botnet attacks and their evolving nature drive the need for continuous improvement of detection techniques. The detection techniques in botnet literature started with simple signature-based approaches and evolved gradually with time. The current proposals focus on behaviourally based approaches and mainly targeting network level information. Majority of these techniques use only network header level information to counter packet payload challenges e.g., encrypted botnet traffic and. processing complexities. The use of machine learning algorithms to extract botnet behavior patterns from the monitored network is common in these approaches. The network flow-based approaches suffer mainly with two issues. First one is flow collection which comes up with many challenges in traditional IP networks. Secondly, the detection techniques that use a diverse dataset with real traffic traces and have a good percentage of unknown traffic in the testing dataset for which the model is not trained, suffer from the relatively high false positive rate.

2.Literature Review:

Miller et al., (2016) Over the past ten to fifteen years botnets have gained the attention of researchers worldwide. A great deal of effort has been given to developing systems that would efficiently and effectively detect the presence of a botnet. This unique problem saw researchers applying mac

Haddadi, Fariba & Zincir-Heywood, A.. (2015) Botnets represent one of the most significant threats against cyber security. They employ different techniques, topologies and communication protocols in different stages of their lifecycle. Hence, identifying botnets have become very challenging specifically given that they can upgrade their methodology at any time.

Thanudas et al., (2015) Botnet comprises of collection of bot-infected computers that allows an attacker to take control and carry out large scale cyber attacks. Botnets have been used to perform various malicious activities such as Distributed Denial of Service (DDoS), information stealing, and cyber physical attacks. Botnets act in a stealthy manner by keeping themselves hidden from the users of compromised systems.

García et al., (2014) The results of botnet detection methods are usually presented without any comparison. Although it is generally accepted that more comparisons with third-party methods may help to improve the area, few papers could do it. Among the factors that prevent a comparison are the difficulties to share a dataset, the lack of a good dataset, the absence of a proper description of the methods and the lack of a comparison methodology.

Eslahi et al., (2014) Recently, MoBots or Mobile Botnets have become one of the most critical challenges in mobile communication and cyber security. The integration of Mobile devices with the Internet along with enhanced features and capabilities has made them an environment of interest for cyber criminals. Therefore, the spread of sophisticated malware such as Botnets has significantly increased in mobile devices and networks.

Beigi et al., (2014) Botnets, as one of the most formidable cyber security threats, are becoming more sophisticated and resistant to detection. In spite of specific behaviors each botnet has, there exist adequate similarities inside each botnet that separate its behavior from benign traffic. Several botnet detection systems have been proposed based on these similarities. However, offering a solution for differentiating botnet traffic (even those using same protocol, e.g. IRC) from normal traffic is not trivial. Extraction of features in either host or network level to model a botnet has been one of the most popular methods in botnet detection.flow-based features employed in the existing botnet detection studies and evaluate their relative effectiveness.

Barazi et al., (2014) Botnet detection and response is currently an arms race. The bot-masters rapidly evolve their botnet propagation and command and control technologies to evade the latest detection and response techniques from security researchers. Researches in botnets in addition to botnet detection include also Tracking, Measurement, Prediction, and countermeasure. In this study, we try to raise main points in the field of botnet detection techniques.

3.Framework of Botnets:

A network, composed of a growing number of interconnected devices, poses a significant security risk to financial and business institutions, resulting in substantial financial losses and damage to their reputation. Malicious software affecting an increasing number of users has become a critical issue, with botnets emerging as a major security threat. Botnets are particularly worrisome because of their ability to infiltrate Internet-connected devices, including digital video recorders (DVRs), and control corporate mainframes. A botnet is a network of compromised host devices, including desktop computers, smartphones, notebooks, and tablets, used for malicious activities. The essential components of a botnet include a botmaster (the attacker), a command and control (C&C) server, and infected machines known as bots. The botmaster requires a C&C channel to command and coordinate malicious attacks, with C&C channels utilizing various communication protocols, such as IRC, HTTP, and P2P. Bots carry out diverse malicious activities, including distributed denial of service (DDoS) attacks, phishing, spam emails, and more. Botnets have attracted significant research attention over the past decade, with an emphasis on detection techniques. These techniques have evolved from signature-based to more advanced behavior-based approaches. However, most current detection strategies focus on offline botnet detection, leaving a gap for real-time detection. The growing interconnectivity of society, along with the increasing number of Internet-enabled devices, has led to heightened concerns about cybercrime. Botnets, collections of infected computers, pose significant threats and engage in malicious activities such as DDoS attacks, spam distribution, and malware dissemination.

3.1.Botnet Elements:

Understanding the fundamental elements of a botnet is crucial. A botnet comprises three primary components: bots, the botmaster, and the command and control channel (C&C). Bots are software programs (malware) installed on compromised hosts and perform malicious activities. Bots are not vulnerabilities in applications or operating systems but rather programs distributed through worms or used to establish backdoors on compromised machines. These bots are initialized at each boot, and actions are executed upon commands sent through the C&C channel by the botmaster. The presence of the C&C channel distinguishes bots from other malware types. Bots in a botnet communicate through HTTP with Masterbots, allowing the botmaster to remotely issue commands and receive reports.

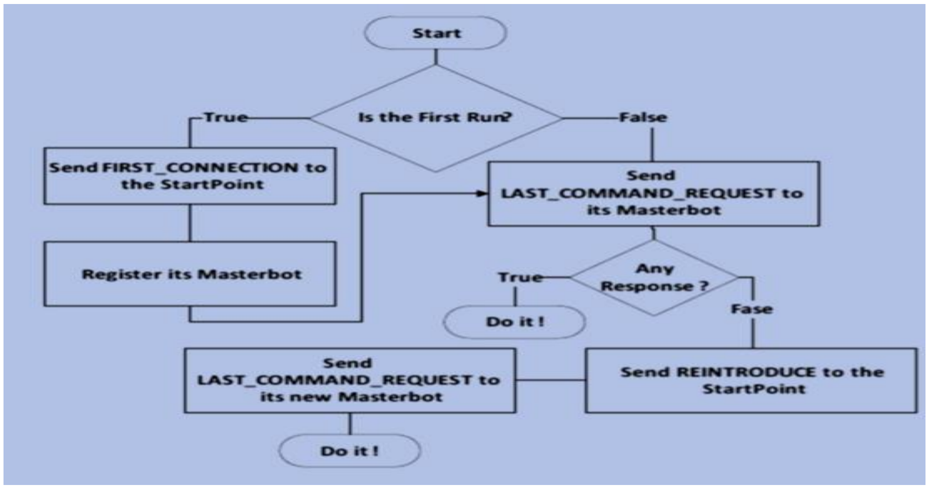

Start
Is the First Run?
True
False
Send FIRST_CONNECTION to the StartPoint
Send LAST_COMMAND_REQUEST to its Masterbot
Register its Masterbot
True
Any Response ?
Fase
Do it !
Send LAST_COMMAND_REQUEST to its new Masterbot
Send REINTRODUCE to the StartPoint
Do it !

Fig 1 Operations of Bot

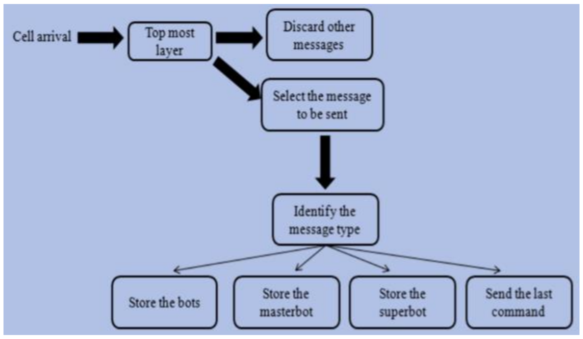
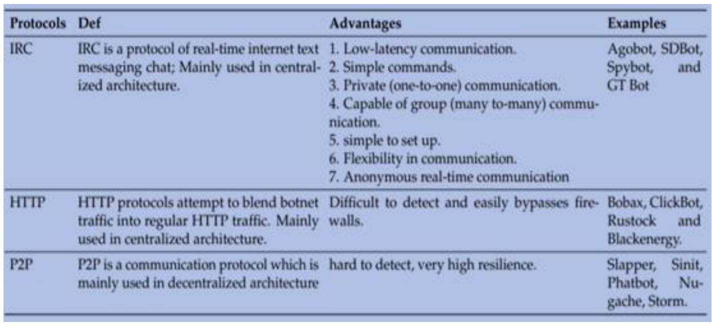


Fig 2 Operation of Masterbot

3.2.Botnet Architecture:

Botnets are categorized into three different structures according to the C&C channel: centralized architecture, decentralized architecture, and hybrid architecture. Ac- cording to the communication protocols used by botnets, botnets can be classified into several protocols. According to the communication protocols used by botnets, botnets can be classified into several protocols



Protocols	Def	Advantages	Examples
IRC	IRC is a protocol of real-time internet text messaging chat; Mainly used in centralized architecture.	1. Low-latency communication. 2. Simple commands. 3. Private (one-to-one) communication. 4. Capable of group (many to-many) communication. 5. simple to set up. 6. Flexibility in communication. 7. Anonymous real-time communication	Agobot, SDBot, Spybot, and GT Bot
HTTP	HTTP protocols attempt to blend botnet traffic into regular HTTP traffic. Mainly used in centralized architecture.	Difficult to detect and easily bypasses firewalls.	Bobax, ClickBot, Rustock and Blackenergy.
P2P	P2P is a communication protocol which is mainly used in decentralized architecture	hard to detect, very high resilience.	Slapper, Sinit, Phatbot, Nugache, Storm.

Fig 3 Botnet protocols

3.3.Centralized Architecture:

In a centralized botnet architecture, the botmaster controls all the bots from a central hub known as a command-and-control server. In this structure, a single point (the C&C server) is used to exchange instructions between the botmaster and the bots. The major benefit of this architecture is that it provides reliable coordination of the bots for their botmaster. Moreover, it makes status monitoring easy for the botmaster, and it speeds up reaction time. In contrast, once the C&C server is identified, it is very easy for a defender to take down this type of botnet. The two protocols most often used in a centralized architecture are Internet relay chat (IRC) and hypertext transfer protocol (HTTP). A centralized architecture can suffer a single point of failure, because of a denial-of-service attack, and the botmaster is no longer able to communicate with the bots when an IRC or HTTP server is taken down. Figure 1 shows a centralized model.

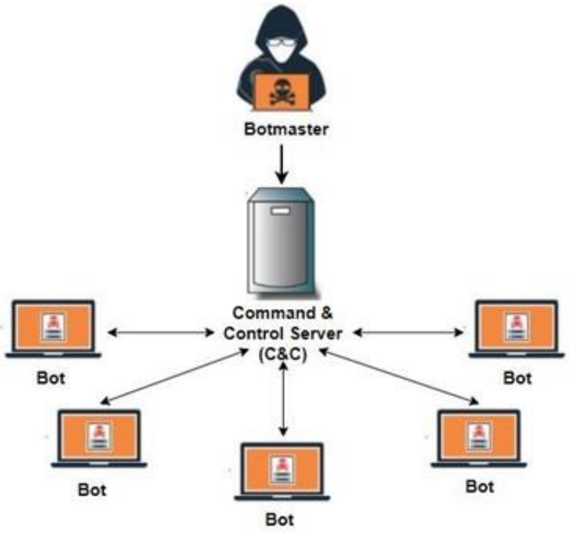


Fig 3 Architecture of centeralized model

4.Machine Learning for Botnet Detection: Machine learning is a field dedicated to developing algorithms capable of learning from data, enabling them to recognize intricate patterns and make informed decisions based on past information. The central challenge in machine learning is to generalize knowledge gained from a limited set of prior experiences, facilitating the generation of valuable decisions for unseen events. This goal is achieved through the creation of algorithms grounded in robust statistical and computational principles. The typical machine learning process comprises five phases, as illustrated in Figure 1.7. It begins with data acquisition, where relevant data is collected for analysis. The second and third phases focus on monitoring system behavior and extracting pertinent features. With these features in hand, the learning and classification stages can commence. Finally, the classifier's performance is evaluated. Our research within the machine learning process is outlined in figure.It starts with raw data, including bot binaries for Windows and APK files for Android. Data preparation follows, where features from APK files and network traffic are identified and stored in a database or flat files. Feature extraction is the next step, leading to one of the core phases in machine learning—classification. The primary objective here is to classify data as normal or botnet and possibly identify specific attack types. Machine learning offers various classification techniques, including rules, decision trees, linear and nonlinear functions, instance-based examples, probability models, and ensembles of classifiers. The primary aim is to uncover hidden patterns, enabling the development of new detection templates. This approach transcends the limitations of signature-based systems, which rely on static intrusion signatures, allowing for a transition from memorization to generalization.

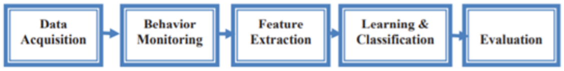


Fig 4 Machine learning process

The final stage involves the various evaluation measures used for measuring the performance of the classifiers. The measures include Accuracy, Precision, Recall and F-measure arrived at by doing a k-fold cross validation. Cross-validation (CV) is a technique for estimating the generalization performance of a predictive model. The main idea behind CV is to split the data, once or several times, for estimating the risk of each algorithm: Part of the data (the training sample) is used for training each algorithm, and the remaining part (the validation sample) is used for estimating the risk of the algorithm. Then CV selects the algorithm with the smallest estimated risk. In our work we have used 10-fold cross validation. A confusion matrix contains information about actual and predicted classifications done by a classification system. Performance of such systems is commonly evaluated using the data in the matrix.

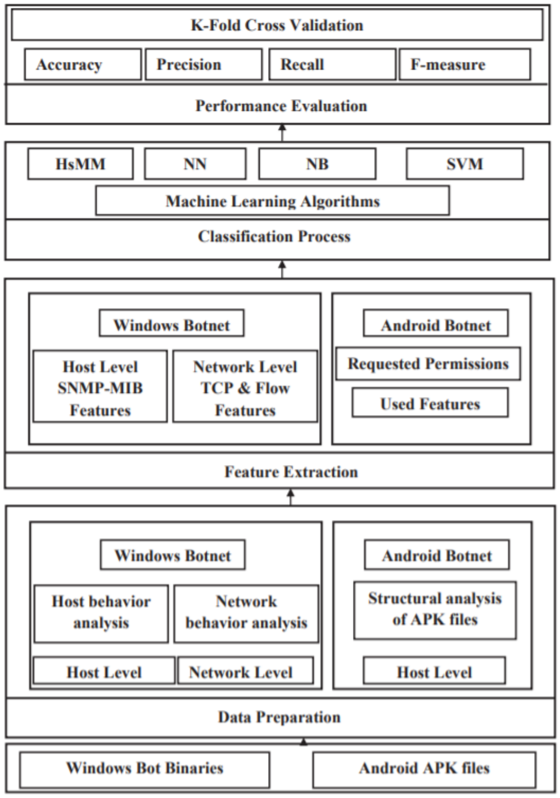


Fig 5 Overall architecture of the proposed research

5.Proposed Methodology of the Study:

This thesis focuses on a detailed analysis of Windows and Android based botnets, and the design of efficient detection mechanisms. Since nowadays, IRC botnets are not much utilized by botmasters, we concentrate on HTTP botnets. But in reality, if the detection system is capable of detecting any type of botnet, then it will be the most suitable one. Hence one of our work focuses on botnet detection irrespective of the C&C structure. A brief summary of the research work carried out is given below.

5.1.HTTP botnet Detection using Hidden Semi-Markov Model with SNMP MIB Variables:

In this chapter, we propose a Hidden semi-Markov Model (HsMM) for the host-based detection of botnets by considering that most of the communications of HTTP botnets are based on TCP related connections. The duration of a system in normal state need not be constant and exponentially distributed. Hence, we use a Hidden semi-Markov Model for modeling the system behavior. In the training phase, the SNMP-MIB variables are first transformed into HsMM observation sequence using forward-backward training algorithm. Next the HsMM is inferred from the observation sequence. In the testing phase, the SNMP-MIB variables are transformed to HsMM observation sequences, and then the HsMM is used to compute the probability of each test sequence in order to determine Average Log Likelihood (ALL) which decides whether it is a normal traffic or botnet communication. Several experiments are conducted using Spyeye, BlackEnergy, Zeus, Athena and Andromeda botnets to validate the model. The proposed model is efficient with high detection accuracy and low false positive rate. Since the proposed method used SNMP-MIB variables as features, it is computationally fast.

5.2.HTTP Botnet Detection using Adaptive Learning Rate Multilayer Feed Forward Neural Network:

In this work, we consider the analysis and detection of botnets at the network level. Most of the communications of web botnets are based on TCP connections, relative and direct features of TCP connections are extracted from the network traffic. The extracted TCP features are passed to the Multi-Layer Feed Forward Neural Network training model which uses Bold Driver Back-propagation learning algorithm. Using this approach, Spyeye, BlackEnergy, Zeus, Sogou, Athena and Andromeda botnets are efficiently identified. The performance of the proposed method is compared with that of C4.5 Decision Tree, Random Forest and Radial Basis function network. The results show an improvement in detection accuracy with neural network when compared to other classification techniques.

6.Botnet Detection via Mining of Traffic Flow Characteristics:

In this work, a method is proposed to detect botnets irrespective of their structures, based on network traffic flow behavior analysis and machine learning techniques. We have analyzed botnet characteristics in a controlled environment and found that a bot in the network will generate a burst of small packets when actively searching for susceptible hosts and exhibit a more uniform pattern when the bot queries for updates or instructions continuously, resulting in many uniforms sized, small TCP/UDP packets. Based on this behavior, traffic flow features in different time windows have been extracted and are used to classify the traffic flows into botnet or normal flows using machine learning techniques. The proposed approach is validated through experiments on a set of publicly available benchmark datasets such as ISOT botnet, CAIDA Conficker, ZeroAccess, Skynet, Citadel, Medfos, Sogou, Kelihos, Rbot, Virut.n, Eldorado and self-generated botnet datasets such as Zeus, Spyeye and BlackEnergy. The experimental results show that the proposed approach is capable of detecting the known and new botnets effectively irrespective of their structures and also it achieves high detection accuracy and low false positive rate. Results show an improvement in detection accuracy in detecting botnets when compared to state-of-the-art detection techniques.

6.1.Structural Analysis and Detection of Android Botnets:

Using Machine Learning Techniques In this work, Android botnets are considered for analysis and a detection mechanism is designed using machine learning algorithms. Unique patterns (i.e combinations of requested permissions and used features) based on malicious activities of botnets are generated by using Apriori association rule mining algorithm and information gain method is used to select the most significant patterns in order to provide a better detection. The selected unique patterns are passed to the machine learning framework to classify the applications as benign or botnet. Diverse sources of Android botnet datasets such as, Android Malware Genome project, Drebin, Droid Analytics, ISCX Android Botnet dataset and dataset from Beijing Jiao-tong University of China have been used in this study. Experiments on real-world benchmark datasets show that the selected patterns produce high detection accuracy compared with similar detection methods.

7.Feature Extraction:

HTTP botnet sends a large number of HTTP requests to the target C&C server automatically. These HTTP requests have legitimate formats and are sent via normal TCP connections and hence these requests are difficult to identify. In addition, the bots request usually is generated randomly or by repeating a few simple HTTP requests via TCP connections. Hence, in this work, we have extracted TCP connection related MIB variables for the effective detection of a botnet without analyzing the packets. These MIB variables are collected through SNMP running on the host.

The collected eight TCP connections related MIB variables are tcpActiveOpens, tcpPassiveOpens, tcpAttemptFails, tcpEstabResets, tcpCurrEstab, tcpInSegs, tcpRetransSegs, and tcpInErrs. Each of these collected SNMP-MIB variables has its own specific task. The MIB variable tcpActiveOpens gives the number of times that TCP connections have made a direct transition to the SYN-SENT state from the CLOSED state, tcpPassiveOpens gives the number of times that TCP connections have made a direct transition to the SYN-RCVD state from the LISTEN state, tcpAttemptFails gives the number of times that TCP connections have made a direct transition to the CLOSED state from either the SYN-SENT state or the SYN-RCVD state, and the number of times that TCP connections have made a direct transition to the LISTEN state from the SYN-RCVD state, tcpEstabResets gives the number of times that TCP connections have made a direct transition to the CLOSED state from either the ESTABLISHED state or the CLOSE-WAIT state. The tcpCurrEstab MIB variable gives the number of TCP connections for which the current state is either ESTABLISHED or CLOSE-WAIT, whereas tcpInSegs gives the total number of segments received, including those received in error. The tcpRetransSegs MIB variable gives the total number of segments retransmitted; that is, the number of TCP segments transmitted containing one or more previously transmitted octets and tcpInErrs MIB variable gives the total number of segments received in error (e.g., bad TCP checksums).

The collected MIB variables are passed through Principal Component Analysis (PCA) for selecting the significant SNMP-MIB variables. It is one of the most widely used dimensionality reduction techniques for data analysis and compression and it is based on transforming a relatively large number of variables into a smaller number of uncorrelated variables by finding a few orthogonal linear combinations of the original variables with the largest variance. After applying PCA, the following four significant MIB variables are selected: tcpActiveOpens, tcpPassiveOpens, tcpCurrEstab and tcpInSegs.

After some exhaustive experiments with normal traffic and botnet communications traffic, we found that summation of the selected MIB variables (SUM-MIB) at different time points provides interesting results and hence that summation is used in our further analysis.

SUM-MIB = tcpActiveOpens + tcpPassiveOpens + tcpCurrEstab + tcpInSegs

8.Hidden semi-Markov Model:

HsMM is an extension of HMM by allowing the underlying process to be a semi-Markov chain with a variable duration or sojourn time for each state. The important difference between HMM and HsMM is that one observation per state is assumed in HMM while in HsMM each state can emit a sequence of observations and the number of observations produced while in a state i is determined by the duration d which is the time spent in the state i as shown in Figure 3.3.

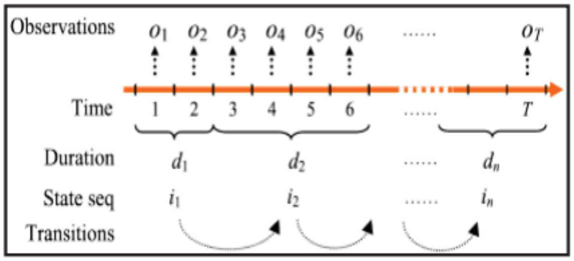


Fig 6 General structure of the HsMM

HsMM can be described as a tuple

$\lambda = (N, M, V, A, B, \Pi)$

where

- N is the size of the state space $\Phi = \{0, 1, 2, \ldots, N - 1\}$ of the hidden semi-Markov chain H_t, $t = 1, 2, \ldots$
- $V = \{v_0, v_1 \ldots, v_{M-1}\}$ is visible symbols, which are nothing but values of the summation count of SNMP-MIB variables.
- M is the number of all visible symbols.
- $A = [a_{ij}]_{N\times N}$ is the state transition probability matrix.
- $B = \{b_i(k)\}$, $i \in \Phi$, $1 \le k \le M$, is the distribution of visible symbols V,

where $b_i(k) = \Pr$ {observed SUM-MIB = v_k | current state i}.

- $\Pi = [\Pi_0, \Pi_1, \Pi_2, \ldots, \Pi_{N-1}]$ is the initial distribution.
- O_t, $t = 1, 2, \cdots, T \in V$ is a sequence of observed visible symbols where T is the number of observed visible symbols.

An HsMM is constructed to build a profile of normal MIB traffic behavior and this model is used to detect the botnet. In this model, only two different states are considered and hence the state space $\Phi = \{0, 1\}$, where 0 represents the normal state of the system and 1 indicates that the system is under the control of the botmaster.

The state transition probability matrix A=

where $a_{ij} = \Pr$ {next_state = j | current state = i}, where i, j € Φ

We assume initially, which means that in a normal process, no matter what current state is, the process will transfer to normal state next time with probability 1.

Let V = $\{v_0, v_1, v_2, \ldots, v_{M-1}\}$ be the set of visible symbols which are nothing but the summation of SNMP−MIB variables at different time points. B =$\{bi(k)\}$, $i \in \Phi$, $1 \le k \le M$, is the distribution of visible symbols.

The initial state distribution $\Pi = [\Pi_0, \Pi_1]$ and it is assumed that $\Pi_0 = 1$and Π_1=0, since the model is constructed for a perfect normal process initially.

9.Experimental Results:

Using the experimental setup, Spyeye, Blackenergy, Zeus, Athena and Andromeda botnets are installed. Zeus, ZeuS, or Zbot is a Trojan horse malware package that runs on versions of Microsoft Windows. While it can be used to carry out many malicious and criminal tasks, it is often used to steal banking information by man-in-the-browser keystroke logging and form grabbing. It is also used to install the CryptoLocker ransomware. Zeus is spread mainly through drive-by downloads and phishing schemes.

Spyeye, the most advanced and dangerous malware kit today, has incorporated the functionalities of the Zeus malware builder kit since early 2011. The Spyeye banking malware continues to plague computers across the world and is proving to be a difficult foe to detect and remove from infected Windows PCs.

Athena is a stable DDoS botnet coded in C++ which is perfect for infecting and herding windows machines. This botnet has advanced DDoS tactics that will take down web servers, gaming servers, VoIP servers and home connections etc.

BlackEnergy is a web-based distributed denial of service (DDoS) botnet used by the Russian hacker underground. BlackEnergy gives the attackers an easy to control web-based bot that can launch various attacks and control the bots using a minimal syntax and structure. The BlackEnergy botnet uses HTTP to communicate to its controlling servers by sending messages to the server.

Andromeda, also known as Win32/Gamarue, is an HTTP based botnet. The Andromeda botnet has a wide reach, which is why many cybercriminals rely on it for distributing malware. In the campaign involving GamaPoS, experts determined that the Point-of-Sale (PoS) malware is downloaded on only 3.8 percent of systems affected by Andromeda. PoS malware is designed to steal payment card data from PoS systems. Since most of the devices infected with Andromeda backdoors are not running any PoS software, it appears that the attackers behind GamaPoS are hoping to catch at least some PoS systems in the large volume of compromised computers.

The feature extraction component is implemented using java and SNMP setup tool is installed in the system. With the help of these components, MIB variables are collected for 12 hrs/day for seven days from the zombie machines. These MIB variables are continuously updated from the outgoing/incoming traffic. Normal traces are also collected during the system's normal activities which include web service, e-mail service, FTP service and remote service and the corresponding MIB variables are collected for 12 hrs/day for ten days from National Knowledge Network (NKN) and from our research lab network. Table 3.1 shows the details of the datasets.

Botnet MIB traces			
Botnet	MIB Trace size	Botnet	MIB Trace size
Spyeye	1.25 GB	Athena	2.73 GB
Blackenergy	2.96 GB	Andromeda	4.59 GB
Zeus	2.57 GB		
Normal MIB traces			
FTP service	4.95 GB	Web service	4.27 GB
E-mail service	3.28 GB	Remote service	2.90 GB

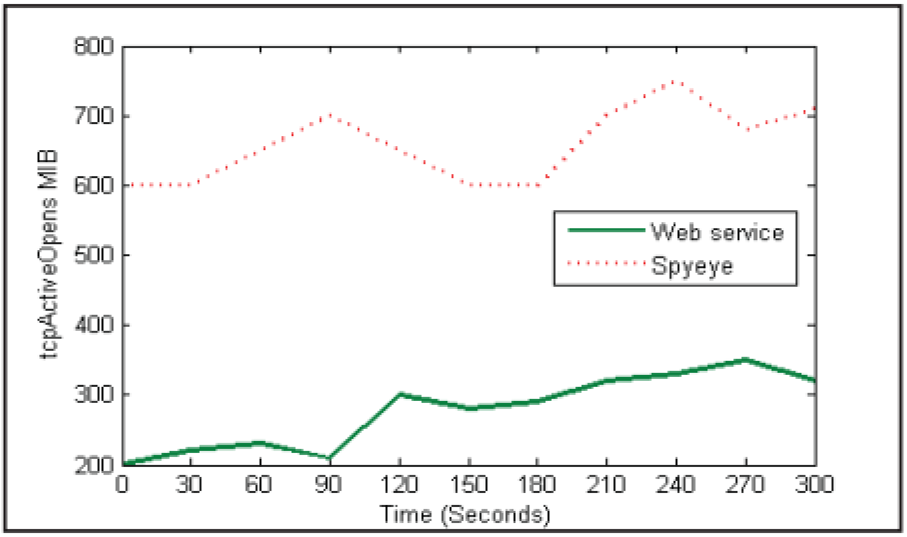


Fig 7 tcpActiveOpens MIB in web service and Spyeye

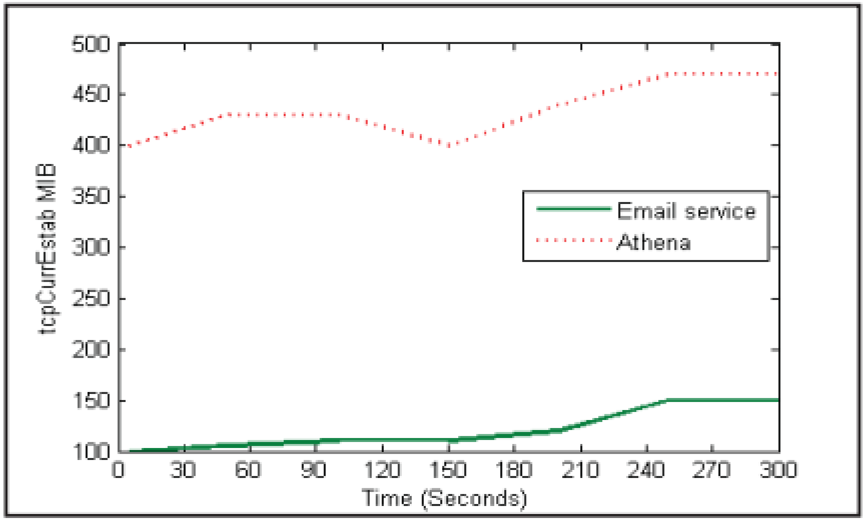


FIG 8 tcpPassiveOpens MIB in FTP service and Blackenergy

10.Conclusion:

In today's network landscape, security threats loom large, presenting a pressing challenge for network management, especially in handling multimedia data. Among these threats, bots have emerged as a particularly insidious form of security attack targeting individuals or groups of network nodes, effectively transforming them into bots. These compromised nodes collectively form a botnet, a network that can be controlled by a bot master. Botnets are notorious for their malicious intent, often programmed to disrupt network activities and collect highly sensitive information, including bank account details and personal data. To combat this menace, intrusion detection systems (IDS) are now employed to efficiently monitor network activities in various organizations and entities. Botnets operate by creating fake or unwanted data, which is then disseminated to all nodes in the network, leading to reduced network efficiency. Their primary objectives are to infect as many devices as possible and propagate malicious programs widely. While botnet attacks pose a significant threat across various technologies, even basic internet security measures such as firewalls and antivirus software offer some protection. However, a dynamic analysis approach is crucial for early detection, seeking signs of infection through behavioral analysis, network traffic anomalies, and individual symptoms of botnet attacks, especially at the network level. In the last decade, botnets have garnered global attention, prompting substantial research efforts to develop effective and efficient detection techniques. This thesis addresses the limitations of conventional botnet detection methods and aims to provide more robust solutions. It encompasses a comprehensive exploration of botnets at various phases to formulate detection mechanisms capable of overcoming the constraints of standard detection approaches, both on Windows and Android platforms. To tackle this unique challenge, the study harnesses the power of machine learning (ML). Machine learning plays a pivotal role in enhancing botnet detection. This thesis offers a concise overview of diverse ML approaches and their specific functions in identifying botnets. The primary objective is to elucidate the precise roles played by various machine learning algorithms in botnet detection. Through this research, a foundation is laid for the development of real-time online detection systems, which are not only more effective but also more robust. In a landscape where cyber threats continually evolve, the insights and knowledge presented here pave the way for more potent defenses against the over-adapting botnet menace.

REFERNCES: -

[1] Botnet. [Internet] 2018 [updated 2018 Mar 5] Available from: https://en.wikipedia.org/wiki/Botnet

[2] A Bijalwan, M. Wazid, E. S. Pilli, and R. C. Joshi, Forensics of random-UDP flooding attacks, Journal of Networks, vol. 10, no. 5, pp. 287-293, 2015.

[3] A K. Soodn, R. J. Enbody, Crimeware-as-a-service—A survey of commoditized crimeware in the underground market, Internation al journal of critical infrastructu protection vol - 6(2013) p 28 – 38.

[4] A. S. da Silva, C. C. Machado, R. V. Bisol, L. Z. Granville and A. Schaeffer-Filho, “Identification and selection of flow features for accurate traffic classification in sdn”, In Network Computing and Applications (NCA), 2015 IEEE 14th International Symposium on, IEEE, (2015), pp. 134-141

[5] Ahlwat, P.. (2018). Detection of botnet based attacks on network: Using machine learning techniques. 10.4018/978-1-5225-4100-4.ch007.

[6] Alparslan, Erdem & Karahoca, Adem & Karahoca, Dilek. (2012). BotNet Detection: Enhancing Analysis by Using Data Mining Techniques. 10.5772/48804.

[7] Arshad S, Abbaspour M, Kharrazi M, & Sanatkar H, (2011). An anomaly-based botnet detection approach for identifying stealthy botnets., In Computer Applications and Industrial Electronics (ICCAIE), 2011 IEEE International Conference on (pp. 564–569). IEEE.

[8] B. Anchit and S. Harvinder, Investigation of UDP Bot Flooding Attack, Indian Journal of Science and Technology, vol. 9, no. 21, 2016.

[9] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer,C. Kruegel, and G. Vigna, Your Botnet isMy Botnet: Analysis of a Botnet Takeover, in ACM Conference on Computer and Communications Security (CCS), November2009.

[10] B., Thanudas & Rajan, Sheena & Sreelal, S. & Bs, Manoj. (2015). A Survey on Botnet Detection Techniques.

[11] Baecher P, Koetter, M, Holz, T, Dornseif M, & Freiling F, (2006). The nepenthes platform: An efficient approach to collect malware. In International Workshop on Recent Advances in Intrusion Detection (pp. 165–184). Springer.

[12] Barazi, Jihan & Jakalan, Ahmad & Wang, XiaoWei. (2014). Botnet Detection Techniques. International Journal of Computer Science and Communication Security (IJCSCS). 4. 61-64.

[13] Barford.P and Yegneswaran.V, “An Inside Look at Botnets,” in Malware Detection, ser. Advances in Information Security, Christodorescu.M, Jha.S, Maughan, Song.D, and Wang.C, Eds. Boston, MA: Springer US, 2007, vol. 27, ch. 8, pp. 171–191. [Online]. Available: http://dx.doi.org/10.1007/978-0- 387-44599-1\ 8

[14] Barthakur P, Dahal M and Ghose M K (2013), An Efficient Machine Learning Based

Classification Scheme for Detecting Distributed Command & Control Traffic of P2P Botnets, I J Modern Education and Computer Science, Vol-10, pp 9-18.

[15] Beigi, Elaheh & Jazi, Hossein & Stakhanova, Natalia & Ghorbani, Ali. (2014). Towards effective feature selection in machine learning-based botnet detection approaches. 2014 IEEE Conference on Communications and Network Security, CNS 2014. 247-255. 10.1109/CNS.2014.6997492.

[16] Belenky.A and Ansari.N, “Ip traceback with deterministic packet marking,” 2003.

[17] Bellovin.S.M, Leech.M, and Taylor.T, “ICMP traceback messages,” Obsolete Internet draft, February 2003. [Online]. Available: http://tinyurl.com/be2sa93

[18] Berger.A and Hefeeda.M, “Exploiting sip for botnet communication,” 2009 5th IEEE Workshop on Secure Network Protocols, pp. 31–36, 2009. [Online]. Available: http://tinyurl.com/8n9rkex