A Framework for Detecting Botnet Command and Control Communication over an Encrypted Channel

Zahian Ismail[1], Aman Jantan[2], Mohd. Najwadi Yusoff[3]
School of Computer Sciences
Universiti Sains Malaysia
Pulau Pinang, Malaysia

***Abstract*—Botnet employs advanced evasion techniques to avoid detection. One of the Botnet evasion techniques is by hiding their command and control communication over an encrypted channel like SSL and TLS. This paper provides a Botnet Analysis and Detection System (BADS) framework for detecting Botnet. The BADS framework has been used as a guideline to devise the methodology, and we divided this methodology into six phases: i. data collection, customization, and conversion, ii. feature extraction and feature selection, iii. Botnet prediction and classification, iv. Botnet detection, v. attack notification, and vi. testing and evaluation. We tend to use the machine learning algorithm for Botnet prediction and classification. We also found several challenges in implementing this work. This research aims to detect Botnet over an encrypted channel with high accuracy, fast detection time, and provides autonomous management to the network manager.**

***Keywords*—Botnet; Botnet Analysis and Detection System (BADS); encrypted channel; machine learning; accuracy; autonomous**

I. Introduction

Botnet has become a significant concern in the computer industry. With users engaging in daily life surfing to the Internet, there was a high risk of becoming a victim of a Botnet attack. The botnet has developed many capabilities, but unfortunately, most of those capabilities are used for attack purposes, such as performing a DDoS attack, spamming, malware spreading, and large computer compromising. Launching a massive DDoS attack is one of the main capabilities of Botnet. For instance, a DDoS attack that happened in the year 2000 is one of the notorious DDoS attacks when the cyber-criminal targeting Yahoo!, Fifa.com, Amazon.com, Dell, Inc., E*TRADE, eBay, and CNN. Another main capability of Botnet is spamming. Several authors like Solomon and Evron (2006) [1] highlight Botnet spamming as a significant concern because of the large amount of distribution of spam, which will use many network resources. Their concern was supported by McAfee Avert Labs [2] which stated that more than 70 percent of spam email caused by Botnet.

Other than DDoS and spamming, Botnet vigorously compromised many computers and tried to develop a vast network of the infected machine through its command and control (C&C) communication. Thus the impact of the attack is enormous. Therefore, Botnet is becoming an increasingly widely-used method by cybercriminals for many purposes, such as gaining recognition from other hackers, financial gain, and many other nefarious activities; hence, all of these affect the users in general. The Botnet is also making antivirus tools ineffective, and bots able to modify registry entries, so they remain active even when the infected machine is booted in a safe mode. Some of the Botnet even respond vigorously if they notice there are efforts made trying to detect their presence.

Considering such capabilities deployed in many Botnets, the effects of Botnet attack are so huge. Botnet brings high risk to national security, intimidates the security of many organizations, either public or private entities, especially caused terrible disturbance and high usage of network resources. Cleaning on the detected system will be very difficult because the volume of network traffic created by bots is massive, thus making it impossible to perform an update on an infected machine (Thomas & Jyoti, 2007) [3]. For this reason, even governments have to spend much money to prevent Botnet attacks. CyberSecurity Malaysia [4] provides the statistic of Botnet attacks in Malaysia, as depicted in Fig. 1. Table I shows Botnet attacks in comparison to other attacks in Malaysia for six consecutive years. Botnet employs many evasion techniques to avoid detection and stay in the network. One of the evasion techniques is by manipulating encrypted channel like SSL and TLS to hide their C&C activities. Zhang (2017) [5] refers Botnet that uses encryption evasion techniques as an advanced Botnet. Burghouwt (2015) [6] states the encryption of C&C traffic as the most crucial evasion technique by Botnet. The Botnet dependency to this evasion technique is due to several reasons; for instance, the increasing number of services and applications that use an encrypted channel to secure the communications and contents.

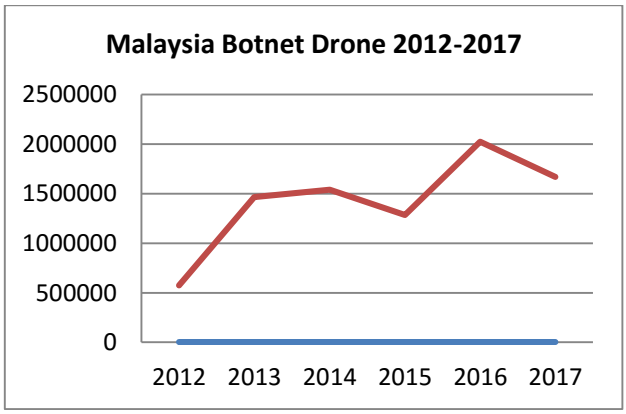

Fig. 1. Malaysia Botnet Statistics 2012-2017 (MyCert, 2017).

TABLE. I. TREATS COMPARISON IN MALAYSIA FROM 2012-2017

Year	DoS	Mal code	Spam	Intrusion	Fraud	Botnet
2012	23	645	526	4326	4001	573401
2013	19	1751	950	2770	4485	1465785
2014	29	716	3650	1125	4477	1539734
2015	38	567	3539	1714	3257	1285605
2016	64	375	518	2328	3612	2026276
2017	40	814	344	2011	3821	1669973

Furthermore, according to Nicholson (2015) [7], social media like Pinterest and Twitter, and email applications use SSL/TLS to encrypt all communications. These social media have many users, thus making them a good target for the attacks because of the potential to compromise the massive host. Moreover, web traffic is allowed almost everywhere. Botnet strains, for example, Storm, Waledac and Rustock, use social media and email applications for their C&C communication to evade detection and stealthily distribute the command and control bots in the network. Hence, Paganini (2014) [8] states that while SSL/TLS encryption improves privacy and integrity, Botnet uses SSL blind spot to avoid detection and leave their C&C covert for as long as possible.

Even though the reports by CyberSecurity Malaysia in Fig. 1 and Table I did not directly state that the Botnet attacks have been performing through Secure Socket Layer (SSL) or any encrypted channel, we consider another report by Gebhart from Electronic Frontier Foundation (EFF) [9] which stated that by 2017 more than 50% of Internet traffic had been protected by HTTPS. The effort of turning the traffic into encrypted one was enthusiastically made since 2010. Therefore, the botmasters are taking advantage of this situation to hide their operation and evade detection. Botnet itself creates a massive impact, and with the implementation of advanced evasion techniques like masquerading in the SSL channel will amplify the impacts. These scenarios have shown the severity of encrypted Botnet attack and therefore become an encouraging factor for developing solutions to detect Botnet over an encrypted channel even though there are other network attacks encrypted in SSL/TLS-enabled protocol.

Additionally, according to Nicholson (2015) [7] and Finley (2017) [10], roughly more than half of all traffic is encrypted mostly by SSL/TLS. By the end of 2016, Gooley (2017) [11] states that 80% of traffic across Google properties was encrypted, and 54% of threats Zscaler blocked are hidden inside SSL traffic. Therefore the use of SSL for the distribution of malicious content is rising too. Recent Botnet strains manipulated this situation and use SSL/TLS channel for their command and control communication. SSL/TLS protects legitimate content but simultaneously provides Botnet with hiding spots, making encrypted channel beneficial for a good guy and bad guy. There are three most active malicious contents referred by Gooley which are Dridex, Vawtrak, and Gootkit; all of them are variants of Botnet, which commonly associated with user credential stealing. Moreover, Rossow & Dietrich (2013) report that detailed C&C traffic analysis shows at least ten prevalent malware families avoided well-known C&C carrier and preferred encrypted channel, and one of them is Zeus. Consequently, the rising of SSL-encrypted traffic increases the Botnet attack trough SSL/TLS channel.

Despite the advancement of Botnet technology and fast evolution, research in finding the solution for Botnet detection is still in its infancy because existing studies remain somewhat limited in scope and do not generally include recent research and development (Silva et al. 2012) [13]. Thus, this indicates that there is still room for improvement in Botnet detection, uniquely encrypted Botnet. Many Botnet detection techniques are based on payload analysis, and these techniques, unfortunately, are inefficient for encrypted C&C channels (Shanti & Seenivasan, 2015) [14]. Zhang et al. (2013) [15] prove that Botnet detection techniques that rely on payload analysis could be foiled by encryption. Payload-based analysis requires decryption, and this leads to a privacy issue. Zhao et al. (2013) [16] state that several challenges in Botnet detection remain unaddressed, such as the ability to design detectors which can cope with new forms of Botnet, therefore they proposed the use of machine learning techniques which proven to increase detection accuracy even for dynamic forms of Botnet. Furthermore, Botnet detection approaches for encrypted traffic were not well established, for instances limited signatures, limited features extracted, limited Botnet detected, and insufficient alarm mechanism (Zhao et al., 2013 [16]; Bortolameotti, 2014 [17]; Larinkoski, 2016 [18]).

The purpose of the study is to propose an approach to detect Botnet in the encrypted channel. The solution was devised to secure the gaps in encrypted Botnet detection system especially for the Botnet detection that base on payload analysis. This study will benefit the system administrator as the detection system assist them in monitoring and protecting system security. This research tends to explore the potential of machine learning techniques which expected to produce a detection system with high accuracy, fast detection and autonomous. The autonomous feature provides minimum supervision and self-learning. The findings also will benefit researchers in this area as it opens up to the exploration of the possible machine learning techniques in developing an effective and efficient Botnet detection system.

The implementation of machine learning in Botnet detection is compelling to overcome the limitation in Botnet detection. Cha & Kim (2017) [19] state that despite the limitations of encrypted Botnet C&C detection, machine learning is a promising approach to detect encrypted Botnet C&C communication. In practice, there are several machine learning techniques implemented in Botnet detection system (Bilge et al. 2011 [20]; Chandhankhede 2013 [21]; Guntuku et al. 2013 [22]; Roshna & Ewards 2013 [23]; Hyslip & Pittman 2015 [24]; Ritu & Kaushal 2015 [25]); however there are still gaps to be fulfilled for the research in detecting encrypted Botnet such as inadequacy of detection features used. In order for the system to achieve a high detection rate and fast detection, it requires techniques that offer high precision and fast pattern recognition capabilities. Autonomous in the detection system requires decision support, situation awareness, and knowledge management. Therefore, this research tends to look at any potential machine learning techniques to fulfill the detection system requirements. In general, machine learning has been proven by previous

researches as being able to solve issues like accuracy (Salvador et al. 2009 [26]; Al-Hammadi 2010 [27]; Bilge et al. 2011 [20]; Guntuku et al. 2013 [22]; Ritu & Kaushal 2015 [25]) and real-time (Salvador et al. 2009 [26]; Guntuku et al. 2013 [22]) in Botnet detection.

We organized the remainder of this paper as follows. In section 2, we provide the related work of encrypted Botnet C&C detection. Then in section 3, we propose the Botnet detection framework to detect Botnet over the encrypted channel. This framework will be used to devise the methods based on several phases. Section 4 highlights the challenges for the implementation of the proposed solution. Finally, we drew some concluding remarks in section 5.

II. Related Work

Many researchers relied on a payload-based analysis (deep packet inspection) to detect encrypted Botnet C&C. For example, Zhang et al. (2013) [15] develop high entropy detectors and analyzed packets based on the determined threshold. They stated that the encrypted Botnet produces high entropy, and it can be detected by using the detectors. The challenge of this approach is how to differentiate entropy produces by encrypted Botnet with other traffic that produces high entropy, for instance, media, executable, and compressed files. Tyagi et al. (2015) [28] also implement deep packet inspection (DPI) in their approach and proposed N-gram based HTTP bot traffic detection. The proposed technique detects encrypted and regular Botnet. This technique was based on the fact that the C&C responds with similar communication patterns, with only slight modifications to an HTTP GET request made by a bot. The communications patterns did not varied unless the bot is updated. Therefore this technique works appropriately only if the bot is not updated.

Other work on deep packet inspection was by Sherry et al. (2015) [29], which propose BlindBox to perform deep packet inspection directly on the encrypted traffic. They demonstrate that BlindBox enables applications such as IDS, exfiltration detection, and parental filtering and supports real rule sets from both open-source and industrial DPI systems. They also implement BlindBox and show that it is practical for settings with long-lived HTTPS connections. However, this approach is not specially designed to detect Botnet over the encrypted channel but only stated Botnet as one of their possible usage scenarios. Therefore there was the possibility that this approach might not work well for Botnet detection. Differently, Burghouwt (2015) [6] uses a Causal analysis of traffic flows to detect covert Botnet, for example, Botnet that hides in the encrypted channel. This approach detected covert Botnet by identifying the direct Causal relationship between network flows and prior events. However, this technique needs user events in addition to network traffic; therefore, it causes deployment complexity. Another researcher that used this method is Zhang (2017) [5].

Instead of deep packet inspection, some researchers use decryption techniques to detect encrypted Botnet C&C, for example, Rossow & Dietrich (2013) [12]. They propose PROVEX, a system that automatically derives probabilistic vectorized signatures. PROVEX learns characteristic values for fields in the C&C protocol by evaluating byte probabilities in C&C input traces used for training. This way, they identify the syntax of C&C messages without the need to specify C&C protocol semantics manually, but purely based on network traffic. Even though PROVEX can detect all studied malware families, the fact that it used payload-based analysis that depends on the studied signature limits the detection to the known bots only. Furthermore, by implementing a brute-force-like decryption technique, it leads to the privacy issue.

Some researchers claim that their general Botnet detection approaches could even detect encrypted Botnet C&C based on the assumption that their approaches did not analyze the payload content, and it was Botnet structure independent. For example, Shin et al. (2012) [30], Khan et al. (2015) [31], and Shanti & Seenivasan (2015) [14] commonly use traffic flow analysis in their works. Consequently, they did not have to inspect the payload. However, only Shin et al. provide the detection result of encrypted Botnet C&C even though it was not that prominent. On the other hand, Shanti & Seenivasan provide Botnet detection results in general.

Many researchers implement a machine learning algorithm (MLA) and data mining techniques in their proposed approaches, which can detect Botnet over the encrypted channel. For example, Warmer (2011) [32], Dietrich et al. (2013) [33], Tegeler et al. (2012) [34], Bortolameotti (2014) [17], Wang (2014) [35], Buriya et al. (2015) [36], Cha & Kim (2016) [19] and Jianguo et al. (2016) [37] are some of the researchers that leverage machine learning. Even so, some of them also use deep packet inspection, for instance, Tegeler et al., Wang, and Cha & Kim. Tegeler et al. propose BOTFINDER that uses MLA to identify the key features of Botnet based on the observing traffic that bots produced in the controlled environment. Cha & Kim propose a machine learning approach with several randomness tests to achieve high accuracy detection of encrypted traffic while requiring low overhead incurred by the detection procedure. They test their approach using four MLAs for classification and recommence CART which produced 99.9% accuracy and 2.9 times more efficient than second-best MLA (Naïve Bayes). Wang proposes a novel meta-level classification algorithm based on content features and flow features of traffic. Then he use Naive Bayes classification algorithms to detect encrypted Botnet traffic.

Saad et al. (2011) [38] study the ability of five different commonly used MLAs to meet online Botnet detection requirements, namely adaptability, novelty detection, and early detection. All five MLAs provide high true positive value; however Support Vector Machine got the highest true positive value, which is 97.8%. Warmer (2011) [32] compare different techniques and based on the result proposes three new techniques for detecting HTTP and HTTPS-based C&C channel. It shows Naïve Bayes got the highest true positive which is 97.3%. Ritu & Kaushal (2015) [25] compare different supervised MLAs for determining peer to peer Botnet detection accuracy. Decision Tree and Support Vector Machine achieved 100% accuracy.

Shanti & Seenivasan (2015) [14] propose a detection methodology to classify bot hosts from the normal host by analyzing traffic flow characteristics based on time intervals

instead of payload inspection. They use the Decision Tree and Naïve Bayes classification. Classification with a decision tree gave a better true positive of 86.69%. Kirubavathi & Anitha (2016) [39] propose an approach to detect Botnet irrespective of their structures. They try several MLAs to their approach, and Naïve Bayes has the highest detection rate of 99.14%.

Zhao et al. (2013) [16] and Bortolameotti (2014) [17] use Decision Tree to their approaches, and both provide a very high detection rate which is 98.5 % and 99.96% with a very low false positive rate of 0.01 % and 0%. Dietrict et al. (2013) [33] develop CoCoSpot use Average-Linking Hierarchical Clustering. 50% of Botnet families were detected by the rate of 95.6%. Buriya et al. (2015) [36] use Naïve Bayes and achieved 98.84% accuracy. Apparently most of the MLAs discussed in this paper have a very high detection rate.

Even though some techniques provide a high detection rate, comparatively they also got a high false positive rate. For example, Richer (2017) [40] proposes an approach using Support Vector Machine and got a 100% detection rate; however the false positive rate is more than 15%. The work by Shanti & Seenivasan (2015) [14] also provides very high false positive which is more than 21%. Above all, Warmer (2011) had the highest false positive value of 44.3% by using Naïve Bayes.

Al-Hammadi (2010) [27] presents a host-based behavioral approach for detecting Botnet based on correlating different activities generated by bots by monitoring function calls within a specified time window. Al-Hammadi uses Dendric Cell Algorithm inspired by the Immune System. The evaluation shows that correlating different activities generated by IRC/P2P bots within a specified period achieves high detection accuracy (100%). In addition, using an intelligent correlation algorithm not only states if an anomaly is present, but it also exposed the source of the anomaly.

One of the most prominent MLA for Botnet detection is Neural Network and its distributions, Self-Organizing Map (SOM). SOM is an unsupervised Neural Network and has been widely used in intrusion detection. Unfortunately, there are limited works discussing SOM for Botnet detection, and instead, more in intrusion detection. However, SOM is a promising approach especially for developing an autonomous Botnet detection system. Langin et al. [i] (2009) [41] use SOM to cluster and classify peer to peer Botnet traffic and other malignant network activity by analyzing firewall log entries. Langin et al. [ii] (2009) [42] use Hexagonal SOM for clustering and then for classification of new firewall log data to look for new bots in the network.

Guntuku et al. (2013) [22] propose and implement a hybrid framework for detecting peer to peer Botnet in live network traffic by integrating Neural Networks with Bayesian Regularization for the detection of newer and unseen Botnet in live traffic of a network. It was conclusively shown through the statistical tests that the trained Bayesian Regularization - Neural Network model can generalize very well and can predict the activity of unknown bots' malicious activity. Thus Botnet detection activities successfully achieved with an accuracy of 99.2%. Nogueira et al. (2010) [43] extend the framework propose by Salvador and develop the Botnet Security System called BoNeSSy. Nogueira et al. develop a Botnet detection system that is based on the collection of flow statistics using Neural Network. The results obtained show that the system is feasible and efficient since it provides high detection rates with low computational overhead.

Many existing approaches to the process of detecting intrusions utilized some forms of rule-based analysis. Expert System is the most common form of rule-based intrusion detection approaches. Most existing behavior-based approaches are not able to detect and predict the Botnet as they change their structure and pattern. Roshna & Ewards (2013) [23] present the AdaptiveNeuro-Fuzzy Inference System (ANFIS), a technique that trains the system for future prediction. However, the limitation of this work is the restriction of fuzzy rules and fuzzy sets for the comparison purpose. Therefore, the proposed work should be able to overcome the limitations by increasing the number of rules generated using the Botnet features and information gain.

Fuzzy pattern recognition proposed by Wang et al. (2011) [44] intends to identify bot-relevant domain names and IP addresses by inspecting network traces. The algorithm involves traffic reduction, feature selection, and pattern recognition. Fuzziness in pattern recognition helps to detect bots that are hidden or camouflage. Performance evaluation results based on real traces show that the proposed system can reduce more than 70% input raw packet traces and achieve a high detection rate (about 95%) and a low false positive rate (0–3.08%). Furthermore, the proposed FPRF algorithm is resource-efficient and can identify inactive Botnet to indicate potentially vulnerable hosts. BotDigger proposed by Al-Duwairi &Al-Ebbini (2010) [45] utilizes fuzzy logic in order to define logical rules that are mainly based on some statistical facts and essential features that identify Botnet activities. The key advantage of the architecture designed in this research is that it allows the integration of a wide range of traffic specifications.

The above machine learning approaches mostly use detection rate, accuracy, or false positive value as the metrics to measure the detection performance. However, other vital metrics are real-time and autonomous. Even though detection able to detect accurately, it is useless without fast detection or real-time detection. Researchers focus on developing a real-time Botnet detection system, for example, Salvador et al. (2009) [26], Wang et al. (2011) [44] and Guntuku et al. (2013) [22].

Autonomous mainly focus on self-learning and self-managing properties. Chandhankhede (2013) [21] proposes the new autonomous model for Botnet detection using the K-means algorithm, one of the most straightforward unsupervised learning algorithms that solve the well-known clustering problem. According to Khattak et al. (2014) [46], the degree of automation can be classified as manual, semi-automated and automated. Semi-automated Botnet detection requires very little human intervention, and most of the detection is performed on automated fashion. However, fully automated Botnet detection should require no human intervention after initial development. Khattak et al. also agree that ideally, any detection method should be as generic and automated as possible.

III. Proposed Botnet Detection Framework

To achieve high accuracy, fast detection, and an autonomous Botnet detection system as stated in section I, we propose a conceptual framework as a guideline to devise a methodology to detect Botnet over the encrypted channel. Fig. 2 shows the Botnet Analysis and Detection System (BADS) framework which consists of three main components namely Network Analysis System (NAS), IDS and Alarm System (AS). Through BADS, we generally divide the process into six phases as depicted in Fig. 3 also shows the expected results of each phase.

Phase 1: Data Collection, Customization and Conversion

This study requires the dataset of encrypted Botnet traffic; however, in the secure network, it is a challenge to get one. Therefore, we convert public Botnet datasets into customizable encrypted Botnet dataset by using BotTalker developed by Zhang & Papadopoulos (2013) [47]. The public datasets use are ISOT, Malware Capture Facility Project (MCFP), and Network Information Management and Security (NIMS). Some datasets are in pcap format; therefore, we need to convert the dataset into CSV format using Wireshark, then to arff using Weka.

Phase 2: Feature Extraction and Selection

In encrypted Botnet, detection by inspecting the payload is a tedious process, especially if it involves extensive traffic data. Furthermore, this method required the decryption of data, and it involved a privacy issue. Because of that, for this research, the approach without inspecting the payload is necessary. Encrypted Botnet itself produces features that can be used for detection, and most importantly, it should not require any decryption. Botnet features are extracted through a feature extraction process, and then if necessary, followed by feature selection. Feature selection reduces the number of features, and these selected features are the most relevant features for the detection. Feature selection is crucial because it is a reliant factor to detection accuracy, as proven by Buriya et al. (2015) [36] and Kirubavathi & Anitha (2016) [39].

Tranalyzer is used to extract the Botnet features. From the literature study, Tranalyzer was proven to capture more features compared to other features extractors. Furthermore, features extracted using Tranalyzer provide better accuracy (Jianguo et al., 2016) [37]. For feature selection, we use Information Gain Attribute Evaluation in Weka and employ Ranker Algorithm to select the features that will give the most relevant features based on the ranking provided.

Phase 3: Botnet Prediction and Classification

We use Weka classify module to perform Botnet prediction and classification. The classification generates decision rules, and these rules are used for Botnet detection.

Phase 4: Botnet Detection

IDS component consists of a Snort-based Botnet detection mechanism, as shown in Fig. 4. Sensors sniff the packets from the network. Packet decoder takes packets from different types of network interfaces and prepares the packets to be preprocessed. The preprocessor arranges or modifies data packets before the detection engine does some operation. It also normalizes protocol headers, detects anomalies, packet re-assembly, and TCP stream reassembly. The detection engine is an essential part of IDS. The detection engine detects Botnet intrusion activity that exists in a packet. There are two detectors use; misuse detector and anomaly detector. The detection engine employs fuzzy inference rules for this purpose. The rules are read into internal data structures or chains where they are matched against all packets. If a packet matches any rule, appropriate action is taken; otherwise, the packet is dropped. The idea of fuzzy rules and fuzzy inference implementation in the detection engine is to determine the severity level of a Botnet attack.

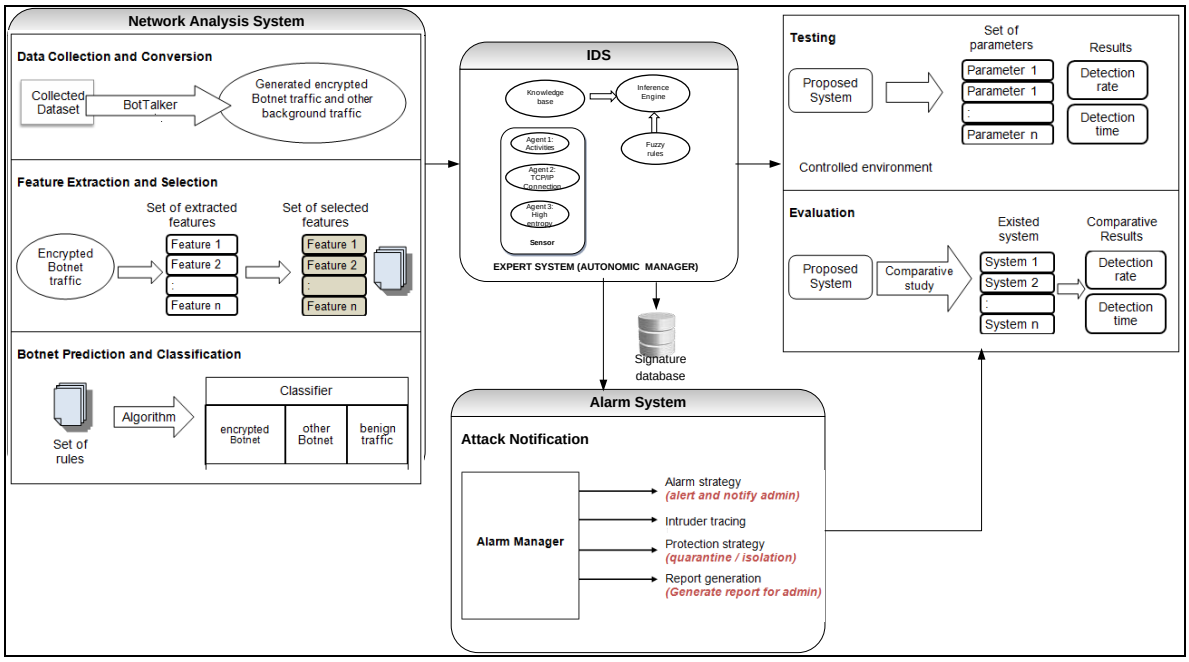

Fig. 2. Botnet Analysis and Detection System (BADS) Framework.

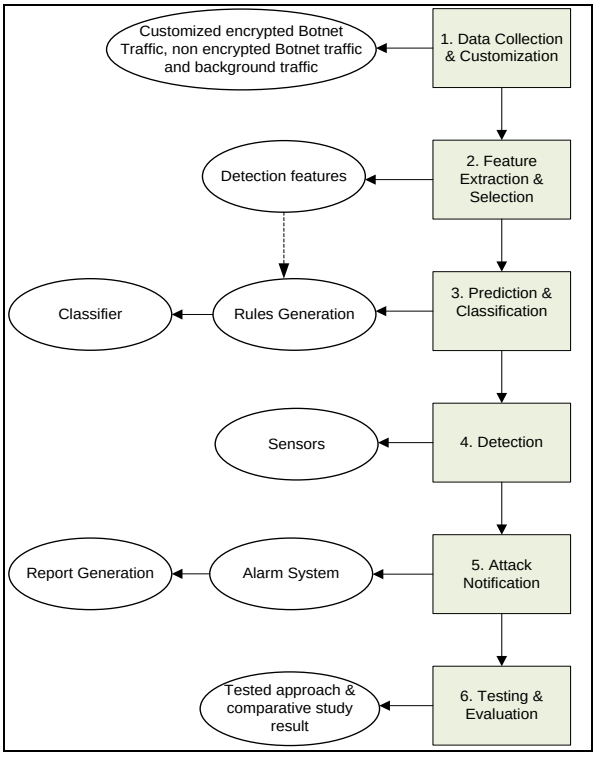

Fig. 3. Phases in Detecting Encrypted Botnet.

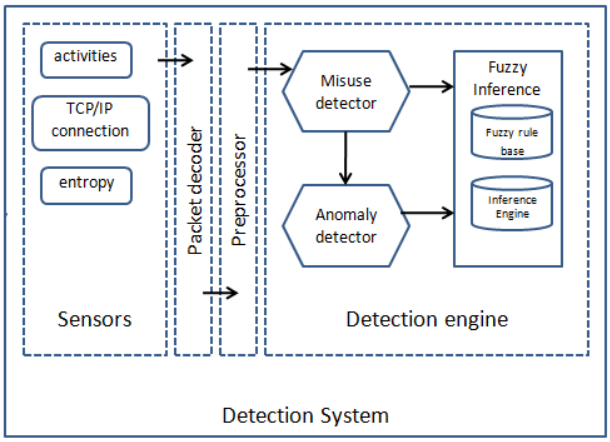

Fig. 4. Fuzzy Inference Snort-Based Detection System.

Phase 5: Attack notification

Intrusion notification assists the network manager to manage the system. There are four sub-components to fulfill that purpose namely intruder tracing, alarm strategy, protection strategy, and report generation. Implementing these components into the system should be able to alert the network manager and notify them of the severity of attacks, suggesting protection strategy, and generate a report. The autonomous mechanism enables the Botnet detection system to works effectively with minimum human intervention.

Phase 6: Testing and Evaluation

The open stack test-bed is set up by using multiple virtual machines. Several virtual machines are used to carry out intrusion attempts and one virtual machine to run the proposed Botnet detection system. This phase happens to test the effectiveness of the proposed approach by using several parameters, for instance, accuracy and false positive. Then, for the evaluation, the comparison study is performed between the proposed approach and other existing Botnet detection system to measure the efficiency of the approach.

The BADS assists the network manager in monitoring the security of the system. The idea of BADS is to minimize human intervention in performing network monitoring and suppose to take appropriate action based on the severity of the Botnet attack. Therefore we endeavor to propose an autonomous Botnet detection system.

IV. Challenges of Implementation

There are several challenges to implement BADS. Firstly, it is data preprocessing parts, which are dataset collection, dataset customization, and conversion. Actually, for this work, we also want to use the Botnet dataset from IMPACT Cyber Trust. However, the dataset is only available to US-based researchers and those in approved locations. Unfortunately, our location is not in that approved locations. Another challenge is for data customization as the reference for BotTalker is quite limited. Then, we have to do data conversion for all the datasets except for the NIMS dataset. Overall, the data preprocessing part is a tedious part and requires many works.

Since we are using various tools in our work, we are expecting conflict because each tool produces different types of outputs. For example, the rules that are retrieved from the classification in Weka to the rules structure in Snort. Furthermore, we also want to employ fuzzy rules into Snort because we want the detection system to be able to determine the severity of Botnet attacks. We try to achieve this because we want the detection system to provide appropriate solutions based on the level of a Botnet attack. This feature will help the network manager to monitor the network and provides automation. We believe fuzzy inference rules can provide the required solution. Currently, we are still looking for solutions to this issue.

V. Conclusion and Future Work

Botnet evolves, and new Botnet strains have developed advanced evasion techniques. It includes the capabilities to manipulate encrypted channels like SSL/TLS for their command and control communication, use social media to spread malware, spamming, and gain credential info (social bot). These avoidance techniques enable Botnet to cover its operation, evade detection, and stay on the system as long as possible. However, existing detection techniques were not well established and had limitations in detecting Botnet especially the Botnet over the encrypted channel. Having an effective and efficient Botnet detection system is essential. This research endeavors to find a solution to enhance the Botnet detection system over the encrypted channel by using machine learning. Machine learning is a promising approach to detect Botnet, especially over an encrypted channel. Therefore, we proposed the BADS framework and devised a methodology based on the framework.

This framework consists of three main components, which are Network Analysis System (NAS), IDS, and Alarm System (AS). Besides the main components, testing and evaluation processes also included in the framework. We devise the methodology from the framework and divide them into six phases. Overall the contribution of this paper is three-fold:

1) The framework of Botnet Analysis and Detection System (BADS).

2) The methodology of devising the techniques for Botnet detection.

3) The design of fuzzy inference Snort-based Botnet detection system.

ACKNOWLEDGMENT

This research is supported by the Universiti Sains Malaysia Research University (RUI) Grant (account number 1001/ PKOMP/ 8014017).

REFERENCES

[1] Soloman, A. & Evron, G., The World of Botnet, Virus Bulletin, September 2006, www.beyondsecurity.com/whitepapers/Solomon EvronSept06.pdf.

[2] Security Press Release, Botnets Threaten National Infrastructure and Security, 24 October 2006, http://www.itsecurity.com/press-releases/press-release-mcafee-botnets-102506/.

[3] Thomas, V. & Jyoti, N., Virus Bulletin, Defeating IRC Bots on the Internal Network, 1 February 2007, https://www.virusbulletin.com/ virusbulletin/2007/02/defeating-irc-bots-internal-network.

[4] MyCERT Incident Statistics, https://www.mycert.org.my/statistics/2018 .php.

[5] Zhang, H., Detecting Advanced Botnets in Enterprise Networks, Ph.D. Thesis, 2017, Department of Computer Science, Colorado State University, USA.

[6] Burghouwt, P., Detection of Botnet Command and Control Traffic in Enterprise Networks, Ph.D. Thesis, 2015, The Hague University of Applied Science, Netherlands.

[7] Nicholson, P., What Lies Beneath: Advanced Attacks that Hide in SSL Traffic, September 30, 2015, https://www.a10networks.com/blog/what-lies-beneath-advanced-attacks-hide-ssl-traffic.

[8] Paganini, P., SSL Blacklist A New Weapon to Fight Malware and Botnet, http://securityaffairs.co/wordpress/26672/cyber-crime/ssl-blacklist-new-weapon-fight-malware-botnet.html, 2014 access on 3 August 2017.

[9] Gebhart, G., We Are Halfway to Encrypting The Entire Web, Electronic Frontier Foundation (EFF) Report, 2017, https://www.eff.org/deeplinks /2017/02/were-halfway-encrypting-entire-web.

[10] Finley, K., Half of the Web is Now Encrypted. That Makes Everyone Safer, 2017, https://www.wired.com/2017/01/half-web-now-encrypted-makes-everyone-safer/.

[11] Gooley, D., The Rise in SSL-based Treats, 2017, https://www.zscaler.com/blogs/research/rise-ssl-based-threats.

[12] Rossow, C. & Dietrich, C.J., (2013). ProVeX: Detecting Botnets with Encrypted Command and Control Channels, International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA), 18-19 July 2013, Berlin, Germany, pp 21-40.

[13] Silva, S. S. C., Silva, R. M. P., Pinto, R. C. G., & Salles, R. M. (2013). Botnets: A survey. Computer Networks, 57(2), 378–403. https://doi.org/10.1016/j.comnet.2012.07.021.

[14] Shanti, K. & Seenivasan, D., (2015). Detection of Botnet by Analyzing Network Traffic Flow Characteristics using Open Source Tools, 2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO), 9-10 January 2015, Andhra Pradesh, India, DOI: 10.1109/ISCO.2015.7282353.

[15] Zhang, H., Papadopoulos, C., & Massey, D., (2013). Detecting Encrypted Botnet Traffic, 2013 IEEE INFOCOM, 14-19 April 2013, Turin, Italy, pp 3453-3458.

[16] Zhao, D., Traore, I., Sayed, B., Lu, W., Saad, S., Ghorbani, A., & Garant, D. (2013). Botnet Detection based on Traffic Behavior Analysis and Flow Intervals. Computers and Security, 39(PARTA), 2–16. https://doi.org/10.1016/j.cose.2013.04.007.

[17] Bortolameotti, R. (2014). C&C Botnet Detection over SSL, Master Thesis, University of Twente, Netherlands.

[18] Larinkoski, L. (2016). Detecting Encrypted C&C using Network Fingerprints

[19] Cha, S., & Kim, H. (2017). Detecting Encrypted Traffic: A Machine Learning Approach (pp. 54–65). Springer, Cham. https://doi.org/10.1007/978-3-319-56549-1_5.

[20] Bilge, L. (2011). EXPOSURE : a Passive DNS Analysis Service to Detect and Report Malicious Domains, V(4).

[21] Chandankhede, P. (2013). Autonomous Botnet Detection, *3*(13), 71–76

[22] Guntuku, S., Narang, P., & Hota, C. (2013). Real-time Peer-to-Peer Botnet Detection Framework based on Bayesian Regularized Neural Network. ArXiv Preprint ArXiv:1307.7464. Retrieved from http://arxiv.org/abs/1307.7464.

[23] Roshna, R. S., & Ewards, V. (2013). Botnet Detection Using Adaptive Neuro Fuzzy Inference System, *3*(2), 1440–1445.

[24] Hyslip, T., & Pittman, J. (2015). A Survey of Botnet Detection Techniques by Command and Control Infrastructure. Journal of Digital Forensics, Security and Law, 10(1), 7–26. https://doi.org/10.1145 /1090191.1080118.

[25] Ritu & Kaushal R. (2015), Machine Learning Approach for Botnets Detection, 3rd Security and Privacy Symposium, 13-14 February 2015, IIIT – Delhi..

[26] Salvador, P., Nogueira, a., Franca, U., & Valadas, R. (2009). Framework for Zombie Detection Using Neural Networks. 2009 Fourth International Conference on Internet Monitoring and Protection, 14–20. https://doi.org/10.1109/ICIMP.2009.10.

[27] Al-Hammadi, Y. A. A. (2010). Behavioural correlation for malicious bot detection. Doctor, (April). Retrieved from http://etheses.nottingham .ac.uk/1359/.

[28] Tyagi, R., Paul, T., Manoj, B. S., & Thanudas, B. (2015). A novel HTTP botnet traffic detection method. 12th IEEE International Conference Electronics, Energy, Environment, Communication, Computer, Control: (E3-C3), INDICON 2015, 1–6. https://doi.org/10.1109/INDICON. 2015.7443675.

[29] Sherry, J., Lan, C., Popa, R. A., & Ratnasamy, S. (2015). BlindBox: Deep Packet Inspection over Encrypted Traffic. Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication - SIGCOMM '15, 213–226. https://doi.org/10.1145/2785956.2787502.

[30] Shin, S., Xu, Z., & Gu, G. (2012). EFFORT: Efficient and effective bot malware detection. Proceedings - IEEE INFOCOM, 2846–2850. https://doi.org/10.1109/INFCOM.2012.6195713.

[31] Khan, A., Ahlawat, C., & Bijalwan, A. (2015). A unified botnet detection framework 1, International Journal of Advances in Electronics and Computer Science (March), Vol. 2, Issue 5, pp 81-87.

[32] Warmer, M. (2011). Detection of web-based command & control channels, (November). Master Thesis, University of Twente, Netherlands. Retrieved from http://essay.utwente.nl/61232/.

[33] Dietrich, C. J., Rossow, C., Pohlmann, N. (2013), CoCoSpot: Clustering and Recognizing Botnet Command and Control Channels using Traffic Analysis, Journal of Computer Networks, Vol. 57, Issue 2, pp. 475-486.

[34] Tegeler, F., Fu, X., Vigna, G., & Kruegel, C. (2012). BotFinder: Finding Bots in Network Traffic Without Deep Packet Inspection. Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies - CoNEXT '12, 349. https://doi.org/10.1145/2413176.2413217.

[35] Wang, Y. (2014). Encrypted botnet detection scheme. Proceedings - 2014 9th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 3PGCIC 2014, 559–565. https://doi.org/ 10.1109/3PGCIC.2014.110.

[36] Buriya, S., Patel, A. K., Yadav, S. S., Buriya, S., Patel, A. K., & Yadav, S. S. (2015). Botnet Behavior Analysis Using Naïve Bayes Classification Algorithm Without Deep Packet, *IX*(Viii), 45–54.

[37] Jianguo, J., Qi, B., Zhixin, S., Wang, Y., & Lv, B. (2016). Botnet Detection Method Analysis on the Effect of Feature Extraction, 1884–1890. https://doi.org/10.1109/TrustCom.2016.286.

[38] Saad, S., Traore, I., Ghorbani, A., Sayed, B., Zhao, D., Lu, W., … Hakimian, P. (2011). Detecting P2P botnets through network behavior analysis and machine learning.2011 Ninth Annual International Conference on Privacy Security and Trust, 174–180. https://doi.org/10.1109/PST.2011.5971980.

[39] Kirubavathi, G., & Anitha, R. (2016). Botnet detection via mining of traffic flow characteristics. Computers and Electrical Engineering, 50, 91–101. https://doi.org/10.1016/j.compeleceng.2016.01.012.

[40] Richer, T. J. (2017). Entropy-based detection of botnet command and control. Proceedings of the Australasian Computer Science Week Multiconference on - ACSW '17, 1–4. https://doi.org/10.1145/3014812.3014889.

[41] Langin, C., Zhou, H., & Rahimi, S. (2009). A Self-Organizing Map and its Modeling for Discovering Malignant Network Traffic, (Mar). https://doi.org/10.1109/CI-CYBS.2009.4925099.

[42] Langin, C., Che, D., Wainer, M., & Rahimi, S. (2009). Visualization of Network Security Traffic using Hexagonal Self-Organizing Maps.

[43] Nogueira, A., Salvador, P., & Blessa, F. (2010). A Botnet Detection System Based on Neural Networks. Digital Telecommunications (ICDT), 2010 Fifth International Conference On, 57–62. https://doi.org/10.1109/ICDT.2010.19.

[44] Wang, K., Huang, C. Y., Lin, S. J., & Lin, Y. D. (2011). A fuzzy pattern-based filtering algorithm for botnet detection. Computer Networks, 55(15), 3275–3286. https://doi.org/10.1016/j.comnet.2011.05.026.

[45] Al-Duwairi, B., & Al-Ebbini, L. (2010). BotDigger: A fuzzy inference system for botnet detection. 5th International Conference on Internet Monitoring and Protection, ICIMP 2010, 16–21. https://doi.org/10.1109/ICIMP.2010.11

[46] Khattak, S., Ramay, N. R., Khan, K. R., Syed, A. A., & Khayam, S. A. (2014). A Taxonomy of Botnet Behavior ,*16*(2), 898–924.

[47] Zhang, H., Papadopoulos, C., & Massey, D. (2013). Detecting encrypted botnet traffic. Proceedings - IEEE INFOCOM, 3453–3458. https://doi.org/10.1109/INFCOM.2013.6567180.