ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 11, November 2017

Distributed Collaborative Approach to Botnet Detection

Zahraddeen Gwarzo[1], Mohamed Zohdy[2], Hua Ming[3], Richard Olawoyin[4], Hany Othman[5]

PhD candidate, Department of Computer Science and Engineering, Oakland University, Michigan, USA[1]

Professor, Department of Electrical and Computer Engineering, Oakland University, Michigan, USA[2]

Assistant Professor, Department of Computer Science and Engineering, Oakland University, Michigan, USA[3]

Assistant Professor, Department of Public and Environmental Wellness, Oakland University, Michigan, USA[4]

Professor, Department of Computer Information Systems, Oakland Community College, Michigan, USA[5]

ABSTRACT: Over the years, there has been rapid advancement in internet technologies, such as the email, the world wide web, VOIP, social networks, etc. Networks of compromised individual and corporate computers called (botnets) have been used to deploy malware, such as Viruses, Worms, Trojans, Spyware etc, to vulnerable computerson a global scale. Botnets are used for various kinds of malicious activities on the internet including: distributed denial of service (DDOS) attacks, massive spam email messages, distributing other malware, click fraud attack and information theft, etc. Better Security decisions are usually associated with experience in cyber security, advanced-technologies, and rich data and information, as such an earnest and determined collaborative approach to botnet detection is likely to have a significant positive outcome in tackling the menace of botnets. In this paper, we propose a novel botnet detection approach that leverages the expertise and experience of several research collaborators, as well as the abundant data and information at each collaborator's disposal, to detect botnets irrespective of command and control protocol, type of architecture, or infection behaviour. We use Python scripts to broadcast diagnosis request to peer collaborators and then use Supervised Machine learning to learn through False Positives ($\mathbf{f_p}$), False Negatives ($\mathbf{f_n}$), True Positives ($\mathbf{t_p}$), and True Negatives ($\mathbf{t_n}$), the detection accuracy of peer collaborators, detect malicious collaborators, and finally, detect costly and unreliable collaborators.

KEYWORDS: Botnet, Malware, Distributed Collaborative Detection, Intrusion Detection System, Supervised Machine Learning, Hashing.

I. INTRODUCTION

Botnets are networks of compromised computers (bots) under the control of a single command and control (C&C) channel called a botmaster [1]. A botmaster is an adversary who leverages the advancement of the communications technology to covertly install malicious software on unsuspecting end users (host computers) in order to turn these end computers (also called Bots or Zombies) into attack arsenals for a wide range of malicious activities. The initial ambition of a botmaster is to gather as many bots as possible so as to have a powerful and effective botnet capable of disrupting a target network. However, the larger the botnet size, the higher the chance of it being detected, as such botmasters have since changed tactics and preferred to create smaller groups of independent botnets. The botmaster can then command these independent botnets from multiple C&C channels to carry out a variety of malicious activities, or just rent out individual botnets to other adversaries.

Botnets pose severe threats to Internet security since they provide the platforms for most large-scale and harmonized cyber-attacks, for deploying malicious activities on the internet, such as; hosting phishing web sites [2], search engine abuse [3], spamming activities [4], facilitate distributed denial of service (DDoS) attacks [5-9], click fraud [10-13] and information theft [14]costing companies, businesses, and governments billions of dollars in losses. More than 100 billion spam messages are believed to be sent daily [15], and botnets are believed to account for 85%of all spam messages [16-21], Botnets have expanded rapidly in recent years both in diversity and population and in 2011, it was estimated that about 40% of world's end computers were part of botnets [22]. As botnets become more robust and

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 11, November 2017

furtive, the contamination of other emerging infrastructure, such as smart phones [23] and industrial control systems [24] is also pervasive. Although there have been significant achievements in the botnet research community, botnet developers continue to develop highly sophisticated botnets and covert ways to control their networks. As such the need for a collaborative approach to botnet detection, mitigation and prevention by experienced stakeholders using all the relevant data and advanced-technologies at their disposal.

In this paper, we propose a novel detection approach that leverages the expertise, experience, and the abundant data at the collaborators' disposal, to detect the wide range of botnets including potentially newer classes of botnets. Our approach is a fully distributed collaborative application that sits on top of an intrusion detection system (IDS). We configure Snort intrusion detection system to generate logs and dump the logs on to a database. Python scripts are then used to fetch data payload from the database, pass the payload to a hash generator, and broadcast an advisory request alongside the computed payload hash to the collaborative network. We then collate and aggregate all responses and apply a supervised machine learning to learn the detection accuracy of collaborators, and detect malicious and costly collaborators.

The remainder of this paper is organized as follows. Section II compares various approaches in the literature. Section III discusses problem statement and our contributions to this work. Section IV presents our methodology and approach for this work, and how the methodology is applied and the anticipated results. Finally, we discuss and conclude the paper in section V.

II. RELATED WORK

In this work, our focus is on a distributed collaborative approach to detect specifically botnets, while the collaborative approaches in the literature focus more on the general intrusion detection of any kind of malware which of course includes botnet; however, both approaches are very relevant.Carol Fung et al. in [25] and [26] proposed a distributed host based IDS (HIDS) particularly focusing on identifying, selecting, and maintaining collaborators using Bayesian learning. However, the papers focus more on learning and fishing out a dishonest collaborator but does little to identify effective and efficient collaborators which are key to creating a robust detection system. Bogus test messages are sent out periodically to collaborators in an effort to fish out a malicious or incompetent collaborator similar to [27]. While this approach may help fish out an incompetent collaborator, it can be defeated by a determined sophisticated malicious collaborator who would anticipate this kind of bogus messages, thus responding correctly over time and then responds misleadingly when it matters most.

It is important to note that unlike this work, much work in the literature such as in [28-30], failed to address the issue of a possible malicious collaborator in the network which will greatly impact the success of the system. Some papers have used majority voting [31] among peer collaborators to detect a malicious insider, which is clearly not practical as it degrades the performance and responsiveness of the system. Other works have used trust management such as in [32-33] in an attempt to fish out a malicious insider.

Guerid et al. [34] proposed a collaborative approach to botnet detection in real-time, by grouping botnet members residing in distributed networks into communities, and then attempting to detect botnets C&C servers by identifying their convergence points using bloom filters similar to [35]. While this may be a good approach, however, using bloom filters to group botnet members could result in too many false positives as more and more members are added to the group list. Moreover, using bloom filters deprives the methodology the chance to remove members that are incorrectly added to the group or no longer have the characteristic of that particular group.

Wang et al. [36] proposed a collaborative architecture for specifically botnet detection using hierarchical levels of collaboration that could help facilitate the detection mechanism. The three levels of collaboration they used are, information collection from collaborators systems, feature extraction and correlation from those systems, and finally the decision level collaboration. While this proposed architecture is decentralized, which makes it promising, However, no practical experiment has been carried out to actually assess the performance of the system, or normalize information sharing among peer collaborators, among other things. The authors also propose other collaborative approaches to botnet detection [37]. Again, these approaches have not been practically implemented and tested, and as such suffer from similar weaknesses which [36] suffers from.

Stevanovic et al. [38] proposed a framework called ContraBot aimed at improving Botnet detection and mitigation through collaborative approach from several collaborators. ContraBot relies on input from a wide range of sources which include network sensors from within and outside the network to perform an in-depth correlation of suspicious

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 11, November 2017

network traffic. The correlator then forwards detected malicious traffic to a warning generator. It will in turn send a warning alert to its subscribers. Again, this hypothesis has not been implemented as such it is not clear how issues such as trust management among collaborators would be managed, or how effective or efficient ContraBot is.

Last but not the least, Quanyan Zhu et al. [39] propose a distributed collaborative detection methodology aimed at finding from the collaborators' responses, decisions which are less costly. However, unlike in this paper, where all responses and no responses are all aggregated, the methodology allows some collaborators not to respond at all and as such no responses are aggregated. This makes the overall collaborators feedback aggregation weaker and potentially increases the cost of decisions.

III. PROBLEM STATEMENT AND CONTRIBUTION

Botnets are being used to carry out various malicious activities on the internet today. They include distributed denial of service (DDOS) attacks, massive spam email messages, distributing other malware, click fraud attack and information theft, among other threats. In 2011, Hogben et al. reported that more than 40 percent of world's end computers were believed to be infected by bots. As such they were part of some kind of botnets [22].

The global impact of botnet activities on governments and businesses is enormous. The Global Threat Intelligence Report (GTRI) reported that, Client botnet activities constitute 34% of 3 billion worldwide attacks in 2013[40]. In 2014, the FBI said in its statement before the senate judiciary committee on cybercrime and terrorism, that the industry estimates botnets have caused over $9 billion in losses to United States victims, and over $110 billion in losses globally [41]. Furthermore, FBI said, approximately 500 million computers are being compromised globally each year, translating into 18 victims per second. More recently in October, 2016, we have witnessed arguably the largest DDOS attacks in historywhich is believed to have been caused by the Mirai botnet. The attack targeted Dynsystems and was responsible for major internet outage on large numbers of internet users in Europe and North America [42].

Researchers and security professionals are having sleepless nights over Botnet occurrences, the threats posed by Botnets, and the difficulty in detecting highly sophisticated botnets. Botnet developers have become more sophisticated over the years as they use several obfuscation techniques to evade detection.

As botnet developers continue to develop highly sophisticated botnets and covert ways to control their networks, researchers face several challenges in doing experiments that would pave the way for the design and development of effective botnet detection systems. Researchers must collaborate with a wide range of internet domains and organizations to share the necessary data and information needed for experiments to finding responsive solutions. Researchers must make sure they have large sets of real network traces gathered from across wide ranges of administrative internet domains that truly represent the high heterogeneity of the internet. Researchers must also possess a wide range of data about infected systems. However, carrying out this ambition is easier said than done, because most organizations would be reluctant to share real network traces because they will most likely contain sensitive private information. Most organizations would also be reluctant to share information regarding infections which their networks suffer from, as revealing them might be used by competitors to take advantage business-wise. These challenges slow down the progress being made by researchers in this domain.

Therefore, the main motivation of this research method, is to provide higher accuracy, effectiveness and efficiency in detecting botnets of any kind while preserving the privacy of collaborators, and thereafter overcoming the weaknesses of existing botnet detection approaches.

A. *Contributions*

Better security decisions are usually associated with experience, advanced technologies, and rich data and information at our disposal. Therefore, an earnest and determined collaborative approach to botnet detection like this one, is likely to have a significant positive outcome. The main contributions of this research are as follows:

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 11, November 2017

- Novel design, configuration, and deployment of the infrastructure, consisting of data source, sink, and the participating nodes as depicted in Fig. 1
- The process of writing novel python scripts that fetch data logs from a database that was dumped by Barnyard2 from the Snort's alert logs server, compute a hash for the payload, and then append a question to the payloadhash, and broadcast the message to the collaborating entities in a secure manner. The collaborating entities then broadcast a reply to all.
- The data is collected, aggregated and a Supervised Machine Learning algorithm is applied, resulting in learning the detection accuracy of collaborators through False Positives ($\mathbf{f_p}$), False Negatives ($\mathbf{f_n}$), True Positives ($\mathbf{t_p}$), and True Negatives ($\mathbf{t_n}$), the detection of malicious collaborator, and finally, the detection of costly and unreliable collaborators, resulting in an EFFICIENT, ACCURATE, AND SECURE DISTRIBUTED COLLABORATIVE BOTNET DETECTION SYSTEM**.**

IV. METHODOLOGY

This research is generally divided into three parts. The first part involves the creation, configuration, and deployment of the infrastructure, consisting ofdata source, sink, and the participating nodes. The second part involves the process of writing python scripts that fetch data logs from the Snort's alert logs server/MYSQL, then attach a question to the payload, compute a hash for the payload, encrypt the payload hash, and broadcast the message to the collaborating entities. The collaborating entities compare payload hash against stored hashes for various threats, then broadcast a reply to all. Finally, the data is collected, aggregated and machine learning technique is applied, and three main decisions are taken; learn the detection accuracy of collaborators, detection of malicious collaborators, and detection of costly and unreliable collaborators. The IDS is subsequently updated with the new signature.

A. Experimental Setup

This research approach involves a fully practical environment as can be seen in Fig. 1 below. This research experiment requires enormous technical expertise to setup and configure the test environment. The test bed is comprised of various operating system platforms, hardware, software, and technologies. It took us 30 days to setup and configure the test bed which includes a Linux box as the firewall and a gateway to and from the internet. For now, the firewall is configured to direct all incoming traffic to the Snort intrusion detection and prevention system which sits on top of the firewall, and is configured on Ubuntu 16.04 Linux operating system. We are only interested in the alert logs generated by snort, therefore we configured Snort in intrusion detection mode instead of prevention mode.

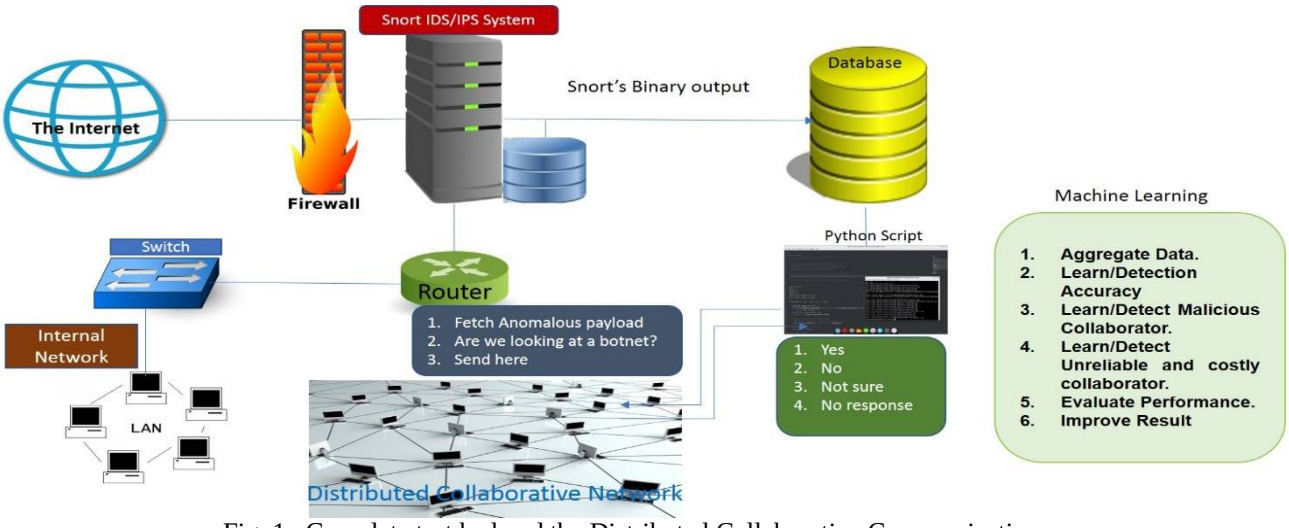

Fig. 1. Complete test bed and the Distributed Collaborative Communication.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 11, November 2017

B. Analysis of the Distributed Collaborative detection

We started by writing script in Python Language to periodically fetch questionable payload data that was dumped into MySQL database by Snort. A hash for the payload is then computed and the script appends a question to the data payload hash, encrypt the hash, and broadcast the data to the distributed collaborative network. Each collaborator analyses the received data with all the resources and technologies at its disposal, and henceforth broadcast a reply with an opinion as to what the collaborator believes about the questionable data i.e if the questionable data has the characteristics of a botnet, or matches a previously stored botnet hashes. A second script handles the reply process. We set up a reply threshold and collate all responses including a no response. We then aggregate all collected data. The data collected includes a "YES" reply, a "NO" reply, a late reply, and finally a "NULL" which is entered against a collaborator who had not replied at all. We then clean up the data using Core Spark, and apply Machine Learning using both Spark ML, and separately using Anaconda data science platform. Fig. 2 is a flow chart diagram that depicts the above scenario.

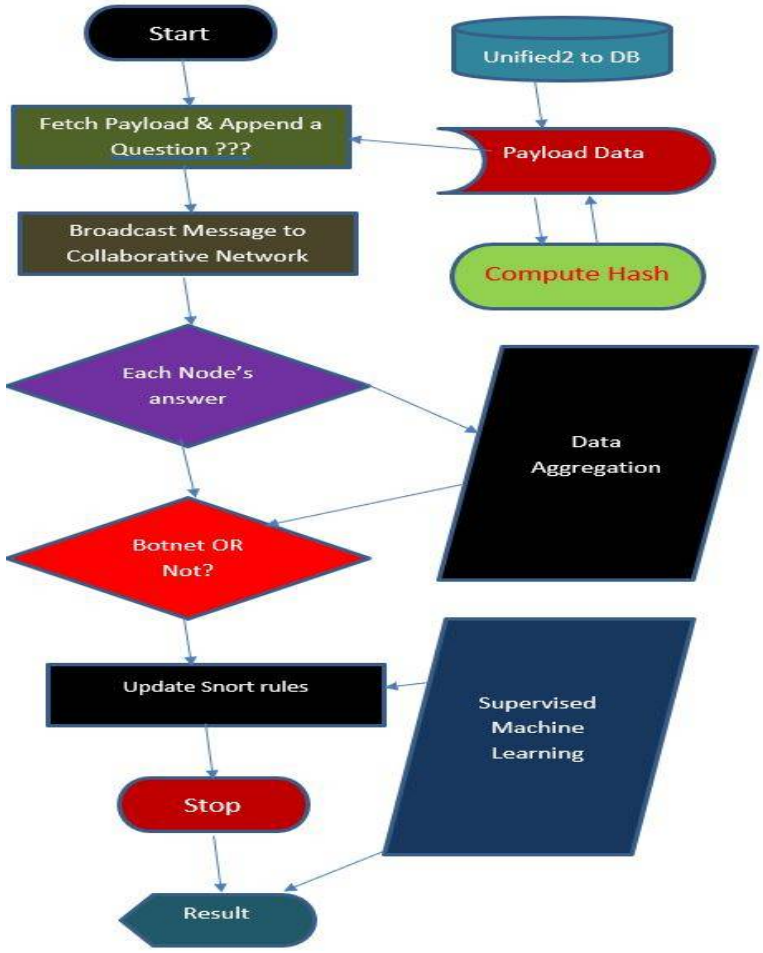

Fig. 2: Distributed Collaborative Detection.

We started by selecting twenty collaborators, and the process of selection was purely manual. This means that we did our homework to find several reputable collaborators that were interested in joining the collaborative network, such as Internet Service Providers (ISPs), Security Companies, intelligent communities, and even government agencies. The process of adding and maintaining the collaborative network is shown below. Our approach has a fixed collaborators

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 11, November 2017

pool P of size N for better management and performance. However, P is not restricted to the fixed size of N, it can be extended as the need arises. But for now, we are limiting the size of P to 20. This is constantly being updated as some collaborators are deleted or replaced based on the risk or cost of keeping them in the collaborative network.

Three different sets are employed in this study: the training set, the validation set and the test set. The training set was used to develop the initial classifier and train the dataset and the different classifier parameters were tested with the validation set. Finally, the classification accuracy was measured using the test set. Three metrics were used to measure the classifier performance which include the Hamming Distance (HD) and accuracy, Mathews Correlation Coefficient (MCC), and the Precision (ψ) and Recall ($\varkappa$). Equation 1 shows the HD between two vectors with binary entries, this calculates the errors from real and predicted classifications and retrieves the number of correct classifications which represents the accuracy.

$$1 - HD \qquad \text{eq. (1)}$$

Our main goal was to learn from the aggregated data the accuracy of responses in the context of f_ps, f_ns, t_ps, and t_ns. If the majority replied with a "YES", then "YES" was a t_p, and a "NO" was a f_n. Whereas, if the majority replied with a "NO", then "NO" was a t_n and "YES" was an f_p respectively. We also learned from the aggregated data, collaborators that either did not reply to a certain request, or did not reply within the threshold, which allowed us to detect incompetent, inefficient and costly collaborators.

For each sample, the quality of the classifier was measured from the real and predicted values using the MCC (equation 2), which considers both the $f_p$andf_n.

$$MCC = \left[\frac{(t_p * t_n) - (f_p * f_n)}{\sqrt{(t_p + f_p)}(t_p + f_n)\sqrt{(t_n + f_p)}(t_n + f_n}\right] \ldots . \text{ eq. (2)}$$

Where; $MCC \geq -1 \leq 1$
$MCC = -1$ $(all\ wrong\ predictions)$
$MCC = 0$ $(classifier\ is\ good\ for\ random\ predictions)$
$MCC = 1$ $(perfect\ predictions)$

It is important also to ensure the precision (ψ) of the classifier so that negative samples are not labeled as positive samples. The precision was calculated from equation 3 and precision is inversely proportional to f_p, the higher the precision value, the smaller the f_p.

$$\psi = \frac{t_p}{t_p + f_n} \quad \ldots\ldots \qquad \text{eq. (3)}$$

The true positive rate or recall ($\varkappa$) represents the unique ability of the classifier to determine all positive samples. This was calculated using equation 4 and it shows that higher $\varkappa$ means that there are smaller numbers of f_n. The ratio of negative samples labeled as positive samples is the fall-out (ξ) which can be calculated from equation 5.

$$\varkappa = \frac{t_p}{t_p + f_n} \quad \ldots\ldots \qquad \text{eq. (4)}$$

$$\xi = \frac{f_p}{f_p + t_n} \quad \ldots . \quad \text{eq. (5)} \ .$$

$\varkappa \approx 1$ $(no\ false\ negatives)$
$\xi \approx 0$ $(no\ false\ positives)$

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 11, November 2017

V. DISCUSSION AND FUTURE WORK

In this paper, we propose a distributed collaborative approach to botnet detection. Our approach leverages the expertise, experience, and resources of several research collaborators to improve the accuracy and efficiency of botnet detection. This collaboration ensures that the system overcomes the weaknesses of existing intrusion detection systems that function in isolation, and therefore have limited capability. Our System has the potential to detect any kind of botnet irrespective of a particular C&C, communication protocol, or infection behavior.

To better manage the collaborators pool, we started with a fixed number of collaborators and expanded as the need arose. The pool was constantly being updated as some members were deleted or replaced based on certain factors. Most collaborative approaches are only proof of concept such as in [34-39], they have not been implemented and tested. Our approach is now being implemented and tested.

Privacy issues are a major concern among collaborators, such as the exchange of payload data which may contain sensitive information. Our approach tackles privacy issues thereby sharing only the computed hash of the payload, instead of the actual payload data. It would be virtually impossible to reverse the hash and retrieve the actual data. Nevertheless, we encrypted the hash before transmitting it across the collaborative network.

It is also crucial to provide a proactive mechanism that will protect the system from malicious insiders. Fung et al. in [27] proposed sending out a bogus test messages periodically to the collaborative network in an effort to detect a malicious insider. While this approach might work well against an amateur collaborator, it would certainly be defeated by a sophisticated malicious collaborator who would anticipate this kind of bogus test message, henceforth replying correctly over time until the malicious collaborator gains the trust of the system, and then becomes malicious when it matters most. In our approach, the mode of communication among collaborators is Broadcast not a Unicast, which means every collaborator gets a copy of a message and its replies. Moreover, a decision is not taken until all replies are aggregated, therefore limiting the impact of a malicious reply. We also set a threshold to receive replies to boost the efficiency of the system. We decided to use a Supervised machine learning approach since our focus is to achieve a desired result from the training dataset. In our future work, we will employ the combination of a supervised machine learning and unsupervised machine learning approach, in other words a hybrid approach similar to [43]. The integration of an unsupervised machine learning approach would help the classifier predict new class of results henceforth making the system more robust.

REFERENCES

1. G. Gu, R. Perdisci, J. Zhang, and W. Lee, “BotMiner: Clustering Analysis of Network Traffic for Protocol- and Structure-Independent Botnet Detection,” in Proc. USENIX Security, pp. 139-154. 2008.
2. C. Whittaker, B. Ryner, and M. Nazif, “Large-Scale Automatic Classification of Phishing Pages,” in Proc. NDSS, vol. 10, p. 2010 2010.
3. J. P. John, F. Yu, Y. Xie, A. Krishnamurthy, and M. Abadi, “deSEO: Combating Search-Result Poisoning,” in Proc. USENIX security symposium, pp. 20-35. 2011.
4. S. Hao, N. A. Syed, N. Feamster, A. G. Gray, and S. Krasser, “Detecting Spammers with SNARE: Spatio-temporal Network-level Automatic Reputation Engine,” in Proc. USENIX security symposium, vol. 9. 2009.
5. Eddy, Wesley M “TCP SYN Flooding Attacks and Common Mitigations.” RFC 4987.2007
6. Lim, Sharon, J. Ha, H. Kim, Y. Kim, and S. Yang. "A SDN-oriented DDoS blocking scheme for botnet-based attacks." In Ubiquitous and Future Networks (ICUFN), 2014 Sixth International Conf on, pp. 63-68. IEEE, 2014.
7. Zargar, SamanTaghavi, James Joshi, and David Tipper. "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks." IEEE communications surveys & tutorials 15, no. 4 (2013): 2046-2069.
8. Alomari, Esraa, Selvakumar Manickam, B. B. Gupta, Shankar Karuppayah, and RafeefAlfaris. "Botnet-based distributed denial of service (DDoS) attacks on web servers: classification and art." arXiv preprint arXiv:1208.0403 (2012).
9. Karasaridis, Anestis, Brian Rexroad, and David A. Hoeflin. "Wide-Scale Botnet Detection and Characterization." HotBots 7 (2007): 7-7.
10. Miller, Brad, Paul Pearce, Chris Grier, Christian Kreibich, and Vern Paxson. "What's Clicking What? Techniques and Innovations of Today's Clickbots." In *DIMVA*, pp. 164-183. 2011.
11. Pearce, Paul, Vacha Dave, Chris Grier, Kirill Levchenko, Saikat Guha, Damon McCoy, Vern Paxson, Stefan Savage, and Geoffrey M. Voelker. "Characterizing large-scale click fraud in zeroaccess." In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, pp. 141-152. ACM, 2014.
12. . Haddadi, Hamed. "Fighting online click-fraud using bluff ads." ACM SIGCOMM Computer Communication Review 40, no. 2 (2010): 21-25.
13. Daswani, Neil, and Michael Stoppelman. "The anatomy of Clickbot. A." In Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets, pp. 11-11. USENIX Association, 2007.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 11, November 2017

14. Stone-Gross, Brett, Marco Cova, Lorenzo Cavallaro, Bob Gilbert, Martin Szydlowski, Richard Kemmerer, Christopher Kruegel, and Giovanni Vigna. "Your botnet is my botnet: analysis of a botnet takeover." In Proceedings of the 16th ACM conference on Computer and communications security, pp. 635-647. ACM, 2009.
15. John, John P., Alexander Moshchuk, Steven D. Gribble, and Arvind Krishnamurthy. "Studying Spamming Botnets Using Botlab." In NSDI, vol. 9, pp. 291-306. 2009.
16. Rao, Justin M., and David H. Reiley. "The economics of spam." The Journal of Economic Perspectives 26, no. 3 (2012): 87-110.
17. Duan, Zhenhai, Peng Chen, Fernando Sanchez, Yingfei Dong, Mary Stephenson, and James Michael Barker. "Detecting spam zombies by monitoring outgoing messages." IEEE Transactions on dependable and secure computing 9, no. 2 (2012): 198-210.
18. Grier, Chris, Kurt Thomas, Vern Paxson, and Michael Zhang. "@ spam: the underground on 140 characters or less." In Proceedings of the 17th ACM conference on Computer and communications security, pp. 27-37. ACM, 2010.
19. Gummadi, Ramakrishna, Hari Balakrishnan, Petros Maniatis, and Sylvia Ratnasamy. "Not-a-Bot: Improving Service Availability in the Face of Botnet Attacks." In NSDI, vol. 9, pp. 307-320. 2009
20. Zhao, Yao, YinglianXie, Fang Yu, QifaKe, Yuan Yu, Yan Chen, and Eliot Gillum. "BotGraph: Large Scale Spamming Botnet Detection." In NSDI, vol. 9, pp. 321-334. 2009.
21. Ramachandran, Anirudh, and Nick Feamster. "Understanding the network-level behavior of spammers." In ACM SIGCOMM Computer Communication Review, vol. 36, no. 4, pp. 291-302. ACM, 2006.
22. Hogben, Giles, Daniel Plohmann, Elmar Gerhards-Padilla, and Felix Leder. "Botnets: Detection, measurement, disinfection and defence." European Network and Information Security Agency (2011).
23. Falliere, Nicolas, Liam O. Murchu, and Eric Chien. "W32. stuxnet dossier." White paper, Symantec Corp., Security Response 5, no. 6 (2011).
24. Mullaney, Cathal. "Android. bmaster: A million-dollar mobile botnet." online], Symantec Security Response, http://www. symantec. com/connect/blogs/androidbmaster-million-dollar-mobile-botnet (2012).
25. Fung, Carol J., Jie Zhang, and Raouf Boutaba. "Effective acquaintance management for collaborative intrusion detection networks." In Network and Service Management (CNSM), 2010 International Conference on, pp. 158-165. IEEE, 2010
26. Fung, Carol J., and Raouf Boutaba. "Design and management of collaborative intrusion detection networks." In Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on, pp. 955-961. IEEE, 2013.
27. Staab, Eugen, Volker Fusenig, and Thomas Engel. "Towards trust-based acquisition of unverifiable information." In International Workshop on Cooperative Information Agents, pp. 41-54. Springer Berlin Heidelberg, 2008.
28. Cai, Min, et al. "Wormshield: Fast worm signature generation with distributed fingerprint aggregation." IEEE Transactions on Dependable and Secure Computing 4.2 (2007).
29. Li, Zhichun, et al. "Netshield: massive semantics-based vulnerability signature matching for high-speed networks." ACM SIGCOMM Computer Communication Review. Vol. 40. No. 4. ACM, 2010.
30. Guerid, Hachem, Karel Mittig, and Ahmed Serhrouchni. "Collaborative approach for inter-domain botnet detection in large-scale networks." In Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 2013 9th International Conference Conference on, pp. 279-288. IEEE, 2013.
31. Ghosh, Arjita, and Sandip Sen. "Agent-based distributed intrusion alert system." In International Workshop on Distributed Computing, pp. 240-251. Springer Berlin Heidelberg, 2004.
32. Duma, Claudiu, Martin Karresand, Nahid Shahmehri, and GermanoCaronni. "A trust-aware, p2p-based overlay for intrusion detection." In Database and Expert Systems Applications, 2006. DEXA'06. 17th International Workshop on, pp. 692-697. IEEE, 2006.
33. Fung, Carol J., Jie Zhang, IssamAib, and Raouf Boutaba. "Robust and scalable trust management for collaborative intrusion detection." In Integrated Network Management, 2009. IM'09. IFIP/IEEE International Symposium on, pp. 33-40. IEEE, 2009.
34. Guerid, Hachem, Karel Mittig, and Ahmed Serhrouchni. "Collaborative approach for inter-domain botnet detection in large-scale networks." In Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 2013 9th International Conference Conference on, pp. 279-288. IEEE, 2013.
35. Locasto, Michael E., Janak J. Parekh, Angelos D. Keromytis, and Salvatore J. Stolfo. "Towards collaborative security and p2p intrusion detection." In Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC, pp. 333-339. IEEE, 2005.
36. Wang, Hailong, and Zhenghu Gong. "Collaboration-based botnet detection architecture." In Intelligent Computation Technology and Automation, 2009. ICICTA'09. Second International Conference on, vol. 2, pp. 375-378. IEEE, 2009.
37. Wang, HaiLong, JieHou, and ZhengHu Gong. "Botnet Detection Architecture Based on Heterogeneous Multi-sensor Information Fusion." JNW 6, no. 12 (2011): 1655-1661.
38. Stevanovic, Matija, Kasper Revsbech, Jens Myrup Pedersen, Robin Sharp, and Christian Damsgaard Jensen. "A collaborative approach to botnet protection." In International Conference on Availability, Reliability, and Security, pp. 624-638. Springer Berlin Heidelberg, 2012.
39. Zhu, Quanyan, Carol J. Fung, Raouf Boutaba, and Tamer Basar. "A distributed sequential algorithm for collaborative intrusion detection networks." In Communications (ICC), 2010 IEEE International Conference on, pp. 1-6. IEEE, 2010.
40. Pierluigi Paganini, Security Affairs- Global Threat Intelligence Report(GTIR), Security, 2014.
41. Joseph Damarest, on Statement Before the Senate Judiciary Committee, Subcommittee on Crime and Terrorism, Washington DC.https://www.fbi.gov/news/testimony/taking-down-botnets, 2014, Accessed 01.21.2017.
42. Kolias, Constantinos, Georgios Kambourakis, AngelosStavrou, and Jeffrey Voas. "DDoS in the IoT: Mirai and Other Botnets." Computer 50, no. 7 (2017): 80-84.
43. Yang, Richard Ruiqi, Victor Kang, Sami Albouq, and Mohamed A. Zohdy. "Application of Hybrid Machine Learning to Detect and Remove Malware." *Transactions on Machine Learning and Artificial Intelligence* 3, no. 4 (2015): 16.