Botnet Detection: A Cooperative Game Theoretical Correlation-Based Model

Noura Al Ebri, Hadi Otrok, Azzam Mourad*, Yousof Al-Hammadi
Department of Electrical and Computer Engineering
Khalifa University of Science, Technology and Research, Abu Dhabi, UAE
*Department of Computer Science and Mathematics
Lebanese American University, Beirut, Lebanon
Email:{noura.alebri, hadi.otrok, yousof.alhammadi}@kustar.ac.ae and azzam.mourad@lau.edu.lb

***Abstract*—In this paper, we address the problem of botnet detection by correlating information from trusted hosts and network. Botnets are groups of compromised computers controlled by a botmaster through a command and control (C&C) channel. They are noted as one of the foremost security threat causing large scale attacks such as Distributed Denial of Service (DDoS), Spam, mass identity theft and click fraud. Various approaches are used to detect botnets and they range from network to host level detection. To enhance the detection rate, a correlation based model was proposed that combines both host and network level information. Such a model is valid in a network made of trusted hosts. The emergence of smartphones with the capability of mobility and being hosts in different networks, open the door of having untrusted hosts that can reveal fake information. As a solution, we propose a trust-based model that uses cooperative game theory to cluster trusted hosts. The trust is built using the reputation value and it is computed using the hosts' marginal contribution which is derived from Shapley value. Simulation results show that our model improves the detection score compared to the traditional correlation model. Where in one of the simulated scenarios we are able to detect a benign cluster of hosts faster than the traditional correlation model.**

***Index Terms*—Botnet, correlation, game theory, cooperative game theory and botnet detection system.**

I. INTRODUCTION

Botnets are a network of compromised computers called "bots" or "zombies" that are remotely controlled by an entity called "botmaster". Nowadays botnets have a great impact on cyber-security especially when it is used by cyber-criminals for a financial gain [9]. New generation botnets are coming out that can unleash severe attacks. To detect such attacks, signature-based and anomaly-based botnet detection systems have been developed where the former analyzes network traffic looking for predefined patterns such as well known bots commands [2]. On the other hand, anomaly based botnet detection system maintains the normal behaviour of the system and generates an alarm when there is a deviation from the estimated normal behaviour [5].

Various botnet detection approaches focus on either the host or the network level, but the need for a better detection system requires data from both levels. Such architecture is suggested by Zeng *et al*. [21] where they detect botnets using a correlation method applied to the combined host and network level information. Their result shows that the proposed framework can detect traditional IRC, HTTP and recent hybrid P2P botnets.

Now a days, smartphones and due to their computer like capabilities and usage growth, attracted botmasters to get benefit from such devices and change their role from being a phone for calls and fun to being bots part of a botnet attack. For instance, spammers have developed a malware that created a botnet from Android mobile devices [8]. Furthermore, many researchers demonstrated how such mobile botnet can be created, such as the SMS-based smartphone botnet [3]. Knowing that such devices are considered as hosts in their corporate network and thus they can be a source of information as in the correlation model of Zeng *et al*. [21].

The inevitability of having mobile botnets in a cooperate network urge us to find ways to detect such mobile bots. As a solution, a trust layer is needed to be added on top of the correlation model [21] so that trusted information can be gathered from trusted hosts. *Note that throughout the paper we will call the model introduced in [21] as a traditional correlation model*. In other words, correlating the data from any host is problematic because with the increase of mobility, many untrusted hosts might connect to various private network. This increase the risk of being infected with bots. Thus, the detection system might correlate untrusted data.

To solve such a problem, we have created a cooperative game that correlates trusted host data by the means of clustering. Such a solution will separate the trusted hosts from the suspicious ones, and hence increases the detection rate. Then, we have introduced a reputation model to create a trust layer. Our model assigns different reputation level to each host, which will evolve according to its host contribution level that will be calculated by the means of a cooperative game model. After that, we correlate the network data with the trusted in-host data collected from the clustered trusted hosts to improve the detection of bots in the network. In summary, the main contribution, of this paper, is a cooperative game model based on Shapley value that:

- Correlates trusted data, which enhances the detection rate.
- Modifies hosts' reputation based on hosts' contribution.

The rest of the paper is organized as the following: Section II presents the related work to botnet detection. Section III illustrates the traditional correlation based botnet detection model.

Section IV shows the Trust based correlation model. Section V shows simulation results. Finally, Section VI concludes the paper.

II. RELATED WORK

A bot is a malicious code which is installed in a victim host without the user knowledge either by opening malicious email attachments or visiting malicious websites. Once the bot is installed in the victim machine, it connects back to an IRC server through a command and control structure (C&C). The bot then joins a predefined channel waiting for the attack commands issued by the botmaster, thus allowing the attacker to control the compromised machine remotely [5], [9].

Since the Eggdrop botnet was created in 1993, many botnets have appeared, the following are some examples of them. HTTP based botnets such as Netbus in 1998, Bagle in 2004, Kraken in 2008 and Asprox in 2008. P2P based botnets such as, Strom in 2007 and Conficker in 2009 that infected $2,708,259$ machine [1].

Many botnet detection systems are employed to detect and count botnets where different types are developed such as IRC based botnet detection systems [2], [19] and botnet detection systems that use knowledge of known botnets' behaviors and signatures based on Snort. Moreover, anomaly based detection systems that detects bots based on several traffic anomalies such as traffic on unusual used ports, high latency, and high volumes of traffic. One example of anomaly-based detection system is the Botsniffer [7] which uses network-based anomaly detection to detect bots. There is also, DNS-based botnet detection systems that use techniques based on certain DNS information generated by a botnet [5]. Honeypot, a computer system that is used as a trap to lure attackers to attack it, is also used as a botnet detection system by monitoring all connections and traffic made to a certain set up honeypot [20]. For example Ono *et al*. [11] set their own honeypot system to capture malware including botnet traffic. In other words, many botnet detection systems are utilized, but each one has its own purpose, benefits and limits.

Game theory [14], [16], [17] applied in various fields such as in economics, biology, politics, computer science and network security. In security field, Game Theory has been employed in many aspects such as in [6] they deploy distributed attacks as security games to analyze its threat. Also, in [14] the authors suggested a moderate to robust game theoretical model for intrusion detection system (IDS) in MANETs that will reduce the resources consumption. In [13] the authors aim at increasing the effectiveness of an intrusion detection system (IDS) for a cluster of nodes in ad-hoc network using game theory. The authors in [12] also apply game theory to analyze the interaction between intruders and the IDS in a wired infrastructure network.

To sum up, botnets have evolved and became more advanced, organized, sneaky, and derived by financial gain. Also, the huge computing power a botnet gains from controlling vast numbers of victims' devices and the anonymity given to the botmaster have motivated cyber-criminals to develop their attacks [9]. In addition, botnets are evolving where many of them changed from centralized to decentralized C&C such as P2P botnets [10]. Furthermore, the current detection systems rely on data gathered at a host level or a network level to detect botnets. Thus, the need for a better detection system became a necessity. One such solution is to deploy both host and network information to increase the detection rate.

III. CORRELATION BASED BOTNET DETECTION

Zeng *et al*. [21] proposed a botnet detection model that combines both host level data (e.g. modification of files in system directory, modification of critical registry key,. . .) and network level data (e.g. Ports, IPs, . . .). The framework consists of a *Host Analyzer*, a *Network Analyzer* and a *Correlation Engine*.

The*Host Analyzer* is installed in every host and it contains two modules which are In-Host Monitor module that monitors runtime system wide behaviour that occurs in the registry, host's network stack and the file system. The other module is a Suspicion-Level Generator that generates a suspicion level by using a certain machine-learning algorithm based on the behaviour reported at each time window and calculates the overall suspicion level. The Host Analyzer sends a suspicion level every 10 minutes together with an average suspicion level generated in an hourly basis. In addition to other network statistics sent to the correlation engine if required.

The *Network Analyzer* works on network traffic collected from a core router and only requires the analysis of Netflow data without accessing the packet's payload. It consists of two modules which are the Flow Analyzer Module that extracts key trigger-action patterns where it looks for suspicious flows. Given that bots under the command of one botmaster receive same commands and acts similarly. The second module is the Clustering Module which identifies a set of suspicious group of hosts by clustering hosts that behave in the same way using network flow analysis. Then if a suspicious group is identified, its host analyzers are instructed to send suspicion level and network statistics to the correlation engine.

The *Correlation Engine* will compare the behaviour data received from the network and from the host to generate a detection score to each host and gives an overall detection result as a score in the interval [0, 1]. The higher the suspicion level the more likely the host is infected by the bot. The correlation engine uses the information sent from the host analyzer and the flow analysis from the network to give better botnet detection result.

This model works well in an internal environment where all hosts are known, but in a dynamic environment (that allows smartphones to join its private network) we cannot trust all hosts. To solve this trust issue, we suggest to add a trust layer to the Clustering Module in a way only trusted hosts are clustered. To do this, we apply cooperative game theory, where the designed game will decide on which hosts should be clustered and this decision is governed by each host's contribution.

IV. TRUST-BASED CORRELATION MODEL

Mobile devices such as smartphones and tablets propagate in today's corporate environments. According to [15], 89% of people have mobile devices connecting to corporate networks and this causes significant concerns. The reason of that many corporate have limited information about these devices unlike the internal company owned devices. Such information could be the type of anti-malware or any installed security measurements in them, firewall existence, software updates among others that can be valuable information for an IT administrator to assess their security. Thus, these devices cannot be trusted to include them in different detection decisions. In other words, if we use the traditional correlation model in a dynamic environment and include the detection score of theses devices, we might get misleading results. Therefore, we need a better correlation model that separates the trusted hosts from the untrusted ones. In our proposed model, we add the trust layer to the clustering model and assign reputation level to each hosts based on their contribution.Note that external hosts might be among the trusted hosts if they exhibits high reputation and contribution values. Thus, our new clustering module will cluster trusted hosts based on their contribution as calculated based on cooperative game. Then, the module will correlate one feature from each host with network features. We capture one feature at each host for efficiency reasons, where we do not want to correlate many features from the hosts and strain them and at the same time we want some in-host data that will support the network BDS detection results. Therefore, to enhance the botnet detection and lower the false alarms, we suggest a model that combines trusted in-host network features together with the network features captured at the network level.

A. Cooperative Game Theory

The design and analysis of our trusted correlation model is done using cooperative game theory [4]. In our game, the l hosts are modeled as a set N of l players in a $N - person$ game with $N = \{N_1, N_2, \ldots, N_l\}$. We define a coalition to be a set of hosts where each host capture one network flow feature. In a cooperative game the quantity $V(S)$ defines the overall amount of value created by a coalition of players S. This overall value is divided up among various players using marginal contribution. To compute such an anticipated marginal contribution for each player (host) in the game with respect to a uniform distribution over the set of all permutations on the set of all players will be presented by Shapley value [4]. To find the contribution of a certain host N_i in coalition S, we consider all the different permutations for the hosts, denoted by R in the coalition. Then we compute the difference between the function including all hosts in the permutations before host N_i joined and included in the function and the function of all hosts prior to N_i, excluding the host N_i. We define ${P_i}^R$ to be the set of all hosts before host N_i in the permutation R. Then taking the average of all these differences, we get the marginal contribution of host N_i in coalition S. In other words, the contribution would be:

TABLE I
FLOW FEATURES

Index	Flow Features
1 to 4	Duration Mean, Variance, Skewness and Kurtosis
5 to 8	Totalbytes Mean, Variance, Skewness and Kurtosis
9 to 12	Number of Packets, Mean, Variance, Skewness and Kurtosis
13	Number of TCP Flows
14	Number of UDP Flows
15	Number of SMTP Flows
16	Number of Unique IPs Contacted
17	Number of Suspicious Ports

$$\phi_i(v) = \frac{1}{|N|!} \sum_{R} [v({P_i}^R \cup \{N_i\}) - v({P_i}^R)] \quad (1)$$

Our characteristic function $V(S)$ specifies the value produced by different group of players (hosts). This value is multiplied by the reputation of the host which is a value from $[0, 1]$, thus we can take the value based on the reputation level of a host. Once we calculate the characteristic function of different groups, we combine the data from the trusted contributors.

B. Illustrative Example

We will analyze our new correlation model using cooperative game theory [4] that will decide on which collaboration gives the greatest total benefits. We let the network BDS capture the same features as the traditional model as seen from Table I. While, the hosts will capture the last three network features.

As an example, consider a cluster N of five hosts that are infected with Waledac worm, thus these hosts are considered as bots in the Waledac botnet network. Symantec threat analysis of Waledac worm [18] contained a detailed analysis of the said worm. Through Symantec investigations they found that Waledac worm uses social engineering (e.g. spreads as an attachment to a spam email) and certain client side vulnerabilities in order to propagate. The worm has different functions and among them are the download and execution of binaries, acting as a network proxy, sending spam, mining infected computers for data, such as email addresses and passwords, and performing denial of service (DoS) attacks. Moreover, to communicate with other peers the nodes in the Waledac botnet network use HTTP protocol for C&C traffic forwarding. In such botnet networks, the botmasters are well hidden behind a P2P network. Furthermore, to initiate a connection with a node, Waledac botnet opens a random local port on the compromised computer and attempts to connect to port 80 of the remote Waledac relay node. Waledac infected hosts can be distinguished form its aggressive spamming traffic pattern.

In this game, we need to set our players and our characteristic function $V(S)$. The game players are considered to be the five infected hosts and we set a coalition $S = \{1, 2, 3\} \in N$ and let each host capture one network feature from the following set of features: {Number of Unique IPs Contacted "IPs" (16), Number of Suspicious Ports "Ports" (17), Number of SMTP Flows "SMTP"(15)} as shown in Table I.

TABLE III
HOSTS $V_2(S)$ WITH THE HIGHEST REPUTATION

Host ID	$V_1(S)$	Reputation	$V_2(S) = V_1(S) \times$ Reputation	Contribution
$Host_1$ (SMTP)	3	0.7	2.1	2.1
$Host_2$ (Ports)	2	0.5	1	1
$Host_3$ (IP)	1	0.6	0.6	0.6

The following specifies our characteristic function $V_1(S)$, where we set the valuation of IP addresses as the least valuable information (value $= 1$) since Waledac bots or benign hosts might visit the same IP address, while we set the valuation of ports at a medium level (value $= 2$) and finally the SMTP flows placed at the highest level (value $= 3$) since Waledac worm is a spam that requires SMTP [18], [21].

$$V_1(S) = \begin{cases} 1, & S \in \{1\} \\ 2, & S \in \{2\} \\ 3, & S \in \{3\} \\ 3, & S \in \{1,2\} \\ 4, & S \in \{1,3\} \\ 5, & S \in \{2,3\} \\ 6, & S \in \{1,2,3\} \\ 0, & S \in Otherwise \end{cases}$$

Since our objective is to correlate information from trusted hosts, we will recalculate the above values based on hosts' reputation (this reputation defines the trust level). We calculate the hosts' reputation based on their contribution as shown in Table II where we increase/decrease reputation with 0.05 to/from hosts based on how they are contributing. If their contribution is helpful then we increase the reputation and decrease otherwise. Hosts that earned zero means that it is the first time to contribute. Thus, the reputation of our five hosts are as follows: $(host_1 = 0.7, host_2 = 0.5, host_3 = 0.6, host_4 = 0.3, host_5 = 0.2)$ where at each of the five hosts we capture the following feature $(host_1 = IP, host_2 = Ports, host_3 = SMTP, host_4 = SMTP, host_5 = IP)$. Table III summarizes our characteristic function $V_2(S)$ of the hosts with the highest reputation. To calculate the marginal contribution of each of the three hosts in the formulated cluster, we use Shapley value in Equation 1.

Thus, once we correlate these trusted in-host level network flow features with the network flow features captured at the network level, we can gain a trusted and a valid detection score. We set the accepted reputation values to be in $[0.5, 1]$. Since the reputation level of the fourth and fifth hosts (< 0.5) we ignore its contribution and compute the highest reputation hosts contributions. Therefore, we correlate the data from the three hosts with the highest contribution to make our detection score. Note that infected hosts will always try to share misleading information such as having low values.

In conclusion, our model is able to cluster trusted hosts based on their reputation and presents a mechanism to modify hosts' reputation based on their contributions.

V. SIMULATION RESULTS

To support our model, we include some simulations scenarios carried out by using our model and the traditional one. In the first scenario, we identified a suspicious cluster of bots where we want to separate the possible benign hosts from the bots. First we use the Traditional model, then we apply our model and evaluate each model's efficiency. This is done by measuring the detection rate. Figure 1 shows the efficiency comparison between the trust-based model and the traditional model over 24 hours. Since our model combines only the data coming from trusted host's, the detection efficiency is higher than the detection efficiency in the traditional one. This is due to the use of reputation as a parameter.

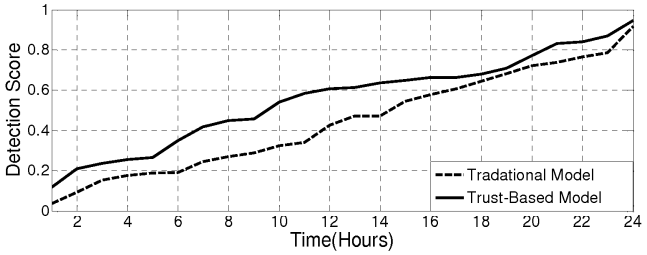

Fig. 1. Efficiency Comparison Between the Trust-Based Model and the Traditional Model over Time

Now, we want to evaluate the overall efficiency of detection by measuring the number of trusted hosts involved in a single detection, to see the affect of having more trusted hosts. As shown in Figure 2, we are measuring the detection rate of bot infested clusters. Having more trusted hosts gives better detection rate until 100 hosts is reached, where the detection rate given by the two model is almost the same. As an example, if we set the bot detection threshold to be more than 0.5 and by combing all the 45 hosts' data will give a detection rate of 0.46, whereas combing only the trusted hosts gives a detection rate of 0.57.

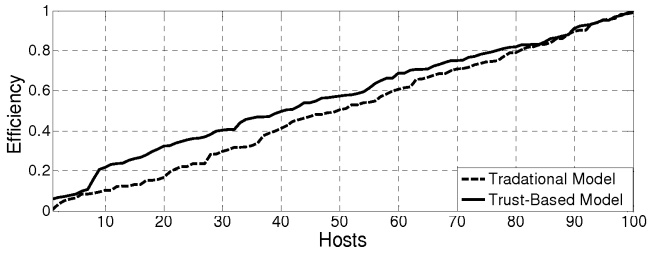

Fig. 2. Efficiency Comparison Between the Trust-Based Model and the Traditional Model by Number of Trusted Users (Hosts)

In the second scenario, we run only the Network BDS to cluster suspicious bots and it identifies a benign cluster of hosts as a suspicious cluster of bots. Thus, depending only in network BDS will result in some false positives. To avoid this we will need more information from a host level which we can get from the host BDS. To do so, we will use the traditional correlation model then the trust-based correlation model to correlate the data gathered in host level with the data

TABLE II
HOSTS REPUTATION COMPUTATION

Host ID	Counter	Previous Contribution	Current Contribution	Previous Reputation	Earned Reputation Points	Current Reputation
$Host_1$ (SMTP)	0	2.1	2.1	0.7	0	0.7
$Host_2$ (Ports)	4	0.65	1	0.45	+0.05	0.5
$Host_3$ (IP)	8	0.55	0.6	0.55	+0.05	0.6
$Host_4$ (SMTP)	2	0.37	0.39	0.35	−0.05	0.3
$Host_5$ (IP)	0	0.23	0.29	0.2	0	0.2

gathered at the network level and give a final detection score. To elaborate more, Figure 3 shows a simulation where we run the BDS network and detect a benign cluster as a suspicious cluster of bots at hour 16, then we run the two correlation model. Although, the two models reach the same result which identifies the cluster as a benign one, the Trust-Based model lowered the detection rate faster than the Traditional model. This happened because the Trust-Based model gives better detection rate as presented in Figures 1 and 2. As a result, the Trust-Based model is more practical and accurate than the Traditional one.

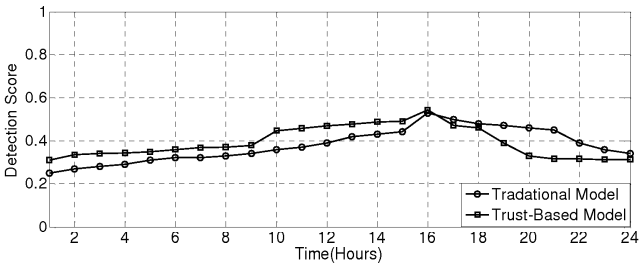

Fig. 3. Cluster Detection Using the Traditional and Trust-Based Models

VI. CONCLUSION

Botnets are evolving and better detection mechanisms are needed. One mechanism suggests combining host and network information. The traditional model might work in a internal environment with known hosts, but it is not appropriate in a dynamic environment that allows smartphones to join their network. The traditional model combines in-host level data of any host disregarding the possibility of existing untrusted hosts. These untrustworthy hosts might tamper with the detection results and affect the detection quality. To solve this, we created a cooperative game model that defines the contribution of each host and based on this calculated contribution we create a reputation model which will set a trust level for each hosts. The higher the reputation, the more trustworthiness are regarded to a host. Then we take the highest contributors to set our detection score. By using this cooperative model, we can select and combine the data of trusted hosts, thus delivering better detection results as illustrated in our simulations.

REFERENCES

[1] R. Borgaonkar. An analysis of the asprox botnet. In *Emerging Security Information Systems and Technologies (SECURWARE), 2010 Fourth International Conference*, pages 148–153, 2010.

[2] E. Cooke, F. Jahanian, and D. McPherson. The zombie roundup: Understanding and detecting and disrupting botnets. In *Proceedings of Workshop on Steps to Reducing Unwanted Traffic on the Internet SRUTI'05*, 2005.

[3] D. Danchev. Researcher demos sms-based smartphone botnet. http://www.zdnet.com/blog/security/researcher-demos-sms-based-smartphone-botnet/8031.

[4] A. E.Roth. The shapley value: Essays in honor of lloyd s.shapley. In *Cambridge University Press*, 1988.

[5] M. Feily, A. Shahrestani, and S. Ramadass. A survey of botnet and botnet detection. In *Emerging Security Information, Systems and Technologies, 2009. SECURWARE '09*, pages 268–273, 2009.

[6] Fultz and Neal. Distributed attacks as security games. In *Master thesis*. US Berkley School of Information, 2008.

[7] G. Gu, R. Perdisci, J. Zhang, and W. Lee. Botminer: Clustering analysis of network traffic for protocol and structure independent botnet detection. In *17th USENIX Security Symposium*, 2008.

[8] B. Jeffrey. Android malware creates smartphone botnet, researchers say. http://www.eweek.com/c/a/Security/Android-Malware-Creates-Smartphone-Botnet-Researchers-Say-256584/.

[9] V. Kamluk. The botnet business. http://www.securelist.com/en/analysis/204792003/Thebotnetbusinessl31.

[10] Y. Namestnikov. The economics of botnets. http://www.securelist.com/en/downloads/pdf/ynambotnets0907en.pdf.

[11] K. Ono, I. Kawaishi, and T. Kamon. Trend of botnet activities. In *Security Technology, 2007 41st Annual IEEE International Carnahan Conference*, pages 243–249. IEEE, 2007.

[12] H. Otrok, M. Mehrandish, C. Assi, M. Debbabi, and P. Bhattacharya. Game theoretic models for detecting network intrusions. *In the Computer Communications journal*, pages 48–60, 2004.

[13] H. Otrok, N. Mohammed, L. Wang, M. Debbabi, and P. Bhattacharya. A game-theoretic intrusion detection model for mobile ad-hoc networks. *In the journal of Computer Communications*, pages 708–721, 2008.

[14] H. Otrok, N. Mohammed, L. Wang, M. Debbabi, and P. Bhattacharya. A moderate to robust game theoretical model for intrusion detection in manets. In *Networking and Communications, 2008. WIMOB '08. IEEE International Conference on Wireless and Mobile Computing*, pages 608–612. IEEE, 2008.

[15] D. Research. The impact of mobile devices on information security: A survey of it professional. http://www.checkpoint.com/downloads/products/check-point-mobile-security-survey-report.pdf.

[16] S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya, and Q. Wu. A survey of game theory as applied to network security. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference*, pages 1–10, 2010.

[17] H. Sally and R. Muhammad. A survey of game theory using evolutionary algorithms. In *Information Technology (ITSim), 2010 International Symposium*, pages 1319–1325, 2010.

[18] G. Tenebro. W32. waledac threat analysis. http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/W32_Waledac.pdf.

[19] L. Wei and A. Ghorbani. Botnets detection based on irc-community. In *Global Telecommunications Conference*, pages 1–5. IEEE, 2008.

[20] H. Zeidanloo, M. Shooshtari, P. Amoli, M. Safari, and M. Zamani. A taxonomy of botnet detection techniques. In *Computer Science and Information Technology (ICCSIT)*, pages 158–162. IEEE, 2010.

[21] Y. Zeng, X. Hu, and K. Shin. Detection of botnets using combined host and network level information. In *Dependable Systems and Networks (DSN)*, pages 291–300. IEEE, 2010.