Proc. of the Fourth International Conference on Advances in Computing, Electronics and Communication - ACEC 2016.
Copyright © Institute of Research Engineers and Doctors. All rights reserved.
ISBN: 978-1-63248-113-9 doi: 10.15224/ 978-1-63248-113-9-01

Botnet Command and Control Traffic Detection Challenges: A Correlation-based Solution

Ibrahim Ghafir, Vaclav Prenosil, Mohammad Hammoudeh

***Abstract*—While high-speed computer networking and the Internet brought great convenience, a number of security challenges also emerged with these technologies. Amongst different computer network security threats, like viruses and worms, botnets have become one of the most malicious threats over the Internet. In this paper, we describe key research challenges in developing effective intrusion detection systems for botnet command and control traffic detection. Then, we outline a new approach to address such challenges, which is based on voting between intrusion detection methods to collaboratively identify command and control traffic. Each detection method analyzes the network traffic to detect one technique used for command and control communications. Four detection methods are initially investigated, these are: malicious IP address, malicious SSL certificate, domain flux and Tor connection detection. Initial analysis shows that the proposed voting-based intrusion detection significantly reduces the number of false positive alerts.**

***Keywords*—Cyber attacks, malware, botnet, command and control server, intrusion detection system.**

I. Introduction

A botnet is a collection of computers connected to the Internet, which have been compromised and are being controlled remotely by an intruder (the botmaster) via malicious software called bots [1]. Amongst different computer network security threats like social engineering attacks [2], targeted attacks [3, 4], and drive-by download attacks [5], botnets have become one of the most malicious threats over the Internet. Financial gains are usually the motive for the design and development of botnets by botmasters, who can reportedly make large sums by marketing their technical services. Experts believe that approximately 16-25% of the computers connected to the Internet are members of botnets [6]. One of the biggest recent distributed denial-of-service (DDOS) assaults the Internet has ever witnessed against KrebsOnSecurity.com shows that the Internet of Things (IoT) is becoming a key target for attackers. The IoT botnet malware, dubbed 'Mirai', spreads to vulnerable connected devices by continuously scanning the Internet for easily hackable IoT systems protected by hard-coded passwords or factory defaults [7, 8, 9, 10, 11, 12, 13, 14].

A typical botnet can be created and maintained in five phases including initial infection, secondary injection, malicious command and control, update and maintenance [15]. This life-cycle is depicted in Figure 1.

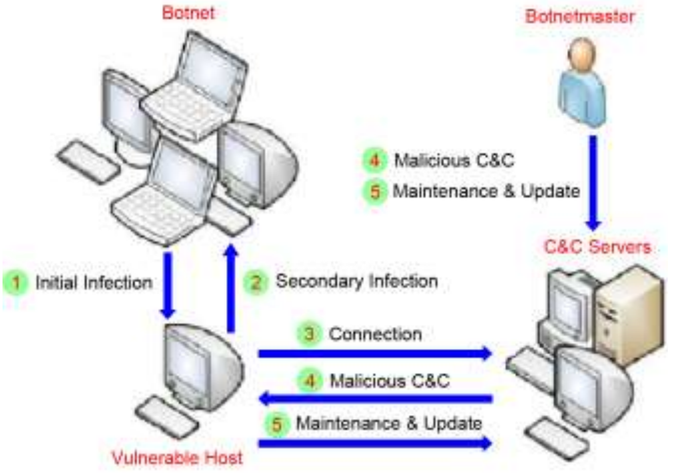

Figure. 1. A typical botnet life-cycle.

During the initial infection phase, the attacker scans a target subnet for known vulnerability and infects victim machines through different exploitation methods. Then, in secondary injection phase, the infected hosts execute a script known as shell-code. The shell-code fetches the image of the actual bot binary from the specific location via FTP, HTTP, or P2P. The bot binary installs itself on the target machine. Once the bot program is installed, the victim computer turns to a *Zombie* and runs the malicious code. The bot application starts automatically each time the zombie is rebooted [16].

In connection phase, the bot program establishes a C&C channel and connects the zombie to the C&C server. Upon the establishment of C&C channel, the zombie becomes a part of attacker's botnet army. After connection phase, the actual botnet command and control activities will be started. The botmaster uses the C&C channel to disseminate commands to his bot army. Bot programs receive and execute commands sent by botmaster. The C&C channel enables the botmaster to remotely control the action of large number of bots to conduct various illicit activities [17].

Ibrahim Ghafir
Faculty of Informatics, Masaryk University, Czech Republic
Faculty of Science & Engineering, Manchester Metropolitan University, UK

Vaclav Prenosil
Faculty of Informatics, Masaryk University
Brno, Czech Republic

Mohammad Hammoudeh
Faculty of Science & Engineering, Manchester Metropolitan University, UK
Manchester, United Kingdom

The last phase is to maintain bots live and updated. In this phase, bots are commanded to download an updated binary. Bot controllers may need to update their botnets for several reasons. For instance, they may need to update the bot binary to evade detection techniques, or they may intend to add new functionality to their bot army. Moreover, sometimes the updated binary move the bots to a different C&C server. This process is called server migration and it is very useful for botmasters to keep their botnet alive [18].

The remainder of this paper is organized as follows. Section II presents the related work to botnet detection. Research questions are stated in Section III. Section IV shows the proposed approach for botnet C&C traffic detection and section V concludes the paper.

II. Related Work

Network security monitoring [19, 20] is a difficult and demanding task that is a vital part of a network administrator’s job. Botnet detection and tracking has been a major research topic in recent years. Different solutions have been proposed in academia. There are mainly two approaches for botnet detection and tracking [15]. One approach is based on setting up honeynets, which is mostly useful to understand botnet technology and characteristics but does not necessarily detect bot infection. The other approach for botnet detection is based on passive network traffic monitoring and analysis. Botnet detection techniques based on passive traffic monitoring have been useful to identify the existence of botnets.

In [21], Baecher et al. introduced Nepenthes, a new type of honeypot that inherits the scalability of low-interaction honeypots but at the same time offers a high degree of expressiveness. Nepenthes is a platform to deploy honeypot modules (called vulnerability modules). This is the key to increased expressiveness: Vulnerability modules offer a highly flexible way to configure Nepenthes into a honeypot for many different types of vulnerabilities. In classical terms, Nepenthes still realizes a low-interaction honeypot since it emulates the vulnerable services.

Another honeypot-based intrusion detection system was proposed by Artail et al. [22]. The system adjusts to changes in the organizational network based on the dynamic deployment and configuration of low-interaction honeypots (honeyds). The main idea is for the honeyds to be deployed using available unused IP addresses such that the distribution of operating, systems and services they emulate mimics that of the operating systems and services of the production hosts in the network. In the majority of cases, the traffic that is directed to the honeyds will be seamlessly diverted to high-interaction honeypots where hackers engage with real services.

A signature-based botnet detection software (Rishi) was proposed by Goebel and Holz [23]. This software matches known nick-name patterns of IRC bots. Rishi is primarily based on passive traffic monitoring for suspicious IRC nicknames, IRC servers, and uncommon server ports. It uses n-gram analysis and a scoring system to detect bots that use uncommon communication channels, which are commonly not detected by classical intrusion detection systems. However, Rishi cannot detect encrypted communication as well as non-IRC Botnets. Moreover, this method is unable to detect bots without using known nickname patterns.

Wurzinger et al. [24] use anomaly detection on aggregate network features to identify a deviation from normal activity. Once identified, a snapshot of the network traffic surrounding the anomaly is taken. Using the intuition that snapshots containing similar anomalies are likely multiple instances of a bot responding to the same botmaster command, the packet payloads leading up to the anomaly are searched for common content to find the command. Once a suitable representation of the command is found, the IDS can build a profile which can then be used to detect future occurrences of the command/response pair.

Another algorithm for detection and characterization of botnets was proposed by Karasaridis et al. [25]. It uses passive analysis based on flow data in the transport layer. This algorithm can detect encrypted botnet communications. It helps to quantify the size of botnets, identify and characterize their activities without joining the botnet.

Host-based approaches also benefit from being easy to deploy and from empowering the end-user directly [26]. In 2014, Balram and Wilscy [27] proposed a detection mechanism for bot C&C traffic by analyzing ”suspicious” flows created after filtering out normal traffic from the traffic generated on a host. The filtering is based on a normal profile of the traffic generated by a user on a host. The profile is built dynamically by examining the behavioral pattern of flows to all destinations. A characterization of bot C&C behavior is also proposed, to derive a set of distinguishing attributes based on which detailed analysis is to be done.

H Xiong et al. [28] proposed a host-based bot detection system for HTTP traffic. The detection system is based on the assumption that users have low diversity on the websites. Out-of-band retrieval and analysis of requested web page is done. Only white-listed web page requests are permitted. The user is informed and asked to take a decision about non white-listed requests. This would be intrusive to the user.

Network-based approaches may require additional cooperation of the network administrator and care must be taken to protect the privacy of the network users [29]. BotMiner, proposed by Gu et al. [30], is a network-based botnet detection system. It relies on the group behavior of individual bots within a botnet for its detection. It exploits the underlying uniformity of behavior of botnets and detects them by attempting to observe and cluster similar behavior being performed simultaneously on multiple machines on a network.

III. Research Questions

To achieve the goal of this work, the following research questions should be answered:

What detection methods can be used for detecting possible techniques used for C&C communications? To answer this question, four detection methods have been proposed, presented in Section 4. Those methods are to be implemented

Proc. of the Fourth International Conference on Advances in Computing, Electronics and Communication - ACEC 2016.
Copyright © Institute of Research Engineers and Doctors. All rights reserved.
ISBN: 978-1-63248-113-9 doi: 10.15224/ 978-1-63248-113-9-01

in the first phase of the research. The proposed modules are not fixed and it is possible to remove or add a new module based on the research progress.

How can the proposed intrusion detection system be made extensible and flexible? The attackers always try to find new techniques for C&C communications, therefore, each detection method should be independent from the other methods, so at any time a new method, for detecting new technique used for C&C communications, can be added to the system and linked with the other methods in a collaborative correlation framework. To achieve the flexibility, in the correlation framework it should be easy to remove or add a new rule for raising an alert on C&C traffic detection.

Is this approach able to handle network traffic in the real-time? The detection system should support the real-time detection, because if an attack, or an attempted attack, is detected quickly, then it can be much easier to trace back the attacker, minimize the damage and prevent further break-ins. To answer this question, in our proposed approach and in the first phase, the detection methods should not depend on storing data and then analyzing it for detection. They should be able to process the network traffic in the real-time and submit their events to the next phase for correlation.

Is this approach effective? The effectiveness of the approach, which is its ability to detect C&C traffic, should be high. This should be combined with a high accuracy resulting into a low number of false warnings. We expect that the chance at a false positive is lower when there is a correlation between the outputs of all detection modules. In order to achieve efficiency for the proposed approach, suitable rules are to be identified for the correlation between the events and this will depend on the evaluation of each detection module and will be done in the last phase of this research.

IV. Proposed Approach

In this section our proposed approach for botnet C&C traffic detection is outlined. This approach is based on the correlation between the events, which are the outputs of the detection modules. As it is shown in Figure 2, the proposed approach consists of two main phases:

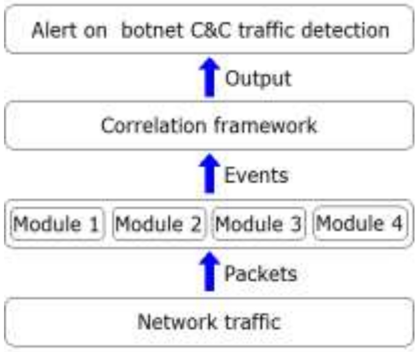

Figure 2. Architecture of proposed approach for botnet C&C traffic detection.

In the first phase, the network traffic is monitored and analyzed to detect possible techniques used in C&C communication. To this end, four detection modules have been proposed: malicious IP address detection module, malicious SSL certificate detection module, domain-flux detection module and Tor connection detection module. Each detection module is independent of the other modules and aims to detect one technique that can be used in C&C communication. The outputs of these detection modules should be submitted to the second phase where they are correlated to raise an alert on botnet C&C traffic detection.

In the second phase, the correlation framework takes events (the outputs of our detection modules) as an input and correlates them to raise an alert on botnet C&C traffic detection. The correlation method is based on voting between the detection methods to make the final decision about the detection.

It is proposed to implement this detection system on top of Bro [31, 32]. Bro is a passive, open-source network traffic analyzer. It is primarily a security monitor that inspects all traffic on a link in depth for signs of suspicious activity. The most immediate benefit that we gain from deploying Bro is an extensive set of *log files* that record a network's activity in high-level terms. These logs include not only a comprehensive record of every connection seen on the wire, but also application-layer transcripts such as, e.g., all HTTP sessions with their requested URIs, key headers, MIME types, and server responses; DNS requests with replies; and much more.

A. Malicious IP Address Detection Module

This module detects any connection between an infected host and C&C server. The detection module is based on a blacklist of malicious IPs of C&C servers. As it is shown in Figure 3, the network traffic is processed and the source and destination IP addresses for each connection are matched with IP blacklist. The blacklist is automatically updated each day based on different intelligence feeds at once [33, 34, 35, 36] and the detection is in the real time.

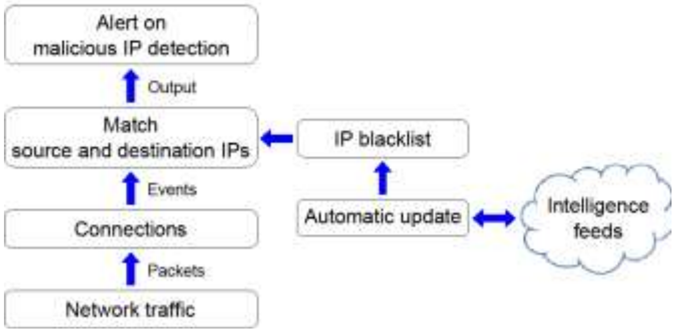

Figure 3. Architecture of malicious IP detection module.

B. Malicious SSL Certificate Detection Module

C&C communications are usually protected by Secure Sockets Layer (SSL) encryption, making it difficult to identify if the traffic is malicious. This detection module is based on a

Proc. of the Fourth International Conference on Advances in Computing, Electronics and Communication - ACEC 2016.
Copyright © Institute of Research Engineers and Doctors. All rights reserved.
ISBN: 978-1-63248-113-9 doi: 10.15224/ 978-1-63248-113-9-01

blacklist of malicious SSL certificates. This blacklist consists of two forms of SSL certificates, *SHA1 fingerprints* and *serial & subject* of bad SSL certificates that are associated with malware and botnet activities. As it is shown in Figure 4, the network traffic is processed and all secure connections are filtered, and then the SSL certificate used in each secure connection is matched with SSL certificate blacklist. The blacklist is automatically updated based on different intelligence feeds [37, 38], and the detection is in real time.

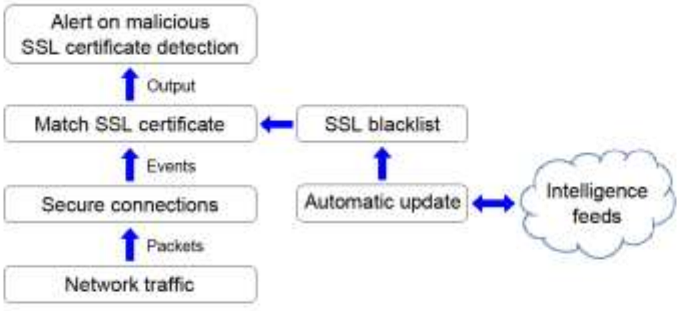

Figure 4. Architecture of malicious SSL certificate detection module.

C. Domain Flux Detection Module

Recent botnets, such as Torpig, BankPatch, Murofet and Conficker, have used more advanced technique for C&C communications. They have used domain flux technique, in which each infected machine separately uses a domain generation algorithm (DGA) to generate a list of domain names [39]. By using domain flux technique, the infected host attempts to query and connect to a large number of domain names which are expected to be C&C servers, while the attacker has to register only one such domain name. This technique makes it difficult for law enforcement to successfully shut down botnets. To prevent infected hosts from updating their malware, law enforcement needs to pre-register all the domains that an infected host queries every day before the botnet owner registers them.

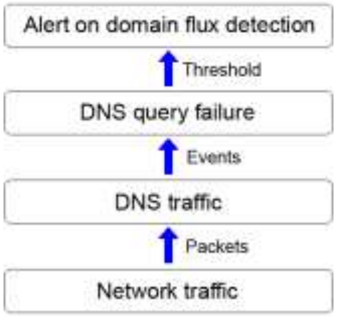

Figure 5. Architecture of domain flux detection module.

This module detects algorithmically generated domain flux. The infected host queries for the existence of a large number of domains, while the owner has to register only one. Therefore, this technique leads to many of DNS query failures because not all of these domain names are registered. The detection module is based on DNS query failures resulting from domain flux technique. As it is shown in Figure 5, the network traffic is processed DNS traffic is filtered. All DNS query failures are analysed and a threshold for DNS query failures from the same IP address is proposed, aiming to detect domain flux technique and identify the infected host.

D. Tor Connection Detection Module

Tor is an anonymous communication network used to secure the privacy of user traffic by encrypting all connections through the overlay network [40]. Tor uses onion routing to direct client's traffic over a circuit of different relays to its destination, denying any single relay to know the complete path of the traffic. However, Tor is not only used for good; it is often misused by criminals and hackers in order to remotely direct and instruct infected machines [41]. For example, researchers at Kaspersky Lab have published a research describing *64-bit version of the Zeus Trojan* that sends traffic through Tor and creates Tor hidden services to hide the attacker's position [42]. Another example is *Trojan.Tbot* malware that uses Tor network to communicate with its C&C server [43].

This module detects any connection to or from Tor network. It is, similar to malicious IP detection module, based on a list of publicly published Tor servers [44]. As it is shown in Figure 6, the network traffic is processed and the source and destination IP addresses for each connection are matched with Tor server list. The list of Tor servers is automatically updated each day and the detection is in the real time.

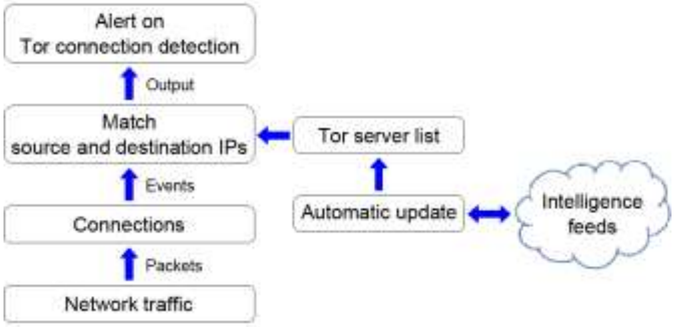

Figure 6. Architecture of Tor connection detection module.

V. Conclusion

In this paper, a novel approach for botnet C&C traffic detection has been proposed and the research questions have been presented. The proposed approach is based on voting between four detection modules to raise an alert on C&C traffic detection. Each detection module processes the network traffic and aims to detect one technique used for C&C communications. It is assumed that the opportunity for using this approach in C&C traffic detection would highly reduce the false positive rate of the detection system.

References

[1] I. Ghafir, J. Svoboda, and V. Prenosil, “A survey on botnet command and control traffic detection,” International Journal of Advances in Computer Networks and its security (ICJNS), vol. 5, no. 1, 2015.

[2] I. Ghafir, V. Prenosil, A. Alhejailan, and M. Hammoudeh, “Social engineering attack strategies and defence approaches,” in 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud). IEEE, 2016, pp. 145–149.

[3] I. Ghafir and V. Prenosil, “Advanced persistent threat attack detection: An overview,” International Journal of Advances in Computer Networks and Its Security (IJCNS), no. I, 2014.

[4] I. Ghafir and V. Prenosil, “Proposed approach for targeted attacks detection,” in Advanced Computer and Communication Engineering Technology. Springer, 2016, pp. 73–80.

[5] I. Ghafir and V. Prenosil, “Malicious file hash detection and drive-by download attacks,” in Proceedings of the Second International Conference on Computer and Communication Technologies. Springer, 2016, pp. 661–669.

[6] B. AsSadhan, J. M. Moura, D. Lapsley, C. Jones, and W. T. Strayer, “Detecting botnets using command and control traffic,” in Network Computing and Applications, 2009.

[7] S. Mount, M. Hammoudeh, S. Wilson, and R. M. Newman, “Csp as a domain-specific language embedded in python and jython.” in CPA, 2009, pp. 293–309.

[8] M. Hammoudeh, O. Aldabbas, S. Mount, S. Abuzour, M. Alfawair, and S. Alratrout, “Algorithmic construction of optimal and load balanced clusters in wireless sensor networks,” in Systems Signals and Devices, 2010 7th International Multi-Conference on. IEEE, 2010, pp. 1–5.

[9] M. Hammoudeh, R. Newman, C. Dennett, and S. Mount, “Interpolation techniques for building a continuous map from discrete wireless sensor network data,” Wireless Communications and Mobile Computing, vol. 13, no. 9, pp. 809–827, 2013.

[10] M. Hammoudeh, R. Newman, and S. Mount, “An approach to data extraction and visualisation for wireless sensor networks,” in Networks, 2009. ICN’09. Eighth International Conference on. IEEE, 2009, pp. 156–161.

[11] A. Abuarqoub, M. Hammoudeh, and T. Alsboui, “An overview of information extraction from mobile wireless sensor networks,” in Internet of Things, Smart Spaces, and Next Generation Networking. Springer Berlin Heidelberg, 2012, pp. 95–106.

[12] M. Hammoudeh and R. Newman, “Information extraction from sensor networks using the watershed transform algorithm,” Information Fusion, vol. 22, pp. 39–49, 2015.

[13] S. Jabbar, A. A. Minhas, R. A. Akhtar, and M. Z. Aziz, “Rear: Realtime energy aware routing for wireless adhoc micro sensors network,” in Dependable, Autonomic and Secure Computing, 2009. DASC’09. Eighth IEEE International Conference on. IEEE, 2009, pp. 825–830.

[14] S. Jabbar, A. A. Minhas, M. Imran, S. Khalid, and K. Saleem, “Energy efficient strategy for throughput improvement in wireless sensor networks,” Sensors, vol. 15, no. 2, pp. 2473–2495, 2015.

[15] M. Feily, A. Shahrestani, and S. Ramadass, “A survey of botnet and botnet detection,” in Emerging Security Information, Systems and Technologies, 2009. SECURWARE’. 2009, pp. 268–273.

[16] Z. Zhu, G. Lu, Y. Chen, Z. Fu, P. Roberts, and K. Han, “Botnet research survey,” in Computer Software and Applications, 2008. COMPSAC’08. 32nd Annual IEEE International. IEEE, 2008, pp. 967–972.

[17] K.-K. R. Choo, Zombies and botnets. Australian Institute of Criminology, 2007.

[18] H. Choi, H. Lee, H. Lee, and H. Kim, “Botnet detection by monitoring group activities in DNS traffic,” in Computer and Information Technology, 2007. CIT 2007, pp. 715–720.

[19] I. Ghafir, J. Svoboda, V. Prenosil, and M. Hammoudeh, “A survey on network security monitoring systems,” in 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud). IEEE, 2016, pp. 77–82.

[20] J. Svoboda, I. Ghafir, V. Pˇrenosil et al., “Network monitoring approaches: An overview,” International Journal of Advances in Computer Networks and Its Security (IJCNS), vol. 5, no. I, 2015.

[21] P. Baecher, M. Koetter, T. Holz, M. Dornseif, and F. Freiling, “The nepenthes platform: An efficient approach to collect malware,” in Recent Advances in Intrusion Detection. Springer, 2006, pp. 165–184.

[22] H. Artail, H. Safa, M. Sraj, I. Kuwatly, and Z. Al-Masri, “A hybrid honeypot framework for improving intrusion detection systems in protecting organizational networks,” computers & security, vol. 25, no. 4, pp. 274–288, 2006.

[23] G. J. Rishi, “identify bot contaminated hosts by irc nickname evaluation,” Cambridge, MA: Proceedings of the HotBots, vol. 7.

[24] P. Wurzinger, L. Bilge, T. Holz, J. Goebel, C. Kruegel, and E. Kirda, “Automatically generating models for botnet detection,” in Computer Security–ESORICS 2009. Springer, 2009, pp. 232–249.

[25] A. Karasaridis, B. Rexroad, and D. Hoeflin, “Wide-scale botnet detection and characterization,” in Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets, vol. 7. Cambridge, MA, 2007.

[26] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, “A survey of intrusion detection techniques in cloud,” Journal of Network and Computer Applications, vol. 36, no. 1, pp. 42–57, 2013.

[27] S. Balram and M. Wilscy, “User traffic profile for traffic reduction and effective bot c&c detection.” IJ Network Security, vol. 16, no. 1, pp. 46–52, 2014.

[28] H. Xiong, P. Malhotra, D. Stefan, C. Wu, and D. Yao, “User-assisted host-based detection of outbound malware traffic,” in Information and Communications Security. Springer, 2009, pp. 293–307.

[29] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, “Intrusion detection system: A comprehensive review,” Journal of Network and Computer Applications, vol. 36, no. 1, pp. 16–24, 2013.

[30] G. Gu, R. Perdisci, J. Zhang, W. Lee et al., “Botminer: Clustering analysis of network traffic for protocol-and structure-independent botnet detection.” in USENIX Security Symposium, vol. 5, no. 2, 2008, pp. 139–154.

[31] Paxson, V.: Bro: a system for detecting network intruders in real-time. Computer networks 31(23), 2435-2463 (1999).

[32] The-Bro-Project: The bro network security monitor. https://www.bro.org/. Accessed: 12-07-2016.

[33] Computer-Incident-Response-Center-Luxembourg: Source and destination ip blocklist. http://misp.circl.lu/. Accessed: 12-07-2016.

[34] Malware-Domain-List: Malware domain list, hostlist ip. http://www.malwaredomainlist.com/hostslist/ip.txt. Accessed: 12-07-2016.

[35] Sentinel-IPS: ci-badguys.txt. http://www.ciarmy.com/list/ci-badguys.txt. Accessed: 12-07-2016.

[36] OpenBL: Abuse reporting and blacklisting. http://www.openbl.org/lists/base.txt. Accessed: 12-07-2016.

[37] Mandiant: Apt1: Exposing one of china's cyber espionage units. http://intelreport.mandiant.com/. Accessed: 12-07-2016.

[38] Security-Affaiairs: SSL blacklist a new weapon to fight malware and botnet. http://securityaaffaiirs.co/wordpress/26672/cyber-crime/ssl-blacklist-new-weapon-fight-malware-botnet.html. Accessed: 12-07-2016.

[39] Stone-Gross, B., Cova, M., Gilbert, B., Kemmerer, R., Kruegel, C., Vigna, G.: Analysis of a botnet takeover. Security & Privacy, IEEE 9(1), 64-72 (2011).

[40] Chakravarty, S., Portokalidis, G., Polychronakis, M., Keromytis, A.D.: Detection and analysis of eavesdropping in anonymous communication networks. International Journal of Information Security pp. 1-16 (2014).

[41] Jagerman, R., Sabee, W., Versluis, L., de Vos, M., Pouwelse, J.: The fifteen year struggle of decentralizing privacy-enhancing technology. arXiv preprint arXiv:1404.4818 (2014).

[42] Kaspersky-Lab-ZAO: The inevitable move - 64-bit zeus enhanced with tor. http://securelist.com/blog/events/58184/the-inevitable-move-64-bit-zeus-enhanced-with-tor/. Accessed: 12-07-2016.

[43] NETRESEC: Detecting tor communication in network traffic. http://www.netresec.com/?page=Blogn&month=2013-04n&post= Detecting-TOR-Communication-in-Network-Traffic. Accessed: 12-07-2016.

[44] Kowalski, J.B.: Tor network status. http://torstatus.blutmagie.de/. Accessed: 12-07-2016.