A Literature Review of Historical and Detection Analysis of Botnets Forensics

Ade Kurniawan*, Ahmad Fitriansyah
Department of Informatics Engineering, Universal University, Batam, Indonesia.

* Corresponding author. Tel.: +6287863777788; email: ade.kurniawan@uvers.ac.id
Manuscript submitted June 14, 2018; accepted September 10, 2018.
doi: 10.17706/ijcce.2018.7.4.128-135

Abstract: In 1988 Botnet was originally developed only as a virtual individual on IRC channels (Internet Relay Chat) while the owner was busy doing activities elsewhere. Furthermore, Botnet incarnated as a frightening specter on the global Internet network with the impact of damage and anxiety on government institutions, private companies, education, health, and even households. According to FBI and cybersecurity agencies estimating losses, the growth and number of attacks using Botnet have reached a worrying point. Therefore, a literature review is needed to uncover all the important spectrums in the Botnet. Hence, this paper will present the two most important spectrums in the Botnet: Historical and Forensic Analysis Botnet. The history of Botnet spectrum was originally developed in 1988 and it was used by operators while they were on IRC channels until the 2006 era - the emergence of the ZeuS Botnet that had made global losses of approximately $110 billion globally. On the spectrum of Forensic Analysis, it is divided into three important aspects: Forensic Analysis Botnet from Software aspect, Forensic Analysis Botnet from Detection aspect, and Botnet Forensic Analysis from Behavior aspect.

Key words: Botnet, detection, forensic analysis, historical.

1. Introduction

Botnets have caused huge financial losses for corporations, governments, Internet service providers, education, and even home users [1], [2]. In June 2014, the FBI estimates that Botnet GameOver ZeuS is responsible for a loss of more than $ 100 million against damage to banking services [3], [4]. In the report, the FBI estimates that 500 million computers are compromised each year, causing a loss of about $ 110 billion globally [4].

A Botnet is a compromised computer network that is controlled by an attacker from a location via a command and control (C & C) channel [5], [6]. The term "Bot" is derived from the English word "Robot" specially designed to perform several functions and perform tasks similar to robots that are performed automatically [7]. Botnets are designed to communicate in a highly confidential manner, its task of avoiding detection on the network then spreading the infection autonomously and the attacker forwarding the commands to the bots via a randomly compromised host on the Internet [8]. This mechanism puts the attacker behind the scenes making it very difficult to trace to the botmaster.

Attackers use Botnets for various malicious applications such as launching Distributed Denial of Service (DDoS) attacks, spamming, phishing, spying, password attacks, and other criminal activities [1], [5], [9]. Botnet detection analysis, traceback or C & C server location of attackers is very difficult to trace, making

law enforcement problems very challenging [10]. Therefore, forensic Botnets are required to collect evidence that will be used to analyze Botnet thoroughly aimed at improving security tools and techniques in the future [11].

Literature Review (LR) is a very determinant factor in a study because a good LR will follow the latest research trends and seek to present state-of-the-art from a comprehensive research topic [12]. Literature Review is a critical study to summarize, make a critical analysis of the synthesis that discusses published information in the field of research and a certain period of time [13]. The findings in this paper using the Literature Review method yield two important spectra: Historical and Detection Analysis from Botnet Forensic.

The first spectrum was to convey the evolutionary history of the early Botnets started by Jarkko "WiZ" Oikarinen from the University of Oulu, Finland in 1989 up to 2006 with Botnet named "ZeuS" which is the most famous Botnet and made the greatest financial loss. The second spectrum is the Botnet Detection Analysis Method. The spread and infecting of Botnets are getting more sophisticated to attend detection, therefore forensic analysis is needed to know where, time, and how to generate forensic reports that can be retained in court.

The structure of this paper is in Section 2 Botnet History passes briefly on the history of Botnet. Section 3 Botnet Detection Analysis Result will explain of briefly on Botnet Detection Analysis. Section 4 is the Conclusion and Future Work of this paper.

2. Botnet History

Bots were originally developed as "virtual individual" in IRC (Internet Relay Chat) channels and doing things while owner owners were busy elsewhere [14]. Botnet evolution seen in Fig. 1, the bot was discovered in August 1988 by Jarkko "WiZ" Oikarinen from the University of Oulu, Finland [15]. The compromised computer is called "Bot", and the attacker controlling the Botnet is called "Botmaster"[14].

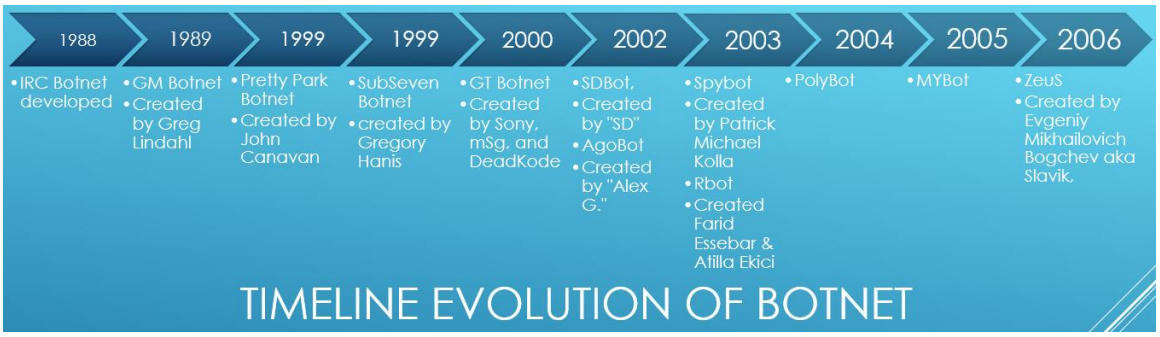

Fig. 1. Timeline evolution from Botnet.

The original IRC bot called GM, according to Wikipedia, was developed in 1989, by Greg Lindahl: an IRC server operator. The first bots were really robots that manifested themselves to other IRC users as other users. One of the prominent characteristics of Botnets is using the command and control channels (C & C) [16]. The main purpose of this communication channel is to send botmaster commands to Botnet [16].

Botnet evolution began in May 1999, Pretty Park (worm) written in Delphi programming language by John Canavan [17]. Introduce the concept of connecting machines to victims through IRC channels to listen to malicious commands Pretty Park has some common functions and concepts found in bots today, including [17]:

1) Ability to retrieve computer name, OS version, user information, and other basic system information.
2) Ability to search and retrieve email addresses and login names.

3) Ability to retrieve username, password, and network settings.
4) Ability to update its own function and/or updating.
5) Ability to upload and/or download files.
6) Ability to direct (tunneling) traffic.
7) Ability to launch various DDoS attacks.

SubSeven Trojan / Bot and some other worms like Jobbo have exploited vulnerabilities on the IRC client side especially mIRC that makes many clients infected with a backdoor at the end of 1999 created by Gregory Hanis [18]. In June 1999, version 2.1 of the Trojan SubSeven was released. This release is important because it allows the SubSeven server to be controlled from bots connected to the IRC server. This sets the network for all the upcoming malicious Botnet. SubSeven is a remote-controlled Trojan, also written in Delphi, touted by its author as a Remote Administration Tool (RAT). This bot has the ability to steal keywords, generate log keys, and hide its identity. SubSeven gives the botmaster full control over the infected system [18].

Sony, mSg, and DeadKode [17], created a Botnet client called "Global Threat" (GT) Bot based on mIRC clients in 2000. Microsoft Internet Relay Chat (mIRC) is an IRC client software package, mIRC has two important characteristics for Botnet construction: run scripts inside the IRC server, and support Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) connections. GT bot has the following capabilities:

1) Port Scanning: Scanning for open ports.
2) Flooding: Perform DDoS attacks.
3) Cloning: Cloning the connection to the IRC server is over and above the first connection.

Bot technology based on mIRC can be said member of GT Bot Familly. The spreading process of the GT Bot does not spread itself directly instead they will use social engineering techniques. Social Engineering commonly used to infect the system is the e-mail that claimed to come from a security vendor. If a user clicks on a link, they are directed to a website that will send Malware to the victim.

The appearance of SDBot was known in 2002. Written in C ++ by a Russian programmer known as "SD" [19]. SDBot makes great strides in the evolutionary bot chain where the author of SDBot publishes a website and opens and releases the source code to the public so that the impact can be accessed by many hackers [20]. Among cyber-criminals / hackers, the SDBot source code is very easy to modify and maintenance because it only produces small files, 40KB [21]. 678,000 PCs were detected infected in June in 2006 according to Microsoft's Malicious Software Removal Tool Report [21].

In 2002, Created by "Alex G." Botnet Agobot aka Gaobot by presenting modular design and significant functionality [19]. There are three modules contained in Agobot:

1) The first module: delivered contains IRC bot clients and remote access backdoor.
2) The second module: attack and kill the antivirus process.
3) The third module: prevent victims from accessing Web sites, usually antivirus vendor sites.

The workings of the Agobot module is to take the next module when the previous module has completed its main task. Agobot uses IRC for C & C, but its proponents use peer-to-peer (P2P) file-sharing applications such as Kazaa, Grokster, and Bear Share [22]. Agobot uses IRC for C & C, but its proponents use peer-to-peer clients from Agobot which can be ordered via IRC. Agobot can also open remote backdoor access to allow each client to be accessed directly by cyber-criminals. Agobot has the following capabilities: (P2P) file-sharing applications such as Kazaa, Grokster and Bear Share:

1) Scan for vulnerabilities
2) launch various DDoS attacks
3) Search for compact disk (CD) Keys in client games.

4) Stop the monitoring and antivirus process.
5) Change the host file to prevent access to the antivirus website
6) Hide by using rootkit technology

Spybot appeared in 2003, with open source code which is a derivative of SDBot [23]. Spybot is easy to customize and adds Spyware capabilities: like collecting event logs, data from web forms, e-mail address lists, and list of visited URLs. Spybot has the following capabilities:

1) Use Port scanning to look for open ports
2) Launch DDoS attacks such as UDP and SYN Flooding
3) Use social engineering to attract P2P users to download Spybot modules
4) Deceive users by installing a fake error message after the user runs an infection module

In 2003, RBot first appeared with 1.9 million PCs infected [21]. RBot scans the system on ports 139 and 445 on Microsoft shares and tries to guess weak password from the administrator.

In March of 2004, Polybot appeared from the Agobot codebase [24]. It is named "metamorphism" because of its ability to perform in various forms. Polybot changes the code on each infection by wrapping the code that is compiled in the "envelope" technique and re-encrypting the entire file each time it runs [24]. Despite successfully hiding its identity, the binary bot can still be detected while executing due to its memory-based detection approach [25]

In 2006, created by Vgeniy Mikhailovich Bogchev aka Slavik, ZeuS surfaced is the most famous Botnet by cybercriminals around the world [26] and is designed to steal important bank account information or credentials and other tasks [26]. ZeuS likely came from Russia or Eastern Europe which is also known as ZBot, PRG, Wsnpoem, Gorhax, and Kneber [26]. ZeuS targets Microsoft Windows operating system with the main purpose of stealing online credentials. This is done by techniques such as logging keystroke, capture screen, or methods such as HTML injection into web pages and exploiting browser-side vulnerabilities [10]. ZeuS collects various system information along with its password and encryption certificate and sends it to the command-and-control server. The server can also send configuration files to the client bot to determine the list of actions to be performed.

3. Botnet Detection Analysis

Intrusion Detection is to determine bad traffic and good traffic. Intrusion Detection can be very difficult, because Botnets are more advanced and sophisticated to avoid detection. To address this, there are two general categories of Botnet detection on the network: Signature-Based and Heuristic-Base [27].

The signature-based intrusion detection system relies on the success of identifying the signature pattern of traffic as it traverses the network. The working principle of the Signature-Based approach equals hosted antivirus programs. When intrusion detection occurs at the network level, Signature-Based can be a port or IP address of an intrusion source. For instance, traffic coming from suspected IP addresses and opening certain ports e.g. port 6667 for the Internet Relay Chat protocol, may need to be marked immediately for wakefulness. Heuristic Approach is an intrusion detection system that sorts out normal and abnormal traffic activities, is also called an anomaly-based detection approach. It is based on classifying normal network traffic, and abnormal traffic as an anomaly. If abnormal traffic is detected by IDS, alerts will appear.

Botnet forensics includes analysis of various Botnet components including C & C servers/channels and compromised hosts. Furthermore, the forensic analysis process includes analysis of bots function, C & C / traffic servers, Botnet attacks, and Botnet design [26]. The nature and behavior of previous Botnet attacks will greatly help identify the intent, purpose, and method of the attacker. Forensic analysis on Botnet as shown in Fig. 2 aims to also classify and link Detection of Botnet attacks.

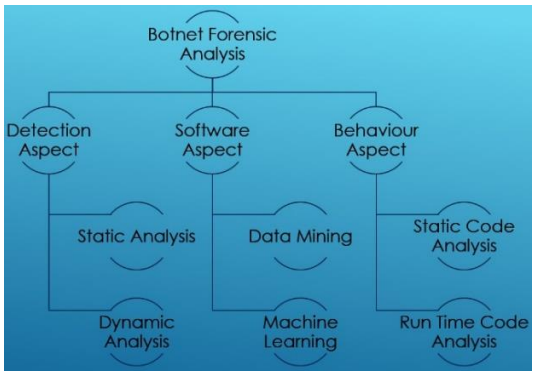

Fig. 2. Three aspects of Botnet forensic analysis.

The explanation of Fig. 2 above about the three aspects of the analysis of Botnet Forensic is:

This analysis can be done through various computing software such as Data Mining or Machine Learning techniques [28]. TCPDump, Wireshark, TCPFlow, TCPTrace, OllyDbg, IDA Pro, NetFlow, TCPXtract, and Snort are tools to support Botnet attack analysis. Various Botnet forensic detection analysis techniques can be studied into two broad categories: Static Analysis and Dynamic Analysis [29].

Static Analysis is a method to understand malware behavior without running it [29]. This method includes analysis of log files, file system analysis, and presence of malware. To gain more insight into malware, reverse engineering of internal structures is done so as not to possess a potential threat to the original environment of the victim.

Dynamic Analysis This approach is done by executing a binary Botnet in a controlled environment, eg Sandbox [30]. This approach examines and monitors external views with the Blackbox testing approach, particularly monitoring Botnet behavior and operations. This method complements static analysis and performance aimed at understanding Botnet functions and features. It is often done in a virtual environment and controls various parameters to study behavior in different situations. Unfortunately, some Botnets are even coded to recognize virtual environments and thus immediately log out and delete logs or provide fake information [25].

Botnet analysis includes two main ways: check the source code and behavior. Botnets always try to avoid detection and also make bots code analysis more difficult [29].

1) Static Code Analysis: There are many tools available for code analysis. Some simple tools, while others require significant effort by investigators, such as Hex WinHex editor. Investigators need a copy of the malware source code for analysis, the malware code can be either a script or a compiled binary code. Simple script file analysis is easier, while binary files that are compiled require a binary decompiler routine that is more difficult to understand/determine its features and functions. Unfortunately, the use of multiple parsers requires a strong programming understanding, and then some Botnet binaries prevent the use of decomposition. Botnet authors also use packers to minimize binary bot code size, disguise binary data, and restrict decompiler functionality [13]. Furthermore, the combination of packaging utilities with encryption makes reverse engineering more difficult, as well as the decryption effectiveness. The use of unusual packaging tools even makes the task of unpacking more difficult. The static analysis tool identifies runtime errors and security vulnerabilities that provide valuable insights and information such as symbol tables, valuable parse trees for Botnet analysis.
2) Run Time Code Analysis: Static code analysis is not a complete solution for obtaining Botnet evidence. Sometimes static bot binary testing cannot analyze bots. To avoid detecting the presence of bots, the binary code itself uses different packets and encryption methodologies. Monitoring

malware actions needs to be done on the victim system, such as file system changes, registry changes, and network activity to gather additional information about malware behavior. Timed code analysis executed will be helpful to identify Botnet information such as bot connection location to accept commands, as well as any username, handle, IRC channel, and password.

Botnets can be detected by analyzing flow characteristics, several strategies have been proposed for analyzing and defeating Botnets. Barford and Yegneswaran [19], do an in-depth analysis of malicious bot source programs and find that the architecture and implementation of Botnet are very complex. They find that Agobot uses tests for the debugger and VMware, kills antivirus, or provides fake information.

Rossow *et al*. [31], proposed a graphical P2P-based Botnet model to capture the properties and vulnerabilities of Botnets. They also analyzed the robustness of Botnets against removal efforts. Furthermore, they propose two counting techniques: the P2P node and the sensor injection to measure the size of the Botnet. Botnet P2P is vulnerable to command injection attacks, for example, Botnet Sality uses reputable peer schemes and is quite resistant to intrusion attacks. There are several possibilities through reverse engineering Domain-Generation-Algorithm (DGA) to register domains before bots can communicate for the future that can help ignore access to bots under their own control. This provides a different perspective between the infected host and the botmaster.

A Machine-Learning Botnet-based detection concept was offered by G. Kirubavathi and R. Anitha [28], the result of the research was a pattern of botnet zombie communication traffic used by the attackers to have the same pattern. The advantages of detection methods using Decision tree approaches, Naive Bayesian and SVM are lightweight, have the ability to detect and classify where traffic is normal, with encrypted bot communications traffic with high detection accuracy and low false positive rates.

A study was conducted by Chen and Lin [8], they proposed an efficient botnet detection system on the network with anomalous traffic monitoring approach. The detection methods offered used two anomalous behaviors, homogeneous responses and group activities. The proposed anomaly score calculated the anomaly level based on the above anomalous behavior. Performance evaluation which was performed on real cases showed that anomaly-based detection could identify infected bots efficiently. Experimental evaluations based on different datasets suggested that the proposed method was capable of achieving the true positive rate of over 90% and a false positive rate below 7%.

The new generation of Botnets is generally very difficult to be detected. Botnets have the ability to survive and find C & C servers by utilizing Domain Generation Algorithm (DGA). Therefore, Wang *et al*. [5] propose DBod to detect a Botnet based Domain-Generation-Algorithm (DGA). DGA uses Domain-Algorithm to avoid detection by generating large Command and Control (C & C) domain-server lists. Botnets DGA-based are very difficult to be detected by using traditional defense mechanisms and hence they propose a Botnet DGA-based detection scheme which is called "DBod". Based on the analysis of DNS traffic query behavior, Botnet DGA-based detection is performed inside the network. Each host is grouped into groups which are corresponding to their behavior. The intensity between them and each cluster is then identified as dangerous or benign depending on the query time distribution and query count distribution characteristics.

4. Conclusion and Future Work

This paper aims to summarize and convey from two important spectrums of Botnets: the history of 1998-2007 and the analysis of software, engineering, and behavioral aspects. The vote was made between 1988 – 2007; whereas, in 1989, the first Botnet surfaced with IRC channels that initially played a substitute role as an IRC operator until the 2006 era with the presence of ZeuS Botnets, that cast the world and descend from the code, is still in use and developed to date. Botnet analysis software can be performed with

Data Mining and Machine Learning. On the other hand, Botnet forensic detection techniques are divided into two broad categories: Static Analysis and Dynamic Analysis. Static analysis briefly understands the behavior of malware without running it while dynamic analysis executes binary Botnet in a controlled environment or run in a Virtual machine or Sandbox. While on the analysis of code, Botnet is divided into two parts: Static Code Analysis and Run Time Code Analysis.

For further research, it is suggested to summarize the derivatives of the ZeuS Botnet which have undergone many developments after 2006 until 2018 from either one of the source code aspects, the attack architecture, or its forensic detection analysis.

Acknowledgment

We are grateful to thank you for support from Universal University, Batam, Indonesia. This work is sponsored by "Universal University Research Funds 2018" (Grant#025/LPPM/UVERS/XI/2018).

References

[1] Sgouras, K. I., Kyriakidis, A. N., & Labridis, D. P. (2017, Oct.). Short-term risk assessment of Botnet attacks on advanced metering infrastructure. *IET Cyber-Physical Syst. Theory Appl., 2(3)*, 143-151.

[2] Kurniawan, A., Riadi, I., & Luthfi, A. (2017). Forensic analysis and prevent of cross site scripting in single victim attack using open web application security project (Owasp) framework. *J. Theor. Appl. Inf. Technol., 95(6)*, 1363-1371.

[3] Rodríguez-Gómez, R. A., Maciá-Fernández, G., & García-Teodoro, P. (2013, Aug.). Survey and taxonomy of Botnet research through life-cycle. *ACM Comput. Surv., 45(4)*, 1-33.

[4] FBI. (2013). GameOver Zeus (GOZ) malware and botnet architecture. *GameOver Zeus/Cryptolocker graphic.*

[5] Wang, T. S., Lin, H. T., Cheng, W. T., & Chen, C. Y. (2017, Jan.). DBod: Clustering and detecting DGA-based botnets using DNS traffic analysis. *Comput. Secur., 64*, 1-15.

[6] Chen, C. M., & Lin, H. C. (2015, Apr.). Detecting Botnet by anomalous traffic. *J. Inf. Secur. Appl., 21*, 42-51.

[7] Zeidanloo, H. R., Manaf, A. B., Vahdani, P., Tabatabaei, F., & Zamani, M. (2014). Botnet detection based on traffic monitoring. *Architecture, 1(8)*, 97-101.

[8] Chen, C. M., & Lin, H. C. (2015). Detecting botnet by anomalous traffic. *J. Inf. Secur. Appl., 21*, 42-51.

[9] Rawat, R. S., Pilli, E. S., & Joshi, R. C. (2018). Survey of peer-to-peer Botnets and detection frameworks. *International Journal of Network Security, 20(3)*, 547-557.

[10] Nagendra Prabhu, S., & Shanthi, D. (2015). Examining Zeus Botnet by adopting key extraction and malicious traffic detection framework using DNS. *Int. J. Appl. Eng. Res., 10(3)*, 6987-7007.

[11] Garg, S., & Sharma, R. M. (2017). Anatomy of Botnet on application layer: Mechanism and mitigation. *Proceedings of the 2017 2nd International Conference for Convergence in Technology (I2CT)* (pp. 1024-1029).

[12] Siddaway, A. (2014). What is a systematic literature review and how do I do one? *Univ. Stirling, (2)*, 1-13.

[13] Wahono, R. S. (2015). A systematic literature review of software defect prediction : Research trends, datasets, methods and frameworks. *J. Softw. Eng., 1(1)*, 1-16.

[14] Zimba, A., Chen, H., & Wang, Z. (2018). Bayesian-boolean logic security assessment model for malware-free intrusions. *International Journal of Network Security, 20(3)*, 558-567.

[15] Hyslip, T., & Pittman, J. (2015). A survey of Botnet detection techniques by command and control Infrastructure. *J. Digit. Forensics, Secur. Law, 10(1)*, 7-26.

[16] Cho, C. Y., Babi ć, D., Shin, E. C. R., & Song, D. (2010). Inference and analysis of formal models of Botnet

command and control protocols. *Proceedings of the 17th ACM Conference on Computer and Communications Security - CCS '10* (p. 426).
[17] Schiller, C., Binkley, J., & Evron, G. (2007). *Botnets: The Killer Web Applications* (1st ed.). Syngress.
[18] Crapanzano, J. (2003). Deconstructing subseven, the Trojan horse of choice. *SANS Inst. InfoSec Reading Room*.
[19] Barford, P., &Yegneswaran, V. (2007). An inside look at Botnets. *Malware Detect., 27*, 171-191.
[20] Kharouni, L., & Micro, T. (2009, December). SDBOT IRC Botnet SDBOT IRC Botnet continues to make waves. *A Trend Micro White Paper*.
[21] Thing, V. L. L., Sloman, M., & Dulay, N. (2007). A survey of bots used for distributed denial of service attacks. *IFIP International Federation for Information Processing, 232*, 229-240.
[22] Lee, C. P. (2009, May). Framework for Botnet emulation and analysis. *Challenges*.
[23] Li, C., Jiang, W., & Zou, X. (2009). Botnet: Survey and case study. *Proceedings of the 2009 Fourth International Conference on Innovative Computing, Information and Control (ICICIC)* (pp. 1184-1187).
[24] Midha, K., Rajawat, K., & Rathore, V. S. (2012). An introduction to Botnet attacks and it's solutions. *Int. J. Comput. Appl. Inf. Technol., 1(2)*, 2278-7720.
[25] Cao, H., Zhao, J., Zhu, P., Lu, X., & Zhao, C. (2013). Worm detection without knowledge base in industrial networks. *J. Commun., 8(11)*, 716-723.
[26] Layton, R., & Azab, A. (2014). Authorship analysis of the Zeus Botnet source code. *Proceedings of the 2014 Fifth Cybercrime and Trustworthy Computing Conference* (pp. 38-43).
[27] Messier, R. (2017). Network forensics. *Computer and Information Security Handbook*.
[28] Kirubavathi, G., & Anitha, R. (2016). Botnet detection via mining of traffic flow characteristics. *Comput. Electr. Eng., 50*, 91-101.
[29] Joshi, R. C., & Pilli, E. S. (2016). *Fundamentals of Network Forensics*. London: Springer London.
[30] Kurniawan, A., & Riadi, I. (2018). Detection and analysis cerber ransomware using network forensics behavior based. *Int. J. Netw. Secur., 20(5)*, 1-8.
[31] Rossow, C., *et al.* (2013). SoK: P2PWNED — Modeling and evaluating the resilience of peer-to-peer Botnets. *Proceedings of the 2013 IEEE Symposium on Security and Privacy* (pp. 97-111).

Ade Kurniawan is a lecture at Department of Informatics Engineering, Universal University. He is a graduated master of computer science from Department of Information Engineering, Islamic University of Indonesia (UII). At present, he is working toward a Ph.D. degree in the Dept. of Electrical Engineering at Nagoya University. His research interested in cyber security, deep learning, digital forensics, machine learning, and network forensics.

Ahmad Fitriansyah is a lecture at Department of Informatics Engineering, Universal University. He is a graduated master of computer science from Department of Informatics Engineering, STTI Benarif Indonesia. His research interested in computer security, information system governance, and expert system.