A SIMULATION STUDY OF SDN DEFENSE AGAINST BOTNET ATTACK BASED ON NETWORK TRAFFIC DETECTION

Rahmadani Hadianto and Tito Waluyo Purboyo
Department of Computer Engineering, Faculty of Electrical Engineering, Telkom University Bandung, Indonesia
E-Mail: hadianto777@gmail.com

ABSTRACT

This paper discusses the Software Defined Networking (SDN) security experiment on Zeus Botnet attacks based on traffic behavior in the network. The development of SDN technology is increasingly in demand today, both on the researcher and industry side. This is inseparable from the SDN feature that gives the developer the flexibility to program the system inside. But on the other hand, the development of network technology cannot be separated from the threat of attack, especially Botnet attacks. Botnets are able to take control of the SDN network by attacking the control plane. This is possible when the botmaster enters a third party into the network and infects the associated device in the SDN network as a bot. This problem is categorized as Integrity in CIA triad (Confidentiality, Integrity, and Availability) used in the evaluation of security performance. Integrity in the CIA triad is a state of information that is always accurate and consistent until a recognized user makes a change. At the end of this paper will be explained about future research based on experimental test results.

Keywords: software defined networking, botnet, integrity, HTTP, zeus botnet.

INTRODUCTION

SDN [1] [2] [3] as the development of new technologies in the network has an important role in the centralized network configuration, control, and operation network. SDN can strengthen the network architecture, cost efficiency, and provide opportunities for application and network functionality by improving the software system [4]. SDN is not well known by the security community [4]. As an evidence of this statement, the number of SDN research papers on the security field is much less than in the improvement of network performance. in 2017, there is still less attention from a security researcher [4].

On the other side, SDN has potential to be attacked along with technological development. Botnet attack is one type of attack that became the main threat to the traditional network that has a chance to become a dangerous threat in SDN. Botnets are able to take control of the plane on the SDN, this has an impact on the integrity factor, Integrity is one of CIA (Confidentiality, Integrity, Availability) Triad component that used as security parameter for network security. Integrity in CIA triad is a condition where information is kept accurate and consistent unless authorize changes are made.

To overcome this problem will be tested botnet detection based on traffic properties on the network using sFlow. There are different characteristics of normal traffic and botnet-affected traffic; this is because the size of the data packet communication between bots and botmasters tends to remain different from normal traffic that tends to be random.

SDN feature

SDN provide benefit with four features that divided as follow:

a) Dynamic flow control

SDN has the capability to dynamically control data flow, with this feature SDN data flow can be controlled efficiently. Malicious network packets flow can be divided from authorized flows without setting up new middle boxes and it can be done with network device only. PBS (Programmable BYOD Security) [5] is an example of this feature, it grantsaccess operator for establishing and set up virtual Software Define Network.

b) Integrate control with extensive visibility

Supervise and manage the overall network to get all status of network and data flow.NetSecVisor [6] has the ability to utilize security devices and affect SDN function to be virtualized.

c) Network with programmability

Network function on SDN control plane can be programmed by Application Programming Interfaces (APIs) [7]. This feature makes network security development become easier by the setup program for security application. One of these feature examples is FRESCO [8].

d) Simple data package

Not like as in traditional network, SDN divided control and data plane. This feature makes SDN data plane become simpler and open opportunity for adding data plane extension for security function. OFX (Open Flow Extension Framework) [9] is an example of this feature, OFX has the ability to enable SDN security application with existing OpenFlow.

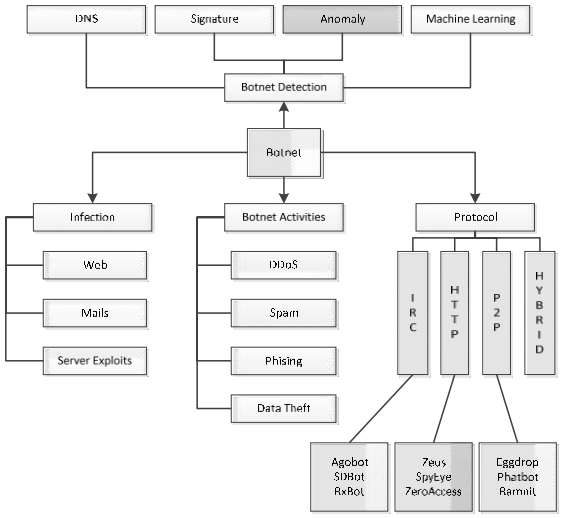

Figure-1. Botnet terminology [21].

Botnet overview

As defined before, Botnet is a collection of the computer that controlled by hackers. Botnets are divided into 3 main cycles in the way its phase of infection, C & C communication phase, and phase of the attack. [10] [11] [12]. Differences botnet than other types of attacks is the existence of C & C that work in giving orders from botmaster to bot. Bots always hide while looking for an unattended target, when bot find the target they will report to the botmaster [13]. Figure-1 shows the botnet terminology in general, some of which will be explained in the next explanation.

Architecture Botnet

Botmastermanages their C&C (Command and Control) to communicate with their bot indirectly in purpose to hide the C&C architecture. There arefour topologies on Botnet attack described as follow:

A. Botnet IRC

Figure-2 shows the Botnet attack scheme through IRC, IRC is a text message sent over the internet [14]. The protocol works based on the model of the client-server that is used on computers in a distributed network. The advantages of using IRC protocol on botnet are:

a) Has a low latency on the communication side

b) Real-time communication is done covertly

c) Have the ability to work in groups and in pairs

d) Easy to set up

e) Simple command instruction, basic command consists of, commands to connect to the server, post messages in the channel.

f) Communication run flexibly

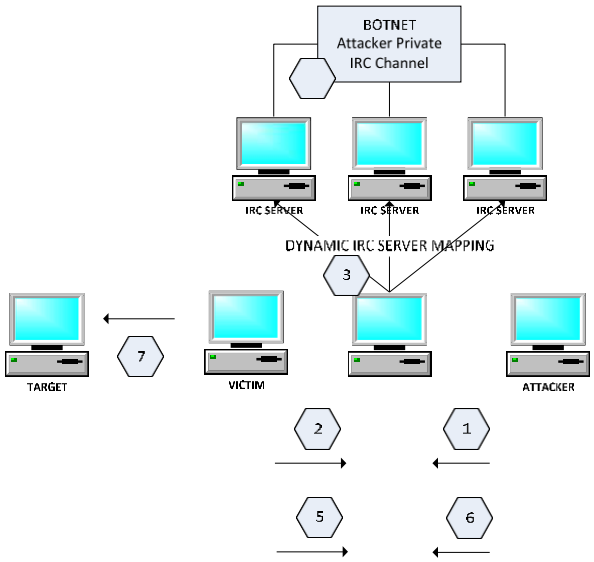

Figure-2. Botnet IRC [15].

B. Botnet HTTP

Since the use of IRC Botnets has been recognized, researchers are beginning to focus on tracking the whereabouts of IRC botnets. Therefore, the attacker initiated the use of HTTP Botnet which can be seen from Figure-3, in this way the botnet becomes hard to find.

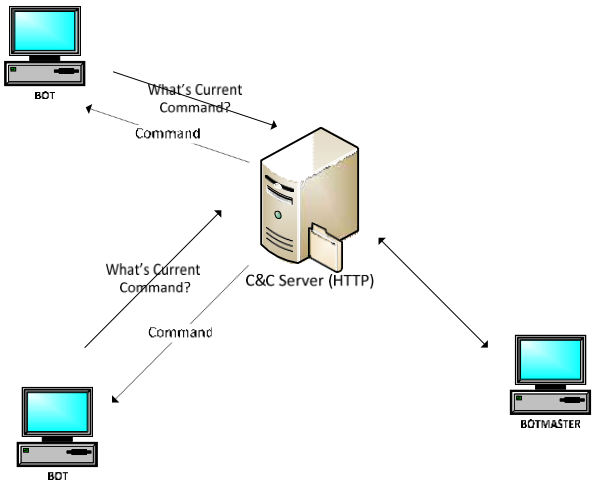

Figure-3. HTTP Botnet [10].

This is because the botnet uses the HTTP protocol to hide the existence of its bot in web traffic, so botnets can easily bypass the firewall using a port-based filtering mechanism and avoid detection from IDS. Under normal circumstances, the firewall will block both incoming traffic and outbound traffic to unwanted ports which include IRC ports. Several examples of HTTP botnet are Rustock, Zeus, Bobax, and ClickBot.

C. Decentralized / peer to peer (P2P) C&C server

The next development on botnet topology is decentralized C&C, this development is done because the centralized botnet has been recognized and resolved by the researcher. In decentralized botnets, attackers use P2P communication systems as a pattern of C&C, which in turn provides an advantage in avoiding network failures [12]. Every bot in this topology has limited size and periodically every bot connects to a neighbor to receive a command from botmaster. So Botmaster in this topology only needs to connect with one bot to send command.

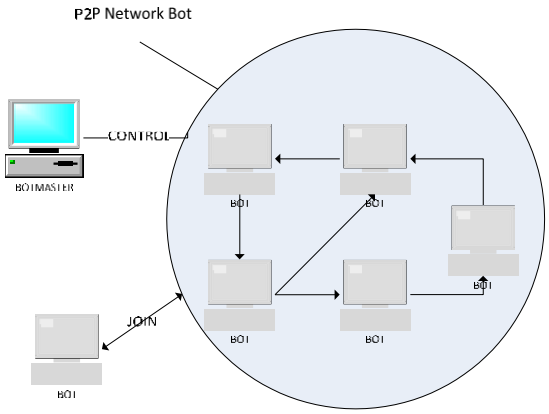

Figure-4. Peer to peer C&C server [13].

Figure-4 shows the absence of a centralized point in communication, each bot keeping in touch with other bots. On the other hand, the bot also acts as a client and server in forwarding information. The newly infected bot must know the other botnet connected to it. If the bot in the botnet is offline, other bots can continue operations under the botmaster command.

D. Hybrid

The next development of P2P botnet is Hybrid P2P botnet which can be seen from Figure-5 [16], bot on this type is classified 2 categories as follows:

a) Servant Bots - the first category referred to as servant bots, the bot acts as a client and server, which has static, routable IP addresses and is easily accessible from the rest of the internet.
b) Client Bots - Bots in the second category act as clients because they do not accept incoming connections [28]. This category contains the remaining bots as follows:

1) Bots with dynamic IP addressing properties
2) Bots with fixed route IP address
3) Bots that work behind a firewall, they cannot be connected to the global internet.

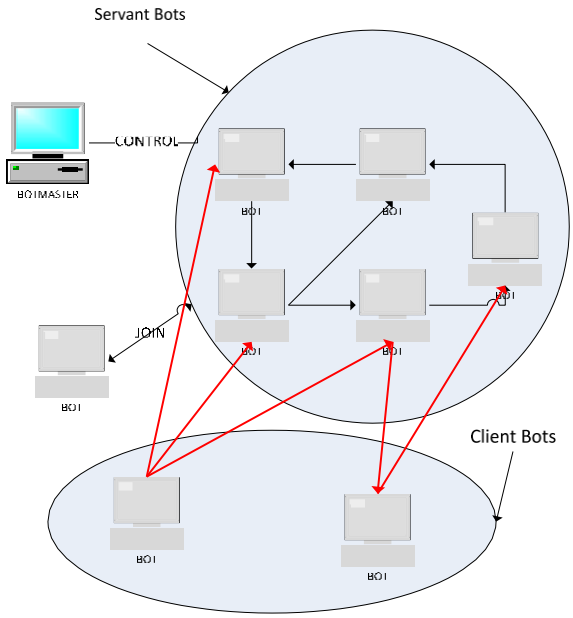

Figure-5. Hybrid P2P Botnet [13].

Zeus Botnet

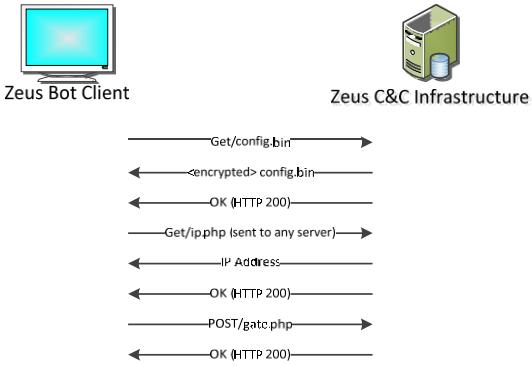

Figure-6. Zeus Botnet communication system [20].

Zeus Botnet is one type of HTTP-based botnet. Figure 6 shows the communication pattern between the C & C server with the infected machine, here is an explanation of the communication pattern:

a) The infected client starts the process by sending a Get /config.bin request to the C&C server. The configuration is a botnet configuration file.
b) C&C server reply request with config.bin that has been previously encrypted.
c) Bots receive messages and encrypt using encryption keys from within the binary bot file.
d) When the botmaster sets up the botnet, the infected machine will provide an external IP address and provide reports on the use of Network Address Translation (NAT).
e) Bot reports status to C&C server Post/gate .php

The entire process above is done with the same packet size every time, so it can be used as a detection parameter of the existence of botnet that will be described in the next section.

Botnet detection

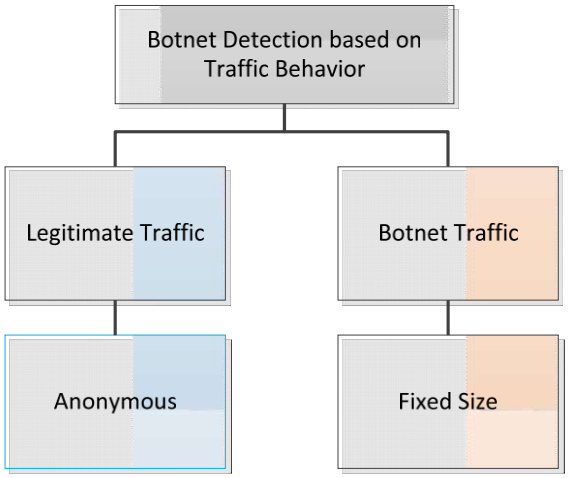

Figure-7. Botnet detection parameters.

In the botnet detection approach, there are several ways of determining it, among them is based on DNS, Signature, Anomaly, and Machine Learning [17] [18] [19] as shown in Figure-1. Figure-7 represents botnet detection based on the nature of traffic or can be said as a way of detection based on anomaly traffic. The size of data transmitted in communication between bots and botmasters tends to be fixed, in contrast to the size of the data transmitted normally within the network.

RESULT AND ANALYSIS

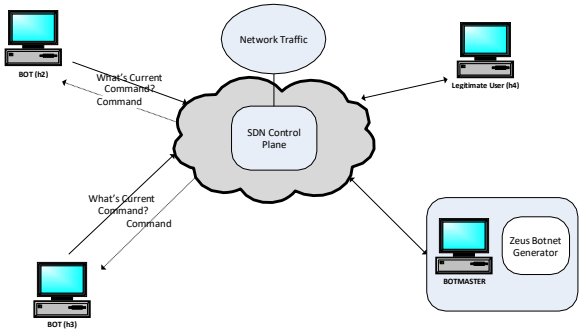

Figure-8. Botnet detection.

Figure-8 shows the scheme used in botnet detection testing experiments on SDN. The scheme consists of 3 computers (h2, h3, h4) connected with SDN Server (h1) and Botmaster. Computers h2 and h3 have been infected by botmaster and act as bots, h1 acts as a switch on the SDN server, while h4 is a computer that has not been infected by botmaster.

Any traffic that goes into control plane on SDN will be captured to be used in detecting the presence of botnet based on the incoming and outgoing traffic between host and switch.

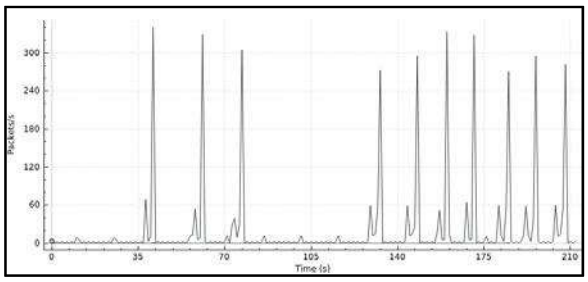

Figure-9. h2 (Bot) to h1 traffic.

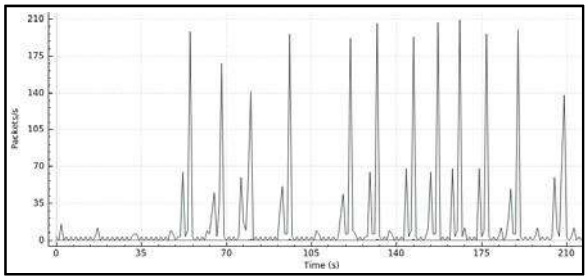

Figure-10. h3 (Bot) to h1 traffic.

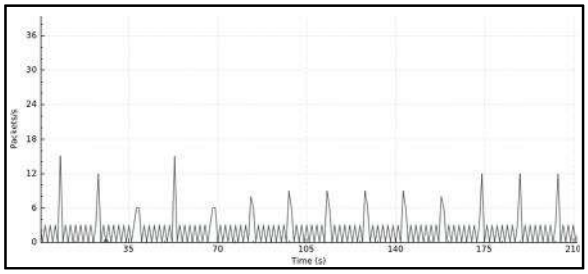

Figure-11. h4 (User) to h1 traffic.

Figure-9 shows traffic monitoring results on the communication traffic between host and switch based on characteristic botnet communication. Application of botnet attacks is done directly through the xterm command on the mininet terminal.

The x-axis represents the size of packets sent between computers to the server, whereas the y-axis is the timeframe for communication to occur. Based on the test results shown in Figures 9 and 10, there is an increase in the traffic each command to the bot is sent. Whereas in h4 uninfected botnets, traffic is random and there is no sudden increase in the magnitude of traffic. So it can be concluded that the existence of botnets can be seen when there is a sudden increase in traffic and patterned the same.

CONCLUSION AND FUTURE WORK

Botnets are an important threat to the SDN communication system because they are able to take control of the plane that is a vital part of the SDN. In this

paper has been discussed how the botnet works and how to detect its existence. In addition, botnet detection experiments have been conducted based on traffic behavior. Based on the experiments conducted, the existence of botnets can be identified based on sudden and patterned increase in traffic. But on the other hand, this is just one simple way of detecting botnets, it needs to be re-developed a more complex detection method because there is a chance of similarity in traffic size between bots and users over time. The development can be the application of methods of botnet detection on the traditional network by adjusting the increase in the SDN by providing certain programs.

REFERENCES

[1] Hao and K. Bao. 2014. A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation. IEEE Communications Surveys & Tutorials. 16(4): 2181-2206.

[2] Y. Jarraya, T. Madi and M. Debbabi. 2014. A Survey and a Layered Taxonomy of Software-Defined Networking. IEEE Communications Surveys & Tutorials. 16(4): 1955-1980.

[3] D. Kreutz, F. Ramos, P. Verissimo, C. Rothenberg, S. Azodolmolky and S. Uhlig. 2015. Software-Defined Networking: A Comprehensive Survey. Proceedings of the IEEE. 14-76, 2015.

[4] S. Shin, L. Xu, S. Hong and G. Gu. 2016. Enhancing Network Security through Software Defined Network (SDN). IEEE.

[5] S. Hong, R. Baykov, L. Xu, S. Nadimpali and G. Gu. 2016. Towards SDN-Defined Programmable BYOD (Bring Your Own Device) Security. NDSS'16.

[6] S. Shin, Wang and G. Gu. 2015. First Step toward Network Security Virtualization: From Concept to Prototype. IEEE Transaction on Information Forensics and Security.

[7] M. Mitchiner. 2014. Software-Defined Networking and Network Programmability: Use Cases for Defense and Intelligence Communities. CISCO White Paper.

[8] S. Shin, P. Porras, V. Yegneswaran, M. Fong, G. Gu and M. Tyson. 2013. FRESCO: Modular Composable Security Services for Software Defined Networks. Proceedings of the 20th Annual Network and Distributed System Security Symposium (NDSS'13).

[9] J. Sonchack, A. J. Aviv, E. Keller and J. M. Smith. 2016. Enabling Practical Software Defined Networking Security Application with OFX NSDI'16.

[10] J. Leonard, S. Xu, R. Sandhu. 2009. A Framework for Understanding Botnets. International Conference on Availability, Reliability, and Security (ARES 2009) 917-922.

[11] S. Silva, R. Silva, R. Pinto, R. Salles. 2013. Botnets: A survey. Computer Networks. 57(2): 378-403.

[12] S. Jain, A.Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Holzle, S. Stuart and A. Vahdat. 2013. B4: Experience with a Globally-deployed Software Defined Wan. In Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM.

[13] Hossein, R. Zeidanloo, Azizah, A. Munaf. 2009. Botnet Command and Control Mechanism. 2nd International Conference on Computer and Electrical Engineering.

[14] Z. Chi, Z. Zhao. 2007. Detecting and Blocking Malicious Traffic Caused by IRC Protocol Based Botnets. IEEE, IFIP International Conference on Network and Parallel Computing Workshop.

[15] R. Puri. 2003. Bots &; Botnet: An Overview. Research on Topics in Information Security.

[16] J.B. Grizzard, V. Sharma, and C. Nunnery. 2007. Peer-to-Peer Botnets: Overview and case study. Proceeding of the 1st Workshop on Hot Topics in Understanding Botnets (HotBots 2007).

[17] J. Kinder, S. Katzenbeisser, C. Schallhart and H. Veith. 2005. Detecting Malicious Code by Model Checking. In: Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA).

[18] Christodorescu M., Jha. 2003. Static analysis of executable to detect malicious patterns. In: Proceedings of the 12th USENIX Security Symposium.

[19] S. Chamotra, R. K. Sehgal, R. Kamal. 2012. Honey Sand: An open source tools based sandbox environment for Bot analysis and Botnet Tracking. In: Special issue of International Journal of Computer Applications on Communication Security. No. 7.

[20] H. Binsalleh, T. Ormerod, A. Boukhtouta, P. Sinha, A. Youssef, M. Debbabi and L. Wang. 2010. On the

Analysis of the Zeus Botnet Crimeware Toolkit. Eight Annual International Conferences on Privacy, Security, and Trust.

[21] S. Kumar, R. K. Sehgal and S. Chamotra. 2016. A Framework for Botnet Infection Determination through Multiple Mechanisms Applied on Honeynet Data. Second International Conference on Computational Intelligence & Communication Technology.