International Journal of Control and Automation
Vol. 11, No. 4 (2018), pp.175-186
http//dx.doi.org/10.14257/ijca.2018.11.4.16

Test Bed Construction for APT Attack Detection

Seong-Uk Park[1] and Sun-Myung Hwang[2*]

[1,2]Daejeon University, Computer Engineering Dept., 62, Daehak-ro, Dong-gu, Daejeon, Korea
[1]pppforte@gmail.com, [2]sunhwang@dju.kr

Abstract

APT attacks cause a lot of confusion in the Internet world, and DDoS attacks still dominate the ranking of cyber threats. Accurate detection is a big challenge. As the social and financial damage of an APT attack increases, a technical solution to the APT attack is needed. However, protecting existing APT attacks with existing security equipment is difficult. You can implement a test bed that can collect attacker data and analyze behavior. In this paper, we will build a test bed for intelligent APT attack detection and check it. We will discuss the introduction of APT, the introduction of DDoS, and the construction and construction of a test bed. This testbed will be able to test DDoS attacks in the future, and it will be difficult to detect attacks and discover the epicenter of the attack.

Keywords: *APT Attack, DDoS, NetFlow, Intelligent Detecting*

1. Introduction

According to a May 2014 survey, 97% of customers with well-established security infrastructures experienced an intrusion. Of these, 25% were hit by APT attacks. The results of the 2015 survey did not change much either. 96% were also hacked and 27% were exposed to APT (Advanced Persistent Threat) attacks. Many companies have noticed an accident in an average of 205 days after a data breach occurred. The attacker takes only 7 minutes to leak information, which takes hundreds of days to detect [1]. The hacker group now extracts confidential information to specific banks and bureaus, rather than an unspecified number. If you search for a social network such as Facebook, which is used by an individual, it is easy to know who is doing what and what company is doing. An attacker can grasp the target correctly and send an email with malicious code to APT (Advanced Persistent Threat) Attack and dig into the corporate or institutional network. Existing security has limitations to cope with APT attacks, so defense technology that replaces existing security structures is required [2]. Nowadays, DDoS (Distributed denial of service) attacks on web sites reward attackers financially or politically because our life tightly depends on web services such as on-line banking, e-mail, and e-commerce. One of DDoS attacks to web servers is called APT flood attack which is becoming more serious. As the social and financial damages caused by APT attack such as cyber terror are increased, the technical solution against APT attack is needed. Most existing techniques are running on the application layer because these attack packets use legitimates HTTP-GET request and malicious requests. We will discuss the construction of a testbed that can study practical detection techniques for APT attacks based on the access behavior of inline objects in a web page using NetFlow.

Following the introduction Chapter 2 describes the existing methods and explanations for APT attacks. Chapter 3 describes scenarios, construction methods, test bed building

Received (January 25, 2018), Review Result (April 10, 2018), Accepted (April 17, 2018)
* Corresponding Author

ISSN: 2005-4297 IJCA

environments and construction methods for test beds. Chapter 4 describes the APT attack simulation system, "Netbot", and describes the components and attack methods of Netbot.

2. APT

2.1. APT Attack

The APT attack, which is being used in cyber terrorism, cyber warfare, and hacktism, is expected to cause social and national threats in the situation of increasing cyber threats. It is used to infiltrate the internal system through the communication network for institutions, industrial facilities, companies, financial institutions, *etc.*, and hide it for a while. In other words, it is not a way to extract information just like a conventional hack, but a type of malicious code that is a kind of hide-and-seek type that waits for some time after an intrusion and disables security, collects information and then spills out [3, 4]. Malicious code can be created to prevent detection by existing antivirus programs, and unlike viruses that propagate themselves, it does not generate a lot of traffic, so monitoring of network administrators is easily avoided. This threat is a threat caused by IT technology such as malicious code, vulnerability, hacking, *etc*. [4], which does not rely solely on simple technology but maneuvers directly on target and attempts various attacks based on it. In addition, major information leakage, control system attacks, and cyber weapons are spreading globally, and threats to APT attacks targeting major information facilities are intensifying, leading to greater social risks. These APT attacks are characterized by intelligent attacks on specific targets. APT attack methods use vulnerable attack methods such as unknown vulnerabilities, zero-day vulnerabilities, system destruction, spear phishing with social engineering techniques, and watering holes. Stuxnet, Operation Aurora, Night Dragon, and EMC / RSA attacks are typical examples of APT attacks [5, 6]. In general, APT attack has the characteristics such as advanced, persistent and threat. The stage of APT attack is as fallows;

2.2. The Stage of APT Attack

2.2.1. Advance Preparation: Prepare the target to attack. How to infiltrate attack targets and customize malicious code and send them by e-mail or file transfer to prepare and find ways to infiltrate. Once this process is complete, it will infiltrate.

2.2.2. Internal Network Intrusion: The infiltration process causes an attacker to infect vulnerable systems with malicious code and infiltrate the internal network. Usually, malicious code is infiltrated by secretly attaching malicious code to the employee's PC or e-mail that does not use the vaccine.

2.2.3. Internal Activity: It collects information or collects information on the internal system, such as collecting information on unprotected systems, interrupting system operation, and disturbing the system. This process is difficult for the target to be attacked.

2.2.1. Goal Achievement: As the attacker's base, the ultimate goal is achieved by disrupting data transmission and system operation and destroying equipment. As a result, the object will be seriously damaged due to the system being disabled or leakage of personal information.

Therefore, various event information on host network and legacy node have to analyze and detect the anomaly cased of attacker on Netflow against advanced APT attack.

2.3. DDoS Attack Technique

One of the APT attacks for DDoS attacks, is an attack by an attacker using a malicious code to infect a PC or other mobile device and send a huge amount of packets to a target server to paralyze the server. DDoS attacks are attacked by various methods such as UDP Flooding, TCP_SYN Flooding, and HTTP Get Flooding [5] - [6]. UDP Flooding is an attack that sends a large number of UDP packets to the target server, making normal service impossible. It is not easy to prevent UDP from being an unconnected communication and because there is no singularity or pattern in the packet itself [7]. TCP_SYN Flooding uses 3-way-handshaking used in TCP communication. When an attacker requests a communication (SYN) to the server and receives a response (SYN + ACK) from the server, the server does not send an acknowledgment (ACK) Makes space for connections in the queue. The attacker continuously attempts to make a new connection to the server in the Half Open phase, and the backlog queue is filled up to accept further connections It is an attack that cannot be done. Every existing HTTP GET flooding attack detection adopts a method that specifically analyzes the contents of the packet. Systems using these algorithms are located and operated in the input of particular website or the input of the web server. In this study, based on the net flow information collected from any network position, this paper proposes a method of detecting HTTP GET flooding attack. First, it is needed to examine the netflow information being generated when normal users accessing a web server. In other words, if profiling the behavior of a normal user well, it will be able to easily distinguish between HTTP GET flooding attack traffic and normal.

2.4. Traffic Majoring tool based on Netflow

2.4.1. FlowScan: Based on the Netflow information, we draw a graph on the time axis of the network for each protocol application service for the desired period. It is also advantageous that most useful analysis such as Custom graph, Top AS usage, Top user and Raw flow dump are available. However, it is difficult to obtain statistical data other than visualized information.

2.4.2. Cflowd: Collect flow information from Netflow and save it as Arts ++ type file. Basically, you are only responsible for the collection, and you can use the Arts utility to get some statistical information in text form.

2.4.3. MADAS: It is a traffic measurement tool developed by Russia for measuring traffic between MIRNET and STARTAP. It is an ideal tool for real-time graphing and query interface. MADAS focuses on inter-country traffic arrangements. This is a unique feature not found in any other measurement tool, but it is suitable for the backbone network in that it analyzes traffic between countries based on the AS. The advantage of displaying the results of a query graphically is that it is vulnerable to the inability to analyze by application services.

2.4.4. Flowtools: Cflowd & Similar to ARTS. You can gather information from Netflow and get 20 predefined statistical information. It is possible to query from the Web using CGI, but it is basically a command-based program.

2.4.5. NetFlow FlowCollector: It is a commercial Netflow-based traffic measurement tool developed by CISCO. It collects, processes, and stores Netflow traffic from an export device. Stored data can be used by other Netflow applications, such as Netflow DataAnalyzer. Through the collection of functions called Thread and Filter, users can

save the information they want accurately and concisely at a low capacity, and can collect flows in various ways. In addition, data can be collected and stored by various methods such as port, host, AS, and protocol. However, it is very expensive because it is a commercial traffic measurement tool, and it is not easy to analyze raw data.

2.4.6 Netflow DataAnalyzer: It is a commercial NetFlow-based traffic measurement tool developed by CISCO that receives and visualizes NetFlow data collected, aggregated and stored by NetFlow FlowCollector. It is a format that receives data from the Netflow FlowCollector, so it is important how to set up and collect data in Netflow FlowCollector. It allows you to view data in various sets, and also provides special functions such as Time Slider, AS Drilling Down and Search. In the graph, there is a disadvantage that the whole traffic is not displayed on the time axis, the web is not supported and it is slow.

3. Construction of Test Bed

3.1. Test Bed Scenario

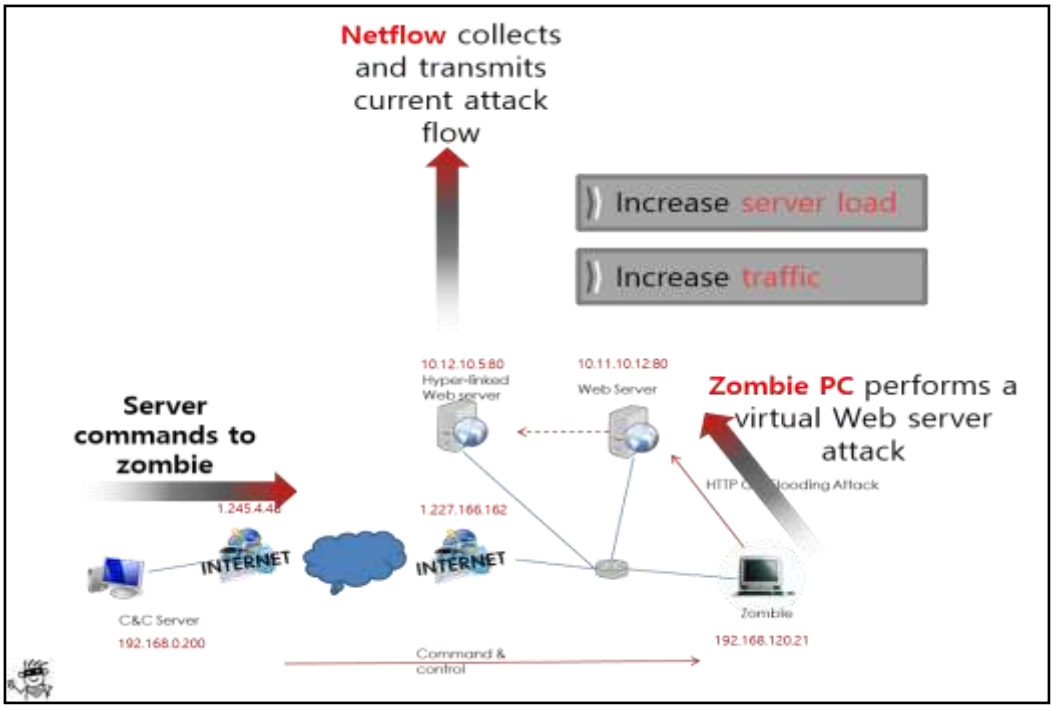

Figure 1. Test Bed Scenario

The We constructed the test environment contains Netbot in Figure 1. The Test bed can test which occurs various APT attacks based on possible scenarios. The C&C server has a separate IP network, attacking the web server by using one C&C server and one zombie PC. These series of attacks are configured to collect flows on the network using Netflow.

Actually test scenarios are as follows.

1. Netflow collects the flow of the current attack and stores it.

2. Check and analyze current network situation using "Flow Visualizer" through saved flow data.

3. Track network flow analysis and C&C server with Flow Visualizer.

4. Implementation and realization of intelligent security technique based on scenario test using intelligent surveillance system.

3.2. Network Configuration

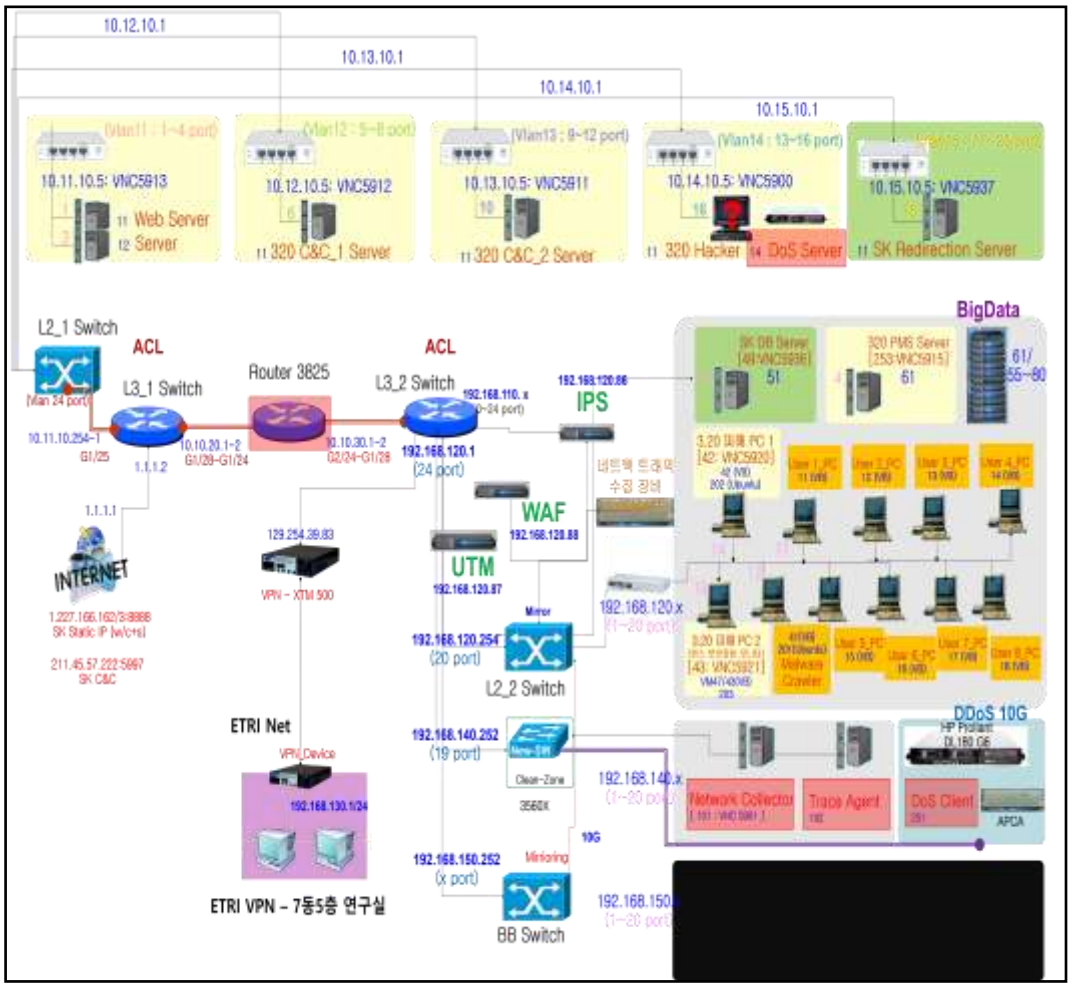

Figure 2. Network Diagram

The network configuration of the test bed is shown in Figure 2. Each Web server, attacker, and zombie are both virtual machines on the main PC. We used 'vmware' for the virtual machine and 'Windows XP' for the operating system. The attacker infiltrates into the zombie PC existing in the internal network of Figure 2 from the external network, issues commands, and directs traffic to the web server and the hyperlink web server. On the other hand, the router located at the center of the testbed is configured to generate NetFlow information for inbound traffic through the interface to which the zombie host is connected. The related parameters for NetFlow operation are 'nosampled NetFlow'

Figure 3 shows the spec configuration of the testbed server, the attacker's PC, the zombie pc, and the server configuration. The attacker and the zombie PC are composed of a general PC. The web server to be attacked, the hyperlink web server and the Netflow device are composed of servers, which are mounted in the server rack and are operated in the server room Respectively. The server room is always at 18 degrees Celsius, and there are various types of servers such as Netflow collection server and black box server. We focus only on the equipment in the server rack shown in Figure 3.

Figure 3. Spec of Test Bed

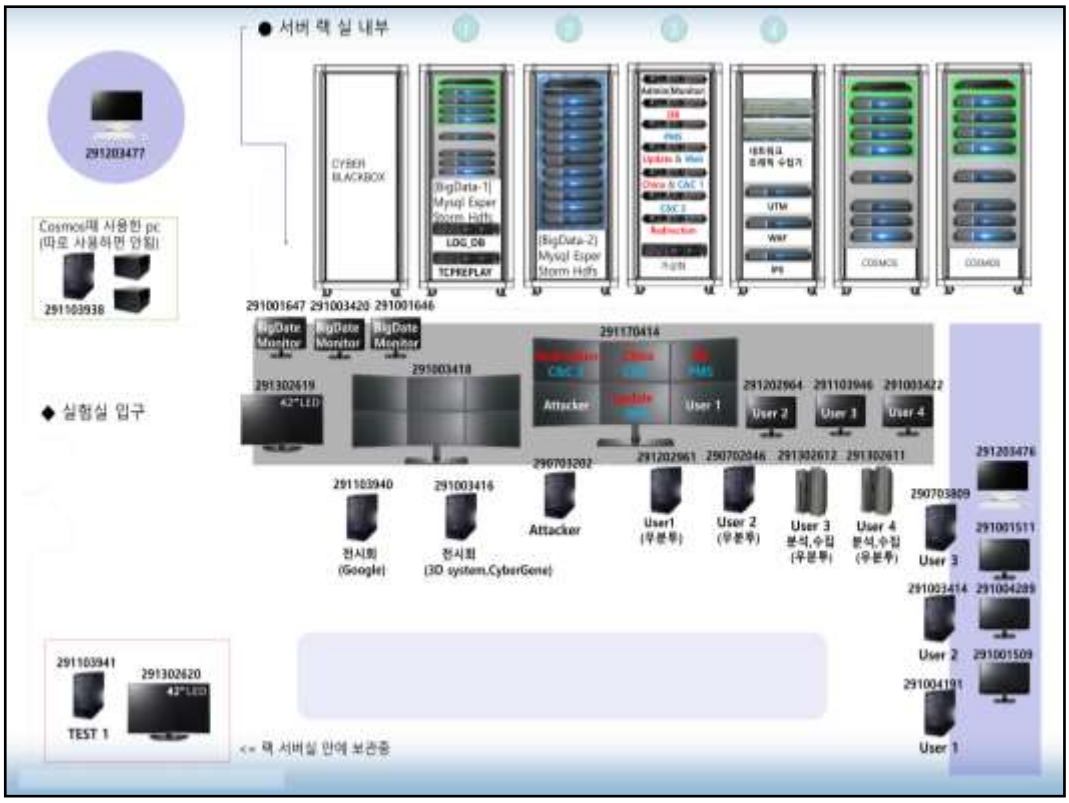

Figure 4. Testbed Components

4. APT Attack Simulation System

4.1 Netbot

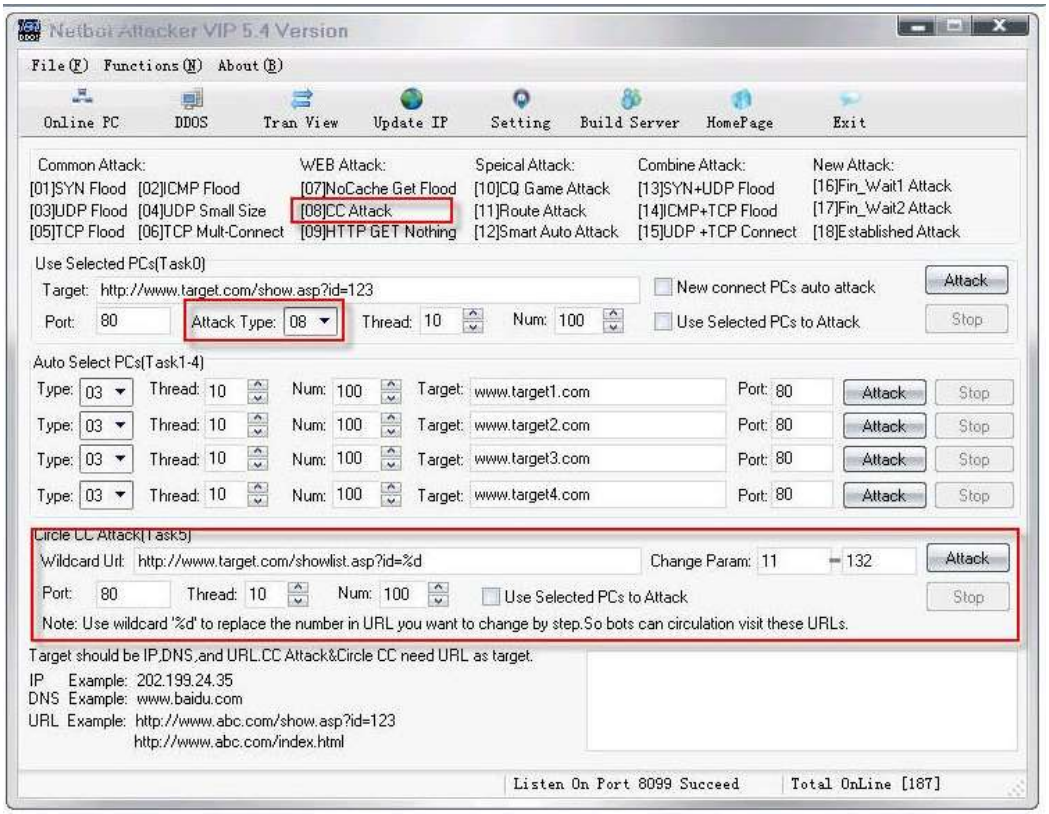

Figure 5. xDos Flow Analysis based on the Netbot Attacker

The Netbot attacker in Figure 5 is a remote control program that uses DDoS attack to the other side using a zombie PC. The system is consist of five types of attack and 18 methods such as 'http-get flood' attack method with the designated ip to the zombie pc created by virus in advance. The HTTP-GET flood attack is an application-level attack. It uses a botnet consisting of thousands or tens of thousands of zombie hosts to continuously send a normal HTTP GET request message to the attack target web server, Load, memory usage, etc.) to prevent normal users from accessing the web server normally. According to recent research [1,9,18], many bots behave like normal browsers, enabling more sophisticated HTTP-GET flood attacks. However, many HTTP-GET flood attacks in the real world are still reported to use binary files created using various cyber attack tools [19]. In other words, it is a real problem for the general public who have no expertise in network or Internet to try DDoS easily using these attack tools. However, these attack tools, which are easily accessible to the public, do not exhibit the highly sophisticated forms of HTTP-GET flood attacks that have been studied in recent years. The web browser used by ordinary users receives a web page and parsing process, and then proceeds to a process of fetching the corresponding inline object. However, the binary created by the existing cyber attack tool merely transmits an HTTP-GET request message and send it to the web server constantly.

4.2. Zombie PC

For the DDOS attack, we created an agent binary file using NetBot Attacker version 5.4, infected the Windows XP computer and created a zombie host as shown in Figure 6.

The zombie host periodically accesses the C & C server and notifies itself of its presence as shown in Figure 7. In the zombie PC, the attacker periodically transmits the presence / absence of communication to the attacker through the SYN / FIN communication, and the attacker displays the current communicable zombie PC as shown in Figure 6.

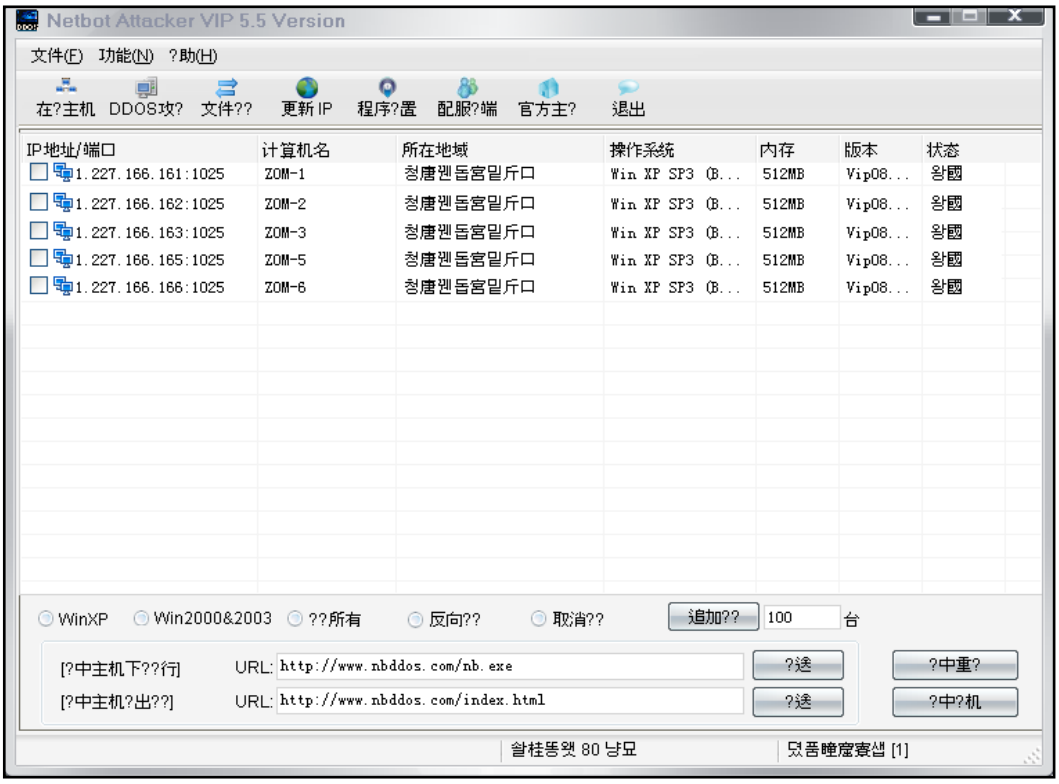

Figure 6. Zombie PC List

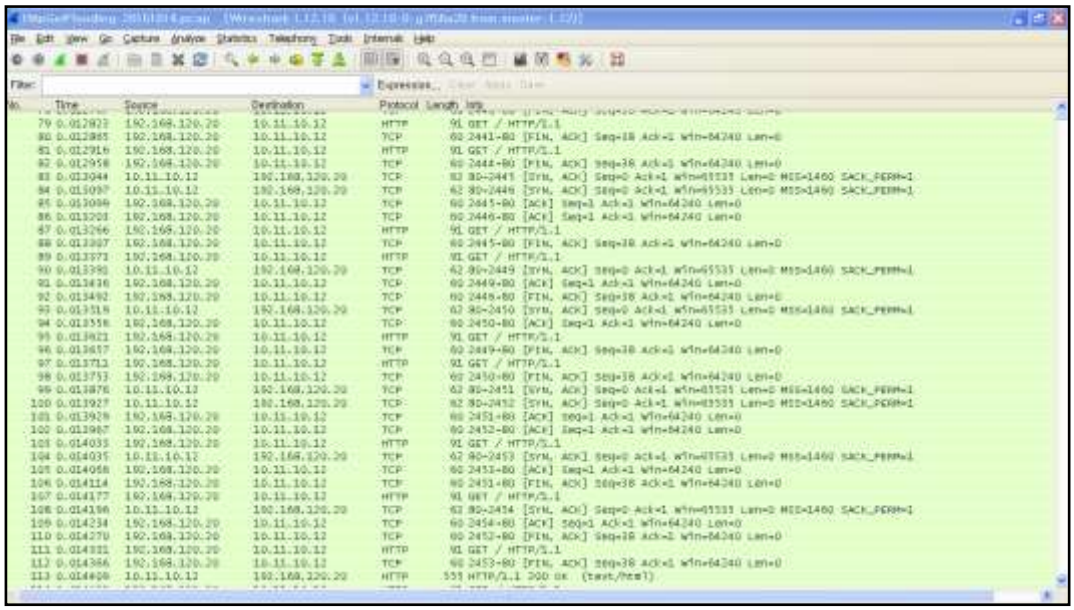

Figure 7. Capture Communication History on Zombie PC using Wireshark

Attackers now control infected zombie hosts remotely through a C&C server, which will attempt five types of attacks. The NetBot Attacker C&C server commanded the

zombie host to attack the web server using the basic parameters of the HTTP GET flood attack (10 threads and 10 attack packets per second). On the other hand, in the main page of the web server, two inline objects are embedded, and another external web server is hyperlinked as shown in Figure 6. Table 1 summarizes the various parameter values used in the experiment.

Table 1. Experimental Parameters

Field	Parameters
NetFlow	Inactive timer = 15 sec active timer = 30 min
Netbot C&C Server	Number of threads = 10 Number of HTTP GET request = 10/s

The experiment scenario is as follows to compare the occurrence of HTTP GET flood attack traffic from the zombie host and normal web server access as follows.

- Run Internet Explorer web browser on the zombie host

- Zombie host attempts to attack HTTP GET flood for 2 minutes by remote control of C&C server

- 2 minutes stop attack

- Attack for 2 minutes

- After a 2 minute attack, a normal web server access from a zombie host

5. Conclusions

APT attacks are long term and persistent attacks aimed at specific organizations, regardless of the means and methods used to achieve them. The attacker performs an intensive investigation into the target organization in advance, and intrudes mainly through e-mail using social engineering. APT attacks are cyber crimes aimed at securing confidential information of organizations and countries. As a result of cyber terrorism, cybercrime has become a serious social problem both in the world and in Korea. In APT attack, Hyundai Capital, Auction, SK Communications, and Nonghyup. These APT attacks are increasing. Therefore, APT security research is needed. as APT attack penetrates, various behaviors such as network behavior change, register behavior change, and OS behavior change are analyzed. However, there is a lack of a countermeasure against APT attack according to user's behavior. Currently, various companies are launching a number of APT attack countermeasure solutions, but the effect is not much different from existing intrusion detection systems. Conventional countermeasures such as firewalls, IDS / IPS, and anti-virus are needed to respond to evolving forms of attacks in introducing specialized APT-enabled technologies. This security solution can be applied to the organization's consistent security policy, and it is possible for an attacker to defend against basic attacks during multiple attacks that APT attacks use. The intrusion detection system based on the decision tree using the behavior analysis to analyze the behavior of the existing APT malicious code in order to defend the APT attack that is intelligently changed after infiltration into the system, Of-sight detection. Therefore, the proposed system recognizes the possibility of initial intrusion and can minimize damage area through quick response from APT attack. Intelligent smartphone security based on user behavior using cloud computing aims to protect smartphone users from leaks and illegal charging of users and to prevent smart phones from being used for APT attacks. Intelligent smartphone security model based on user behavior intelligently responds to malicious codes that are difficult to detect. Behavior-based techniques have a series of processes that detect, analyze, block, and alarm actions, such as process, network,

registry, driver, and authority, after detecting malicious code. In addition, step-by-step processes and user behavior-driven techniques enable in-depth detection and prevention. Since NetFlow information is the default information provided by most routers and switches running on the Internet today, there is no need to install a new device specifically on the network to extract attack traffic information. Because application level attacks, such as HTTP GET flood attacks using a huge number of zombie hosts that make up botnets, use the normal network layer protocol, most of the existing attack detection methods are handled through deep packet inspection in application clusters. However, attack detection techniques that operate at the application layer are implemented near victim hosts that are vulnerable to attack rather than the origin of the attack. In addition, detection methods at the application layer are very complex in their implementation and operation.

In this paper, we constructed testbad for APT attack detection intelligently and proposed a intelligent technique to analyze pattern and behavior using Net Flow data. And we generated test data based on scenarios to detect anomaly that has network traffic flow information. A test bed was constructed and a tool for analyzing source IP, packet, byte and source port / destination port based on the traffic quality was developed by test scenarios of simulation attack. we will study use this testbed to create test scenarios for various APT attacks, and to provide an intelligent search method for attacks.

References

[1] C. Tankard, "Persistent threats and how to monitor and deter them", Network security, vol. 2011, no. 8, **(2011)** August, pp. 16-19.
[2] P. Chwalinkski, R. Belavkin and X. Cheng, "Detection of Application Layer DDoS Attack with Clustering and Likelihood Analysis", Proceedings of Globecom, **(2013)**.
[3] T. Yatagai, T. Isohara and I. Sasase, "Detection of HTTP-GET flood Attack Based on Analysis of Page Access Behavior", Proceeding of IEEE Pacific Rim Conference, **(2007)**, pp. 232-235.
[4] D. Dittrich and F. Sven, "P2P as botnet command and control: a deeper insight", Proceedings of the 3rd International Conference on Malicious and Unwanted Software, **(2008)**, pp. 41-48.
[5] B. Claise, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information", IETF working group, **(2013)**.
[6] B. Mah, "An Empirical Model of HTTP NetworkTraffic", Proceedings of INFOCOM'97, **(1997)**, pp. 592-600.
[7] W. Lu and S. Yu, "An HTTP Flooding Detection Method Based on Browser Behavior", Proceedings of Computational Intelligence and Security, **(2006)**, pp. 1151-1154.
[8] Y. Choi, I. Kim, J. Oh and J. Jang, "AIGG Threshold Based HTTP GET Flooding Attack Detection", Proceedings of WISA, **(2012)**.
[9] M. Srivatsa, A. Iyengar, J. Yin and L. Liu, "Mitigating application-level denial of service attacks on Web servers: A client-transparent approach", ACM Trans. on the Web, vol. 2, no. 3, Article 15, **(2008)** July.
[10] C. M. Chen, B. C. Jeng, C. R. Yang and G. H. Lai, "Tracing denial of service origin: Ant colony approach", Applications of Evolutionary Computing, Springer Berlin Heidelberg, **(2006)**, pp. 286-295.
[11] X. Yin, W. Yurcik, M. Treaster, Y. Li and K. Lakkaraju, "VisFlowConnect: netflow visualizations of link relationships for security situational awareness", Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security, **(2004)**, pp. 26-34.
[12] D. Huistra, "Detecting Reflection Attacks in DNS Flows", Proceedings of 19th Twente Student Conference on IT, **(2013)**.
[13] L. Bilge, D. Balzarotti, W. Robertson, E. Kirda and C. Kruegel, "Disclosure: detecting botnet command and control servers through large-scale netflow analysis", Proceedings of the 28th Annual Computer Security Applications Conference, **(2012)**, pp. 129-138.
[14] C. Estan, K. Keys, D. Moore and G. Varghese, "Building a better NetFlow", ACM SIGCOMM Computer Communication Review, vol. 34, no. 4, **(2004)**, pp. 245-256.
[15] H. Choi and J. O. Limb, "A Behavioral Model of Web Traffic", Proceedings of Seventh International Conference on Network Protocols, **(1999)**, pp. 327-334.
[16] S. Yu, G. Zhao, S. Guo, Y. Xiang and A. V. Vasilakos, "Browsing Behavior Mimicking Attackson Popular Web Sites for Large Botnets", Proceedings of IEEE INFOCOM WKSHPS, **(2011)**, pp. 947-951.
[17] K. S. Han and E. G. Im, "A Study on the Analysis of Netbot and Design of Detection Framework", Proceedings of Joint Workshop on Information Security, **(2009)**, pp. 1-12.
[18] P. Giura and W. Wang, "A Context-Based Detection Framework for Advanced Persistent Threats", International Conference on Cyber Security, IEEE, **(2012)**, pp. 69-74.

[19] J. Shenk, "Learning from Logs: SANS Eighth Annual 2012 Log and Event Management Survey Results", SNAS, **(2012)**.
[20] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh and A. Hung Byers, "Big Data : The Next Frontier for Innovation, Competition, and Productivity", McKinsey Global Institute (MGI)., **(2011)**.
[21] W. Lee, S. J. Stolfo and K. W. Mok, "A Data Mining Framework for Building Intrusion Detection Models", IEEE Symposium on Security and Privacy, **(1999)** May, pp. 120-131.
[22] W. Wang and S. Gombault, "Efficient Detection of DDoS Attacks with Important Attributes", Risks and Security of Internet and Systems, CRiSIS '08. Third International Conference, **(2008)** October, pp. 61-67.
[23] R. Vijayasarathy, B. Ravindran and S. V. Raghavan, "A System Approach to Network Modeling for DDoS Detection using Naive Bayesian Classifier", Communication Systems and Networks (COMSNETS), 2011 Third International Conference, **(2011)** January, pp. 1-10.
[24] M. Zubair Shafiq, S. Momina Tabish and M. Farooq, "PE-Probe: Leveraging Packer Detection and Structural Information to Detect Malicious Portable Executables", Virus Bulletin Conference, **(2009)**, pp. 29-33.
[25] J. Lee, K. Jeong and H. Lee, "Detecting Metamorphic Malwares using Code Graphs", Proceedings of the 2010 ACM symposium on applied computing, **(2010)**, pp. 1970-1977.
[26] D. Krishna Sandeep Reddy and A. K. Pujari, "N-gram analysis for computer virus detection", vol. 2, no. 3, **(2006)**, pp. 231-239.
[27] P. Giura and W. Wang, "A Context-Based Detection Framework for Advanced Persistent Threats", International Conference on Cyber Security (CyberSecurity), **(2012)**, pp. 69-74.
[28] Y. Fukushimayz, A. Sakaiy, Y. Horiyz and K. Sakuraiyz, "A Behavior Based Malware Detection Scheme for Avoiding False Positive", 2010 6th IEEE Workshop on Secure Network Protocols (NPSec), **(2010)**, pp. 79-84.
[29] P. Trinius, T. Holz, J. Gobel and F. C. Freiling, "Visual Analysis of Malware Behavior Using Treemaps and Thread Graphs", 6th International Workshop on Visualization for Cyber Security, **(2009)**, pp. 33-38.
[30] R. Tian, L. M. Batten and S. C. Versteeg, "Function Length as a Tool for Malware Classification", 3rd International Conference on Malicious and Unwanted Software (MALWARE), **(2008)**, pp. 69-76.

Authors

Seong-Uk Park, Graduate Student
Computer Engneering, Daejeon University
62,Daehak-ro, Dong-gu, Daejeon, S.Korea
Major: Software Engneering
Email: pppforte@gmail.com

Sun-Myung Hwang, received received his M.S. and Ph.D. from the Department of Computer Engineering University, Seoul, Korea, in 1984 and 1987, respectively. he was a Dean of College of Engineering in Daejeon University, and a Professor in a Computer Engineering department in Daejeon University. His recent research interests include SPI (Software Process Improvement), SW quality evaluation and testing technique.

