HTTP Get Flooding Detection Technique based on Netflow Information

Youngsoo Kim, Jungtae Kim and Ikkyun Kim
Information Security Research Division
Electronics & Telecommunications Research Institute
Daejeon, Republic of Korea
e-mail: {blitzkrieg, jungtae_kim, ikkim21}@etri.re.kr

Koohong Kang
[2]Dept. of Information and Communications Engineering
Seowon University
Cheongju, Republic of Korea
e-mail: khkang@seowon.ac.kr

Abstract— A variety of attacks by botnets on a web server has become the most significant threat. One of the DDoS attack, HTTP get flooding attack, is especially difficult to distinguish because HTTP based attack to web server access is similar to the normal accesses of user. In this paper, in order to detect the HTTP get flooding attack, we propose detection technique using netflow information, which can be distinguished from the normal characteristics.

Keywords-HTTP Get Flooding Attack; Netflow; Botnet; Command & Control Server; Flow Pattern; Zombie Host.

I. INTRODUCTION

Considering the social turmoil, economic benefits, and showing off the hackers have targeted, it is the most effective for hackers to attack web server that is most widely used and provides important services these days. HTTP Get flooding attacks are being exploited in the most efficient way among denial-of-service type attacks aimed at these web server application layer [1][2]. HTTP Get flooding attack is to send a large amount of HTTP-GET requests to the target Web server by virus-infected computers or Bot under the control of Command and Control (C&C) server in order to deplete the processing resources so it disables normal user's requests. Since these attack packets maintain the normal HTTP payload, servers cannot easily distinguish between normal user's HTTP-GET request messages and their malicious request.

These attacks aimed at the application layer can be divided into three classes as follows: [3]. 1) Request Flooding Attack: each attack session creates a large amount of request rate compared with the normal session; 2) Asymmetric Workload attack: each attack session increases the request rate in the form of increasing the operation workload of the server resource. For example, it increases the ratio of the request that causes the database access. This type of attack can lower request rate than Request Flooding attack so it is more effective for hackers; 3) Request One-Shot attack: it is from Asymmetric workload attack, rather than sending multiple requests, it sends one request causing overload to one session. Thus, these attacks will be able to easily avoid a threshold-based DoS defense system, and also after the session ends, it can continue to give damage to the performance of the server.

As stated above, HTTP-GET flooding attacks use normal HTTP protocol so it is not easy for common Intrusion Detection System (IDS Intrusion Detection System) to detect. Since IDS detects attacks based on the attack signature, it is not easy to detect the HTTP-GET flooding attack. Therefore, it adopts a method of blocking an input request message if it exceeds maximum amount of traffic that the web server can support. However, this simple method has a problem that also blocks the normal traffic. Recently, a variety of detection methods to overcome this problem have been proposed.

The paper reviews various conventional methods suggested for detecting the HTTP Get Flooding Attack in Section II. Details of the proposed detection technique using netflow information with the analysis results are described in III. Finally, the Section IV concludes the paper with future works.

II. THE TRADITIONAL METHODS

HTTP / 1.1 sessions support the persistent connection. Therefore, a client sends and receives requests to a web-cluster without opening a new TCP connection for each request. As a result, one normal HTTP / 1.1 session is composed of a number of requests for the session. Requests can be closed loop type that the client waits for response before it sends the next request, or can be pipelined type that the client does not wait and sends numbers of requests. One page brings one main request for text context and image files included in main page through embedded requests. Main request is typically dynamic so contains a processing, such as database processing but embedded requests are shown as static, simply handle web-cluster processing.

The DNS query and response pattern of the normal user are important information. It can be used to set a reference value for the attack patterns based on these user baseline models. Therefore, in this section, this paper analyzes the traffic characteristics of a normal user.

One client request is processed as follows. 1) If the request is received, reverse proxy server will parse the requested URL and forward the request to a web server according to the load balancing policy. If the request is for static web pages or image files, the server will service the requested page.

ISBN: 978-1-61208-516-6

If the request is the e-commerce function, it is handled by the application scripts such as PHP, JSP or Javascript. These requests are being composed of a multiple of database queries, these results are synthesized to make the response page. Following Fig. 1 Ranjan et al [3] shows the typical victim system model for the web based applications and servers within a Content Distribution Networks (CDN).

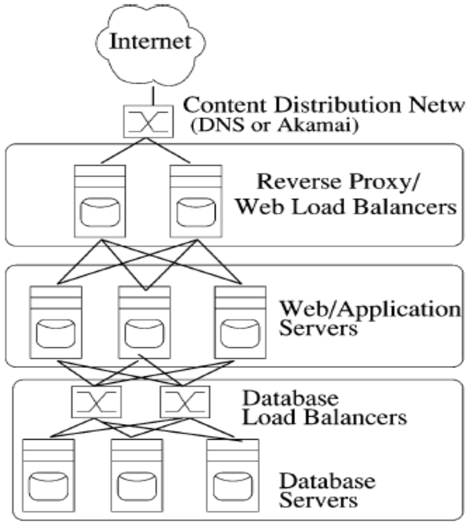

Figure 1. Victim Modeling

For the web applications, the HTTP attacks can be made by changing the session parameters such as Session inter-arrival time, request inter-arrival time, or workload-profile. Ranjan et al. [3] proposed a method of detecting the HTTP-GET flooding attacks by detecting misbehavior for these three changes. Also Yatagai et al. [1] suggests a method of detecting hosts, which maintain the order of the same page, by recording the web page browsing procedure for each source IP address as shown in the Fig. 2. This uses the fact that clients, which are infected or used as zombie hosts, browse the web page of the same order.

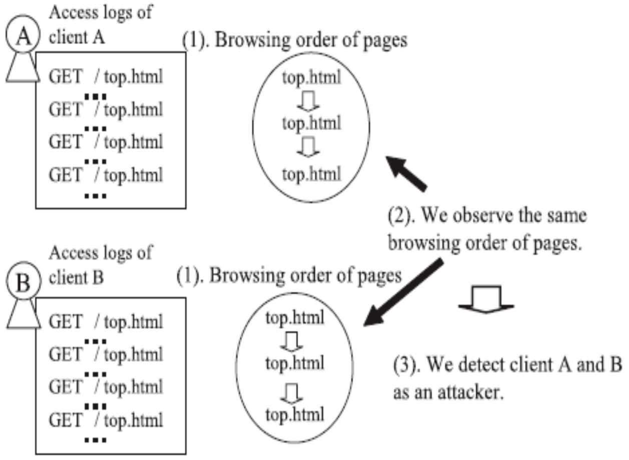

Figure 2. Browsing Oder of Web Pages for HTTP Get Flooding Attack [1]

In addition, it detects the attack using the connection between the Web page sizes and browsing time. This is because normal users access to a large amount of information, it should take longer to browse.

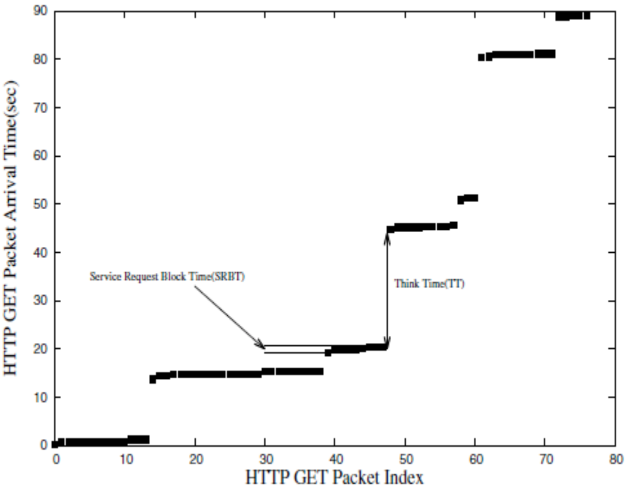

Figure 3. HTTP GET request packet arrival time for the main page access

Choi et al. [2] proposed a method of detecting attacks by checking a series of HTTP GET request packet between the main request and the sub-request. Above Fig. 3 shows the HTTP GET request packet arrival time for the main page access by legitimate users. Choi et al. [2] proposed a method for detecting attacks using these time characteristics.

III. DETECTION TECHNIQUE USING NETFLOW

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. Fig. 4 shows the observed results of the behavior based on related netflow information by monitoring the traffic of a normal user who accesses representative portal site in Korea. On the other hand, a similar result shows the number of packets based on time that flow generated by collecting all flow-records for a major overseas shopping mall site in Fig. 5. Since it contains a lot of external hyperlinks within the portal main page, it shows that flow of pipeline type, which does not wait for a response to the request at the same time as the approach of the user, and closed loop type by parsing are taking place in the first 1 minute. It is possible to observe the packet changes and the time difference between flow of the user thinking time and the new page accessing time.

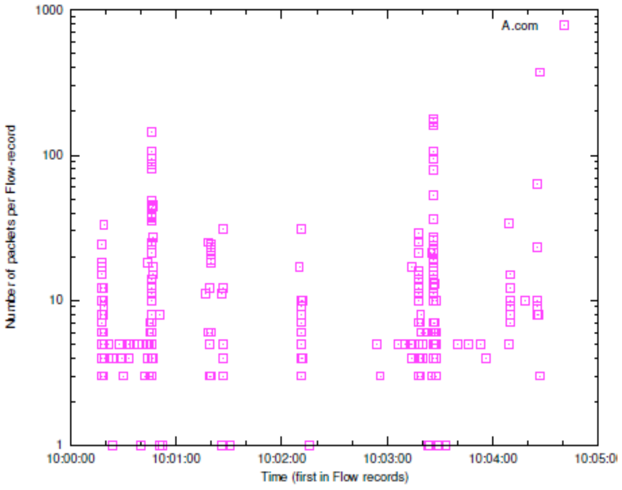

Figure 4. Number of packets based on flow-records beginning time (large domestic portal sites)

There are a variety of tools that can attempt to HTTP GET flooding attack. Fig. 6 shows the flow information for the HTTP GET flooding attacks caused by using NetBot Attacker. As shown in the Fig. 6, it looks very formal attack traffic pattern. That flow is generated at regular time intervals, and the number of packets within flow look very constant. As a result, HTTP-GET flooding attack done by a normal tool has very simple form, but it can be inferred that attacks can be detected easily only by the netflow information.

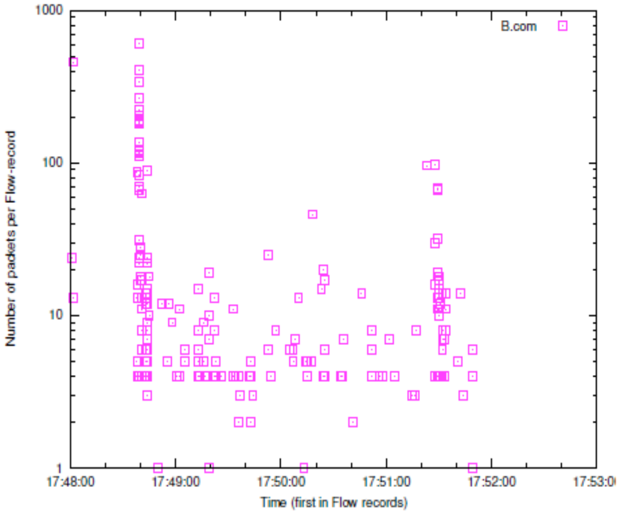

Figure 5. Number of packets based on flow-records beginning time (large overseas online shopping mall sites)

In fact, considering accidents and incidents caused by using the public attacking tools that can be obtained via the Internet are increasing every year, attacks of this level can be easily detected using only the netflow information. Fig. 7 shows the average number of bytes within flow that have been collected during the first one minute to the Fig. 4.

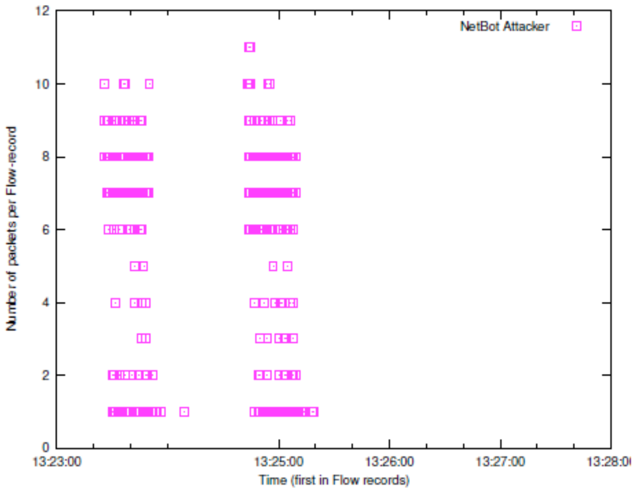

Figure 6. Number of packets per flow for HTTP GET flooding attack patterns using NetBot Attacker tools

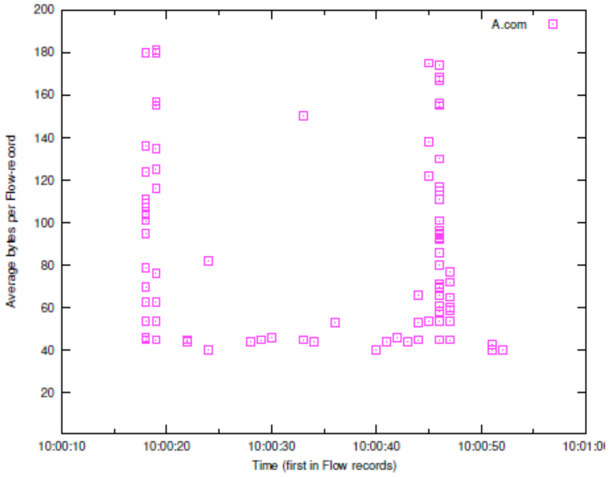

Figure 7. Average number of bytes base on flow-records starting time (domestic large portals)

As explained in Fig. 4., it shows pipelined and closed loop type to access main page as the pattern of the flow bytes. (Fig. 5., Fig. 8., and Fig. 9. represents the average number of bytes within flow in Fig. 6.) As seen in Fig. 9., HTTP GET flooding attack made by the NetBot Attacker tool can be found significant differences compared to the flow patterns of a general user.

Fig. 10 shows the overall flow of the HTTP GET flooding attack. As shown in Fig. 10., Hackers constitute Botnet [4] for an effective attack, and Bots (zombies PC) form command and control channels to C & C servers. Eventually Hackers (botmasters) deliver the orders to attack bots in a botnet using these channels. Traditionally, the C & C server is a centralized type using Internet Relay Chat (IRC) protocol but, since protection is strengthened for this so these days it is used based on the HTTP [6], the P2P or tree-layered method [5] besides centralized type C & C server.

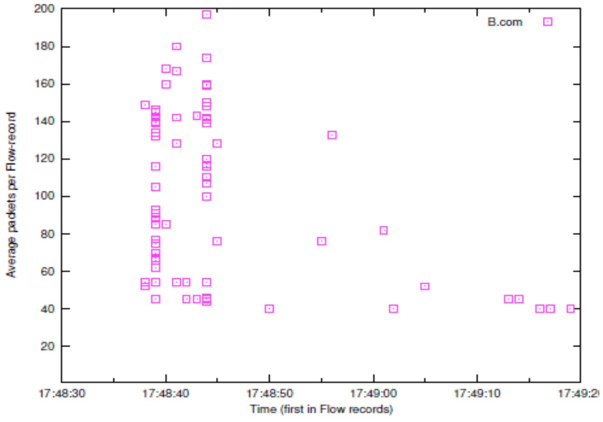

Figure 8. Average number of bytes based on flow-records starting time (large overseas online shopping mall sites)

This section may remain at the level of detecting the bot (zombie PC), as previously described. Of course, it is also important to detect these bots and block the HTTP GET flooding attack generated by them, but in order to defense more effectively, it is important to detect the C & C server.

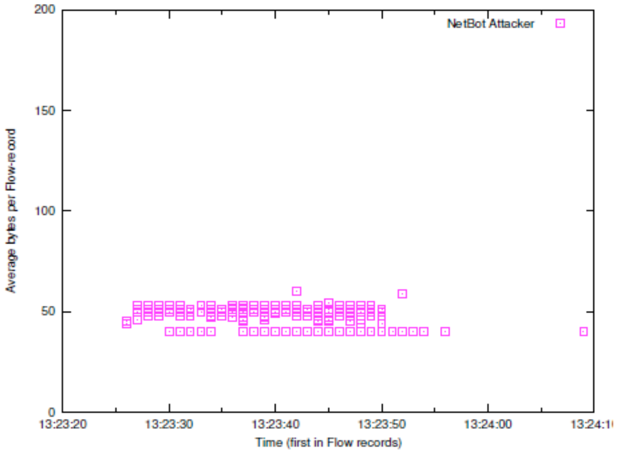

Figure 9. Average number of bytes per flow for HTTP GET flooding attack pattern by NetBot Attacker tools

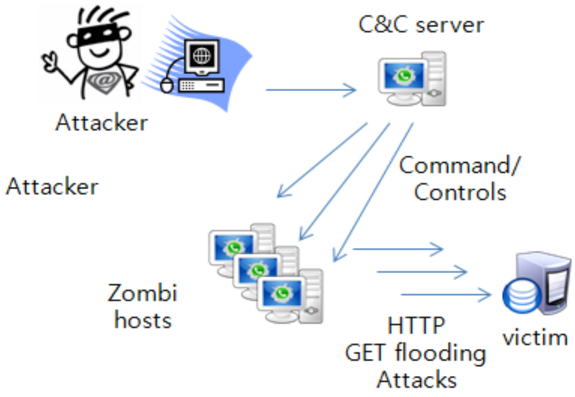

Figure 10. Connection of HTTP GET Flooding

That is, after detecting the C & C server and blocking the traffic, eventually botnets will not be able to proceed with the malicious acts longer. Existing methods for detecting the connection channels between bots and C & C servers, they are required to retrieve all of the traffic based on the messages using the corresponding protocol. For example, IRC session is a message, such as PASS, NICK, USER and etc. HTTP protocol is a signature, such as GET, POST, or HEAD. However, in this study, it will be able to extract another bots in corresponding botnet by configuring fingerprint using the netflow information that can be extracted from traffic between detected bot hosts and C&C servers. Also, it can also be applied to the Connection Based Tracking Algorithm, developed to track the botmaster access to the C&C server. These ideas are left for further study.

IV. CONCLUSION AND FUTURE WORK

DoS attacks still dominate the ranking of cyber threats. It is a great challenge to accurately detect. HTTP-GET flooding attacks use normal HTTP protocol so it is not easy for common intrusion detection system to detect. In this paper, we show a method of detecting HTTP GET flooding attack from normal behavior based on the net flow information. And it shows that most attacks can be detected easily only by the net flow information. We will study the method of extracting netflow information between bots and C&C server and find the other bots for future work.

ACKNOWLEDGMENT

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No.R-20160222-002755, Cloud based Security Intelligence Technology Development for the Customized Security Service Provisioning).

REFERENCES

[1] T. Yatagai, T. Isohara, and I. Sasase, “Detection of HTTP-GET flood Attack Based on Analysis of Page Access Behavior,”, in Proc. of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, pp.232-335, 2007.

[2] Y. Choi, I. Kim, J. Oh, and J. Jang, “AIGG Threshold Based HTTP GET Flooding Attack Detection,” in Proc. of WISA 2012, pp 270-284, 2012.

[3] S. Ranjan, R. Swaminathan, M. Uysal, A. Nucci, and E. Knightly, “DDoS-Shield: DDoS-Resilient Scheduling to Counter Application Layer Attacks,” IEEE/ACM Trans. On Networking, Vol. 7 No.1, pp.26-39, 2009.

[4] G. Gu, J. Zhang, and W. Lee, “BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic,”In Proc. of the 15th Annual Network and Distributed System Security Symposium, pp.1-18, 2008.

[5] G. Ollmann, “Botnet Communication Topologies: Understanding the intricacies of botnet Command-and-control,” White Paper, Damballa Inc., 2009.

[6] J. Lee, H. Jeong, J. Park, and M. Kim, “The Activity Anaylsis of Malicious HTTP-Based Botnets Using Degree of Periodic Repeatability,” In Proc. of International Conference on Security Technology, pp.83-86, 2008.