ICIC Express Letters
Part B: Applications
Volume **9**, Number **5**, May **2018**

ICIC International ©2018 ISSN 2185-2766
pp. 399–404

IMPLEMENTATION OF THE PERFORMANCE EVALUATION AND OBSERVATION SYSTEM FOR WEB SERVER UNDER ATTACKS

Hiroto Nakashima[1] and Takuo Nakashima[2]

[1]School of Industrial and Welfare Engineering
Tokai University
9-1-1, Toroku, Higashi-ku, Kumamoto 862-8652, Japan
4big2107@mail.tokai-u.ac.jp

[2]ICT Education Center
Tokai University
4-1-1, Kitakaname, Hiratsuka-shi, Kanagawa 259-1292, Japan
taku@ktmail.tokai-u.jp

Received October 2017; accepted January 2018

Abstract. *The number of DDoS (Distributed Denial of Service)/DoS attacks increases in recent years. Robustness of a Web server should be evaluated due to the evaluation of machine specification under frequent attackers, and actual performance evaluation systems are required. In this paper, we proposed the implementation of the packet sending system, the system observation system and packet observation system. As the results of experiments, we could get the unique parameter values of Web server responding behavior. These results will be useful to detect cyber attacks.*

Keywords: SYN flooding attack, DDoS, Performance evaluation

1. **Introduction.** In recent years, the number of DDoS (Distributed Denial of Service)/DoS attacks does not decrease even if end networks are moved into private networks. Current cyber attacks do not consist of simple DDoS/DoS attacks, but complicated attacks combining other types such as malware attacks, attacks for open ports, and attacks for target Web servers. Web servers should especially be built tightly for DDoS/DoS attacks. Robustness of a Web server should be evaluated due to the evaluation of machine specification under frequent attackers. The performance, however, depends on the server hardware and software specifications, and network speed. The actual performance evaluation systems are required. In this paper, we have proposed the implementation of the performance evaluation and observation system for the Web server under attacks.

2. **Previous Researches.** The research [1] implemented an attacking program, and observed response packets from the server on different OSs. Its performance estimation explored the metric to detect a condition caused by SYN flood attacks. The other research [2] developed a queueing model to study the system performance under the most prevalent SYN-flooding attack, with reference in terms of different parameters and different situations, such as arrival rate, traffic rate, timeout period, the maximum capacity of system, the standard deviation and the distribution of the service time. This research, however, is limited in virtual execution using the simulator. Another research [3] proposed a fuzzy logic based system for detecting SYN flooding. Performance of the proposed system had been compared with Cumulative Sum (CUSUM) algorithm. This research, however, does not consider the performance of real computer system. We discuss the performance on real network system. Other researches such as the research [4] investigated the resistance to DoS attacks under virtual environments. Performance under various virtual environments has been pursued in this research, but performance under real circumstances has

not been investigated. As the virtual environments consume a large memory and CPU resources, the accurate evaluation is required. The research [5] investigated the impact of CPU and network bandwidth under flooding based Denial of Service (DoS) attacks. In this experiment, it was found that the attack interval and the packet number were the most influent metrics on the target machine. The total tolerance of attacked servers, however, has not yet been investigated. The research [6] proposed the detection method on slow HTTP Denial of Service (DoS). Detection on slow HTTP Denial of Service (DoS), however, has been valid only for Web servers.

3. **Proposed Implemented System.** We customized three machines, the attacker, victim and system observation machine, and implemented the packet sending system, the system observation system and packet observation system respectively. The attacker machine was built on Linux kali 4.6.0-kali-amd64, and able to send packet function within 10 milliseconds. Firstly, we implemented the packet sending system using the raw socket on this attacker machine. In this system, the forged source IP address is inserted in the sending IP packets. The victim machine built on OS is CentOS-7. Secondly, the system observation system is implemented based on the TOP command gathering system resources such as CPU consumption rate, and memory consumption size. The packet observation machine watches network traffics, such as the attacking and responding packets flowing over the network. Finally, the packet observation system is implemented based on the dump command catching and showing the IP packets from capturing on mirror port of the router. Our system can analyze the packet flow focusing on the sequential number with the exact time. Figure 1 shows the proposed system architecture and network configuration.

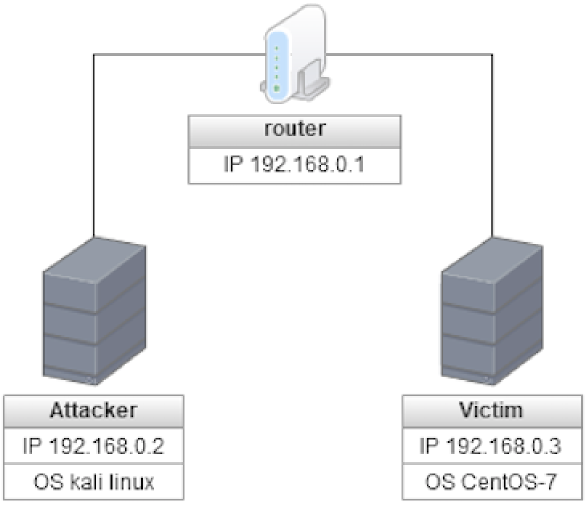


Figure 1. Network system configuration

In this implementation, an attacking user can change each field of TCP/IP packet leading to a consideration that the source IP address field has been forged. When the attacker uses the fake source IP address, the SYN packet on TCP is sent to the victim host. The victim host allocates memory storage as the TCB (TCP Control Block), and response SYN+ACK packet to the forged source IP address shown in Figure 2. The TCB is thought to be kept in memory until the server receives the ACK response from the associated source IP address. Repeated SYN packet causes the memory overflow by the huge allocation of TCBs.

The proposed systems of ours can capture the current network traffic and TCP state transition and the performance of victim machine. If the machine is exposed under

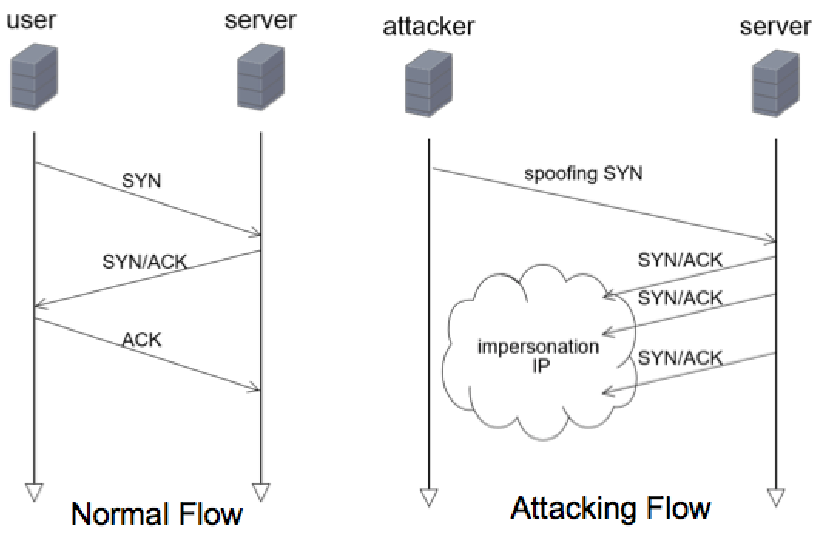
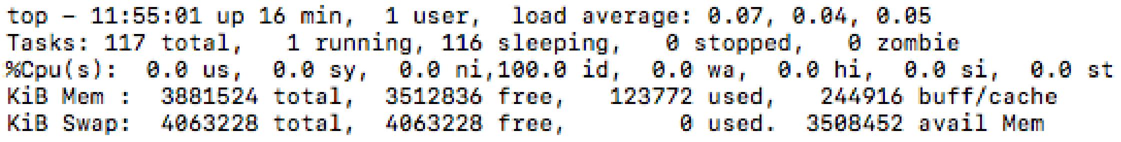


Figure 2. TCP packet flow on two different conditions

attacks, system manager should quickly decide whether this machine should continue to run or not based on the machine conditions. At that time, the condition of TCP state and machine resources are required to decide the shutdown or not. Our proposed systems are easily implemented to other types of machine, and effectively works under such conditions to get the TCP state and machine condition.

4. **Experimental Results.**

In this environment, we repeatedly sent huge packets from the attacker machine to the victim machine. In addition, we conducted the evaluation of the server performance using TOP command – which leads us to collect the value of memory usage, CPU available rate and so on. Figure 3 shows the results of TOP command.



```
top - 11:55:01 up 16 min,  1 user,  load average: 0.07, 0.04, 0.05
Tasks: 117 total,   1 running, 116 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  3881524 total,  3512836 free,   123772 used,   244916 buff/cache
KiB Swap:  4063228 total,  4063228 free,        0 used.  3508452 avail Mem
```

Figure 3. Results of the TOP command

4.1. **Normal state of the Web server.**

Firstly, we conducted normal request for the Web server using a Web browser, and observed the packets over the experiment network. Figure 4 shows the SYN+ACK reply message of the Web server. The reply message repeatedly responded to the forged source IP address until receiving the ACK packet. The Web server replied six times, and the time interval of two reply messages made the value larger than that of previous time interval with factorial manner.

4.2. **Under attacking state of the Web server.**

Figure 5 shows the time of the SYN+ACK reply packet splitting the reply times. The repeated SYN+ACK reply packets used the same acknowledge number in TCP header field.

Attacking packets arrived on the Web server between 1.1 second and 61.1 second. This experiment led to the following results. Firstly, Web server was not able to respond to the first reply packet, but was able to respond to the 2nd and 3rd packets – meaning that the server quickly produced multi thread of HTTP daemon under an attacking or increased request for the Web server. Secondly, almost three packets replied to the spoofed IP

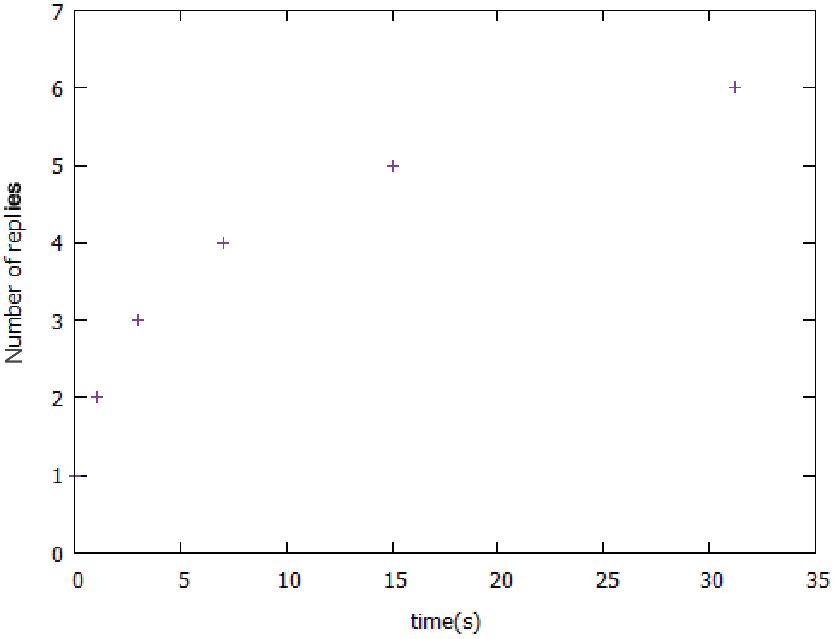


FIGURE 4. Reply time and count under normal condition

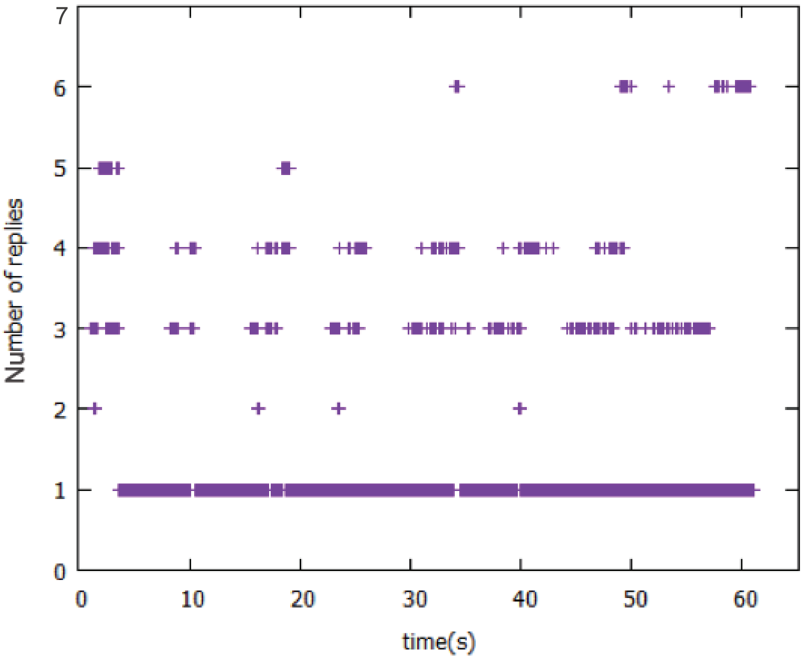


FIGURE 5. Reply time under attacks

address even if the six packets were replied under normal condition. The web server controls the number of SYN+ACK reply packets under attacking condition.

Figure 6 shows the used memory space of CPU under attacks from 1.1 second to 61.1 second. Firstly, the arrival of a large number of attacking packets caused the sudden increase of memory usage after time 1.1. After starting the attacks, memory was used mainly by http daemons and TCBs. Memory spaces were used during the attacking time until 61.1. After the end of attacking time, http daemons can be thought to have kept TCBs in order to respond to the forged source IP addresses to finish the TCP connection stage in 35 seconds during which TCP sends the response packet six times.

If we set the threshold to the difference value of each memory usage, our system can effectively detect the start of attacks indicating that the huge increase of memory usage

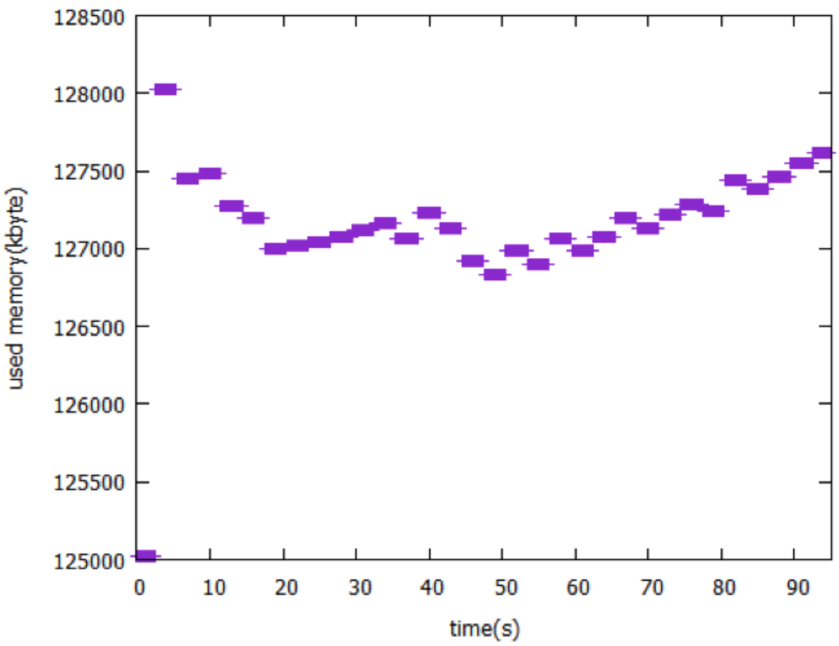


Figure 6. CPU used memory space under attacks

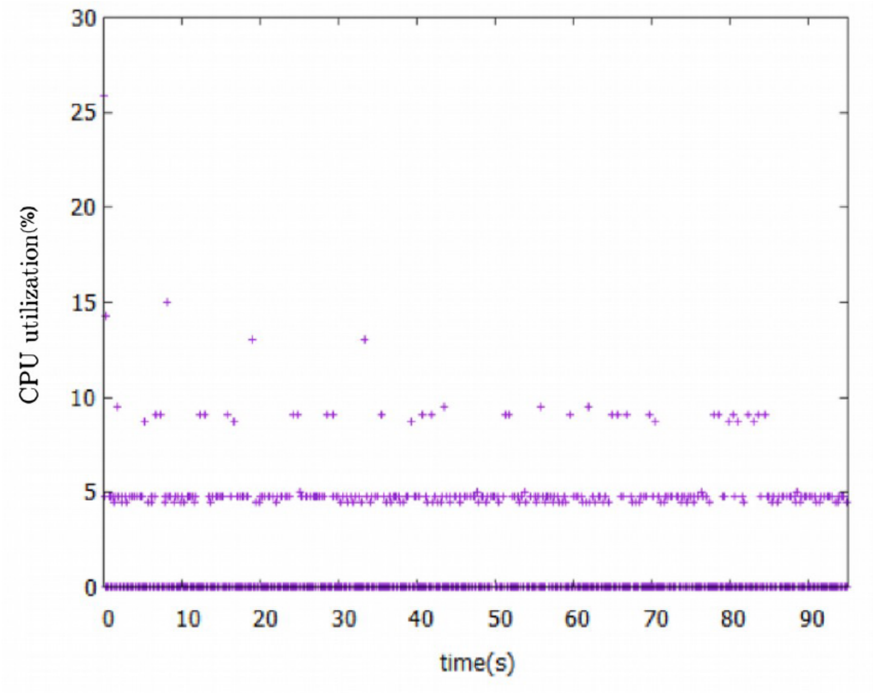


Figure 7. CPU utilization rate under attacks

occurred. In addition, if we set the threshold to some value of memory usage, our system can detect the continuation of arrivals of attacking packets.

Figure 7 shows the CPU utilization rate under attacks. The CPU is mainly used by the http daemons to enable to accept http request from the remote clients. The http daemons, however, do not largely consume the CPU resource. The current CPU performance can overcome the attacks with 10m second interval time.

5. **Conclusion.** We proposed implementation of performance evaluation and observation system for the Web server under attacks. We implemented sending packet application with ability to forge the source IP address field and to change the sequential field on

IP header. In addition, we developed an observation machine to capture the attacking and responding packets, and analyzed the IP and TCP header fields. As the results of experiments, we extracted the following results. 1) The observed Web server replied six times for one SYN request consuming 35 times on the normal condition. 2) The server quickly produced multi thread of HTTP daemons and then daemon controlled the total number of SYN+ACK reply. 3) The http daemons mainly consumed the memory and the CPU resources and continue the consumption after the end of attacking period for the end of keeping the TCBs of the SYN+ACK reply messages. Using these quantic values and our proposed system, we will easily develop the detection system for DDoS/DoS attacks, and evaluation system whether the system operates under attacks or not.

In future works, we will compare the performance of the Web server on different operating systems.

REFERENCES

[1] T. Nakashima and T. Sueyoshi, Performance estimation of TCP under SYN flood attacks, *Proc. of the 1st International Conference on Complex, Intelligent and Software Intensive Systems (CISIS'07)*, pp.92-99, 2007.

[2] Z. Lian and J. Lin, Simulation analysis of the DoS attack in Internet service, *International Conference on Wireless Communications, Networking and Mobile Computing*, pp.6298-6301, 2007.

[3] T. Tuncer and Y. Tatar, Detection SYN flooding attacks using fuzzy logic, *Proc. of the 2008 International Conference on Information Security and Assurance*, pp.321-325, 2008.

[4] R. Shea and J. Liu, Performance of virtual machines under networked denial of service attacks: Experiments and analysis, *IEEE Systems Journal*, vol.7, no.2, pp.335-345, 2013.

[5] A. Aborujilah, S. Musa and M. N. Ismail, Detecting TCP SYN based flooding attacks by analyzing CPU and network resources performance, *The 3rd International Conference on Advanced Computer Science Applications and Technologies*, pp.157-161, 2014.

[6] N. Tripathi and Y. Singh, How secure are web servers? An empirical study of slow HTTP DoS attacks and detection, *The 11th International Conference on Availability, Reliability and Security*, pp.454-463, 2016.