Bulletin of Computer Science and Electrical Engineering
Vol. 1, No. 2, December 2020, pp. 63~72
ISSN: 2722-7324, DOI: 10.25008/bcsee.v1i2.7

TCP SYN Flood (DoS) Attack Prevention Using SPI Method on CSF: A PoC

I Putu Agus Eka Pratama[1]
[1,3]Department of Information Technology, Faculty of Engineering, Udayana University, Bali, Indonesia

Article Info

Article history:

Received Jun 26, 2020
Revised Jul 29, 2020
Accepted Aug 10, 2020

Keywords:

TCP SYN Flood
Denial of Service (DoS)
Configserver Security and Firewall (CSF)
Stateful Packet Inspection (SPI)
Proof of Concept (PoC)

ABSTRACT

TCP SYN Flood as one kind of Denial of Service (DoS) attack, still popular to flood the server connection, by sending SYN packets to the target. Because of the risk caused by this attack, there is a need for a network security mechanism. In this paper, one of the security mechanisms proposed is using Stateful Packet Inspection (SPI) method on Configserver Security and Firewall (CSF). By using SPI method, CSF has capabilities to responsible for separating packets of data, that may be entered with data packets that should not be entered into the server. For example: port to be opened, port closed, and IP Address that may access the server for anywhere. This paper combines both of CSF and SPI method to prevent TCP SYN Flood (DoS) with Proof of Concept (PoC) at the Linux operating system. The security process is done in 3 ways: configuring a maximum connection from an IP Address to a server, securing an incoming SYN packet per second, and counting how many times an IP Address violates the minimum SYN packet rule per second before being blocked by a firewall.

Corresponding Author:

I Putu Agus Eka Pratama
Department of Information Technology, Faculty of Engineering, Udayana University
Jl. Raya Kampus Unud, Bukit Jimbaran, Badung, Bali, Indonesia
Email: eka.pratama@unud.ac.id

1. INTRODUCTION

Most of the services on the internet are websites that are run through a web server. Web server is a program on computer network that uses Hyper Text Transfer Protocol (HTTP) and Hyper Text Transfer Protocol Secure (HTTPS) to present files in the form of web pages to users and also increasing the security of transactions within it [1]. The web server will respond to user requests sent using the user's HTTP client computer by sending files in the form of these web pages. server must be having high availability, so that it can provide services without having problems [2].

Nowadays, Denials of Service (DoS) attacks are well-known as one of the major threats in today's internet services [3]. Most of web services and applications are suffering and targeted by attacker using the execution of DoS attack. DoS attacks of website is significantly increasing day by day. One of the famous of DoS attacks is TCP SYN Flood, in which attackers flood the server with the SYN package [4]. It's means that in SYN Flood, an attacker sends many TCP packets in a short time by sending only SYN packets, so that it is not enough to do a TCP three-way handshake to establish a proper connection. The server sent the SYN-ACK packet to the attacker, but the attacker who was supposed to reply with the ACK packet did not reply, causing the server to wait for the ACK package. At the same time another SYN packet arrived, it is caused the server to become tired and down. Illustration of comparison between normal TCP connection and TCP SYN flood attack is shown in the Figure 1 below:

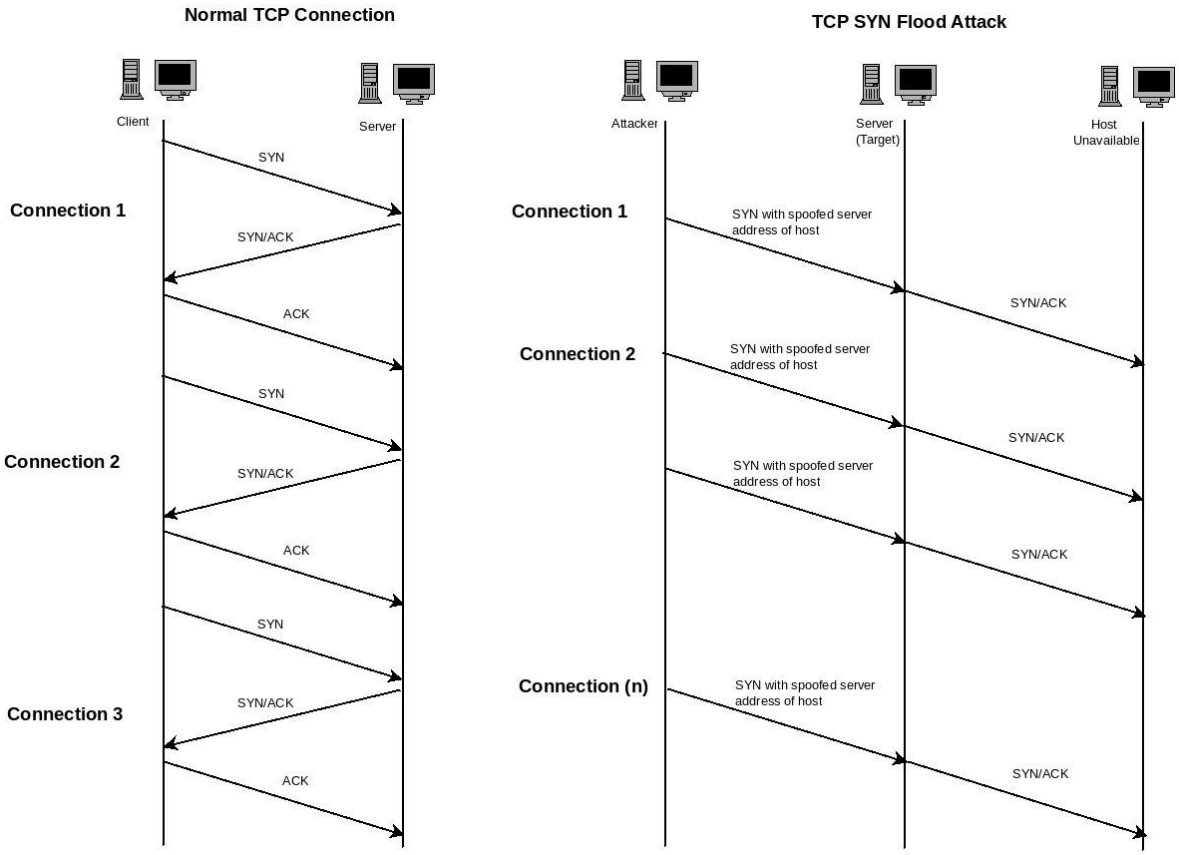

Figure 1. The comparison between normal TCP connection and TCP SYN flood attack

Based on the background described earlier, the purpose of this research is to utilize Configserver Security and Firewall (CSF) as a tool used to secure server from TCP SYN Flood (DoS) Attack. CSF is a firewall configuration script created to provide better security on servers with an easy-to-use interface. CSF most of the services on the internet in the form of websites that are run through a web server, have many features for server security such as the Login Failure Daemon which functions to monitor excessive login failure activities, and other configurations to secure the server from DoS, as well as manage permitted connections.

As a state of the art, there are 20 previous researches about Denial of Service (DoS) attack and Distributed Denial of Service (DDoS) attack that related with this research with various of case studies. Kshirsagar, *et al.*, at their publication describe the proposes and implementation of Denial of Service (DoS) detection framework which consists of a packet sniffer, feature extraction, attack detection, and output module, that can detect TCP SYN Flood based on threshold and misuse detection [5]. Manna and Apmhawan review the state-of-the art of detection mechanisms for SYN flooding and performance measure [6]. Bogdanoski, *et al.*, analyzes systems vulnerability targeted by TCP segments when SYN flag is ON, which gives space for a DoS attack called SYN flooding attack or more often referred as a SYN flood attack [7]. Siregar analyze the cause of the Denial of Service (DoS) attack on a web system using literature study [8]. Polat, *et al.*, detect the Distributed Denial of Service (DDoS) attacks in Software Define Network (SDN) using machine learning-based models [9].

Fadil doing new approach in detecting Distributed Denial of Service (DDoS) attacks that is expected to be a relation with Intrusion Detection System (IDS), to predict the existence of DDoS attacks based on average and standard deviation of network packets in accordance with the Gaussian method [10]. Prajapati, *et al.*, review recent of detection methods for HTTP DDoS attacks and recognizing detection attacks at the application layer [11]. Gavric, *et al.*, in their paper describes

some of the most common Denial of Service (DoS) attacks and potential methods of protection against them [12]. Vinnarasi, *et al.*, proposed the host-based IDSs (HBIDS) as a security solution for Software Define Network (SDN) using host-based Intrusion Detection System (IDS) over Distributed Denial of Service (DDoS) Attack [13]. Poongothai, *et al.*, simulate an environment by extending NS2, setting attacking topology and traffic, which can be used to evaluate and compare the methods of DDoS attacks and tools, so that bssed on the simulation and evaluation results, more efficient and effective algorithms, techniques and procedures to combat these attacks can be developed [14].

Jafaar, *et al.*, viewed 12 recent detection of DDoS attack at the application layer published between January 2014 and December 2018 [15]. Alomari, *et al.*, present a comprehensive study to show the danger of Botnet-based DDoS attacks on application layer, especially on the web server and the increased incidents of such attacks that has evidently increased recently [16]. Mahjabin, *et al.*, at their paper provide a systematic analysis of this type of attacks including motivations and evolution, analysis of different attacks, protection techniques, mitigation techniques, possible limitations and challenges of existing research [17]. Keromytis, *et al.*, propose an architecture called Secure Overlay Services (SOS) that proactively prevents DoS attacks, geared toward supporting Emergency Services or similar types of communication [18]. Masdari and Jalali presents an in-depth study of the various types of the DoS attacks proposed for the cloud computing environment and classifies them based on the cloud components or services,which they target [19]. Hong, *et al.*, in their paper propose a network-based Slow HTTP DDoS attack defense method, which is assisted by a software-defined network that can detect and mitigate Slow HTTP DDoS attacks in the network [20].

Shaar and Efe published their result of survey that discuss in detail the classification of DDoS attacks threatening the cloud computing components and make analysis and assessments on the emerging usage of cloud infrastructures that poses both advantages and risks [21]. Cheng, *et al.*, propose an Adaptive DDoS Attack Detection Method (ADADM) based on Multiple Kernel Learning (MKL) [22]. Nathiya in his paper describe about The DDOS attack, DDOS attack architecture and DDOS attack various technologies, mitigation of DDOS, countermeasures of DDOS attack, how to process hardware checking methods and to detect and preventing DDOS attacks in tools [23]. Dao, *et al.*, proposes a novel solution called Adaptive Suspicious Prevention (ASP) mechanism to protect the controller from the Denial of Service (DoS) attacks that could incapacitate a Software Define Networking (SDN) [24]. Linux operating system (CentOS, Ubuntu, Debian, SUSE) is widely used as an operating system for the server [25]. In this research, both of server and attacker using Ubuntu Linux and at server it also running XAMPP (Apache webserver, MySQL database server, PHPMyAdmin), Wireshark, and CSF. There are two objectives to use both of CSF and SPI method as a tool to secure the server from TCP SYN (DoS) attack: 1.) To streamlining the performance of the server to serve its users, 2.) To improve server security to realize the security principle of a service.

2. RESEARCH METHOD

The research using Experimental Methodology that consist of some steps of implementation and analysis of result, including literature review, solution design and testing scenario, implementation, testing, analyzing, discussion, documentation, and publication [26]. In this research, both of server and attacker computer using Ubuntu Linux. After this process, the next process is to attack from the attacker to the server in form of TCP SYN Flood (DoS) attack using Xerxes. The next step is to install CSF on server and configure it to prevent TCP SYN Flood (DoS) attack. Server which has been installed CSF is attacked again using Xerxes from attacker. Finally, a comparison of the results of the two attacks in terms of resource used and server availability to serve clients. A sequence chart of this research process is shown in the Figure 2:

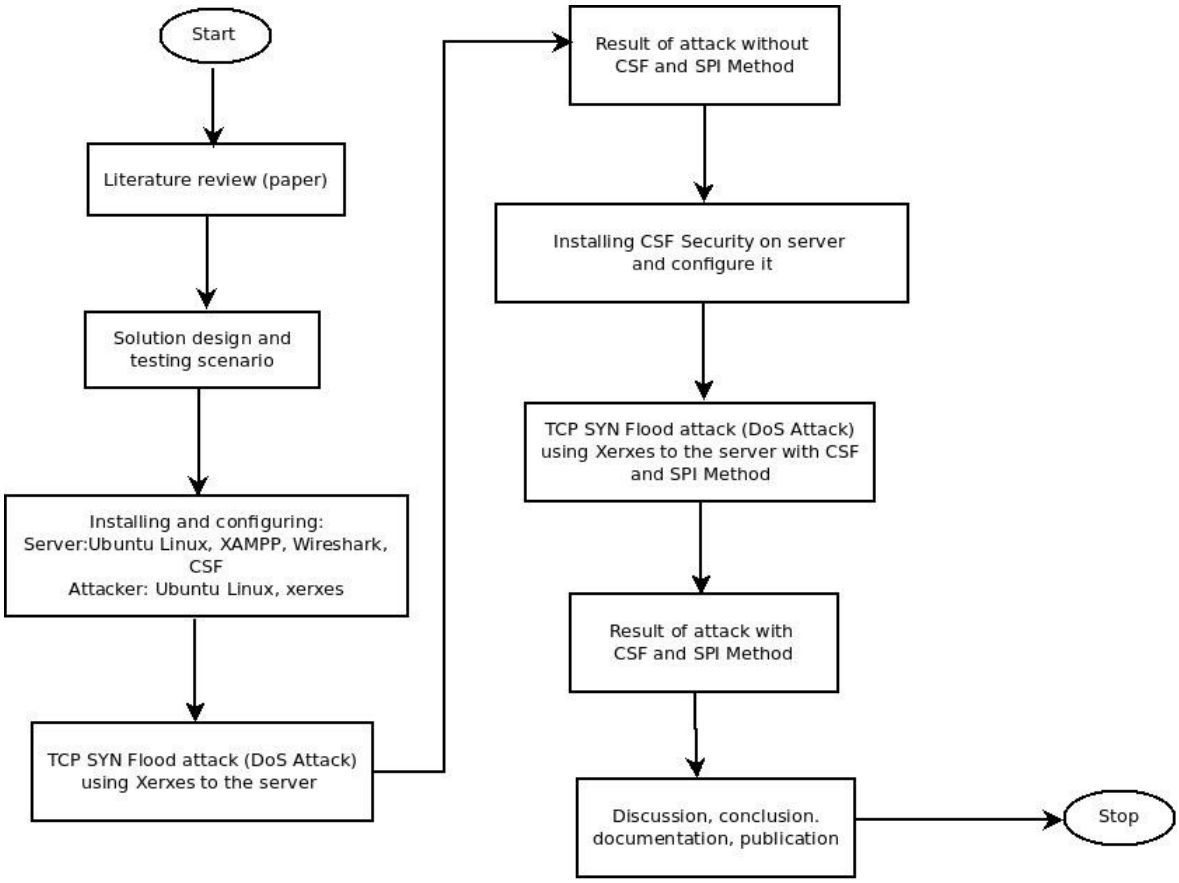

Figure 2. Sequence chart of the research process based on Experimental Methodology

The network topology in this research was designed with wireless point to point topology. The network topology design for this research shown at Figure 3 below:

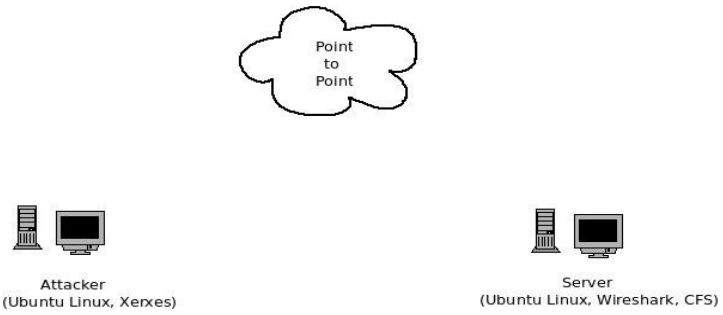

Figure 3. Network topology design

3. TESTING, RESULTS AND DISCUSSION

Based on the sequence chart, testing is done twice for servers without CSF and SPI method and using CSF with SPI method. After that, the result of testing both without CSF and using CSF (with SPI method) is compared. Attacker has an IP Address 10.42.0.185 with Netmask 255.255.255.0 and Broadcast Address 10.42.0.255. Server (target) has an IP Address 10.42.0.1 with Netmask 255.255.255.0 and Broadcast Address 10.42.0.255. Both of them are connected wireless point to point. Server running Apache webserver, MySQL database server, and PHPMyAdmin using XAMPP on Ubuntu Linux, shown in Figure 4:

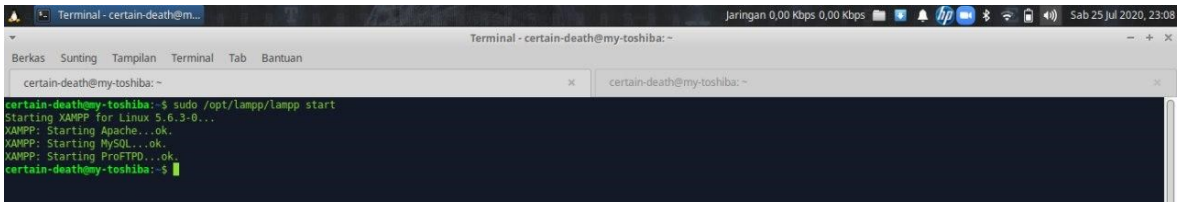

Figure 4. Server running Apache webserver, MySQL database server, PHPMyAdmin

Attacker using Xerxes to attack server with TCP SYN Flood (DoS) Attack. The compile process of Xerxes in C using this command:

```
deathknell@my-small-machine:~$ gcc -o xerxes xerxes.c
```

After that the compiled xerxes file is given permissions so that it can be running, using this command:

```
deathknell@my-small-machine:~$ sudo chmod +x xerxes
[sudo] password for deathknell:
deathknell@my-small-machine:~$
```

Finally, running Xerxes using this command to the server with IP Address 10.42.0.1 and port number 80 :

```
deathknell@my-small-machine:~$ ./xerxes 10.42.0.1 80
```

The process of TCP SYN Flood (DoS) Attack from attacker to the server shown in Figure 5 below:

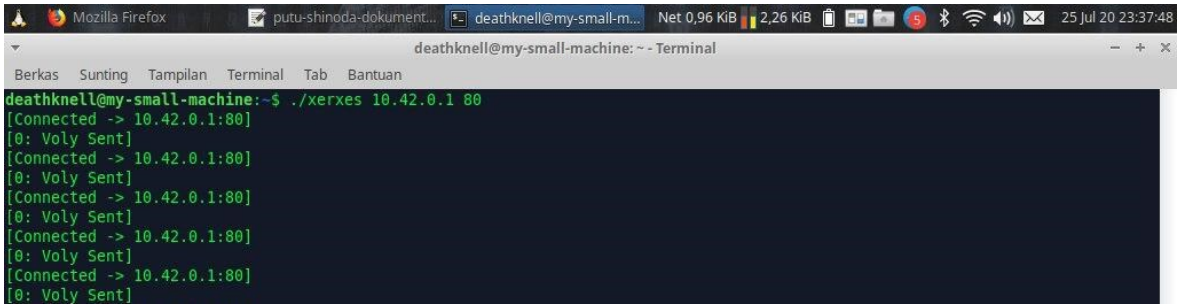

Figure 5. TCP SYN Flood (DoS) Attack from attacker to the server

3.1. Testing and Result Without CSF and SPI Method

Testing is done by flooding the server with many of packages, using TCP SYN Flood (DoS) Attack. The first condition carried out in testing is to attack the server without CSF protection and from the client trying to make an HTTP request to the server. The attack process has been carried out through the attacker connected to the server. The attacking process sent a TCP packet to the server successfully. On the other hand, the server without CSF seems overwhelmed by the attack. Using Terminal and Top command, it can show the result of attack without CSF protection. Evidenced by the HTTP daemon process (httpd) on the server increases in terms of memory and CPU usage. The following is a capture of the condition of the server resource on server before the attacking process and during the attacking process, using Top command on Terminal, shown on Figure 4 and 5:

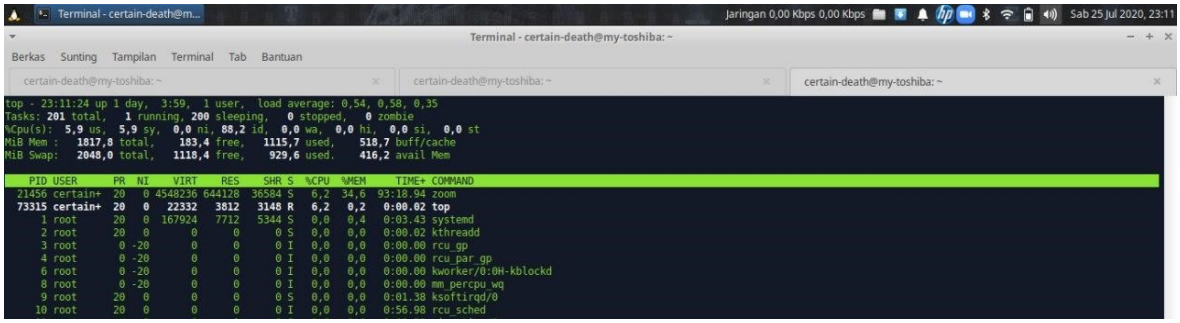

Figure 6. Condition of server resource before attack (without CSF and SPI)

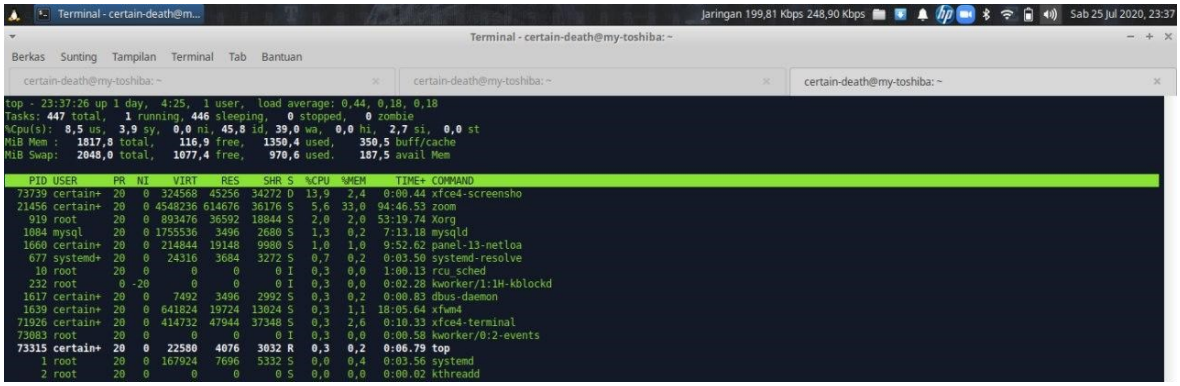

Figure 7. Condition of server resource during attack (without CSF and SPI)

During the TCP SYN Flood (DoS) attacking process, the process that appears associated with httpd increases rapidly, so that CPU consumption rises and memory usage also increases. The increase in memory increased from 115,7 MB to 1350,4 MB. Existing processes in the sleep state also increase from 200 processes to 446 processes. It is means that 246 processes are waiting for something. We can capture and analyzing them using Wireshark on eth0 interface. Network capture from the server side proves that there are many TCP packets sent continuously to the server. The server replies it with the SYN-ACK packet. Capture results on the server-side show that the TCP packets with SYN ACK are used to connect to the server, that have not been replied by the server, so that connections do not form. In this case, the attacker keeps trying to connect to the server. The results of the attacker network traffic capture to the server using Wireshark can be shown at the Figure 5 below at TCP Protocol from IP Address 10.42.0.185 (attacker) to IP Address 10.42.0.1 (server) :

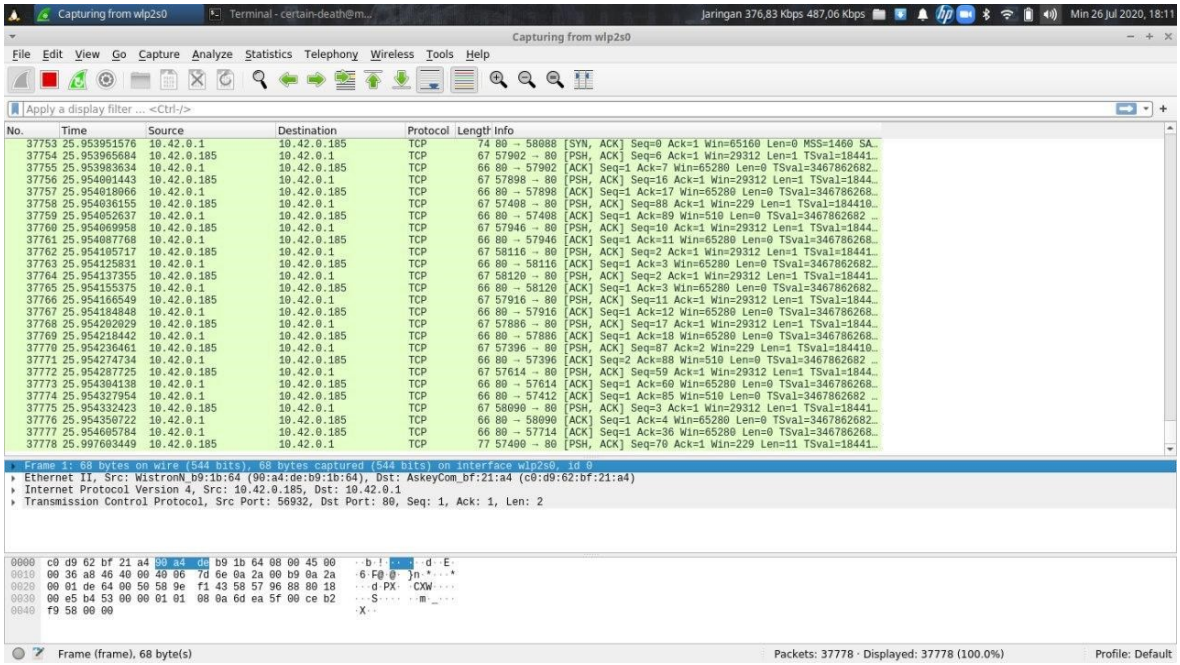

Figure 8. Capture of network traffic to the server (without CSF and SPI) using Wireshark

3.1. Testing and Result Using CSF With SPI Method

The second test was carried out using CSF and SPI method on the server. CSF can be download by wget using this command:

```
certain-death@my-toshiba:~$ wget http://download.configserver.com/csf.tgz
```

This file saved at /home and can be extract using this command:

```
certain-death@my-toshiba:~$ tar -xzf csf.tgz
```

After the extracting process, it creates the scf directory at /home and change to csf directory usng this command:

```
certain-death@my-toshiba:~$ cd csf/
```

After that, the install.sh should be give the execution permission using this command, so that SCF can be installed:

```
certain-death@my-toshiba:~/csf$ chmod +x install.sh
```

CSF should be installed using sh command with root privillege:

```
certain-death@my-toshiba:~/csf$ sudo sh install.sh
```

After the installation process, CSF can be running using this command (with root privillege):

```
certain-death@my-toshiba:~/csf$ sudo perl /usr/local/csf/bin/csftest.pl
```

While it running, CSF can be configured. The configuration file of CSF with SPI method located at /etc/csf/csf.conf, edited using nano editor on Terminal. There are three ways to configure CSF using SPI method. First, a state CSF is set up along with TCP and UDP packets. For CSF conditions, it is changed from testing to non-testing, by changing the TESTING line from value 1 to 0 (TESTING = "0") so that security can be done. Second, the next is setting the port number on the network that is allowed to communicate using the TCP protocol and receiving TCP packets that originate from that port number. The port numbers that allowed in this research are 20, 21, 22, 25, 53, 80, 110,143,443,465,587,993 and 995. For this reason, the port numbers are entered in the TCP_IN line (TCP_IN = "20,21,22,25,53, 80, 110,143,443,465,587,993,995 "). As for receiving TCP packets from allowed port numbers, this is done via the TCP_OUT line. In this research, the number of ports allowed for sending TCP packets is 20,21,22,25,53,80, 110,113,443,587,993,995, so the setting is TCP_OUT="20,21,22,25,53,80, 110,113,443,587,993,995". The same is done for sending and receiving UDP packets for allowed port numbers. For access to UDP packet access is given on port 20.21. and 53, which are arranged via the UDP_IN line (UDP_IN = "20,21,53"). Whereas outgoing access for UDP packets is applied to port numbers 20,21,53,113 and 123, through the settings on the UDP_OUT line (UDP_OUT= "20,21,53,113,123").

Second, prevention of flooding which is commonly done through TCP SYN Flood (DoS) Attack, is done in the PORTFLOOD line. The PORTFLOOD line is filled in with a value of 80; tcp; 50; 10 (PORTFLOOD="80; tcp; 50; 10"). This value means that the CSF and SPI method only allows connections to port 80 for 50 connections per 10 seconds. In addition, the CONNLIMIT line is set to a value of 80; 10 (CONNLIMIT="80; 10"). This means that the CSF and SPI method limits connections to port 80 per 10 seconds. The SYNFLOOD protection option is also activated by giving a value of 1 (SYNFLOOD="1"). Limiting the number of packets to 50 data packets per second is done via the SYNFLOOD_RATE line with a value of 50 / s (SYNFLOOD_RATE = "50 / s"). Finally, blocking of IP Addresses that carry a packet limit above 10 , via the SYNFLOOD_BURST row with a value of 10 (SYNFLOOD_BURST = "10").

Third, to limit the number of connections, the CT_LIMIT line is set to a value of 50 (CT_LIMIT="50"), which indicates that the CFS security and SPI method will limit connections from any computer (along with IP Address) that has more than 50 connections. restrictions on the number of connections of certain port numbers are made. In this research, restrictions were placed on port number 80, via the CT_PORTS line (CT_PORTS = "80"). The list of line with some of parameter and value at CSF using SPI method, shown in the following configuration:

```
TESTING=“0”
TCP_IN="20,21,22,25,53,80,110, 143,443,465,587,993,995"
TCP_OUT="20,21,22,25,53,80,110,113,443, 587,993,995"
UDP_IN="20,21,53"
UDP_OUT="20,21,53,113,123"
PORTFLOOD=“80;tcp;50;10”
CONNLIMIT="80;10”
CT_LIMIT=“50”
CT_PORTS="80"
SYNFLOOD="1"
SYNFLOOD_RATE="50/s"
SYNFLOOD_BURST="10"
```

After CSF configuration using SPI method, the scenario begins with the attack from the attacker using Xerxes. At the same time, attacker made HTTP requests to the server. The result show that when an attack is carried out, the Xerxes tools seem unable to connect to the server. By utilizing CSF and SPI method, the resource of server (CPU and memory used) also did not increase dramatically during the attacking process. Also, the existing processes in sleep states do not increase. This can be seen from the output displayed via the Top command on the Terminal, that shown at Figure 9 below:

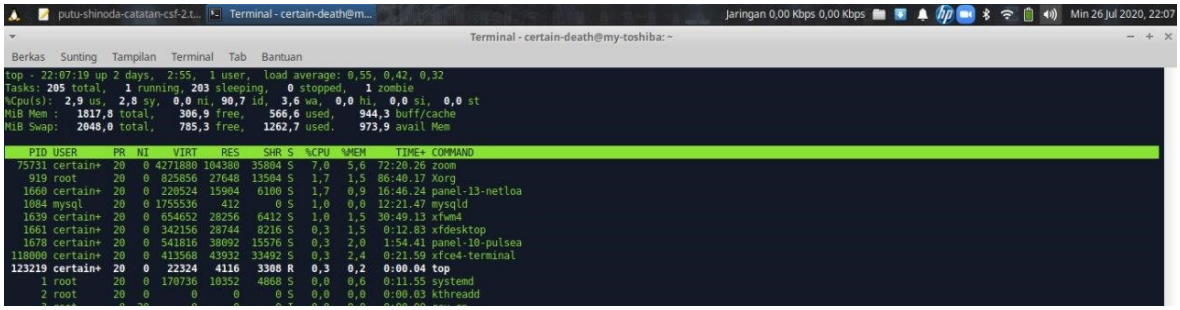

Figure 9. Condition of server resource after attack (using CSF and SPI)

Using Wireshark, an observation is made of network traffic from the attacker's side. Observations show that with the addition of CSF and SPI methods, the server can limit and stop replies to packets sent by the attacker, before it is disconnected. Figure 10 shows the capture results using Wireshark:

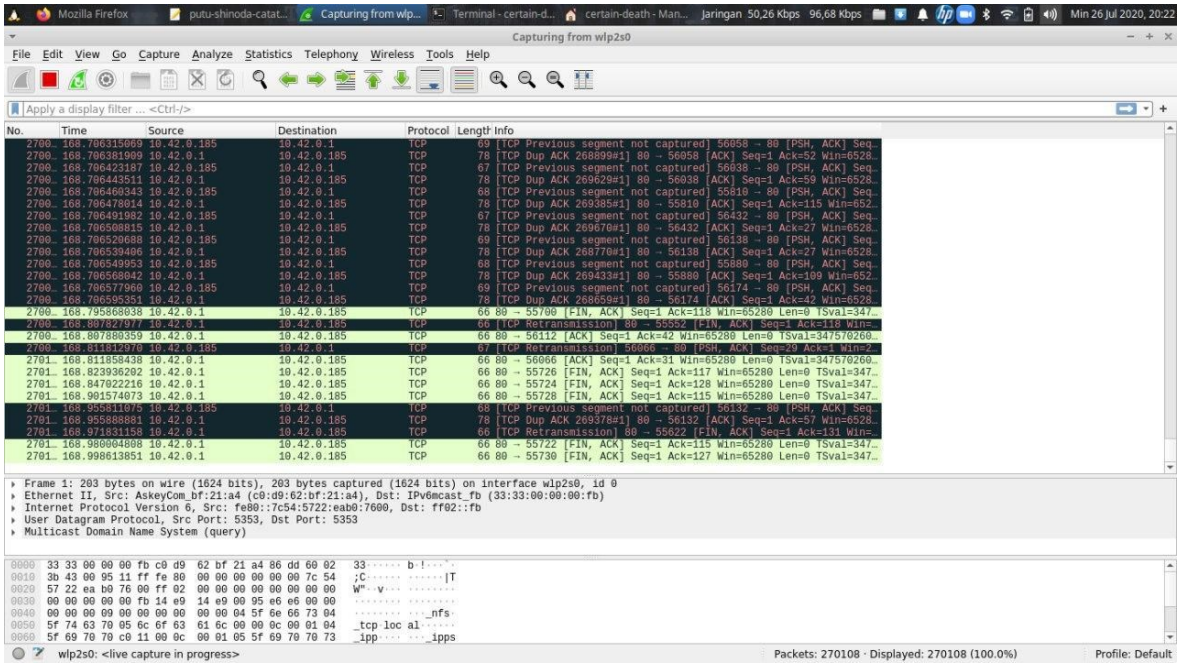

Figure 10. Capture of network traffic to the server (using CSF and SPI)

4. CONCLUSION

Based on the testing result, it shows that when the server is not protected by CSF and SPI method, it cannot block TCP SYN Flood (DoS) attacks that lead to port 80. So that when attacker want to make HTTP requests to the server, it cannot be served because too many packages are being sent. The packet for the connection between the attacker and server has not been replied to by server, so that in capturing network traffic, the attacker is seen trying to repeat the connection to the server. Because it is not connected, the page requested by the attacker cannot be displayed. Besides that, from the server-side, the server resource used, among of memory usage, CPU, and the number of

running processes, increases dramatically. When CSF and SPI methods implemented in servers with various configurations, the server is safer. The configuration causes when there is an excessive connection to the server, then the next connection will be disconnected and the IP Address from the attacker will be blocked for the 30 minutes. This makes the client request to the server does not experience any problems. Resource server used also was not increase. CSF and SPI method provides several configurations to protect Linux from DoS attacks. Based on the testing result at this research, it can be concluded that CSF is a smart firewall tool on Linux operating system that can effectively deal with TCP SYN Flood type DoS Attack, combine with the SPI method. CSF also offers a simple and faster way to mitigate the kind of TCP SYN Flood (DoS) attack as a small case attack. There are also other methods that can be used in this similar case, i.e.: Advanced Policy Firewall (APF), firewall, null-route method for the server's IP address, and Cloudflare service, but CSF offer more benefits based on its configuration at /etc/csf/csf.conf. CSF is possible to ward off types of DoS regarding the expert of the administrator to configure csf.conf file in case to prevent the attack.

ACKNOWLEDGEMENTS

This research was carried out independently in 2020 during the Corona pandemic, using Linux operating systems and some open source applications at two computers point to point connected to each other. Gratitude to all supports from the Linux Community and Open Source Community during this research.

REFERENCES

[1] A.A. Zabar and F. Novianto, Keamanan HTTP dan HTTPS Berbasis Web Menggunakan Sistem Operasi Kali Linux," Jurnal Ilmiah Komputer dan Informatika (KOMPUTA) Vol.4 No.2, 2015.
[2] F. Komariyah1 and H. Argyawati, "Implementasi Server Cluster Hight Availability Pada Web Server Dengan Sistem Operasi Turnkey Linux Menggunakan Heartbeat," Jurnal Penelitian Ilmu Komputer, System Embedded and Logic 4(2):78-88, 2016.
[3] S.Q. Mir, "Investigating the Denial of Service Attack: A Major Threat to Internet and the Security of Information," JK Research Journal in Mathematics and Computer Sciences, Vol.(1) No.(1), 2018.
[4] M. Bogdanoski, "TCP-SYN Flooding Attack in Wireless Networks," 2012 International Conference Innovations on Communication Theory (INCT), 2012.
[5] D. Kshirsagara, et al., "CPU Load Analysis and Minimization for TCP SYN Flood Detection,", 2016 International Conference on Computational Modeling and Security (CMS), pp. 626 – 633, 2016.
[6] M.E. Manna and A. Amphawan, "Review of SYN Flooding Attack Detection Mechanism," International Journal of Distributed and Parallel Systems (IJDPS) Vol.3, No.1, January 2012.
[7] M.Bogdanoski, et al., "Analysis of the SYN Flood DoS Attack," International Journal of Computer Network and Information Security (IJCNIS), Vol.8, pp.1-11, 2013.
[8] J.J Siregar, "Analysis Eksploitasi Keamanan Web Denial of Service Attack," ComTech Journal BINUS, Vol.4 No.2, 2013.
[9] H. Polat, et al., "Detecting DDoS Attacks in Software-Defined Networks Through Feature Selection Methods and Machine Learning Models," MDPI Journal Sustainability, 2020.
[10] A. Fadil, et, al, "A Novel DDoS Attack Detection Based on Gaussian Naive Bayes," Bulletin of Electrical Engineering and Informatics, Vol.6, No.2, pp.140-148, 2017.
[11] P. Prajapati, et al., "A Review of Recent Detection Methods for HTTP DDoS Attacks," International Journal of Scientific and Technology Research, Vol.8, Issue 12, 2019.
[12] Z. Gavric, et al., "Overview of DOS attacks on wire-less sensor networks and experimental results for simulation of interference attacks," Ingeniería e Investigacion, 38(1), 130-138, 2018.
[13] J. Vinnarasi, "Security Solution for SDN Using Host-Based IDSs Over DDoS Attack," International Journal of Emerging Technology and Innovative Engineering, Volume 5, Issue 9, 2019.
[14] M. Poongothai, et al., "Simulation and Analysis of DDoS Attacks," 2012 International Conference on Emerging Trends in Science, Engineering and Technology (INCOSET), 2012.
[15] G.A. Jaafar, et al., "Review of Recent Detection Methods for HTTP DDoS Attack," Hindawi Journal of Computer Networks and Communications, 2019.
[16] E. Alomari, et al., "Botnet-based Distributed Denial of Service (DDoS) Attacks on Web Servers: Classification and Art," International Journal of Computer Applications, Volume 49–No.7, 2012.
[17] T. Mahjabin, et al., "A Survey of Distributed Denial-of-Service Attack, Prevention, and Mitigation Techniques," Journal Sage Pub Volume: 13 issue: 12, 2017.

[18] A.D. Keromytis, et al., “SOS : An Architecture For Mitigating DDoS Attacks,” Journal on Selected Areas in Communications,Vol.21,2013.
[19] M. Masdari, and M. Jalali, "A Survey and Taxonomy of DoS Attacks in Cloud Computing," Security and Communication Networks, 2016.
[20] K. Hong, et al., "SDN-Assisted Slow HTTP DDoS Attack Defense Method," IEEE Communications Letters Volume: 22 , Issue: 4 , 2018.
[21] F. Shaar and A. Efe, "DDoS Attacks and Impacts on Various Cloud Computing Components," International Journal of Information Security Science, Vol.7, No.1, 2018.
[22] J. Cheng, et al., "Adaptive DDoS Attack Detection Method Based on Multiple-Kernel Learning," Hindawi Security and Communication Networks, 2018.
[23] T.Nathiya, "Reducing DDOS Attack Techniques in Cloud Computing Network Technology," International Journal of Innovative Research in Applied Sciences and Engineering (IJIRASE) Vol.1(3), 2017.
[24] N.N. Dao, et al., "Adaptive Suspicious Prevention for Defending DoS Attacks in SDN-Based Convergent Networks," PLOS journal, 2016.
[25] N. Nelmiawati, et al., "Rancang Bangun Lab Komputer Virtual Berbasis Cloud Computing Menggunakan Openstack Pada Jaringan Terpusat," JAIC, vol.2, no.1, pp.11-17, 2018.
[26] L.B. Christensen, "Experimental Methodology, 10th Edition," Willey. 2012.