Denial of Service Attack Generator in Apache JMeter

Stepan Grabovsky
Departement of Telecommunications
Brno University of Technology
Brno, Czech Republic

Petr Cika
Departement of Telecommunications
Brno University of Technology
Brno, Czech Republic
cika@feec.vutbr.cz

Vaclav Zeman
Departement of Telecommunications
Brno University of Technology
Brno, Czech Republic

Vlastimil Clupek
Departement of Telecommunications
Brno University of Technology
Brno, Czech Republic

Milan Svehlak
Departement of Telecommunications
Brno University of Technology
Brno, Czech Republic

Jan Klimes
Departement of Telecommunications
Brno University of Technology
Brno, Czech Republic

***Abstract*—Cyber attacks are currently a major threat to individuals as well as businesses or institutions. Denial of service (DoS) attacks are very common cyber attacks that focuses on availability of network services or network devices. This paper deals with developing of new modules of DoS attacks and network traffic generator into Apache JMeter that is designed to load test functional behavior and measure performance. Modules developed together with Appache JMeter tool can test a network infrastructure and find weak localities prone to DoS cyber attacks. One goal of this paper is to compare tools for stress testing of web servers and services. The network generator module of DoS attacks allows stress testing of networks based on the Transmission Control Protocol/Internet Protocol (TCP/IP). With using the generator developed it is possible to test the availability of network devices when exposure to DoS attacks. Selected cyber attacks targeting on the availability of network resources (DoS attacks) that can test a resistance of a network infrastructure are described in the article. DoS attacks described have been implemented in the Apache JMeter tool together with the design and the development of the network generator of DoS attacks. New modules can be used to test a resistance of network infrastructure. Finally, the measured values using the generator developed are presented and discussed.**

***Index Terms*—denial of service, distributed denial of service, load test, network performance**

I. Introduction

Denial of Service (DoS) attacks are very powerful technique to attack the network devices and services. This type of attack can unplug different services from the internet and is very simple to implement. Distributed Denial of Service attack (DDoS) is more powerful type of DoS. The number of DDoS attacks in all over the world rises up every year. DDoS attack can be devided into three types. One type uses massive amounts of bogus traffic to down a resource such as a website or server, another type uses packets to target the network infrastructure and infrastructure management tools. The third type uses application layer services. [1]

Recently, very common attacks have been HTTP Flood (Hypertext Transfer Protocol), SYN Flood, ICMP Flood (Internet Control Message Protocol), NTP (Network Time Protocol) Flood, UDP (User Datagram Protocol) Flood, DNS Server Flood (Domain Name Server), DNS Amplification, Slowloris. [2] These attacks are described below.

A. Common DDoS Attacks

1) HTTP Flood: HTTP flood is common attack sending a lot of HTTP requests to web server. This attack is common in most Botnet software programs. [3] Before sending an HTTP request, a valid TCP connection must be established. HTTP flood attack becomes very similar to normal web traffic, HTTP GET and HTTP POST requests are used. This type of DoS attack is extremely difficulty to detect. [4]

2) SYN Flood: The SYN flood attack uses SYN attribute of Transmission Control Protocol (TCP) during TCPs three-way handshake mechanism (SYN, SYN-ACK, ACK). Each SYN request create half-open connection. An attacker (client) sends TCP SYN requests to server that looks like legitimate traffic. These requests contain non-existing IP adresses. Server creates new thread for each TCP SYN request, allocates buffer and memory and tries sending of TCP SYN-ACK response back to client. TCP ACK response is never sent back to the server. After that server has a lot of half-open connection and no memory. [5]

3) ICMP Flood: ICMP flood works on the fourth level of ISO/OSI reference model and cooperates with ICMP protocol. Client sends to server a lot of ICMP requests during the attack, most often the Echo requests well known as ”ping”. Server tries to answer but finally does not have a memory and network capacity. [6]

4) NTP Flood: NTP flood also called NTP amplification works on application layer of TCP/IP and uses Network time Protocol (NTP). The principle of the attack consists of

Research described in this paper was financed by the National Sustainability Program under grant LO1401 and by the Ministry of the Interior of the Czech Republic under grant no. VI20152018002. For the research, infrastructure of the SIX Center was used.

978-1-5386-9361-2/18/$31.00 ©2018 IEEE

concealing the victim by NTP server responses to the query ”get monlist”, which returns the list of the last 600 users who joined the NTP server. Attacker sends this message to the server with IP address of victim. [7]

5) UDP Flood: UDP flood attack is based on sending a lot of udp datagrams to various server ports with false source IP address. Server cannot process all received UDP datagrams and uses all capacity to send of ICMP message ”destination unreachable”. [8]

6) DNS Flood: DNS flood is attack used against DNS server. An attacker sends many DNS requests with false IP addresses to DNS server and DNS server answer. DNS requests of legitimate users cannot be processed. [9]

7) DNS Amplification: DNS Amplification uses DNS server as mediator. An attacker sends requests to DNS server with sender IP address of the victim. DNS requests are selected to Internet domain with many DNS records. [10]

8) Slowloris: Slowloris is based on opening many sockets with target web server and keeps this connection for as long as possible. An attacker repeats sending incomplete HTTP messages and socket must stay open. Thanks to correct sending process the slowloris attack is difficult to detect. [11]

B. Tools for Stress Testing

There is currently a number of Open Source-based tools that can be used to stress testing of web servers or web services. The article focuses on tools suitable to generate HTTP / HTTPS load (GET, POST and PUT) and FTP load (PASV and RETR file transfer). Only the tools with open license distribution (GNU, Apache License, etc.) and executables on Linux kernel operating systems were selected for the following performance analysis. Apache Bench, Httperf, Siege, and Apache JMeter were selected to performance analysis. The purpose of this performance analysis was to select the appropriate default software environment for developing a DoS attack network generator.

1) ApacheBench: ApacheBench is one thread software controlled by command line suitable for web server stress testing. The limit of AppacheBench uses only one thread. [12]

2) Httperf: Httperf is a web server performance testing tool that allows you to generate different HTTP loads. Httperf uses only one processor thread and is therefore inappropriate for DoS attacks. [13]

3) Siege: Siege is an HTTP web server testing tool for multiple-threaded work. The tool allows to load the server by a set number of simulated users. [16]

4) Apache JMeter: Apache JMeter is an open-source software designed for stress testing of servers. The software is written in Java. Originally, this tool has been developed to test web applications, but has gradually expanded with other features. [16]

C. Performance testing

For performance testing of ApacheBench, Httperf, Siege, Apache JMeter, the Avalanche 3100 device was used in the Web Server role (Apache HTTP Server 2.0.46 UNIX, 2 x Intel Xeon X5580 3.33 GHz CPUs, 48 GB RAM) and CentOS Linux 7 was used as the load generator (Intel (R) Xeon (R) CPU, E5506 @ 2.13 GHz, 24 GB RAM). Devices were connected to a 10 Gbps laboratory network.

Within the test, a user has been defined to transfer a total of 15 HTTP requests and his web browser performs 5 competing transactions (each of these transactions transmits 64 B data). Table 1 lists the maximum number of transactions per second for each tool. Apache JMeter of all analyzed tools was able to generate most transactions per second, approximately 34.000 transactions per second.

TABLE I
MAXIMUM NUMBER OF TRANSACTION - STRESS TEST TOOLS

Stress test tool	**Maximum number of transaction per second**
ApacheBench	20 493
Httperf	10 000
Siege	14 596
Apache JMeter	34 069

Due to Table I it can be considered that the Apache JMeter is suitable for development of DoS generator.

II. DENIED OF SERVICES ATTACKS GENERATOR

Apache JMeter is open source software written in Java and designed to variable network and services tests. Any DoS attacks module has been included in Apache JMeter. The main point of our work was to develop new modules suitable for DoS attacks. Trafgen [16], multithread network packet generator, was chosen to cooperate with JMeter.

During research the connector between Apache JMeter and trafgen was developed. Intercommunication takes place as follows (Fig. 1):

1) User sets parameters of DoS attack using Apache JMeter GUI and starts test plan,
2) parameters are processed by the developed Apache JMeter-Trafgen connector, the configuration file for trafgen is created,
3) Trafgen is started through Apache JMeter-Trafgen connector together with parameters from configuration file,
4) After the attack is done, trafgen is closed using KILL Linux comman and trafgen Process Identificator (PID).

Next to Apache JMeter-Trafgen connector a new DoS attacks modules for Apache JMeter were developed: HTTP, SYN, ICMP, NTP, UDP, DNS Server, DNS Amplification, universal DoS Slowloris.

III. TESTING OF DEVELOPED GENERATOR

Performance of DoS generator developed was tested on 2 configurations:

1) personal computer 1: eight core AMD FX-8350, 4GHz, 8GB RAM.
2) personal computer 1: two core Intel Core 5 6200U, 2,3 GHz, 8GB RAM.

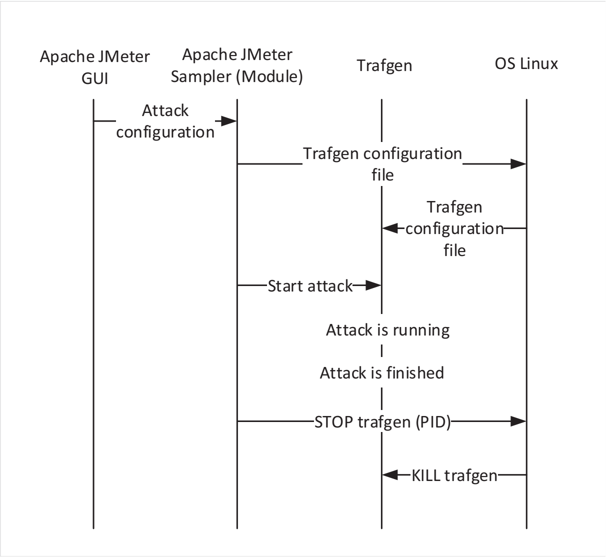

Fig. 1. Apache JMeter-Trafgen Connector.

TABLE II
PERFORMANCE OF HTTP FLOOD MODULE

Resources	**Time [s]**	**Upload speed [pps]**	**Number of packets** *packets*	**Network throughput [Mbps]**
4 CPU, 4GB RAM	362	45	16342	1.300
6 CPU, 6GB RAM	409	56	23078	1481

Both computers used virtual machine with Linux CentOS 7 and Appache webserver.

HTTP Flood attack was tested on personal computer 1 with different number of processors and RAM. The results of tests with different resources are shown in Table II.

More system power means more transmission speed as can be seen in Table II. Totally, with adding fifty percent system power, the power of HTTP Flood module developed rises around thirteen percent. The priority of throughput during HTTP Flood attack is not so big. The force of the attack lies in its proper direction.

Table III shows performance test results for SYN Flood, ICMP Flood, NTP Flood, and Universal DoS Module. Table 3 lists the duration of the test in seconds, the average packet upload speed, and the average throughput. The PC1 virtual machine on which the DoS modules were started, had available system resources for 6 CPUs and 4 GB of RAM. Always ten milion packets have been sent at the highest upload speed.

TTL was set to 64, and the request type was set to 8 (Echo Request) during the ICMP Flood test. The ICMP flood attack configuration file was uploaded to test the universal module that was later changed. The TTL value was set to 36, and the request type was set to 13 (time stamp request).

TABLE III
PERFORMANCE OF SYN FLOOD, ICMP FLOOD, NTP FLOOD AND UNIVERSAL DOS MODULE

DoS module	**Time [s]**	**Upload speed [pps]**	**Network throughput [Mbps]**
SYN Flood	118	88.495	36.0
ICMP Flood	121	86.206	34.4
NTP Flood	110	93.457	133.6
Universal	120	86.965	34.4

TABLE IV
PERFORMANCE OF SYN FLOOD AND NTP FLOOD DEPENDING ON NUMBER OF CPUS

	SYN Flood			**NTP Flood**		
CPU	**Time [s]**	**Upload Speed [Mbps]**	**Network Usage**	**Time [pps]**	**Upload Speed [Mbps]**	**Network Usage**
1	436	23.201	11.2	451	22.421	40.8
2	210	48.780	24.8	203	50.251	92.8
3	196	51.020	28.0	193	53.191	102.4
4	178	57.803	36.0	169	60.975	122.4
6	118	88.495	38.4	125	83.333	133.6

The packet size was 66 B for SYN Flood module, 64 B for ICMP Flood module and 234 B for NTP Flood module. The performance of the individual modules was comparable. Only the NTP Flood module due to its larger packet size achieved greater load. NTP Flood also has a slightly higher packet speed than other DoS modules. This phenomenon is attributed to the absence of dynamic IP address generation (NTP Flood is one destination address and one NTP address). Trafgen tool developers report that the use of these dynamic functions affects the performance of the generator.

Table IV shows the results of performance test for the SYN Flood and NTP Flood DoS modules, depending on the number of CPUs. The greatest increase in performance was when switching from one kernel to two kernels. RAM does not affect module performance because memory consumption of the generator is very low. RAM memory consumption was approximately 200 MB.

The Slowloris attack functionality was tested by opening a single communication socket at the Apache HTTP server (on the PC2 virtual machine) using a common HTTP query. The Slowloris attack continued sending HTTP headers every fifteen seconds to ensure an open connection to the web server. The Slowloris attack has been dropped from experimental performance testing because it works on a completely different principle than flooded DoS attacks.

Experimental performance testing of UDP Flood, DNS Server Flood and DNS Amplification Flood DoS modules was performed using personal computer with a 2.2 GHz Intel Core i55200 processor,16 GB of RAM, a 240 GB solid state drive, and Windows 10 Pro x64 operating system. Two Linux CentOS 7 virtual machines were created in the VMware Workstation 11 environment. The first virtual machine was equipped with a developed DoS attack network generator. The Apache HTTP server and the BIND DNS server7 were installed on the second virtual machine. The results of the experimental performance testing are shown in Tables V, VI, VII. Table V shows the UDP Flood Performance test results

TABLE V
PERFORMANCE OF UDP FLOOD DEPENDING ON NUMBER OF CPUS

CPU	Time [s]	Upload speed [pps]	Network throughput [Mbps]
1	155	6.452	3.10
2	23	43.478	20.87
4	19	52.631	25.26

TABLE VI
POWER OF DNS SERVER FLOOD DEPENDING ON NUMBER OF CPUS

CPU	Time [s]	Upload speed [pps]	Network throughput [Mbps]
1	156	6.410	4.26
2	22	45.454	30.18
4	21	47.619	31.61

depending on the CPU count. Table VI shows the DNS Server Flood Performance Test results depending on the number of CPUs. Table VII shows the performance test results of the DNS Amplification Flood DoS depending on the CPU count.

Tables V, VI and VII includes the number of CPUs assigned to the first virtual machine, the duration of the test from start to end in seconds, the average packet sending rate in packets per second, and network traffic in megabits per second. The first virtual machine on which the DoS module was started with 1, 2, and 4 CPU sequentially. Always 1.000.000 packets have been sent at the highest upload speed. In performance testing of UDP Flood, DNS Server Flood, and DNS Amplification Flood DoS it was found that the greatest gain in performance was when moving from 1 CPU to 2 CPUs. When switching from 2 CPU to 4 CPU, performance gains have not been so robust.

IV. CONCLUSIONS

This article discusses the design and implementation of a DoS network attack generator such as HTTP, SYN, ICMP, NTP, UDP, DNS Server, DNS Amplification, and Slowloris, which also enables the creation of a customized DoS attack configuration. Developed DoS Network Generator is an appropriate tool for ethically testing Web, NTP, and DNS server resilience against DoS attacks. The article describes the principles of selected DoS attacks, an analysis of tools for web server performance testing has been performed and on the basis of which a DoS network generator with a graphical user interface has been designed and created. The developed generator has performed experimental performance testing of individual DoS attacks, depending on available system resources. Testing has confirmed their functionality and suitability for testing the availability of Web, NTP and DNS servers or network infrastructure in both corporate and school environments.

TABLE VII
PERFORMANCE OF DNS AMPLIFICATION FLOOD DEPENDING ON NUMBER OF CPUS

CPU	Time [s]	Upload speed [pps]	Network throughput [Mbps]
1	167	5.988	3.98
2	21	47.619	31.62
4	19	52.631	34.95

REFERENCES

[1] P. Dzurenda, Z. Martinasek, and L. Malina, -Network Protection Against DDoS Attacks-, *International Journal of Advances in Telecommunications, Electrotechnics, Signals and Systems*, vol. 4, no. 1, p. -, Mar. 2015.

[2] K. N. Mallikarjunan, K. Muthupriya, and S. M. Shalinie, -A survey of distributed denial of service attack-, in *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, 2016, pp. 1-6.

[3] N. Hoque, D. K. Bhattacharyya, and J. K. Kalita, -Botnet in DDoS Attacks: Trends and Challenges-, *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2242-2270, 2015.

[4] B. Wang, Y. Zheng, W. Lou, and Y. T. Hou, -DDoS Attack Protection in the Era of Cloud Computing and Software-Defined Networking-, in *2014 IEEE 22nd International Conference on Network Protocols*, 2014, pp. 624-629.

[5] Haining Wang, Danlu Zhang, and Kang G. Shin, -Detecting SYN flooding attacks-, in *Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, 2002, pp. 1530-1539.

[6] C. Douligeris and A. Mitrokotsa, -DDoS attacks and defense mechanisms: classification and state-of-the-art-, *Computer Networks*, vol. 44, no. 5, pp. 643-666, 2004.

[7] J. Czyz, M. Kallitsis, M. Gharaibeh, C. Papadopoulos, M. Bailey, and M. Karir, -Taming the 800 Pound Gorilla-, in *Proceedings of the 2014 Conference on Internet Measurement Conference - IMC '14*, 2014, pp. 435-448.

[8] S. Aarti, D. Junea. Agent Based Preventive Measure for UDP Flood Attack in DDoS Attacks. *International Journal of Engineering Science and Technology*, 2010, pp. 3405-3411

[9] H. Ballani and P. Francis, -Mitigating DNS DoS attacks-, in *Proceedings of the 15th ACM conference on Computer and communications security - CCS '08*, 2008, pp. 189-198.

[10] G. Kambourakis, T. Moschos, D. Geneiatakis, and S. Gritzalis, -Detecting DNS Amplification Attacks-, in *Critical Information Infrastructures Security*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 185-196.

[11] E. Damon, J. Dale, E. Laron, J. Mache, N. Land, and R. Weiss, -Hands-on denial of service lab exercises using SlowLoris and RUDY-, in *Proceedings of the 2012 Information Security Curriculum Development Conference on - InfoSecCD '12*, 2012, pp. 21-29.

[12] -Ab - Apache HTTP server benchmarking tool-, *Apache HTTP server project*, -. [Online]. Available: ab - Apache HTTP server benchmarking tool. [Accessed: 17-Aug.-2018].

[13] D. Mosberger and T. Jin, -Httperf—a tool for measuring web server performance-, *ACM SIGMETRICS Performance Evaluation Review*, vol. 26, no. 3, pp. 31-37, Dec. 1998.

[14] -Siege Home-, *Joe Dog Software*, -. [Online]. Available: https://www.joedog.org/siege-home/. [Accessed: 17-Aug.-2018].

[15] -Apache JMeter-, *The Apache Software Foundation*, -. [Online]. Available: https://jmeter.apache.org/. [Accessed: 17-Aug.-2018].

[16] -Trafgen (8) - Linux Man Pages-, *Linux Man Pages*, -. [Online]. Available: https://www.systutorials.com/docs/linux/man/8-trafgen/. [Accessed: 17-Aug.-2018].