ZESZYTY NAUKOWE POLITECHNIKI RZESZOWSKIEJ 294, Elektrotechnika 36
RUTJEE, z. 36 (1/2017), styczeń-czerwiec 2017, s. 67-76

Mariusz NYCZ[1]
Tomasz SZELIGA[2]
Piotr HAJDER[3]

ASSESSMENT OF THE VULNERABILITY OF THE APACHE SERVER TO DDOS ATTACKS

The article presents an analysis of the vulnerability of the Apache server with regard to common DDoS attacks. The paper begins with presenting the statistical overview of the issue of denial-of-service attacks. We also discuss the methods used for performing DDoS attacks. Working with the virtual systems, the authors designed a test environment, where the assessment was conducted of the vulnerability of selected WWW systems. At the end of the article, actions are proposed to implement effective methods of defending against the denial-of-service attacks. The paper is written for the specialists in the field of web systems security.

Keywords: DDoS Attack; security; the Apache; web server

1. Introduction

With every passing year, ensuring the reliability of the operation of WWW servers is becoming a more and more important issue. The high requirements set for the web services, such as e-banking, e-transaction systems or electronic trading result in setting the accessibility at 99.9%. Along with the increase of the importance of the web systems, we can now observe a dynamic development of new threats and techniques aimed at lowering or even paralyzing the accessibility to the system. Properly performed dispersed denial of service attacks (DDoS) pose a serious threat to all the services on the Internet. The main idea of the denial-of-service attacks is to use up all the WWW server resources, which results in the implemented WWW services being inaccessible, which in turn leads to serious financial losses. We can expect the number of the attacks to increase in the next couple of years, and the methods of attacking to become more and more advanced. Currently used mechanisms of detection and protection against this kind of threats are not fully sufficient.

[1] Autor do korespondencji: Mariusz Nycz, Politechnika Rzeszowska, Zakład Systemów Złożonych, mnycz@prz.edu.pl
[2] Tomasz Szeliga, Politechnika Rzeszowska
[3] Piotr Hajder, Akademia Górniczo-Hutnicza w Krakowie

2. Denial-of-service attacks – a statistical overview

The presented results were formulated on the basis of analysing 459 web-sites belonging to multiple owners from the Subcarpatian voivodeship. Fig. 1 presents the market share of each web server software program. Achieved results were compared with data provided by Netcraft (world) and amudom (Poland). Results show overrepresentation of the Apache server on the Polish market.

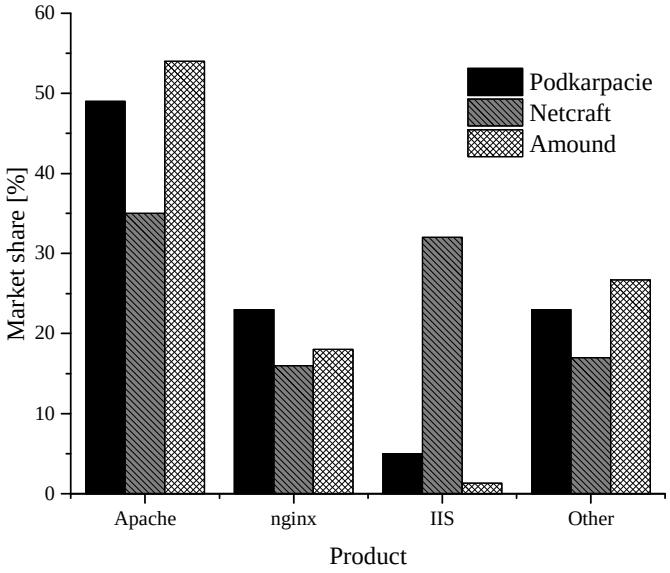

Fig. 1. Web-server software market in 2016 [1, 2]

According to DDoS Attack Report by Prolexic, sales and financial services were the most popular targets for the attacks in the area of web applications [3]. Akamai organization published a report about denial of service attacks. Fig. 2 presents the statistical data on the percentage of different types of attacks. This kind of attacks is performed usually for financial gain. According to the report [3], one of the purpose of this kind of attacks is to buy valuable goods at lower prices.

The attacks began to be aimed also at the financial services provided by the e-banking companies. Denial-of-service attacks are usually performed to steal the records from the existing databases of the banking institutions, which results in making their market situation worse. According to the report [3], DDoS attacks aimed at the sales industry make up 40% of all the application-layer attacks. By comparison, most of the attacks at the web layer are aimed at the game industry, where the protection against the application-layer attacks is very often weaker.

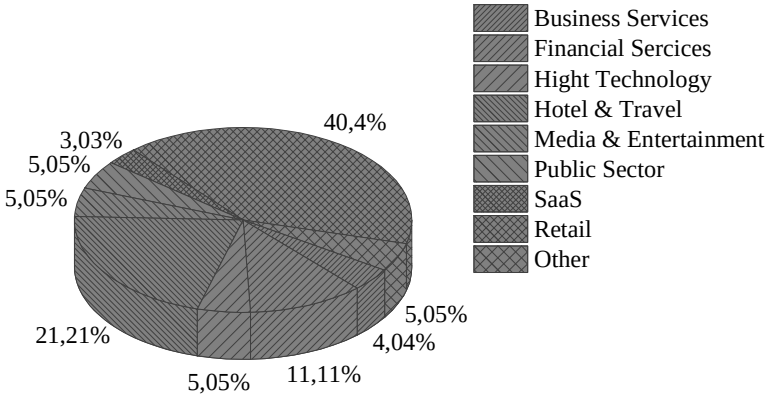

Fig. 2. Percentage of the application-layer attacks in the industry [3]

The number of the attacks increases along with the development in the field of technology. The report [3] includes statistical data regarding the direction of DDoS attacks in the recent years. Fig. 3 presents the data regarding 4rd quarter of 2016.

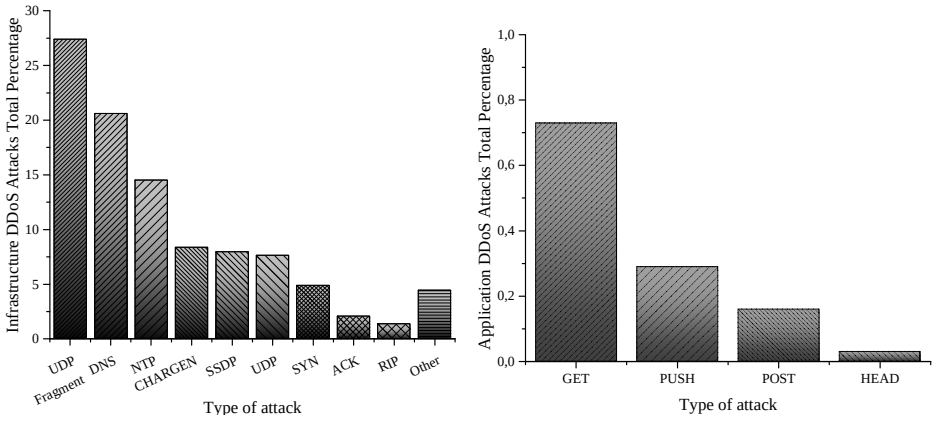

Fig. 3. Directions of denial-of-service attacks [3]

3. The types of DDoS attacks

The analysis performed allowed for the assessment of the effectiveness of different methods of the attack. Slowloris and R.U.Dead.Yet scripts were tested, as well as Syn Flood attack [2, 5]. Below you can find the description of the types of the attacks used for the tests:

- Slowloris – attack with HTTP Get queries. The main task of the script it to maintain the connection by completing the fragments of the header.

The maximum time for sending a package which maintains the connection is 300 sec. Slowloris does not destroy the hardware or the data of the server. It is impossible to detect the attacker by the Apache server logs only [10]. The attack was performed with the following request:

perl slowloris.pl –dns 192.168.0.109 –port 8080 –time 50 –num 1000 (1)

Where: dns is the destination address for the attack, port is the HTTP port number, time is the delay between the requests sent and num is the number of threads used during a single attempt at connecting [16].

- R.U.Dead.Yet – a free script using HTTP Post queries. The attack is performed by sending incomplete but justified headers. Every time, the fragments of the header body are sent, 1 byte each, at a very low speed. The task of every server is to compare the number of the bytes received with the value of the Content-Length field. Adding new information to the header body continuously results in the server being unable to complete the connection, which makes it use more and more of its resources to perform the operation of checking the number of the bytes of the received information. The post header is not being buffered [12]. The authors used Slowhttptest application to generate HTTP Post traffic (2).

Slowhttptest –c 600 –B –i 1 –t POST –g –o Slowhttptest -r 300 –l 400 –u http://192.168.0.101/ -x 20 (2)

Where: c is the total number of connections, -B is the type of attack, -i is the time between the connections, -t is the type of the response, -g -o indicate the parameters used to generate the charts, -r is the number of connections per second, -l is the target length of the text in a second, -u is the address of the victim and –x determines the maximum length of the bits.

- Syn Flood – takes advantage of TCP/IP protocol weaknesses. During the attack, a number of half-open connections with the server is maintained. Each of the connections can be maintained from 3 up to 4 minutes, depending on the used configuration. At the beginning, the client assigns a port for TCP/IP and sends SYN message to WWW server, requesting a connection. The resources are “booked” by the system, which responds by sending a SYN-ACK package to the encrypted IP of the client. During the test, it was necessary to generate a traffic like one during a Syn Flood attack. The authors used Scapy packet generator for that purpose (3).

```
IP= IP(dst="192.168.0.101")
IP= IP(src="192.168.0.111")
t=TCP()                                    (3)
t.sport=(RandNum(1024, 6565)
send((IP/t), loop=1)
```

Where: dst – target address for the attack, src – encrypted source address for the attack, t – TCP traffic variable and sport – setting pseudorandom port numbers. The last request makes the TCP requests being sent continuously [14].

4. WWW server defense mechanisms

Proper choice of the attack mechanisms made it possible to discover the weaknesses of the Apache server. Each of the implemented defence mechanisms is a free tool for protecting against the violations of safety protocols [4, 7-9, 11].

- mod_security module – a complex firewall for the web applications. Responsible for the monitoring and analysis of the HTTP traffic. Allows using the existing firewall rules and creating the new ones. The model also enables to block the access to the rest of the Apache server configuration files. The rules used while performing the analysis of the vulnerability of the WWW server filtered the traffic by: checking the number of the requests sent from one IP address, limiting the number of loggings-on to one per minute, checking the number of active connections for one IP address and checking the request paths. Every time any of the defined values is exceeded, the task of the module is to add the address of the potential attacker to the blacklist. After 5 minutes, the blocked address will be removed from the turn-on list.
- mod_qos module – introduces a mechanism reducing the speed of sending the fragments of the headers of the requests. Activating mod_qos module makes performing a Slow HTTP attack effectively impossible, as well as any other attacks using large bandwidth. Mod_qos module architecture makes it possible to add a new functionality in the form of a separate module, gathering information regarding the operation of the Apache server in real-time. After the installation of mod_status module, the access to the data can be gained through a browser, by providing the address of the server. Moreover, it is possible to generate a report including basic information on the number of the connected addresses or the maximum number of the addresses from where the connection request may be sent. Many other parameters are also displayed. Combining mod_qos and mod_status modules allows controlling the safety of the Apache

server more effectively. A detailed description of the parameters is given in Tab. 1.

Table 1. Mod_qos module setup parameters

Parameter	Description
QS_ClientEntries	Maximum number of clients
QS_ClientEvent RequestLimit	Maximum number of simultaneous requests for a single IP address
QS_SrvMaxConn PerIP	Maximum number of connections for a server address
MaxClients	Maximum number of active TCP connections
QS_SrvMaxConn Close	Maximum number of keep-alive connections. If the limit is exceeded, the connection is closed for every request.
QS_SrvMinData Rate	Minimum bandwidth: the number of bytes sent per second / the number of bytes received per second
QS_LimitRequestBody	Maximum size of the body of the request
LimitRequestFields	Maximum size of the header of the request

- mod_evasive module – module responsible for monitoring of the number of incoming HTTP connections. The architecture of this solution allows communication with ipchains, routers, and complex firewalls. Every client attempting to connect is checked for the number of the requests incoming for one website, as well as for the number of simultaneous requests for a single process or thread of the server. The module also blocks the attempts to establish a connection for the IP addresses included in the blacklist.

5. Assessment of the vulnerability of the Apache server

The main task of the server is to process the requests of the HTTP communication protocol. Web servers are used for websites, e-mail accounts, databases and many other web services [9], [11], [15]. A free tool, the Apache server is used by many companies. The server may operate in one of the two modes: MPM Prefork and MPM Worker. The main difference between the two is the manner of processing the request. The server working in MPM mode uses child processes to accomplish the task. Processing every incoming request is performed with a new child process. This kind of architecture ensures the safety of the rest of the requests processed. The main drawback of this method of request processing is that the resources of the server are limited. The second mode in which the Apache server can operate is MPM Worker. The server creates a sepa-

rate child thread for every request. This solution makes it possible to process a higher number of clients with smaller hardware resources [6], [13]. A properly designed DDoS attack is capable of making the server deny a service irrespective of the mode in which the Apache server has been operating.

Failure to ensure a proper level of securing the Apache server may result in the attacker's being able to use the server for a denial-of-service very soon after the attack is commenced. Three kinds of attack were used for the analysis. Each of them was to show a different security gap in the Apache server. The vulnerability tests of the Apache server proved that an unsecured server cannot filter out a dangerous action. The default WWW server configuration does not allow using any mechanisms of detecting DDoS attacks [6], [15]. Therefore, installing some protective tools is necessary to ensure safety in case of any attempts to violate the safety protocols. The vulnerability tests of the Apache server indicated that the server in its default configuration is vulnerable to all the used mechanisms of attack. During the test conducted with R.U.Dead. type traffic-generating tool, a denial of service occurred. A website activated on the server did not respond to a request by the user. In case of the rest of the other types attacks aimed at an unsecured server, the websites loaded longer than usual. Moreover, the server did not always respond to the requests sent by the user. Fig. 4 presents the results of conducted analysis.

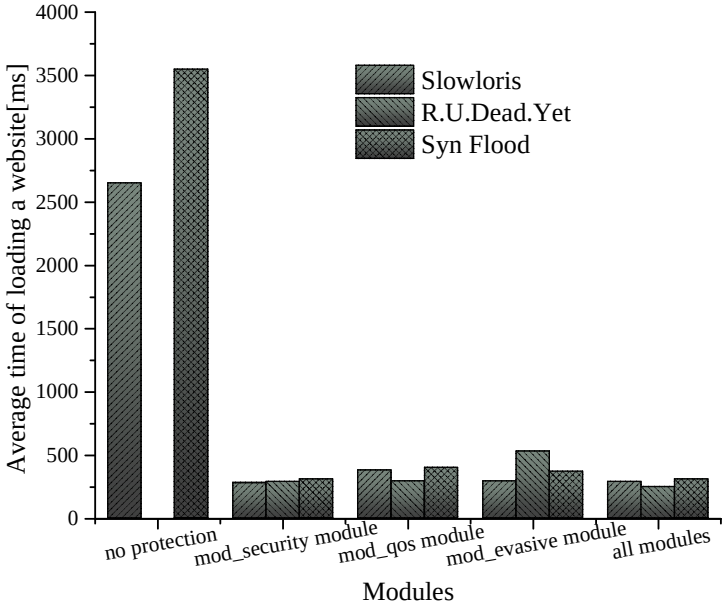

Fig. 4. Analysis of the time of loading pages during DDoS attack for different defence mechanisms

It should be remembered that the testing and analysis of Apache was carried out in a virtual test environment. To carry out the previously mentioned analysis has been used 2 virtual machines, on each of which had been installed Kali Linux. For a virtual machine performing the role of the web server has been set the RAM to 2GB of value, created a virtual SCSI drive size 40GB. In addition, the network card is set to bridge mode. The server at all time has at his disposal two core processor. Apache server the whole duration of the test was operated MPM Prefork which allowed for the safe close requests classified as a probable attack. Use MPM Prefork mode that caused the server can only handle 375 active connections at the same time. Apache also had set up parameters with values: Timeout 200 s, KeepAive On, MaxKeepAliveRequests 400, Max Clients 150.

Using proper mechanisms of protecting against DDoS attacks made WWW server resistant to some extent to the denial-of-service attacks. Unfortunately, the mechanisms implemented did not protect the server in 100%. Additional configurations have to be introduced in the configuration files of the Apache server in order to increase the protection of WWW server.

Summary

Performed vulnerability tests of the Apache server indicated that an unsecured WWW server is not equipped with any mechanisms limiting the effects a DDoS attack. During the stress tests, the server was incapable of executing any of the actions generated, being exceptionally vulnerable to HTTP Post, Get or Syn Flood attacks. Implementing only three protective mechanisms ensured proper operation of the server and the access to the services.

The analysis was not performed to indicate which of the chosen protective mechanisms is the best, but to show the difficulties regarding the protection against denial-of-service type of attacks.

References

[1] Web Server Survey - Web server developers: Market share of active sites. Available: https://www.netcraft.com/internet-data-mining/ [Access: 10.03.2017]

[2] W. Stallings: „Kryptografia i bezpieczeństwo sieci komputerowych. Koncepcje i metody bezpiecznej komunikacji", Helion, Gliwice 2012.

[3] Akamai's [state of the internet] / security – Q4 2016 report. Available: https://www.stateoftheinternet.com/downloads/pdfs/2015-cloud-security-report-q3.pdf [Access: 15.03.2017]

[4] S.T. Zargar, J. Joshi, D. Tipper: "A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks", IEEE communications surveys & tutorials, vol. 15, no. 4, fourth quarter 2013.

[5] Ch. Douligeris, A. Mitrokotsa: "DDoS attacks and defense mechanisms a classification", Department of Informatics University of Piraeus, Piraeus, Greece.

[6] M. Poongothai, M. Sathyakala: "Simulation and Analysis of DDoS Attacks", International Conference on Emerging Trends in Science, Engineering and Technology, 2012.

[7] Security Labs: How to Protect Against Slow HTTP Attacks [Online]. Available: https://blog.qualys.com/securitylabs/2011/11/02/how-to-protect-against-slow-http-attacks [Access: 30.03.2017]

[8] cunetix: How To Mitigate Slow HTTP DoS Attacks in the Apache HTTP Server [Online]. Available: https://www.acunetix.com/blog/[Access: 30.03.2017]

[9] Apache Security: Denial of Service Attacks [Online]. Available: https://www.feistyduck.com/library/apache-security/online/apachesc-CHP-5.html [Access: 01.04.2017]

[10] Ataki Slow HTTP DoS (cz. 1.) – Slowloris, [Online]. Available: http://sekurak.pl/ataki-slow-http-dos-cz-1-slowloris/ [Access: 01.04.2017]

[11] Securing the Apache, Part 8: DoS & DDoS Attacks, [Online]. Available: http://opensourceforu.efytimes.com/2011/04/securing-apache-part-8-dos-ddos-attacks/ [Access: 10.04.2017]

[12] R.U.D.Y. (R-U-Dead-Yet): DDoS Attack Glossary [Online]. Available: https://www.incapsula.com/ddos/attack-glossary/rudy-r-u-dead-yet.html [Access: 10.04.2017]

[13] Understanding the Apache 2 MPM (worker vs prefork) [Online]. Available: https://www.garron.me/en/blog/apache2-mpm-worker-prefork-php.html [Access: 06.04.2017]

[14] K. Geetha: SYN flooding attack — "Identification and analysis", Information Communication and Embedded Systems (ICICES), 2014 International Conference on, 2014.

[15] N. Shipilov, K. Borisenko, A. Shorov: "Simulation of DDoS-attacks and protection mechanisms against them", Young Researchers in Electrical and Electronic Engineering Conference 2015 IEEE NW Russia, 2015.

[16] J. Brynielsson: "Detectability of low-rate HTTP server DoS attacks using spectral analysis", International Conference on Advances in Social Networks Analysis and Mining, 2015.

BADANIE PODATNOŚCI SERWERA APACHE NA ATAKI ODMOWY USŁUGI

Streszczenie

W artykule przedstawiono analizę podatności serwera Apache w odniesieniu do popularnych ataków DDoS. Praca rozpoczyna się od przedstawienia statystycznego ujęcia problemu, jakim są ataki odmowy usług. Ponadto przedstawiony został problem rozpowszechniania metod wykorzystywanych do przeprowadzania ataków DDoS. Autorzy, bazując na systemach wirtualnych opracowali środowisko testowe, na którym zrealizowano badania podatności wybranych systemów WWW. Publikację kończą propozycje działań mających na celu zaimplementowanie efektywnych

metod obrony przed atakami odmowy usługi. Artykuł jest adresowany do osób zajmujących się bezpieczeństwem systemów webowych.

Słowa kluczowe: bezpieczeństwo, Apache, ataki DDoS

DOI: 10.7862/re.2017.6

Tekst złożono w redakcji: maj 2017
Przyjęto do druku: czerwiec 2017