

SLOW DOS ATTACKS DETECTION AND MITIGATION

Marek Sikora

Doctoral Degree Programme (2.), FEEC BUT

E-mail: marek.sikora@vutbr.cz

Supervised by: Václav Zeman

E-mail: zeman@feec.vutbr.cz

Abstract: This article investigates the detection and mitigation methods against Slow DoS (Demand of Service) attacks. This research is focused on Slowloris, Slow POST, Slow Read, and Apache Range Header attacks. Detection methods are based on network traffic analysis and anomalous traffic monitoring. When the attack is detected, the attacker is blocked and web server resources are released. Methods are implemented as an intrusion prevention system software.

Keywords: Slow DoS, network traffic analysis, network monitoring, Slowloris, Slow POST, Slow Read, Apache Range Header, detection, mitigation

1 INTRODUCTION

The DoS attack tries to make network services unavailable to users, or reduce the quality and speed of service. The attacks are most often targeted to web servers. The complexity of DoS attacks increases. The DoS attacks have evolved from primitive flooding attacks into sophisticated application layer attacks [1]. Due to the valid use of lower-layer protocols, these attacks can avoid detection. Service vulnerability depends on the security level of a used web server (Apache, lighttpd, Nginx, IIS, etc.) [2].

There are several mechanisms to protect web-servers from DoS attacks: using backup servers, honeypots, and load balancers; modifying web servers configuration; using specialized security modules and updates for web servers; using firewalls or intrusion prevention system (IPS) [3]. However, these mechanisms are not effective in all situations. Thus, it is necessary to develop an advanced approach based on traffic analysis and detecting anomalies [4].

2 SLOW DOS ATTACKS

Slow DoS attacks can be done from one attacker station by sending a small number of requests at a very slow rate [1]. This type of attack is difficult to detect because it is similar to the communication of a legitimate user with a slow internet connection. Slow DoS attack fills the web server input queue by establishing a large number of Transmission Control Protocol (TCP) connections. An attacker has to communicate sufficiently to prevent the connection from being closed. If all the web server resources are occupied, the web server will not be able to communicate with other users [5].

2.1 SLOWLORIS

Slowloris attack, also known as Slow GET, uses Hypertext Transfer Protocol (HTTP) GET request [1]. The attacker sends an invalidly terminated request to the web server. This request does not include termination character `\r\n\r\n`, as follows:

```
GET / HTTP/1.1 \r\n
```

```
Host: 10.0.0.20 \r\n
Content-Length: 32 \r\n
```

The web server expects to receive the next part of the request, which will be terminated validly. The attacker continues sending a small number of random characters to keep the TCP connection open as long as possible. In this way, the attacker will establish as many connections as possible, causing exhausting all available web server resources [1] [6].

2.2 SLOW POST

This attack, also known as RUDY, an acronym of R-U-Dead-Yet, is similar to the Slowloris attack. The attacker sends validly terminated HTTP POST request with `content-length` value in the header specifying the size of the transmitted data. This value is set to an enormous number while the actual size of the data is minimal [6]. An example of this request is as follows:

```
POST /index.php HTTP/1.1 \r\n
Host: 10.0.0.20 \r\n
Content-Length: 10000000 \r\n
\r\n
name=a
```

The web server is forced to wait for receiving the rest of the data specified in the `content-length` value. Due to hold the connection open, the attacker sends a small number of random characters which may represent the next part of the data. In this way, the attacker opens more connections causing exhausting all available web server resources [1].

2.3 SLOW READ

This attack uses the TCP window size to limit a data bitrate. The TCP window size specifies the amount of data the sender can send without waiting for confirmation from the recipient. During this attack complete HTTP request is sent to the web server. The web server replies with an HTTP response, but it is forced to split the response into many parts and send them slowly because the attacker has set the small TCP window size in the initial TCP SYN packet sent to the web server. In this way, the attacker opens a high number of connections which may cause a denial of service [1] [7].

2.4 APACHE RANGE HEADER

The attack uses a byte range parameter in the HTTP request header to specify a large amount of requested data ranges from the server. The server allocates space in memory for each of the ranges separately which will cause server memory exhaustion. An attacker can use the `Accept-Encoding: gzip` parameter to force sending a response in a compressed format, which will increase processor usage. A denial of service is possible by an HTTP design error that causes the web server has to respond to each requested range separately [6] [8]. An example of the Apache Range Header HTTP request header is as follows:

```
HEAD / HTTP/1.1 \r\n
Host: 10.0.0.20 \r\n
Range: bytes=0-,5-0,5-1,5-2,[...],5-1298,5-1299 \r\n
Accept-Encoding: gzip \r\n
\r\n
```

3 DETECTION AND MITIGATION METHODS

The investigated Slow DoS detection methods were based on network traffic analysis. An incoming packet was parsed into individual layer protocol headers, then it was analyzed and compared with known attack features described above. If match found, remote host's Internet Protocol (IP) address and TCP connection details were saved and the host's traffic was monitored for further analysis.

Due to the easy confusion of attackers and users, limit parameters had to be set to identify attacks with certainty. These parameters were set for each attack separately due to a different type of traffic. Monitored parameters were: the number of open TCP connections from one host; maximum receiving time of one HTTP request; minimum data bit rate; maximum number of HTTP request parts. In this research, limit parameters were based on legitimate and attack traffic observation.

3.1 SLOWLORIS DETECTION METHOD

If an incoming HTTP GET request without valid termination was detected, the Slowloris attack could be involved. The following parameters of specific host communication were monitored:

- the number of unfinished parts of the HTTP GET request received on the TCP connection,
- elapsed time since the first part of the unfinished HTTP GET request in the connection was received,
- the number of open TCP connections from one IP address on which an unfinished HTTP GET requests were received.

If any of the parameters were exceeded, the attack was detected and the source IP address was blocked.

3.2 SLOW POST DETECTION METHOD

For incoming HTTP POST requests, the `content-length` value was compared with the actual received data size. If the values did not match, the beginning of the Slow POST attack was detected and the remote user was tracked. The same traffic parameters as with the Slowloris attack were monitored. The bit rate decrease was also monitored. An attack was detected when limit values were exceeded.

3.3 SLOW READ DETECTION METHOD

The TCP SYN segment was captured in each TCP handshake between the web server and the host. The `Window Scale Factor` value in the TCP SYN segment header was used to calculate the TCP window size. If the window size was suspiciously small, the connection was monitored. The following parameters of the host's communication were monitored:

- duration of an open TCP connection,
- a decrease in data transfer rate,
- the number of open connections from the specific IP address from which a TCP segment with a very small window size was received.

If any of the parameters were exceeded, the attack was detected and the source IP address was blocked.

3.4 APACHE RANGE HEADER DETECTION METHOD

Different types of HTTP requests could be used for this attack. The occurrence of the `byte range` parameter was monitored. If multiple requests with excessive data ranges in the `byte range` parameter were received, an attack was detected and the source IP address was blocked.

3.5 ATTACK MITIGATION METHOD

In the beginning, the implemented network filter applied a firewall rule to limit the number of concurrently established TCP connections from one host. This mechanism protected the web server from the overload caused by the initial attacker's requests. Subsequently, a traffic attack analysis was performed. If the attack was detected by the methods described in the previous chapter, further communication was interrupted and the web server resources were released.

In the first step, a new firewall rule was defined to block all traffic incoming from the attacker IP address. In the second step, the established TCP connections were released by sending TCP RST segments from the network filter and then system resources were available to other legitimate users.

4 TESTING

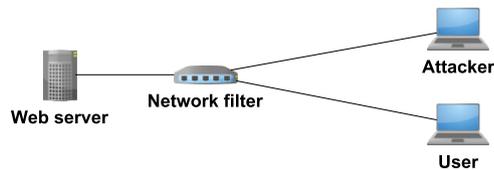


Figure 1: Test network.

Laboratory network topology is shown in figure 1. SlowHTTPTest tool was used for Slow DoS attacks generation. Apache 2.4.18 was used as a target web server. All attacks were tested in two scenarios. Firstly, without using any additional protection. Secondly, protected using the proposed network filter. The results were similar for Slowloris, Slow POST and Slow Read attacks. In the case of the Apache Range Header attack, the Apache web server was already immune. As an example, the results of testing the Slowloris attack are shown in figures 2 and 3. Web server availability is expressed by a green line which takes only two values: maximum – the server is available; minimum – the server is unavailable. TCP connections are expressed by the orange, red and blue lines.

The web server could handle around 280 TCP connections. When an attack was launched, all free web server resources were occupied. From this moment, a legitimate user could not establish com-

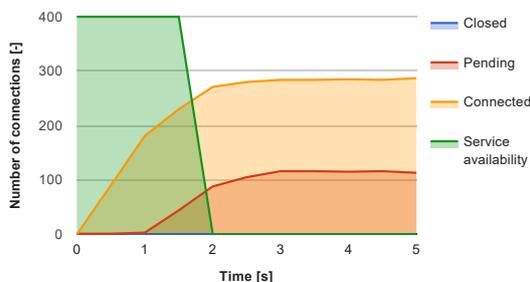


Figure 2: Slowloris without the network filter.

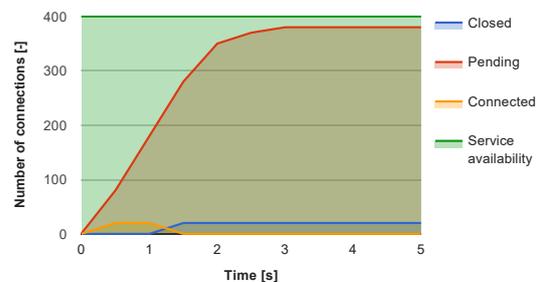


Figure 3: Slowloris through the network filter.

munication with the web server. If the designed network filter were used, the attacker was able to establish only 20 TCP connections. The presence of suspicious HTTP requests was detected by traffic analysis and the attack was revealed. Web server resources were released by sending TCP RST segments to the web server and other incoming traffic from the attacker was blocked by a new firewall rule. The web server was available to legitimate users.

5 CONCLUSION

Testing has demonstrated the vulnerability of the Apache web server to Slowloris, Slow POST, and Slow Read attacks. In contrast, the Apache web server is already resilient to the Apache Range Header attack by default. The proposed methods of traffic analysis and attack mitigation are effective. It has been proven that using the proposed methods prevent causing a denial of service. However, a real network test run demonstrates the need for further research on the Slow POST detection method due to reducing false positives. More precise detection parameters will bring more accurate results.

6 ACKNOWLEDGMENT

The research was supported by project FEKT-S-17-4184 “Information and communication systems security research”. Research described in this paper was financed by the Ministry of Interior under grant VI20172019093.

REFERENCES

- [1] Cambiaso, E., Papaleo, G., Aiello, M.: Taxonomy of Slow DoS Attacks to Web Applications, Recent Trends in Computer Networks and Distributed Systems Security, Berlin, 2012, 195-204, ISBN: 978-3-642-34134-2
- [2] Apparatus and method for detecting slow read DoS attack, Computer Weekly News [online], Atlanta, United States, NewsRx, 30. 10. 2014, 5, URL: <<https://goo.gl/oHkS5R>>
- [3] Method and Protection System for Mitigating Slow HTTP Attacks Using Rate and Time Monitoring, Computer Weekly News [online], Atlanta, United States, NewsRx, 21. 3. 2013, 6, URL: <<https://goo.gl/sgRcCy>>
- [4] Duravkin, I., Loktionova, A., Carlsson, A.: Method of slow-attack detection, 2014 First International Scientific-Practical Conference Problems of Infocommunications Science and Technology [online], IEEE, 2014, 171-172, ISBN 978-1-4799-7342-2
- [5] Hirakawa, T.; Ogura, K.; Bista, B. B. a Takata, T.: A Defense Method against Distributed Slow HTTP DoS Attack, 2016 19th International Conference on Network-Based Information Systems (NBiS) [online]. IEEE, 2016, 152-158, ISBN 978-1-5090-0979-4
- [6] Cambiaso, E., Papaleo, G., Chiola, G., Aiello, M.: Slow DoS attacks: definition and categorisation, International Journal of Trust Management in Computing and Communications, vol. 1, no. 3/4, 2013, ISBN 2048-8378
- [7] Shekyan, S.: Are you ready for slow reading?, Qualys Blog: Security Labs [online], 2012, URL: <<https://goo.gl/Ju3UWn>>
- [8] Baldwin, M.: Mitigating the Apache Range Header DoS Vulnerability, Infosec Island [online], 2011, URL: <<https://goo.gl/U17PGI>>
- [9] Sikora, M., Blažek, P.: Systém prevence průniku Slow HTTP DoS a DDoS útok, Elektrovrevue - Internet journal, 2017, vol. 19, no. 4, p. 1-8, ISSN: 1213-1539