UDC 004.056.57

ANALYSIS OF REALIZATION AND METHOD OF DETECTING LOW-INTENSITY HTTP-ATTACKS. PART 1. FEATURES OF REALIZATION OF LOW-INTENSITY HTTP ATTACKS

A. CARLSSON
Blekinge Institute of Technology
E. DURAVKIN, A. LOKTIONOVA
Kharkiv National University of Radioelectronics

Abstract *- The analysis of realization features of low-intensity HTTP-attacks was performed. Three types of low intensity attacks were highlighted: Slowloris, Slow POST attack and Slow READ attack. Scenarios of each type of low-intensity attacks were described. Features of this type of attacks in comparison with low-level attacks such as "denial of service" were selected: they do not require a large Number of resources from the attacking machine, and they are difficult for the detection, since their parameters are similar to legitimate traffic. All three types of attacks have been implemented using the tool "slowhttptest". The web server Apache parameter by which can be realized these vulnerabilities were obtained. Different configurations of the server Apache, which exposed to attacks of this type, have been investigated. The basic parameters of the attacks, in which server Apache transforms into a state, where it cannot service requests have been allocated. For each type of attacks the characteristic features were highlighted. Parameters of http-request, which assume the detection of this type attacks highlighted. The analysis of mathematical tools of building the models for the systems for these types of attacks detection on the basis of the obtained parameters was performed.*

Анотація *– Виконано аналіз особливостей HTTP - атак низької інтенсивності. Виділено три типи атак: Slowloris, Slow POST і Slow READ і описано сценарії реалізації атак кожного типу. Особливостями HTTP - атак низької інтенсивності в порівнянні з DOS- атаками, що реалізовані на мережному і транспортному рівнях, є те, що вони не вимагають потужних ресурсів від зловмисника, а діагностика є важкою. Це є наслідком того, що реалізація атаки виконується засобами легального неаномального трафіку. Виділено конфігурації веб-сервера Apache, що дозволяють реалізувати атаки даного типу, а так само параметри HTTP - запитів, за допомогою яких передбачається виконувати виявлення даних атак. Проведено аналіз математичних апаратів, на основі яких може бути побудована система виявлення низько інтенсивних атак.*

Аннотация *– Выполнен анализ особенностей HTTP-атак низкой интенсивности. Выделены три типа атак: Slowloris, Slow POST и Slow READ и описаны сценарии реализации атак каждого типа. Отмечено, что особенностями HTTP-атак низкой интенсивности по сравнению с DOS-атаками, реализуемыми на сетевом и транспортном уровнях, является то, что они не требуют мощных ресурсов от злоумышленника, а диагностика является затруднительной. Это является следствием того, что реализация атаки выполняется посредством легального неаномального трафика. Выделены конфигурации веб-сервера Apache, позволяющие реализовать атаки данного типа, а также параметры HTTP-запросов, посредством которых предполагается выполнять обнаружение данных атак. Проведен анализ математических аппаратов, на основе которых может быть построена система обнаружения низкоинтенсивных атак.*

Introduction

The development of the Internet is accompanied by an explosive growth of technological resources, such as branching and capacity of channels or power increase of equipment. This also inevitably leads to a power increase of network layer attacks. Attackers are able to clog all channels of a target`s ISP (Internet Service Provider).

Such an opportunity to attack an ISP is provided by DoS (Denial of Service) -attacks. The purpose of DoS-attacks is to create conditions of a website functioning in which the user

cannot access it. In addition, DoS-attacks are directed to exhaust system resources. Attackers usually achieve this by submitting a huge amount of website requests so that users can no longer get through to their website.

I. Trends of application layer attacks development

The efforts of providers worldwide were originally focused on cleaning the channels from "trash traffic". For example, many ISPs used specialized solutions such as: installing and configuring the firewall, routing into "black holes", using intrusion detection systems (IDS), access control lists, etc. These solutions coped with trash traffic in the channel, but noticed almost no application layer anomalies. In connection with this, DoS-attacks of application (seventh) layer and attacks implemented by using tools such as LOIC and HOIC eventually began to gain popularity. In contrast to attacks of the third and fourth layers, attacks of application layer do not require a large attacking botnet and reliably "drop" attacked resource, while remaining virtually invisible to specialized equipment installed by provider. They belong to the type of attacks called "lack of resources", because they are directed to fill all possible server sessions with client. According to data received by specialists of "Kaspersky Lab" [1] on the results of security service Kaspersky DDoS Prevention, from 2008 to 2011 the number of application layer attacks has increased significantly (fig. 1).

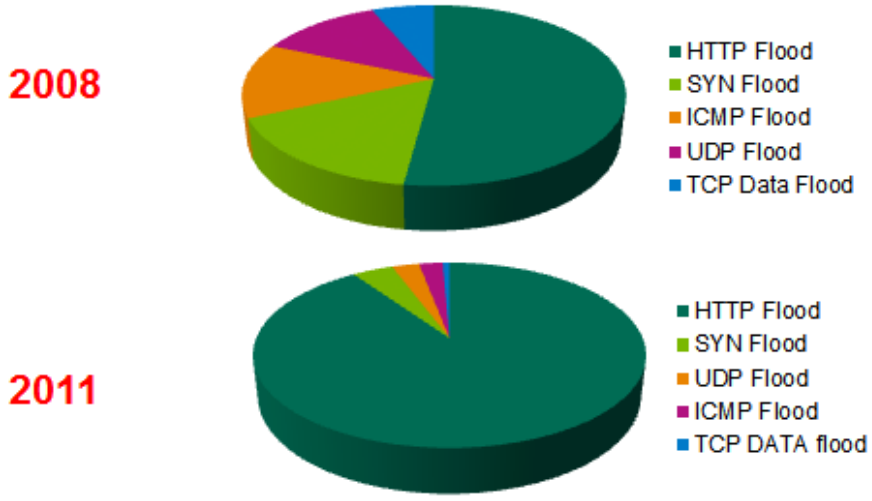

Fig. 1. Distribution of attack types in 2008 and 2011

The effort into implementing such attacks is in opening a connection with a server without sending even a single byte. Opening a connection and waiting for a reply requires almost no resources from the attacker, but it always binds a single server process to be waiting in executing a request. The server will wait until timeout expires and then the connection will be closed.

Opening only one connection will not cause severe damages, but opening hundreds of connections at the same time will occupy all available server processes. On reaching maximum number of processes, the server will log this event into a error log ("server has reached maximum number of queries (Max Clients), consider raising the number of Max Clients") and

will start saving new connections in a queue. If opening of new connections continues at a high rate, legitimate requests will not be maintained. If opening of these connections continues at even higher rate, queue will overflow and it will lead to rejection of new connections.

The main advantages of low intensity application layer attacks are [2]:

- these connections appear as legitimate user connections.
- traditional attack detection systems usually ignore them.
- existing signature-based IPS / IDS solutions, as a rule, do not recognize them.
- they require very few resources and low bandwidth for implementation.
- such attacks can "drop" web-server regardless of attacker`s hardware capabilities.

Different stages of query can be used to implement different types of low intensity attacks. Hence, there exist three types of low intensity attacks:

1. Slowloris attack

Information security specialist Robert Hansen designed an instrument for carrying out attacks such as "Denial of Service" on application layer called Slowloris or "slow headlines". This attack uses discovered vulnerability in architecture of Apache servers and other popular web-servers.

Unlike low-layer DoS-attacks, which allowed to "drop" any website by bombarding server and communication channels with packets, Slowloris can achieve the same results by sending a relatively small number of packets.

Practiced by modern hackers, this approach requires large amount of computing resources. In order to block one website, hackers usually use thousands of compromised computers. The Slowloris technology has minimal resource requirements [3]. It is enough resource to send 200 to 300 packets per minute in order to keep a website in an inoperative state. A standard personal computer can easily be used to execute the Slowloris attack.

The Slowloris attack will force attacked servers to serve large number of opened connections by continuously sending unfinished HTTP-requests. If such requests are sent with desired frequency, the server will wait for completion of each of opened connections. When the server is not overloaded – the processor can remain relatively idle, it just simply does not serve the next connections and requests.

The situation is that many web-servers, such as Apache, provide limitations for the number of simultaneously opened connections, which unfortunately is their main vulnerability. A developed method can be used to block servers like Apache 1.x, Apache 2.x, dhttpd, Go Ahead Web Server and Squid. But Slowloris does not present any particular risk for servers like IIS6.0, IIS7.0 or lighttpd. These solutions are equipped with effective mechanisms of load distribution and use "workerpools", which allow holding any number of opened connections upon availability of resources[2].

2. Slow Request Bodies or Slow HTTP POST attacks

Another type of low-intensity attacks on application layer is called “Slow Request Bodies” or Slow HTTP POST attack. This attack was demonstrated to the public at the OWASP 2010 Application Security Conference. Researcher Wong Onn Chee first discovered this attack together with a team of researchers from Singapore in 2009.

This type of attack is based on the vulnerability in the HTTP protocol. The Slow HTTP POST has the following operation algorithm: an attacker sends a POST header with legitimate "Content-Length" field, which allows a web-server to understand how much data it is going to receive. Once the header has been sent, the body of the POST message is transmitted at a very slow rate that allows using server resources for much longer than it is necessary and, consequently, prevents processing other requests. Few thousand of these connections can make a web server become inoperable in just a few minutes.

Such attacks are very easy to implement. For example, a simple Java applet can be used that will run during an online game. Once a victim accepts a self-signed applet, it starts to execute the attack while a user plays an online game. The attack terminates and then the applet is removed just after exiting the game and closing the browser. It is difficult for a user to detect the fact, that he is the source of the attack, because the computer is not infected in the classic sense of the term and it is very hard to distinguish such traffic from legitimate HTTP traffic [4]. Furthermore, the Internet channel remains almost unused. Moreover, it is the only known DoS-attack that can be organized through a proxy.

This attack leads to failure of web-servers with Microsoft IIS, Apache, etc. within the HTTP or HTTPS protocols, and obviously any "safe" connections like SSL, VPN, and others. Also, attack can be adapted to work with SMTP and even DNS-servers [5].

3. Slow Read attack

Previous low intensity attacks are aimed at slowing the rate of request transmission to a web-server. The Slow Read attack uses a method of slowing down the rate at which the client (attacker) is able to read data of query response that comes back from web-server.

The client sends a complete request, but when the server responds, it advertises a very small TCP window for response data [6]. Thus, server sends data to the client slowly, keeping its sockets opened. It keeps probing the client to check its receive window size while the client always advertises a small window size, slowing down the transfer. The larger the file, the longer it takes to complete such connections. Several requests of this type for a large file can very quickly lead server inoperable.

II. Implementation of low intensity attacks

In August 2011, Sergey Shekyan wrote a tool called «slowhttptest», which tests web servers for vulnerabilities connected with processing slow HTTP requests, such as Slowloris, slow HTTP Post and Slow Read [2]. All these types of attacks were carried out using above-mentioned tool.

Scenario of HTTP Slowloris attack implementation (fig. 2).

The attacker sends a request by parts of a very small size (13 bytes) with 5 second delay between them, allowing him to keep connection opened for a long time.

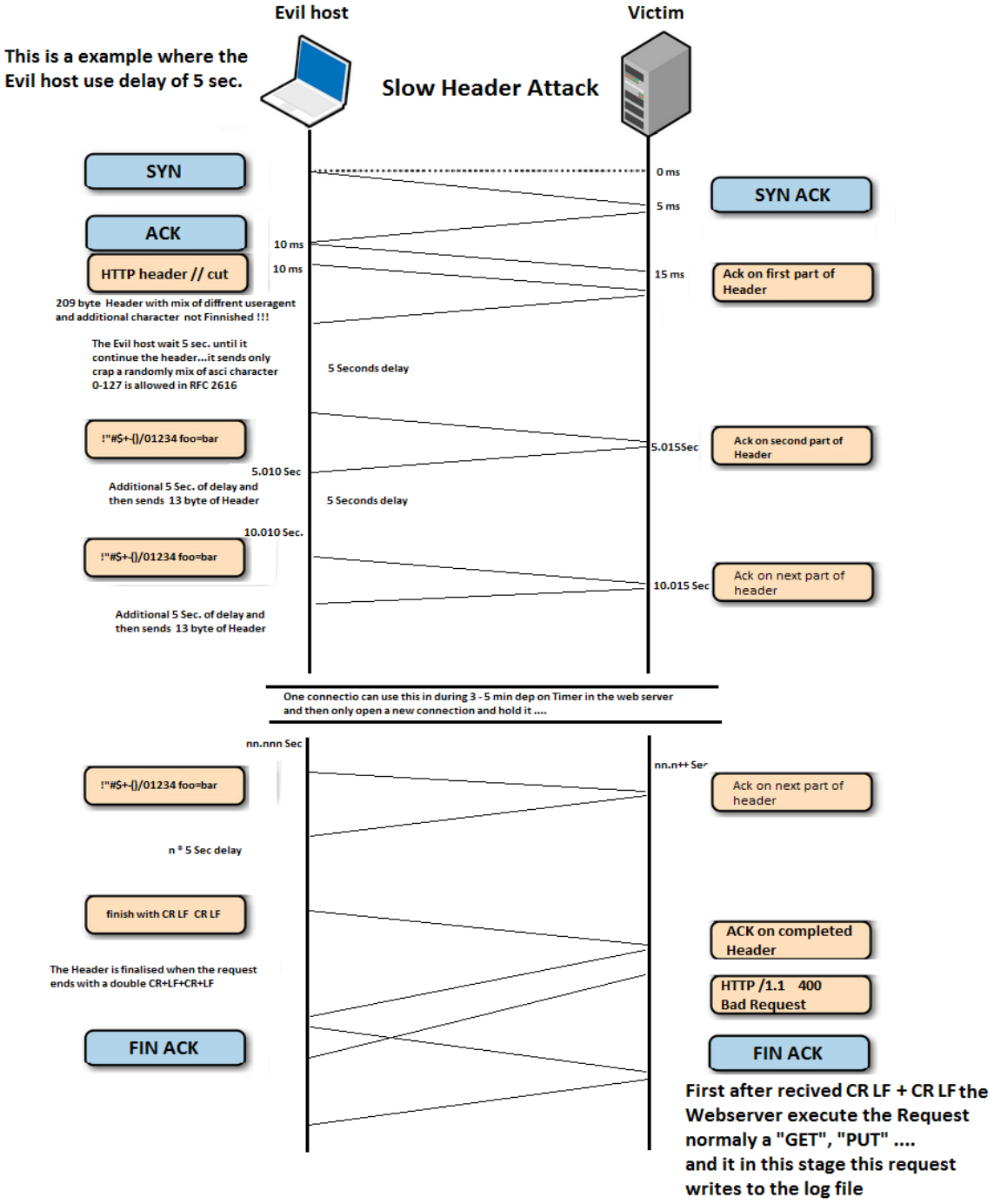

Fig. 2. Scenario of HTTP Slowloris attack implementation

At the end of the query, a client must send to the server the final CRLF, which informs him that request is complete. If this is not done, the web-server will wait for the rest of the request data until it reaches timeout. Thus, the Slowloris can keep web-servers in a constant standby mode, sending new requests until reaching timeout.

Scenario of HTTP Slow Body attack implementation (fig. 3):

The attacker sends the body of request by parts of a very small size (15 bytes) with 10 second delay between them, allowing him to keep connection opened for a long time.

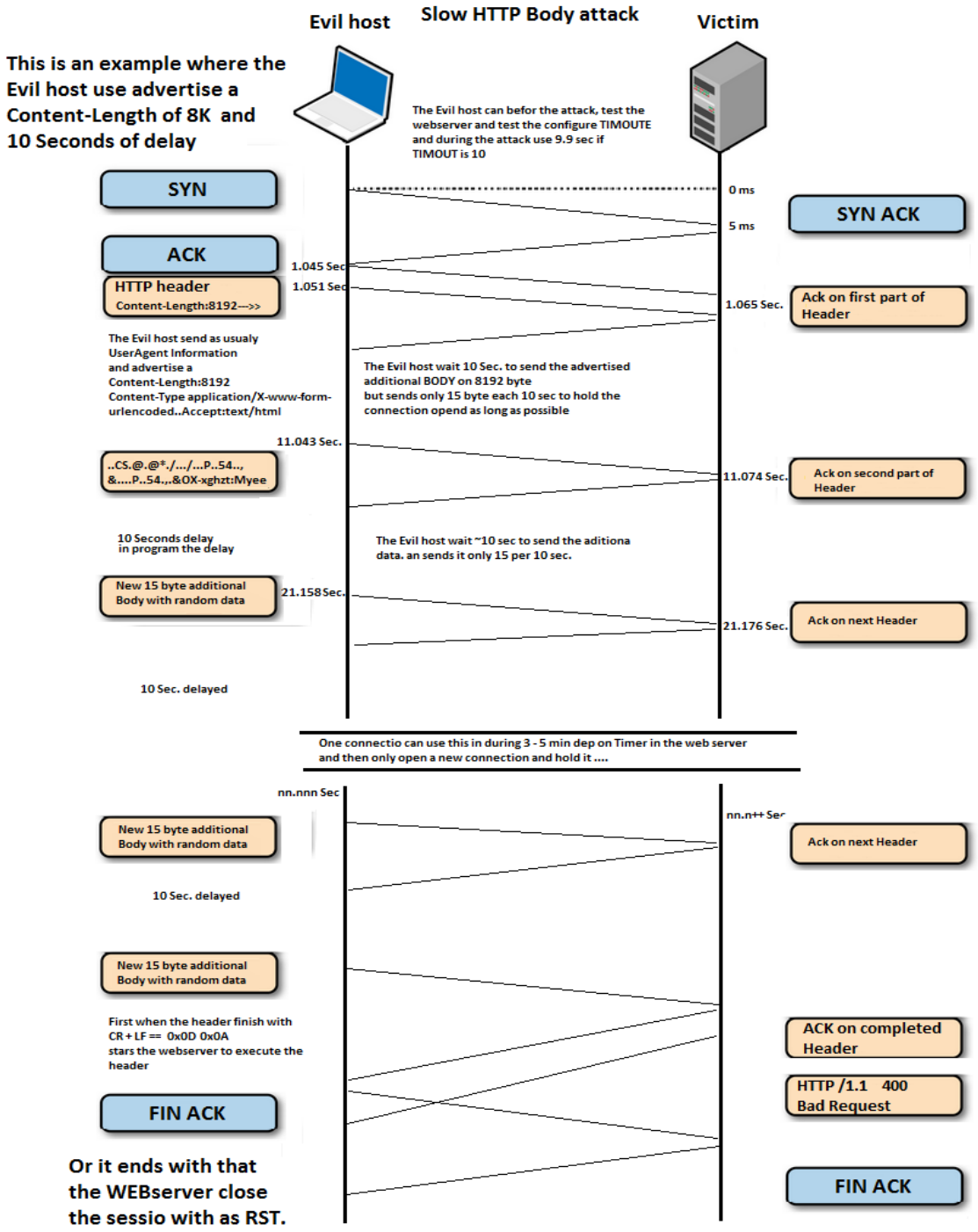

Fig. 3. Scenario of HTTP Slow Body attack implementation

Implementation of this attack requires creation of a normal test connection to the server in order to obtain information about the connection timeout of the server. Server timeout was equal to 10 seconds, so implementation will take 9,9 seconds.

First after that Time Out occurred or a full Body is upload the web-server will execute and respond to the request from client, the Web-server execute the BODY part and if the server not understand the BODY part, it returns a HTTP/1.1 400 Bad Request. And the information delivers to the «LOG» system.

Scenario of HTTP Slow READ attack implementation (Figure 4):

Attacker sends a request to the server, where it indicates window size of 572 bytes. However, when a server starts to send data to the client, the client reports that its window size is 0.

Accordingly, the server is constantly trying to send data to the client, but the client reads them very slowly that allows keeping the connection opened.

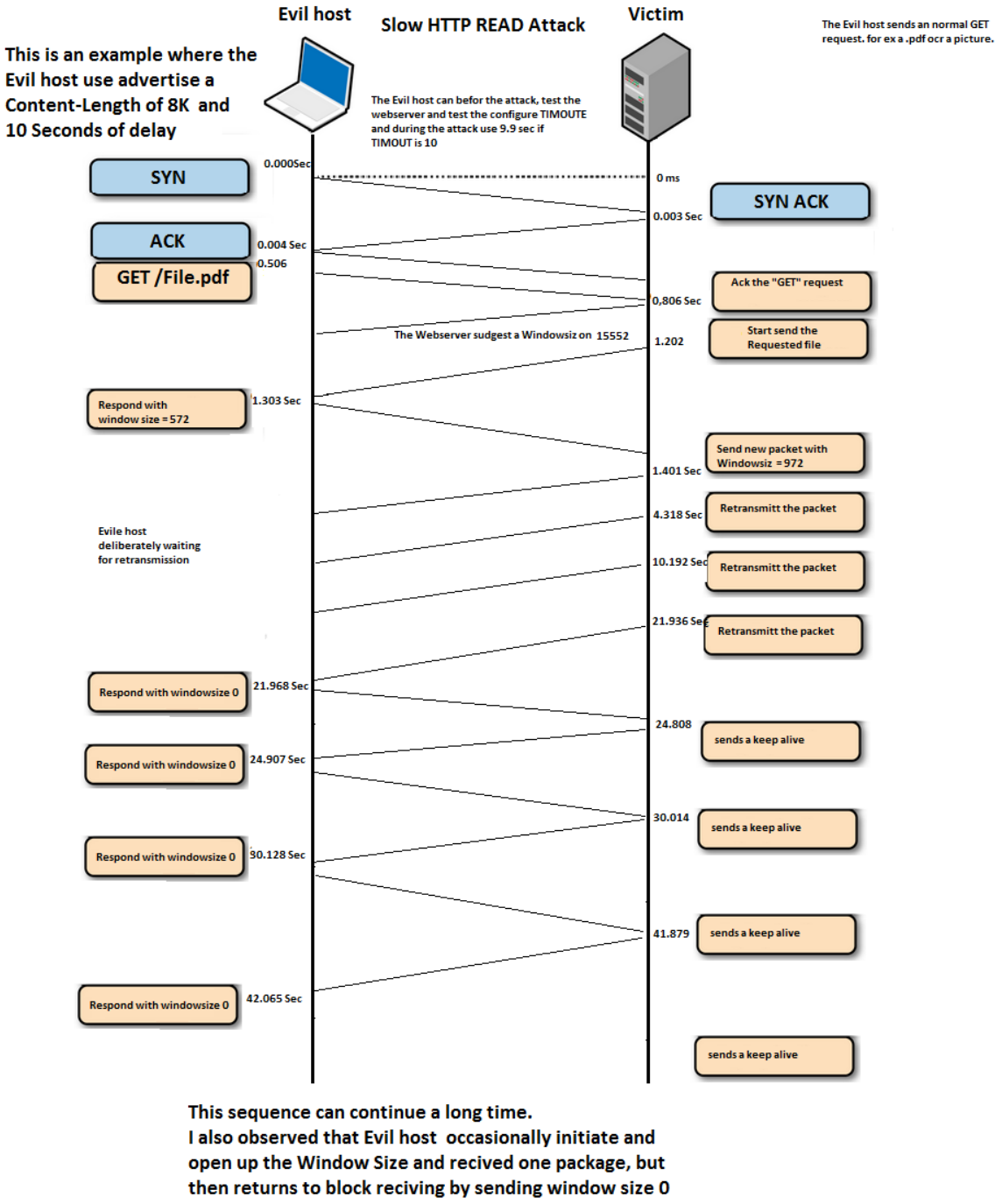

Fig. 4. Scenario of HTTP Slow READ attack implementation

The table below describes the main parameters in the implemented attacks that led to the server into an inoperable state (table 1).

Table 1. The main parameters in the implemented attacks

Attack type	Slowloris	Slow BODY	Slow READ
Number of connections	1000	410	1000
Request type	GET	POST	-
Receive window size	-	-	20-572
The number of simultaneous requests on the same connection	-	-	1
Reception buffer reading speed	-	-	32bytes/5 sec
Value of Content-Length field in the header (length of message body in bytes)	209	8192	-
Additional data field	52	66	-
Interval between successive data (sec)	5	10	-
Connections per second	300	200	100
Timeout for test connection	5	5	5
Test duration (sec)	200	240	240
Using proxy	no	no	no

The implemented types of attacks on the application layer using slow http test tool are: Slowloris, slow HTTP Post and slow Read. The first two attacks are intended to send HTTP request with a long delay between its parts, in the first case - slow request header transfer, in the second case - slow request body transfer. The third attack, unlike the previous two, does not send a request to the server, but executes the next step of connection - slowly reads server response. Slow Read attack advertises a large TCP window size to the server for receiving the response data, then, when the server starts to respond, it changes window size to very small value. All three attack types help in keeping the server connection opened for a long time. Simultaneous opening of a few hundred of such connections will lead small, medium and large sites into an inoperable state in a matter of a few seconds, blocking all possible connections to the server, and thus not allowing the legal sources to be served.

Attacks of this type are a huge threat for web-servers nowadays, as is difficult to diagnose the attack, because traffic does not exceed normal values. The similarity of such attack traffic from legitimate traffic complicates filtering "bad" packets and makes it easy to organize a hard detectable botnet. In addition, the implementation of these attacks does not require powerful resources from the attacker: sometimes it can be enough to use only one computer in order to "drop" the server.

Conclusion

Most existing DoS-attack detection methods and systems can effectively recognize and deal with snowballing DoS-attacks on network and transport layers, which are aimed at filling bandwidth (Smurf, UDP-flood, etc.) and exceeding the normal load of individual network nodes (SYN-flood, Teardrop, Ping of death, etc.). However, existing methods and means of DoS-attack detection are ineffective against modern DoS-attacks on application layer, directed at specific network services. It is quite difficult to distinguish traffic that was generated during such attacks from legitimate application layer traffic, making it difficult

to use signature method of intrusion detection. Furthermore, low activity DoS-attacks do not lead to formation of statistical anomalies, because data transfer channels are not overloaded.

Application layer attack detection requires constant monitoring of all systems in real time with drawing up the reports on flow and allocation of resources. Driving continuous analysis, protective tool, that supports resource accounting, must detect such deviations from the norm, as large number of simultaneous connections to one ip-address or incomplete processes, which have to be closed.

These facts necessitate the development of specialized method for detecting distributed low-intensity application layer attacks such as "denial of service" in computer networks. Set of the following conditions, resulting from ongoing network monitoring, can give prerequisite for thinking that client is intruder and tries to attack the server (table 2):

Table 2. The main parameters in the implemented attacks

Slowloris	Slow POST	Slow READ
Number of concurrent requests from one IP-address (varies from hundreds to thousands of queries, depending on server magnitude)		Keep-Alive and HTTP conveyer are turned on
Delay between transmissions of parts approaches the value of connection timeout, but does not reach it;		Delay between receiving portions of server response approaches the value of connection timeout, but does not reach it;
Request header is sent in small pieces (10-20 bytes);	Parts of request are sent in small portions (10-20 bytes);	The initial size of receive window is sufficiently large;
Waiting for double CRLF till almost reaching the timeout	Sufficiently large content length field, with relatively small portions of transmitted data	Server receives SYN-packets with an abnormally small TCP window size.

Analysis of all the values of these conditions allows allocating a set of parameters, which are specific for Slow HTTP attacks on application layer:

Characteristic parameters for Slowloris and Slow POST are:

1. Number of requests;
2. Number of IP-addresses;
3. Connection speed;
4. The interval of client activity;
5. The ratio between claimed window size and transmitted data.

Characteristic parameters for Slow READ are:

1. Number of requests;
2. Number of IP-addresses;
3. Connection speed;
4. The interval of client activity;
5. The difference between initial and subsequent (tends to 0) TCP window size;
6. Presence or absence of permanent connections (Keep-Alive) and HTTP conveyer.

There are several mathematical tools, which allow implementing Slow HTTP attack detection models, based on the received parameters: approach based on Markov chains, deterministic factor approach based on the rules, approach based on indistinct conclusion rules and approach based on neural networks.

References

1. *Maria Garnaeva*. DDoS-attack in the first half of 2012-Securelist [Electronic resource]. – Access mode: http://www.securelist.com/ru/blog/207764220/DDoS_ataki_v_pervom_polugodii_2012.

2. *Kelly Jackson Higgins.* Researchers to Demonstrate New Attack That Exploits HTTP [Electronic resource]. – Access mode: http://www.darkreading.com/attacks-breaches/researchers-to-demonstrate-new-attack-th/228000532.

3. Slowloris HTTP DoS [Electronic resource]. – Access mode: http://ha.ckers.org/slowloris.

4. *Nicole Henderson.* Slow Read DOS Attack Created by Software Engineer Shows HTTP Server Vulnerability [Electronic resource]. – Access mode: http://www.thewhir.com/web-hosting-news/slow-read-dos-attack-created-by-software-engineer-shows-http-server-vulnerability.

5. *Martin Jartelius.* The Slow Denial of Service - the vulnerability management blog [Electronic resource]. – Access mode: http://blog.outpost24.com/2013/02/25/the-slow-denial-of-service.

6. *Ian Muscat.* How to mitigate Slow HTTP DoS Attacks in Apache HTTP Server [Electronic resource]. – Access mode: https://www.acunetix.com/blog/web-security-zone/articles/slow-http-dos-attacks-mitigate-apache-http-server.