UDC 004.056.57

Ie.V. Duravkin[1], Anders Carlsson[2], A.S. Loktionova[1]

[1] *Kharkov National University of Radioelectronics, Kharkov*
[2] *Blekinge Institute of Technology, Karlskrona*

METHOD OF SLOW-ATTACK DETECTION

The values of the quality characteristic of TCP connections for each type of Slow-http attacks are highlighted. A mathematical model for formalizing the behavior of the web-server when implementing the various types of Slow HTTP-attacks was developed. The relations that enable us to estimate the probability and the transition to the web-server into overload condition, with the current network activity settings were defined. The detection algorithm of Slow-http attack and the classification of its type are developed. The problem of detecting the source of the attack and the development of measures to protect the web-server from overload are solved. Architecture Slow HTTP-attacks, allowing to implement the developed algorithm is proposed.

Keywords: *Denial of service, network attack, Slow-http, web-server.*

Introduction

Conventional DOS/DDOS attacks introduce massive amount of packet's, bursts with 10 000'sof packet's per time frame and in DDOS attacks 1000's of attacking host with multiple 1000 packet from each host.

Slow attacks, on the other hand, leverage modern devices as smart cell phones, tablets that are in far from the access points or base stations in a modern mobile networks.

One good example is to have a cell phone/tablet in a Metro Train Station where the connectivity is good, and then, to continue, going in and out of Metro Tunnel's, where the connectivity is almost completely broken. Packet were transmitted to negotiate down the bandwidth such as lowering TCP window size, length of packet, with may be only a few character in each point of connectivity, while trying to keep the attacks themselves alive as long as possible. This is compounded by the "base stations" in the mobile networks lowers the priority of nodes in area of bad connection, to have the possibility to give higher connectivity to nodes in area of good connectivity.

The main implementation features of slow HTTP attacks include (1) not requiring high intensity traffic generation and (2) performing direct attacks at the application layer of targeted web-servers.

Most existing DoS-attack detection systems on network and transport layers were based on finding attack signatures or traffic anomalies leveraging methods such as wavelet analysis, regression analysis, etc. [3].

Another important feature of slow HTTP attacks is the traffic similarity between the attacks and legitimate traffic. Unlike ordinary DoS attacks, slow HTTP attacks do not fill up the network bandwidth, but they deplete resources (memory, CPU time) at the application layer (e.g., web-servers). Consequently, existing DOS-attack detection systems are ineffective for detecting slow HTTP attacks.

There have been several possible mechanisms proposed to protect web-servers against slow HTTP attacks:

- using programs like «Flying frog», which is based on the monitoring of HTTP-traffic and real-time event analysis;
- creating rules that limit the number of simultaneous streams with the same IP-address;
- query caching;
- using filters like DDoS GUARD.

However, these mechanisms are only partial solutions as they cannot protect servers in their entirety. Thus, it became necessary to develop a systematic approach, which can not only detect the attack incidences, but also identify attack sources and thus protect the targeted servers.

Development of Slow HTTP attack detection system

The model of low-intensity DoS-attack detection on web resources, based on the mathematical apparatus of Markov chains, queuing theories, and generating functions was described in [1, 2].

The architecture of our model is shown in Fig. 1.

IDS consists of 6 modules.

- traffic collection (sniffer);
- calculation of various traffic parameters;
- generation of network statistics;
- calculation of web-server parameters;
- marking of potentially attack traffic statistics;
- attack classification.

IDS workflow consists of the following stages:

1. At the first stage traffic enters the input of the SOA.

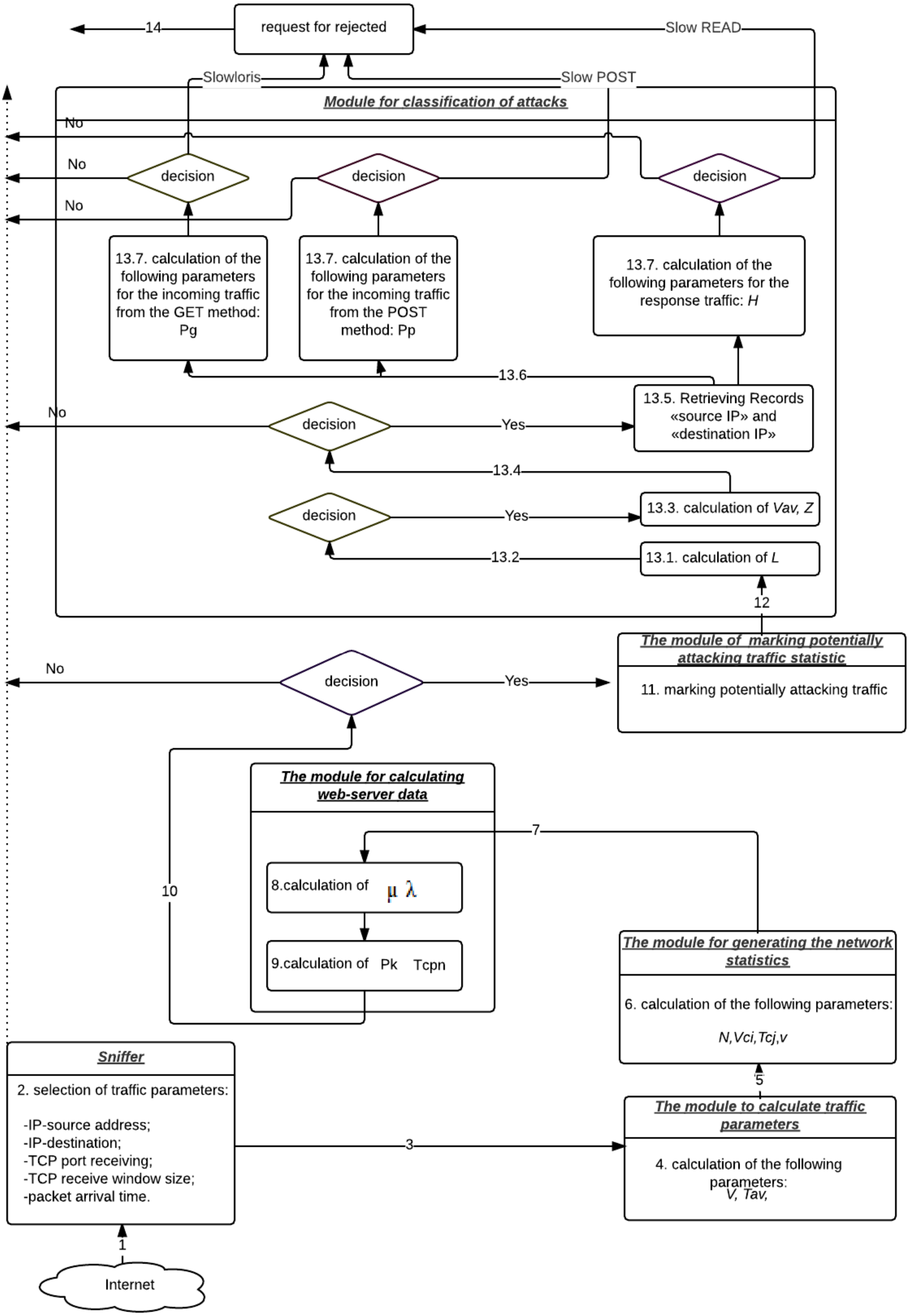
request for rejected
14
Slow READ
Slowloris
Slow POST
Module for classification of attacks
No
No
No
decision
decision
decision
13.7. calculation of the following parameters for the incoming traffic from the GET method: Pg
13.7. calculation of the following parameters for the incoming traffic from the POST method: Pp
13.7. calculation of the following parameters for the response traffic: H
13.6
13.5. Retrieving Records «source IP» and «destination IP»
No
decision
Yes
13.4
13.3. calculation of Vav, Z
decision
Yes
13.2
13.1. calculation of L
12
The module of marking potentially attacking traffic statistic
11. marking potentially attacking traffic
No
decision
Yes
The module for calculating web-server data
7
8.calculation of μ λ
9.calculation of Pk Tcpn
10
The module for generating the network statistics
6. calculation of the following parameters:
N,Vci,Tcj,v
5
Sniffer
2. selection of traffic parameters:
-IP-source address;
-IP-destination;
-TCP port receiving;
-TCP receive window size;
-packet arrival time.
3
The module to calculate traffic parameters
4. calculation of the following parameters:
V, Tav,
1
Internet

Fig. 1. Architecture of Slow HTTP attack detection

2. For a specified period of time, the module for traffic collection allocates traffic parameters required for further IDS calculations:

- source IP-address,
- destination IP-address,
- TCP port of reception,
- TCP window size,
- packet arrival time.

3. Selected parameters are passed to the module for calculating the parameters of traffic.

4. At the next stage the following traffic characteristics are calculated for each IP-addresses:

– The total amount of data transmitted over the analyzed period of time

$$V=\sum_{i=1}^{k}V_{i|}\,|_{\Delta t}\,, \tag{1}$$

where:

V – total volume of data;

V_i – volume i-th data packet;

k – number of packets;

Δt – analyzed period of time.

– The average time interval between transmitted packets.

$$Tcp=\frac{\sum_{i=1}^{k}(t_{i+1}-t_i)}{i-1}|_{\text{Д}t}\,, \tag{2}$$

where:

Tav – average time interval between transmitted packets;

t_i – the i-th packet arrival time;

t_{i+1} – the $(i+1)$-th packet arrival time;

k – number of packets, received during the analyzed period of time;

Δt – analyzed period of time.

5. Calculated parameters are passed to the module of formation the network statistics.

6. Generation module extracts a set of entries of statistics about traffic and calculates the following parameters:

– number of sessions for a given time interval

$(N \mid \Delta t)$;

– the amount of data for each session, with reference to the host session

$$Vci=\sum_{i=1}^{k}V_i\,, \tag{3}$$

where:

Vci – volume of data on the i-th session;

k – number of packets per session;

i – session number.

– the delay between packets with in a session

$$Tcj=t_{j+1}-t_j\lim_{\delta x\to 0}\,, \tag{4}$$

where:

Tcj – delay between packets within a session;

t_j – the j-th packet arrival time;

t_{j+1} – the $(j+1)-$ th packet arrival time;

j – packet number in the session.

– connection speed (v)

The built-intimer allows us to record the start and end of the session, which makes it possible to monitor the duration of open connections.

7. Calculated parameters are passed to the module for calculating parameters of web server.

8. At the next stage, the module for calculating parameters of web-server calculates intensity of income and processing of HTTP-request sat each time interval: the ratio of received packets with a certain IP-addresses for the given time interval.

Intensity of inflow of http-requests:

$$\lambda=\frac{k_i}{t}\,, \tag{5}$$

where:

λ – intensity of inflowing http-requests;

k_i – number of inflow requests, analyzed on a given interval;

t – reporting interval.

$$\mu=\frac{k_j}{t}\,, \tag{6}$$

where:

μ – intensity of http-request processing;

k_j – number of processed requests, analyzed on a given interval;

t – reporting interval.

9. Based on calculated intensities of incoming and processing requests, the load of web-server and the time at which it reached an overload condition are determined [2].

10. At the next stage of the IDS, the legality of a potential attack incidence is determined on the basis of calculated parameters. Statistics of connection with IP-addresses, whose parameters exceed defined static thresholds, are transmitted to the module of marking potential attack traffic.

11. Detailed traffic statistics, whose calculated parameters exceed static thresholds, is marked as "potentially attacking traffic statistics."

12. This statistics is transmitted to the module of attack classification.

The task of the module for attack classification is to classify attacking traffic assignment to a class of Slow HTTP attacks, or removal of the marker attack, if connection parameters do not exceed the thresholds.

In order to solve this problem, an algorithm was created to run in two ways – for applicant and response from web-server traffic flows.

13. The process of this module consists of several stages:

13.1 At the first stage the ratio of IP-addresses marked traffic to the amount of HTTP-requests for a predetermined time interval is calculated:

$$L = \frac{N}{k}\Big|_{\text{Д}t}\,, \qquad (7)$$

where:

L – ratio of IP-addresses of the marked traffic to the number of HTTP - requests for a given time interval;

N – number of IP-addresses of the marked traffic;

k – number of requests in a given time interval;

Δt – given period of time.

13.2 At the second stage calculated parameters are compared with predetermined threshold values:

– if the value L does not exceed static thresholds, a decision that traffic belongs to a valid user and token statistics potential attack traffic is removed;

– if the value L exceeds predetermined threshold, statistics compound for warded to the next stage of attack classification.

13.3 At the third stage, average connection speed and delay between packets is counted for each IP-address:

$$Vcp = \sum\nolimits_{i=1}^{n} V_i \Big/ i\,, \qquad (8)$$

where:

Vi – transfer rate of the i-th part of the request;

n – amount of transmitted parts per query.

13.4 At the fourth stage calculated parameters are compared with predetermined threshold values:

– if the values Vav and Z do not exceed a predetermined threshold, it is decided that traffic belongs to a valid user and token 'statistics potential attack traffic' is removed;

– if the values Vav and Z exceed a predetermined threshold, statistics compound for warded to the next stage of attack classification.

13.5 At the fifth stage of attack classification there cords «source IP» and «destination IP» are retrieved from potentially attacking traffic statistics.

13.6 At the sixth stage of the calculated parameters of statistics potentially attacking traffic is classified, according to the extracted method query.

13.7 At the seventh stage, the statistical analysis of the incoming connection and return traffic:

$$Pg = \frac{Vg}{\sum_{i=1}^{n} Vg(i)/i}\,, \qquad (10)$$

where:

Pg – ratio of total size of request header to the mean value of transmitted data in specified time for GET requests;

Vg – total size of request header;

$Vg(i)$ – the i-th portion of transmitted request header;

n – number of pieces of transmitted request header.

$$Pp = \frac{Vp}{\sum_{i=1}^{n} Vp(i)/i}\,, \qquad (11)$$

where:

Pp – ratio of total size of request body to mean value of transmitted data in specified time for POST requests;

Vp – total size of header;

$Vp(i)$ – i-th of the transmitted request header;

n – number of transmitted request header pieces.

If the values Pg for traffic with method GET or Pp for traffic statistics with method POST does not exceed predetermined threshold, it is decided that traffic belongs to legitimate user and token of "statistics of potential attack traffic" is removed.

If the value Pg exceeds predetermined threshold, it is decided to classify incoming traffic to the class of low-intensity attacks of Application level Slowloris, and intrusion detection system sends request to web-server to close connection from specified IP-addresses.

If the value Pp exceeds a predetermined threshold, it is decided to classify the incoming traffic to the class of low-intensity attacks of Application level Slow POST, and intrusion detection system sends request to Web server to close connection from specified IP-addresses.

For response traffic flow the ratio of the initial size of the TCP receive window to the size of parts of the flow return traffic is calculated:

$$H = \frac{Vtcp}{\sum_{i=1}^{n} V(i)/i}\,, \qquad (12)$$

where:

H – ratio of initial TCP receive window size to the size of parts return traffic flow;

$V(tcp)$ – initial size of TCP receive window;

$Vp(i)$ – th of the the response of the request;

n – number of service able parts request for the specified time interval.

If the value H of response from the web-server does not exceed predetermined threshold, a decision is made that return traffic belongs to legitimate user and token of statistics of potential attack traffic is removed.

If the value H exceeds predetermined threshold, it is decided to classify traffic to the class of low-intensity attacks of Application level Slow READ and intrusion detection system sends a Web server request to close connection from specified IP-address.

Conclusions

The developed system allows detect the fact of Slow HTTP attack implementation, identify the source of the attack and block malicious traffic.

At the heart of the proposed system of protection is the analysis of the behavior of a web server in normal operation and in the implementation of Slow HTTP attack type differences. The analysis allowed to identify the most sensitive parameters to Slow HTTP attack attacks, as well as set their thresholds, depending on the channel capacity, equipment performance and configuration settings of the Web server.

The process of attack detection is implemented based on a Markov model the behavior of the web server, the model parameters are the statistical characteristics of incoming, outgoing traffic, as well as the dynamics of resource use web server.

Feature of the implementation Slow HTTP attack is to use one source of the attack. Formation of traffic statistics with reference to IP-based source and destination addresses enables identification of intruders, and thus block the malicious traffic.

The advantage of the proposed system is that it allows you to detect an attack to the server status of failure, which makes it possible to implement security mechanisms in a timely manner.

References

1. Carlsson A. Analysis of realization and method of detecting low-intensity HTTP-attacks [Электронный ресурс] / A. Carlsson, E.V. Duravkin, A.S. Loktionova // Проблеми телекомунікацій. – 2013. – № 3 (12). – С. 61-70. – Режим доступа к журналу: http://pt.journal.kh.ua/ 2013/3/1/133_carlsson_attack.pdf .

2. Carlsson A.A. Analysis of realization and method of detecting low-intensity HTTP-attacks. Part 2. Method of detecting Slow HTTP attacks [Электронный ресурс] / A.A. Carlsson, I.V. Duravkin, A.S. Loktionova // Проблеми телекомунікацій. – 2014. – № 1 (13). – С. 96-100. – Режим доступа к журналу: http://pt.journal.kh.ua/2014/1/1/ 141_carlsson_attack.pdf.

3. Denial of Service Attack – Prevent DoS Attacks with Palo Alto Networks [Electronic resource]. – Access mode: https://www.paloaltonetworks.com/resources/learning-center/what-is-a-denial-of-service-attack-dos.html

4. Albert R. Meyer, Ronitt Rubinfeld Generating Functions // Mathematics for Computer Science. – 2005. – P. 9-12.

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

6. 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.

7. Щерба М.В. Обнаружение низкоактивных распределенных атак типа «отказ в обслуживании» в компьютерных сетях: дис. ... канд. техн. наук / Щерба М.В. – Омск, 2007. – 130 с.

Поступила в редколлегию 3.09.2014

Рецензент: д-р техн. наук, проф. Д.В. Агеев, Харьковский национальный университет радиоэлектроники, Харьков.

МЕТОД ОБНАРУЖЕНИЯ НИЗКОИНТЕНСИВНЫХ АТАК

Е.В. Дуравкин, Андерс Карлссон, А.С. Локтионова

Выделены значения показателей качества TCP соединений характерные для каждого типа Slow-http атак. Разработана математическая модель, формализующая поведение web-сервера при реализации Slow HTTP-атак различного типа. Выведены формульные соотношения, позволяющие оценить вероятность и время перехода web-сервера в состояние перегрузки при текущих параметрах сетевой активности. Разработан алгоритм обнаружения Slow-http атаки и классификации ее типа. Решена задача обнаружения источника атаки и выработки мер по защите web-сервера от перегрузки. Предложена архитектура системы обнаружения Slow HTTP-атак, позволяющая реализовать разработанный алгоритм.

Ключевые слова: *отказ в обслуживании, сетевая атака, низкоинтенсивные -HTTP атаки, web-сервер.*

МЕТОД ВИЯВЛЕННЯ НИЗЬКОІНТЕНСИВНИХ АТАК

Є.В. Дуравкін, Андерс Карлссон, А.С. Локтіонова

Виділено значення показників якості TCP з'єднань що характерні для кожного типу Slow-http атак. Розроблено математичну модель, яка формалізує поведінку web-сервера при реалізації Slow HTTP-атак різного типу. Виведено формульні співвідношення, що дозволяють оцінити ймовірність і час переходу web-сервера в стан перевантаження при поточних параметрах мережевої активності. Розроблено алгоритм виявлення Slow-http атаки і класифікації її типу. Вирішена задача виявлення джерела атаки і вироблення заходів по захисту web-сервера від перевантаження. Запропонована архітектура системи виявлення Slow HTTP-атак, що дозволяє реалізувати розроблений алгоритм.

Ключові слова: *відмова в обслуговуванні, мережева атака, низькоінтенсивні-HTTP атаки, web-сервер.*