AiMT

Advances in Military Technology
Vol. 13, No. 1 (**2018**), pp. 87-94
ISSN 1802-2308, eISSN 2533-4123
DOI 10.3849/aimt.01220

Detection of Attacks Causing Network Service Denial

V. Ďurčeková, L. Schwartz*, V. Hottmar and B. Adamec

University of Žilina, Slovak Republic

The manuscript was received on 30 November 2017 and was accepted after revision for publication on 13 April 2018.

Abstract:

This article deals with ICT security and particularly the Denial of Service (DoS) executed on the application layer. The main objective of the article is to describe the original algorithm designed for timely detection of DoS application attacks and, subsequently, on the results of experimental verification of the designed process. This algorithm is focused on the detection of HTTP GET Flood attack, which will cause a crash of the attacked server. Appropriate detection of attack from the analysis of incoming traffic is able to prevent a crash of server from happening. To detect such an attack, an original algorithm designed by our team was used.

Keywords:

Denial of Service – DoS, Distributed Denial of Service – DDoS, HTTP GET Flood attack, algorithm, simulation, theory of queuing systems

1. Introduction

This article deals with both DoS and DDoS attacks. In [1], an overview of existing systems performing various DoS detection systems is presented. These systems are systematically classified in the following way:

- Audit Source: Network-based and Host-based
- Usage Frequency: Real-time and Delayed
- Model Generation: Programmed and Learned
- Detection Paradigm: Signature-based, Anomaly-based and Classification-based

An overview of types of DDoS attacks is presented in [2]:

- Flooding attacks
- Vulnerability attacks:
  - On the basis of attacks mechanism: Direct DDoS attacks and Indirect DDoS attacks,

* *Corresponding author: Department of Multimedia and Information-Communications Technology, University of Žilina, Univerzitná 8215/1, 010 26 Žilina, Slovak Republic.*
Phone: +421 41 513 2214, E-mail: ladislav.schwartz@fel.uniza.sk

- On the basis of target protocol: Network Transport-level DDoS attacks and Application-level DDoS attacks.

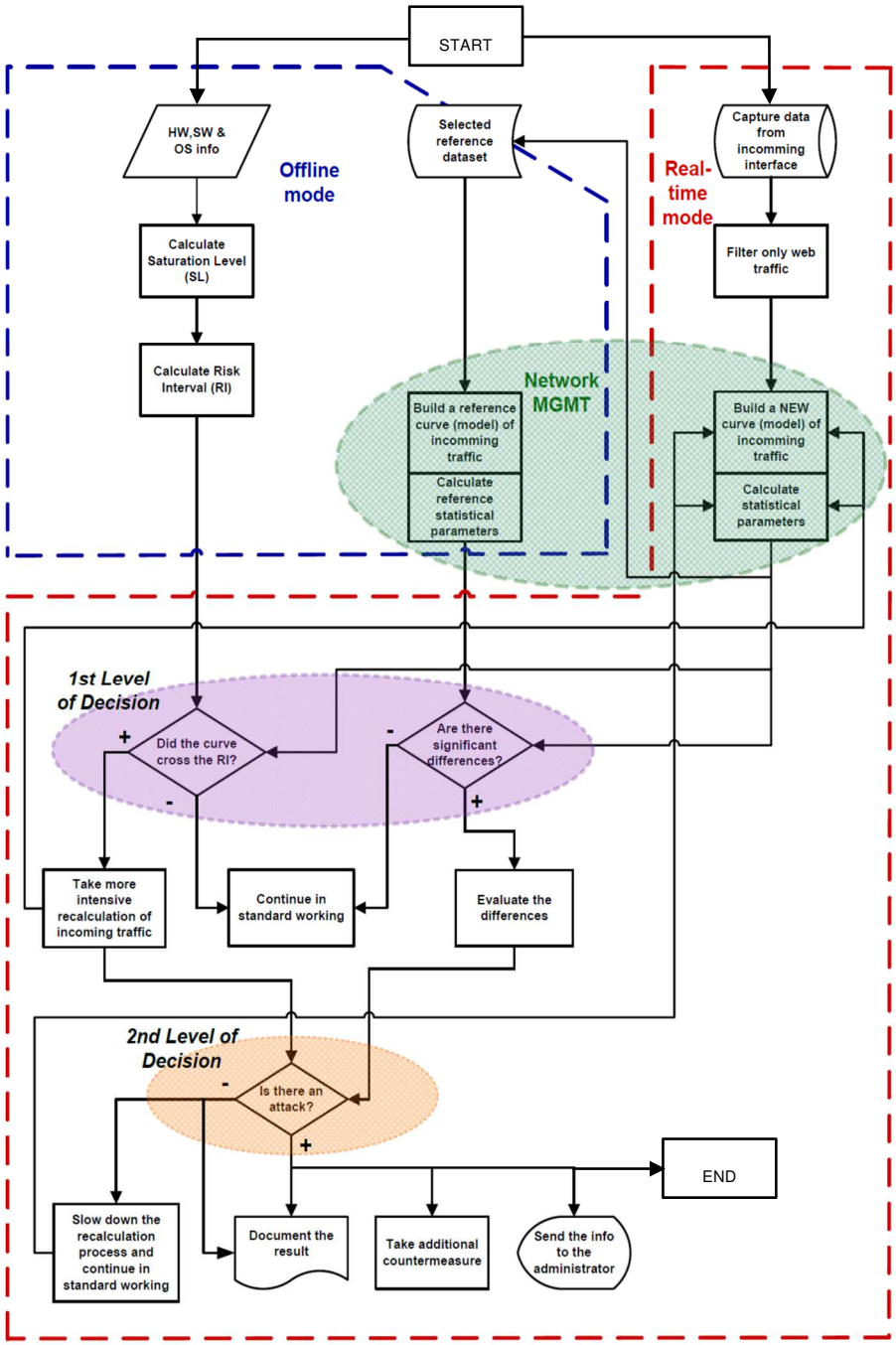

Fig. 1 Flow diagram of the designed detection algorithm

This article describes the development of an original detection algorithm on the application layer. The first designed methods of DoS / DDoS attack detection include:

- Client Puzzle Protocol (CPP),
- Input Filtration [3-5],
- Intrusion Detection Systems (IDS) [6, 7],
- Threshold Limits [8].

These methods are unable to provide a complex, effective and timely detection of application attacks. The related problems are described in [3 to15]. The currently developed procedures for the detection of application DoS and DDoS attacks can be divided into two directions, based on the following principles:

- the detection based on the signature of attack,
- the detection based on anomalies in the network traffic.

The result of combining these two directions is a hybrid system encompassing the advantages of the individual directions. The effective detection mechanism based on the identification of anomalies in the network traffic can be considered a more efficient and perspective tool which is described in the following parts of the paper.

2. Development of Detection Algorithm

The article presents a newly designed, original flow diagram for the detection algorithm of DDoS attacks. It was created on the basis of detailed analyses of application attacks focusing on the denial of services, in particular on the service offered by HTTP protocol. This designed detection algorithm combines the older detection method defining saturation level of the monitored device with the latest trend of application attack detection based on the identification of anomalies in the network traffic. We have produced a reference mathematical model of server traffic by applying the queuing theory and the model of real server traffic. These models were compared and the measured dissimilarities were evaluated. The threshold limit of traffic of the monitored server, which states the maximum number of requests processed by the server without saturating its services, was also defined.

The designed flow diagram of the original detection algorithm is shown in Fig. 1. The flow diagram consists of two blocks:

- reference off-line mode,
- real-time monitoring mode.

In the off-line reference mode, the calculation of saturation level (SL) and risk interval (RI) is as follows. Saturation level is determined by the maximum of serving capacity of server which is given by the configuration of its hardware. That level is compared with actual intensity of incoming traffic. The overflow defined by SL indicates an irregular situation on server. Risk interval defines the limit for the capacity of incoming traffic. The defined limit of RI directly depends on the saturation level. The upper limit of RI lies under the limit of SL and is defined by percentage. The overflow of RI then indicates that on the server, there is a risk situation in receiving requests on processing that can lead to serving maximum of server, or even to its crash.

3. Experiments and Verification of the Designed Algorithm

The functionality of the newly designed detection algorithm for DoS / DDoS application attacks of HTTP GET Flood type was experimentally verified in laboratory conditions. For the purpose of the experiment, we used an Apache web server as an attack target,

which is at present the most widely used solution for Web services [10]. From the configuration of the WEB server, it was possible to derive important parameters used in the queuing theory model, namely the number of serviced positions and further queue dimension, which is defined by 50 waiting positions using a FIFO system [12].

The modern multifunction device AVALANCHE 290 was used as a generator of both nominal and attack traffic. For experimental verification of the activity of the designed detection algorithm, a simple network topology was used, as depicted in Fig. 2 [17].

The use of the proposed topology guaranteed that between the generator of traffic and the monitoring server there was no additional delay caused by the activity of network components (routers or switches). Such direct interconnection of devices thus guaranteed that the specific profile and properties of generated traffic would not be modified by network components themselves. The aim of our experiments was to supply the input of the monitored WEB server traffic with "GET requests" type of message of varying intensity.

This was done with the aim not to influence the experimental load tests by the transit of artificially generated traffic from source to target.

4. Reference Models

For the reference model (RM) of the monitored WEB server, queuing system type M/G/1/50 was selected. Here the parameter M defines exponential distribution of the arrival time of requests, parameter G defines the general distribution of service time, number 1 defines that there is one service place in the system and number 50 defines the length of the waiting queue.

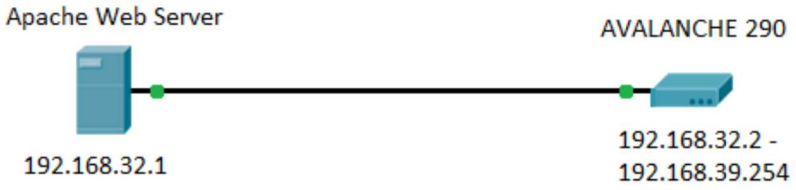

Fig. 2 Topology of network experiments

Model 1 – Middle

In this case, the load factor ρ of the system obtains values from the range of 0.3 to 0.8. The RM falls within this range and the load factor of system ρ reaches the average value of 0.546. The results of the experiments of the model M/G/1/50 with the parameters representing the middle load (ML) are shown in Tab. 1 and are written in line with ML index.

The symbols of the parameters of RM models:

λ – the intensity of arrival requests [arrival requests/sec],
μ – the intensity of service requests [serviced requests/sec],
ρ – the load factor of system [regular range from 0 to 1],
L – the average number of requests in the system,
χ^2 – the parameter of Pearson's chi-squared test,
P_B – the probability of request denial,

Dif – the difference of the number of received requests,
R – the percentage of repeated messages [%],
E – the presence of log errors in Apache web server [% of errors].

Tab. 1 Comparison of Values of Attack Identifiers for Reference Model (RM) and Middle Load (ML), Boundary Load (BL) and the Model Attack (MA)

Model	λ	ρ	L	χ^2	P_B	*Dif*	R	E
RM	1.664	0.546	0.902	3.072	0.353	1.341	0	0
ML	1.561	0.519	0.822	4.333	0.341	1.432	0	0
BL	2.657	0.832	3.03	0.875	0.454	1.806	0	36.6
MA	237.0	43.85	—	152.61	0.975	261.200	38.9	100.0

Model 2 – Boundary Load

The model boundary load (BL) expresses the state when the load factor of the system ρ obtains values ranging from 0.8 to 1. The model M/G/1/50 for the case of boundary load is the next mode of the designed detection algorithm. The results are compared with the RM and are written in Tab. 1 in line with BL index.

In the first alternative block of the first level of decision (1st Level Decision – 1st LD) in Fig. 1, the request arrival intensity of the present boundary model is compared to the initial boundary interval of risk. The intensity λ of the model boundary situation about the size of 2.657 requests per second does not exceed initial boundary interval of risk that is defined as 50 requests per second. As a result, the block 1st LD does not indicate a dangerous situation. The comparison of intensity values of the incoming traffic for the RM and the boundary load (BL) are shown in Tab. 1.

In the second alternative block of the first level of decision (1st LD), the parameters of the RM and the present model of boundary load (BL) are compared, as shown in Tab. 1. In this case, we observe differences between the compared models and the increase of the request arrival intensity λ, as well as the increase of the intensity of load factor of system ρ that has the average value of 0.832 for a model boundary situation. The index of the load increase of the system is also the parameter defining the average number of requests in the system in the optional time L and also the size of request denial probability P_B.

By comparing the values of the model that emulates the situation of the boundary load (BL) to the values of the RM, it can be observed that the allowed variances do not exceed the middle value of the load factor $\rho = 0.832$. The final decision of the designed algorithm is *the declaration about the absence of attack* that is based on the comparison of selected indicators of attack from Tab. 1.

Model 3 – Attack

And finally, the model of attack (MA) is an emulation of the application of HTTP GET Flood attack that is focused on the monitored WEB server. In this case, it is not possible to fulfil the condition and to accept the hypothesis that the request input flow in this modelling situation obtains exponential distribution. This discovery indicates a non-standard change of request input flow in the network interface of the WEB server. The

index of the loss of exponential character χ^2 of input requests in the designed detection algorithm is considered as supporting the identifier of attack. It is independent from the queuing theory.

In the first alternative block of the first level of decision (1st LD) of the designed detection algorithm, we can compare the intensity of arrival requests of the present model with the interval risk and saturation level of the monitored WEB server. In this case, the intensity of arrival requests on average 237.031 requests per second exceeds the initial boundary interval of risk, which was defined by the dimension of 50 requests per second. This intensity of arrivals also exceeds the saturation level of the monitored WEB server, defined by the dimension of 67 requests per second. The result of the first alternative block of the first level of decision (1st LD) is therefore an indication of a high-danger situation. It results in a significant overload of the established service capacity of the WEB server.

In the second alternative block of the first level of decision (1st LD), the parameters of the RM and the present model of load from Tab. 1 are compared.

The explicit index of attack is the probability of blocking or request denial expressed by parameter P_B. In this case of model situation, the presence of attack achieves the average dimension of 97.55 %, which expresses a high probability of failure.

Another important indicator of the attack presence is the average difference in the number of received requests in individual seconds of test *Dif*. The difference between the compared models expressed in this way clearly indicates a very fast change of the input flow, which is with high probability the cause of malicious activity of the attack flow. The extreme changes and anomalies also occur in comparison of percentage of retransmitted request quantity in the input flow of incoming requests. In the case of reference load, retransmission of unserved requests does not occur. On the contrary, such a state occurs when there is a fully reserved waiting queue in the server and other requests are coming with intensity λ higher than intensity μ of the server service. The requests that cannot be stored in the waiting queue due to its overflow are thrown out. The consequences are repetitions of requests (messages) for dropped requests (messages) in the server. In that case the model represents the presence of attack and it incurs up to 38.914 % retransmissions. In the case of the designed detection algorithm, the percentage of tolerated retransmissions is defined as 15 %. If this value is exceeded, the algorithm indicates the state of danger and increased receiving of requests causing system overload with resulting request denial.

The last model completely exceeds the permitted values of dissimilarities of the selected detection parameters. Owing to this, the result in the third alternative block of the second level of decision (2nd Level Decision – 1st LD) of the designed detection algorithm is *the declaration about the presence of attack.*

5. Conclusion

On the basis of the analysis of the well-known attacks of DDoS type on the application layer and their prevention, we designed an original universal algorithm, described in Fig. 1 and in detail in [16].

The submitted article provides the results of a complex theoretical analysis of the problem of attacks of DoS and DDoS types on the Internet, as well as the decisions about their detection with specific focus on the application layer.

The function of the new original detection algorithm of attacks on WEB server was described theoretically and subsequently verified by a series of experiments in the form

of studies of 3 model cases of traffic load. The results of the designed detection algorithm are presented in Tab. 1 and evaluated above.

In the RM, ML and BL models (when load factor of system ρ is in the interval of 0 to 1) parameters λ, μ, L, χ^2, P_B, *Dif*, *R*, *E* are in regular boundaries. *This is the declaration about the absence of attack.* When ρ is more than 1, the parameters are rapidly growing. *This is the declaration about the presence of attack.*

Acknowledgement

The work is supported by the Scientific Grant Agency of the Ministry of Education, Science and Research of Slovak Republic and the Slovak Academy of Sciences under VEGA project No. 1/0427/15.

References

[1] LIN, D. *Network Intrusion Detection and Mitigation Against Denial of Service Attack* [Technical Report]. University of Pennsylvania, 2013. [cited 2017-03-04]. Available from: <http://repository.upenn.edu/cgi/viewcontent.cgi?article=2027&context=cis_reports>.

[2] KAUR, P., KUMAR, M. and BHANDARI, A. A Review of Detection Approaches for Distributed Denial of Service Attacks. *Systems Science & Control Engineering,* 2017, vol. 5, no. 1, p. 301-320. DOI 10.1080/21642583.2017.1331768.

[3] JUNG, J., KRISHNAMURTHY, B. and RABINOVICH, M. Flash Crowds and Denial of Service Attacks: Characterization and Implementations for CDNs and Web Sites. In *11th International Conference on World Wide Web*, Honolulu: ACM, 2002, p 293-304. DOI 10.1145/511446.511485.

[4] FERGUSON, P. and SENIE, D. *Network Ingress Filtering: Defeating Denial of Service Attacks which Employ IP Source Address Spoofing*. Network Working Group, 2000, 10 p. DOI 10.17487/RFC2827.

[5] PATRIKAKIS, CH., KALAMARIS, T. and KAKAVAS, V. Performing Integrated System Test Using Malicious Component Insertion. *Electronic Notes in Theoretical Computer Science,* 2003, vol. 82, no. 6, p. 11-21. DOI 10.1016/S1571-0661(04)81021-1.

[6] ANKALI, S.B. and ASHOKA, D.V. Detection Architecture of Application Layer DDoS Attack for Internet. *International Journal of Advanced Networking and Applications,* 2011, vol. 3, no. 1, p. 984-990.

[7] CHEN, C.M., OU, Y.H. and TSAI, Y.C. Web Botnet Detection Based on Flow Information. In *IEEE Symposium on Security and Privacy – ISC*, Tainan: IEEE, 2010. DOI 10.1109/COMPSYM.2010.5685482.

[8] DAS, D., SHARMA, U. and BBHATTAHCARYYA, D.K. Detection of HTTP Flooding Attacks in Multiple Scenarios. In *International Conference on Information Communications and Computing & Security – ICCCS,* Rourkela: ACM, 2011, p. 517-523. DOI 10.1145/1947940.1948047.

[9] BHATIA, S., MOHAY, G., TICKLE, A. and AHMED, E. Parametric Differences between a Real-world Distributed Denial of Service Attack and Flash Event. In

Sixth International Conference on Availability, Reliability and Security – ARES, Vienna: IEEE, 2011, p. 210-217. DOI 10.1109/ARES.2011.39.

[10] ALMGREN, M. and LINDQVIST, U. Application-integrated Data Collection for Security Monitoring. In *International Workshop on Recent Advances in Intrusion Detection – RAID 2011*, Menlo Park: Springer, 2011 p. 22-36.

[11] PRABHA, S. and ANITHA, R. Mitigation of Application Traffic DDoS Attacks with Trust and AM Based HMM Models. *International Journal of Computer Applications*, 2010, vol. 6, no. 9, p. 26-34. DOI 10.5120/1101-1443.

[12] BAKER, F. and SAVOLA, P. *Ingress Filtering for Multihomed Networks*. Network Working Group, 2004, 16 p. DOI 10.17487/RFC3704.

[13] KEHE, W., TONG, Z., WEI, L. and GANG, M. Security Model Base on Network Business Security. In *International Conference on Computer Technology and Development*, Kota Kinabalu: IEEE, 2009, vol. 1, p. 577-580. DOI 10.1109/ICCTD. 2009.160.

[14] HOLMES, D. *2016 DDoS Attack Trends*. Seattle: F5 Networks, 10 p. [cited 2017-03-04]. Available from: <https://f5.com/Portals/1/PDF/security/2016_DDoS_Attack-Trends.pdf>.

[15] DENER, M. and BAY, O.F. Practical Implementation of an Adaptive Detection-Defense Unit against Link Layer DoS Attacks for Wireless Sensor Networks. *Security and Communication Networks,* 2017, p. 9. DOI 10.1155/2017/1531928.

[16] ĎURČEKOVÁ, V. *Detection of Attacks Causing Denial of Services* [PhD Thesis]. Žilina: University of Žilina, 2014.

[17] The Apache Software Foundation. *Apache HTTP Server Documentation*. [cited 2017-03-04]. Available from: <http://httpd.apache.org/docs/2.2/mpm.html 2012>.

[18] ČEPČIANSKY, G. and SCHWARTZ, L. *Stochastic Processes with Discrete States*. Balti: LAP Lambert Academic Publishing, 2013, 117 p. ISBN 978-3-659-38320-5.

[19] HANULIAK, I. and HANULIAK, P. Performance Evaluation of Iterative Parallel Algorithms. *Kybernetes*, 2010, vol. 39, no. 1, p. 107-126. DOI 10.1108/0368492 1011021309.

[20] HANULIAK, M. and HANULIAK, I. To the Correction of Analytical Models for Computer Based Communication Systems. *Kybernetes*, 2006, vol. 35, no. 9, p. 1492-1504. DOI 10.1108/03684920610688504.

[21] HANULIAK, J. and HANULIAK, I. To Performance Evaluation of Distributed Parallel Algorithms. *Kybernetes*, 2005, vol. 34, no. 9/10, p. 1633-1650. DOI 10.1108/03684920510614858.

[22] DOLEV, S., KATE, M. and WELCH, J.L. A Competitive Analysis for Retransmission Timeout. In *Proceedings of the 15th International Conference on Distributed Computing Systems*, 1995, p. 450-455.

[23] PINTER, T. and KULČAR, L. *Numerical and Statistical Methods in Astronomy (in Slovak)*. Hurbanovo: Slovenská ústredná hvezdáreň, 2006.