P.Jagannadha Varma, P.Suresh Babu, A.Yugandhara Rao / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 3, May-Jun 2012, pp. 550-553

A Group Testing Based Approach for Detecting Application Denial of Service Attacks

P.Jagannadha Varma*, P.Suresh Babu, A.Yugandhara Rao*****

*(Department of CSE, Kaushik College OF Engg., Visakhapatnam-531163)
** (Professor, Department of CSE, Kaushik College OF Engg., Visakhapatnam- 531163)
** (Asst Professor, Department of CSE, LIET, Vizianagaram-535005)

ABSTRACT:
Detecting the Application DoS attacks is a new class of DoS attack, which aims at disrupting the application service rather than depleting the network services has emerged as a severe threat to Internet Security. Detection and prevention of these attacks are harder compared to classic dos attacks. These attacks have high similarity with legitimate traffic so tracing the attack origin is more difficult. We proposed a new novel Group testing(GT) based approach deployed on back-end servers, which provide short detection delay and low false positive/negative rate. Based on this framework we propose a two mode detection mechanism using some dynamic threshold for identifying the attackers efficiently. The focus of this work lies in the detection algorithms proposed and the corresponding theoretical complexity analysis.

Keywords: Application DoS attack, Denial of Service attack (Dos), Group testing (GT), and Internet Security.

1. Introduction:

Internet has become a integral part of everybody's life. Our daily routines like banking, transport, health, etc., are dependent upon internet partially or completely. People connecting to internet have increasing by exponential rate over the few years. With approximately 2.1billion people connecting to internet and with 1 billion searches on goggle [1]. Any inconvenience in these services can cost us significantly. Denial of service attacks(Dos) which aims at legitimate users, clients, customers from successfully accessing the internet has posing a serious challenge to the network security. Application Dos attacks is a new class of Dos attacks which exploits the flaws in either application design or its implementation. These attacks are harder to trace than Classical Dos attacks because

- These attacks do not consume huge amount of bandwidth.
- These target on creating bottlenecks and resource limitation within application by focusing on weakest link in the application.
- These attacks normally use https as their transport to hide their true origin.

Some of the Application Dos attacks we have seen are Jolt2 an attack targeting Microsoft Systems it sends a continuous stream of ICMP ECHO_REPLY fragments with specially tuned fragmentation parameters to the attacked host and Microsoft IIS suffering from URL parsing bug the decoding of escape sequences in the URL strings was implemented very inefficiently submitting long strings with large amounts of escape characters effectively stopped web server [2] and there are numerous attacks on apache web servers like Apache MIME flooding and Apache Sioux attack [3]mainly these Application Dos attacks are targets towards web servers. Vulnerabilities in web application can allow attacks to exhaust available resources and there by deny access to legitimate users. It is observed that attacks can be indentified much faster if we can find out them by testing them by group rather than testing one by one. This method is called Group testing. It aims at detecting the defective items in a large population with minimum number of tests it was proposed in World war II and was used successfully in medical testing, Computer Networks and Molecular Biology[4][5][6]. The advantages of GT lie in its prompt testing efficiency and fault-tolerant decoding methods [7].

In a system viewpoint, our defense scheme is to embed multiple virtual servers within each physical back-end server and map these virtual servers to the testing pools in GT, then assign clients into these pools by distributing their service requests to different virtual servers. By periodically monitoring some indicators (e.g., average responding time) for resource usage in each server and comparing them with some dynamic thresholds, all the virtual servers can be

judged as "safe" or "under attack." By means of the decoding algorithm of GT, all the attackers can be identified. Therefore,
the biggest challenges of this method are threefold: 1) How to construct a testing matrix to enable prompt and accurate detection. 2) How to regulate the service requests to match the matrix in practical system. 3) How to establish proper thresholds for server source usage indicator, to generate accurate test outcomes.

2. Approaching methods

In our proposed system we use the concept of group testing where we test the clients as a group rather than testing individually. The classic GT model consists of t pools and n items. This model is represented by t x n matrix M. Where rows represent the pools and columns represent the items. An entry M [i j]=1 if and only if ith pool contains the jth item. Otherwise M[i j]=0.

$$M=\begin{bmatrix}1&0&1&1&0\\0&1&0&0&1\\0&0&0&1&0\\0&0&1&0&1\\0&1&0&1&1\end{bmatrix} \xrightarrow{\text{Testing}} v=\begin{bmatrix}1\\1\\0\\0\\1\end{bmatrix}$$

Fig 1: Binary testing matrix M and test out V

The t-dimensional binary column vector V is denotes the test outcome of the t pools, where

If v[i]=1 received a malicious request from at least one attacker if v[i]=0 all clients assigned to virtual server are legitimate.

Two traditional GT methods are adaptive and non-adaptive. Adaptive methods or sequential GT use the results of previous tests to determine the pool for the next test and complete the test within several rounds. While non-adaptive GT methods employ d-disjunct matrix to run multiple tests in parallel and finish the test within only one round.

3. Size Constraint Group Testing

Each testing pool is mapped to a virtual server within a back-end server machine. Although the maximum number of virtual servers can be extremely huge, since each virtual server requires enough service resources to manage client requests, it is practical to have the virtual server quantity (maximum number of servers) and capacity (maximum number of clients that can be handled in parallel) constrained by two input parameters K and w, respectively. Therefore, the traditional GT model is extended with these constraints to match our system setting. We consider the case each client provided with a non spoofed ID which is used in identifying client during our detection period. Attackers are assumed to launch the application service request either at high interarriaval rate or high workload or even both. By periodically monitoring the average response time to service requests and comparing them with the specific threshold values fetched from a legitimate profile each virtual server is associated with a negative or positive outcome by this we can identify a attacker from the pool of legitimate users.

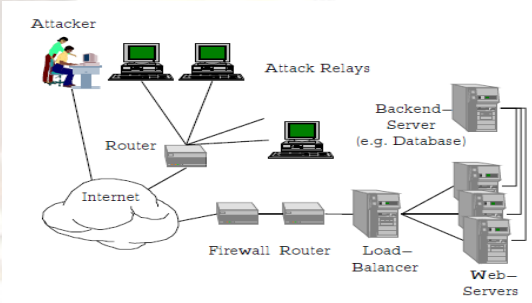

Figure 2: Overview of Attack Scenario

The above figure describes the architeture of web application and its infrastrucure .the requests are generated from users which consists of both legitamate and attackers are send to the proxy server via router The front end proxy server works as load balancer servers and distributes the requests to the back end servers depending on their usage The backend server cylces between two states which are refered as NORMAL mode and DANGER mode

If the estimated response time (ERT) of any back end server exceeds profile based threshold the system transfer to danger mode

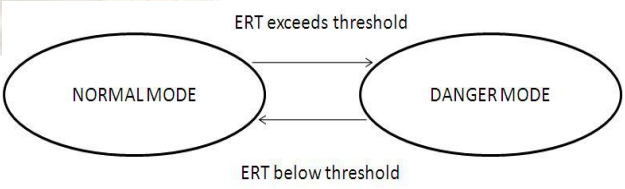

Figure 3: Two state diagram of a system

The ERT value can be calculated using the formulae

ERT= (1-α) ERT+ αART

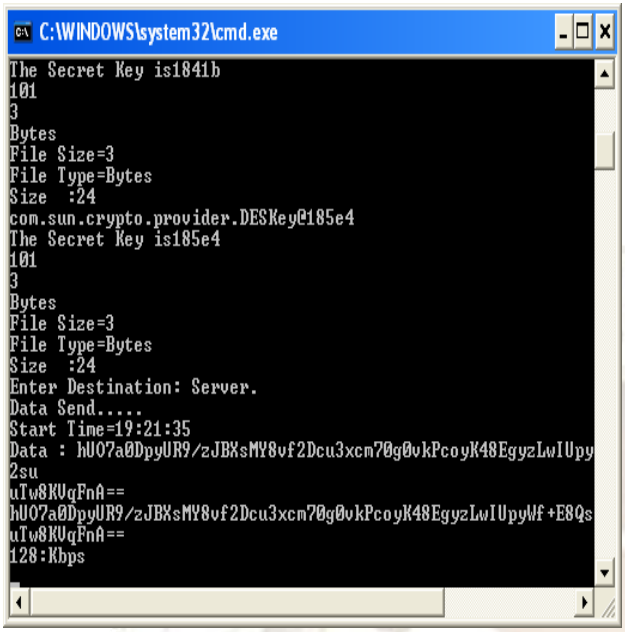

Figure 4: DOS attack calculation.

If any virtual server has ERT>μ+4σ (μ and σ refer as expected value and standard deviation of the ART distribution). The backend server is probably under attack and it transferred to danger mode for detection. After the detection it is returned to the normal mode.

4. Related Work in DoS Detection

Numerous defense schemes against DoS have been proposed and developed [8], which can be categorized into network-based mechanisms and system-based ones.

Existing network-based mechanisms aim to identify the malicious packets at the intermediate routers or hosts [9],[10], by either checking the traffic volumes or the traffic distributions. However, the application DoS attacks have no necessary deviations in terms of these metrics from the legitimate traffic statistics; therefore, network-based mechanisms cannot efficiently handle these attack types.

On the other hand, plenty of proposed system-based mechanisms tried to capture attackers at the end server via authentications [12], [13] or classifications [11], [14]. Honey pots [13] are used to trap attackers who attempt to evade authentications, and can efficiently mitigate flood attacks. However, this mechanism kind relies on the accuracy of authentication. Once the attacker passes the authentication, all the productive servers are exposed and targeted.

5. CONCLUSION

A novel technique for detecting application DOS attack by means of a new constraint-based group testing model. Motivated by classic GT methods, three detection algorithms were proposed and a system based on these algorithms was introduced. Theoretical analysis and preliminary simulation results demonstrated the outstanding performance of this system in terms of low detection latency and false positive/negative rate. Our focus of this paper is to apply group testing principles to application DOS attacks, and provide an underlying framework for the detection against a general case of network assaults, where malicious requests are indistinguishable from normal ones. For the future work, we will continue to investigate the potentials of this scheme and improve this proposed system to enhance the detection efficiency. Some possible directions for this can be:

1. The sequential algorithm can be adjusted to avoid the requirement of isolating attackers.

2. More efficient d-disjunct matrix could dramatically decrease the detection latency, as we showed in the theoretical analysis. A new construction method for this is to be proposed and can be a major theoretical work for another paper.

3. The overhead of maintaining the state transfer among virtual servers can be further decreased by more sophisticated techniques.

4. Even that we already have quite low false positive/ negative rate from the algorithms, we can still improve it via false-tolerant group testing methods. This error-tolerant matrix has great potentials to improve the performance of the PND algorithm and handle application DOS attacks more efficiently

6. FUTURE ENHANCEMENT

We will continue to investigate the potentials of this scheme and improve this proposed system to enhance the detection efficiency. The sequential algorithm can be adjusted to avoid the requirement of isolating attackers. More efficient d-disjunct matrix could dramatically decrease the detection latency, as we showed in the theoretical analysis. A new construction method for this is to be proposed and can be a major theoretical work for another paper. The overhead of maintaining the state transfer among virtual servers can be further decreased by more

sophisticated techniques. Even that we already have quite low false positive/ negative rate from the algorithms, we can still improve it via false-tolerant group testing methods.

REFERENCES

[1] INTERNET SOCIETY REPORT www.internetworldstats.com/stats.htm
[2] Microsoft security bulletin (MS00-29) patch for "MYRAID ESCAPED CHRACTERS VULNARABILITY"
[3] Apache dos attack www.securityfocus.com
[4] "Approximation Algorithms of Non unique Probes Selection for Biological Target Identification "by M.T.Thai, P.Deng
[5] Two Models of Non Adaptive Group testing for designing screening experiments by D.C Torney
[6] D.Z. Du and F.K. Hwang, Pooling Designs: Group Testing in Molecular Biology World Scientific, 2006
[7] M.T. Thai, P. Deng, W. Wu, and T. Znati, "Approximation Algorithms of No unique Probes Selection for Biological Target Identification," Proc. Conf. Data Mining, Systems Analysis and Optimization in Biomedicine, 2007.
[8] J. Mirkovic, J. Martin, and P. Reiher, "A Taxonomy of DDoS Attacks and DDoS Defense Mechanisms," Technical Report 020018, Computer Science Dept., UCLA, 2002.
[9]. Service Provider Infrastructure Security, "Detecting, Tracing, and Mitigating network-wide Anomalies,"www.arbornetworks.com, 2005.
[10] F. Kargl, J. Maier, and M. Weber, "Protecting Web Servers from Distributed Denial of Service Attacks," Proc. 10th Int'l Conf. World Wide Web (WWW '01), pp. 514-524, 2001.
[11] S. Ranjan, R. Swaminathan, M. Uysal, and E. Knightly, "DDos- Resilient Scheduling to Counter Application Layer Attacks under Imperfect Detection," Proc. IEEE INFOCOM, Apr. 2006.
[12] S. Kandula, D. Katabi, M. Jacob, and A.W. Berger, "Botz-4-Sale: Surviving Organized DDoS Attacks That Mimic Flash Crowds,"Proc. Second Symp. Networked Systems Design and Implementation (NSDI), May 2005.
[13] S.M. Khattab, C. Sangpachatanaruk, D. Mosse, R. Melhem, and T. Znati, "Honey pots for Mitigating Service-Level Denial-of- Service Attacks," Proc. Int'l Conf. Distributed Computing Systems (ICDCS '04), 2004.
[14] F. Kargl, J. Maier, and M. Weber, "Protecting Web Servers from Distributed Denial of Service Attacks," Proc. World Wide Web Conf., pp. 514-524, 2001.