International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

A Survey Paper on an On-Line Intrusion Detection Approach to Identify Low-Rate Dos Attacks

Dipali Vaidya[1], Prof. Sonal Fatangare[2]

[1]M.E.Student, Department of Computer Engineering, RMD Sinhgad School of Engineering, Pune, India
[2]Asst Prof., Department of Computer Engineering, RMD Sinhgad School of Engineering, Pune, India

Abstract: ***High rate Denial of service attacks, happen in relatively small amount of time, low rate DoS attack consume resources relatively at slower rate but cause eventual crash of the service providing server. The problem of detection of "Slow Denial of Service" attacks within small time is a challenging task because the approaches either have scalability limitations due to inherent computational costs or these approaches lack timely detection. To overcome this problem, here analysis focuses on the quantity of data directed from the transport layer to the application layer. Frequency of such transfers is taken as input and analysis has been done to identify patterns that show possible Low rate DoS attacks within expected time. It has been discovered in frequency domain analysis that patterns differ between legitimate and anomalous transactions in time horizon proving that fast detection is possible.***

Keywords: denial of service, anomaly detection, Fourier transform, slow dos attack.

1. Introduction

Denial of Service (DoS) threats work so as to make services over network or web unavailable for genuine users. Based on how attacks work, DoS attacks can be categorized into "High Rate Attacks" where the server is overwhelmed with requests through excessive number of service requests. Apart from methods that work behind the scene, various methods like requesting "Capta Code" or "Cognitive" answers from end users have been effective methods of identifying and trapping such attacks. In second category of DoS attacks, also called low rate DoS attacks or Slow DoS Attacks (SDAs); the attacker tries to continuously consume server resources and lower bandwidth; creating impression of legitimate traffic. SDAs work to generate incongruities in protocols in turn reducing concurrent connections by the server - that cases eventual crash of server daemons.

As bandwidth used by SDAs is very less and these attacks take considerable time to show effect. It is important to have detection mechanism that works faster while not consuming significant amount of resources. One of the common detection methodologies for SDAs monitors application layer by inspecting payload along with protocol state machine. This method leads to heavy consumption of resources in turn becoming less scalable solution. In other mechanisms that monitor transport layer, anomalies for TCP flags. These and other approaches do work in certain ways but these do not work in timely detection. One of the reasons is transport flags get switched at end of the connection detection and till connection is closed, detection cannot happen. This is why there is need of a detection mechanism that is scalable, efficient and fast to detect SDAs.

2. Literature Survey

A number of previous researchers have proposed LDos detection methodology but they are failing to detect LDos attack in small amount time. The proposed systems give some metrics which detects attack in desirable time.

In 2009, S. Dolev, Y. Elovici [1] It Limits the Traffic rates under desired proportion to find out LDoS attack. Two algorithms are present here.The first algorithm filters traffic in polynomial time. In the second algorithm only local traffic is considered. In 2009 Y. Xie and S. zheng Yu [2] describes anomaly detector based on hidden semi-Markov model is proposed to describe the dynamics of Access Matrix and to detect the attacks. In 2010 R. Braga, E. Mota, and A. Passito [3] proposed a lightweight method for DDoS attack detection based on traffic flow features, in which the extraction of such information is made with a very low overhead compared to traditional approaches. This is possible due to the use of the NOX platform which provides a programmatic interface to facilitate the handling of switch information. In 2011 Rejo Mathew and Vijay Katkar [4] proposed a Software based LDOS attack detection mechanism which could be integrated with existing Intrusion Detection system and does not require any change in existing infrastructure and protocol. In 2013 D. Moustis and P. Kotzanikolaou [5] proposed a method which considered only limited connection to detect LDoS attack.

3. Slow Rate DoS Attack Type

3.1 Slow Loris

The Slow Loris attack represents one of the most-known slow DoS threats. In this case attacker sends legitimate but incomplete HTTP requests to the victim/server to make server wait for end of the requests. Subsequently attacker never sends the request making the server wait for long. So in general, it may be seen as the SYN flood attack working at the application layer.

3.2 SlowReq

The SlowReq also forces the server to an endless wait. Here in this case also, payload is not made compliant the specific protocol deliberately while keeping size of the payload minimal (e.g. a space). The server waits for complete

message or wastes processing on telling that the message is not valid. Both way, the server resources get wasted and legitimate users cannot use the system effectively.

3.3 Slow Read

The Slow Read attack works by sending legitimate HTTP requests to the server at usual rate while aiming to keep connections alive. As purpose of the attack is to maintain connections alive, it slowly reads the replies received by the server. Server assuming that client is deprived of resources, releases data slower and ends up holding data for much more time that it should be. Mocking slow read from client side is possible by specifying a small client side reception buffer-in the initial SYN packets sent to the victim during the connections establishment. The server keeps holding resources to crash ultimately.

3.4 SlowNext

Slow Next exploits the protocol connection persistency. To minimize overhead of connection chatter, protocols use persistency so that clients can send more than a single request over the same connection. Slow Next works by seizing a specific amount of connections. For each connection, a legitimate request is sent to the server at the rate available along the end to-end path. In turn, a legitimate response is sent by the server at the same rate making server hold resources.

4. System Architecture

4.1 Existing System

One of the biggest challenges in this case is to find right set of features that are available without being too intrusive and resource eating on server side. For this purpose, we explored various features that are available. Discussion below tries to provide journey towards zeroing in on the right feature that would have a good correlation with existence of SDA.

Signal Flow Related Metric: Signal Flows related metrics at pocket level (e.g. the number of bytes sent or received at socket) does have correlation with the traffic happening between client and the server. Problem with socket level features is these can be too resource heavy and can increase implementation complexity, so we want to stick to server level feature.

TCP FIN/RST Flags: These flags arise when connections are terminated, as we want to get feedback before end of the connection - especially with Slowreq and SlowNext in mind - these flags are not right bet.

TCP URG Flag: URG flag is one of the most popular flags in use with malicious intention. In hunt for originality of the solution we are keeping blind eye on URG flag.

TCP SYN and PSH Flags: Use of SYN and PSH flags are very simple. In case of legitimate traffic, you would see few number of SYN is set in random amount of time. But in case of attack number of SYN flag is set in small amount of time. In case of legitimate user, PSH flag is also set in random amount of time but when attack is going on then PSH flag is set in specific amount of time.But these flag could not work on SSh protocol.

Bytes Received by Server: Unlike "Signal Flow Related Metric" that is pocket level; number of pockets/bytes received gets evaluated at server level. As SDAs aim to minimize the traffic, this metric does not show good correlation.

4.2 System Architecture

As depicted in Overall Approach diagram, we will have systems connecting to the server as legitimate users or an attacker. We can use a single system to mock itself as attacker or a legitimate user or even mock as multiple users. We will have a service that will run continuously on server reading data from TCP/IP registers.

This allows server log processor to pick up data for current OH even before Current OH is over. Server Log processor again is a service that does cleanup of data for current and previous OHs. Basic comparison and simple detections are carried out at this stage. For complex detection, transformation process is executed that does Fourier transform for current OH and Previous OH. It also calculates Mutual Information Metrics. This process writes alter file in case any alert is generated. Alert processing can start on based on presence of the file or presence of data in the alert file.

5. Mathematical Model

5.1 Detection Mechanism

Numbers of received packets on web server are used for anomaly-based analysis of web traffic. Anomaly-based detection may be more adoptive vis. a vis complicated methods derived from machine learning. Since here we are hunting for variance in flow statistics; these techniques also can provide good detection rates. Detection needs to happen in a time window where we observe the patterns. We call the time window under monitor Observation Horizon (OH). Two OHs are defined, current OH for the window that is being monitored and previous OH for the window that was monitored last. So both windows look at temporal behavior of the signal S (t) over time. Signals are sampled every second so that comes our sampling frequency - this is in line with law of sampling as we do not expect SDAs to occur in sub-second time frame. Difference between the signal in

current OH $s_0(t)$ and previous OH $s_{-1}(t)$ is compared to trap any variance.

5.2 Metrics

Metrics are able to find out anomalies when attack is going on. Two metrics are considered here.First is simple average method and second is Mutual Information.

1) Simple Average

Simple average of current signals S0 (t) and previous signal S-1(t) are considered.

$$\Delta^{l,a}_{E[.]} = (E[s_0(t)] - E[s_{-1}(t)])^2 \qquad (1)$$

2) Mutual Information

In Mutual Information FFT (Fast Fourier Transform of current signals S0 (t) and previous signal S-1(t) are considered.

$$\Delta^{l,a}_{I(F(.))} = I(F(s_0(t)), F(s_{-1}(t))) \qquad (2)$$

6. Conclusion

TCP flags SYS and PSH are considered to detect Low rate denial of service attack, but these flags are not good for secureshell (SSh) protocol. The number of bytes send from user to server are also not considered because in slow DoS attack throughput is very low, legitimate throughput and DoS throughput are nearly same. Trivial Detection (simple Average) method is very simple but in that threshold of legitimate traffic is required in advance, this is not possible always, so this method is not acceptable. Another method is mutual information in which analysis of spectral feature of traffic is done in small observation horizon. Payload inspection is not considered therefor computational cost is low.

References

[1] S. Dolev, Y. Elovici, A. Kesselman, and P. Zilberman, “Trawling traffic under attack, overcoming ddos attacks by target-controlled traffic filtering,” in Parallel and Distributed Computing, Applications and Technologies, 2009 International Conference on, Dec 2009, pp. 336–341.

[2] Xie and S. zheng Yu, “Monitoring the application-layer ddos attacks for popular websites,” Networking, IEEE/ACM Transactions on, vol. 17,no. 1, pp. 15–25, Feb 2009.

[3] R. Braga, E. Mota, and A. Passito, “Lightweight ddos flooding attack detection using nox/openflow,” in Local Computer Networks (LCN), 2010 IEEE 35th Conference on, Oct 2010, pp. 408–415

[4] Rejo Mathew, Vijay Katkar, Software based LDOS attack detection mechanism, International Journal computer applications 2011

[5] D. Moustis and P. Kotzanikolaou, “Evaluating security controls against http-based ddos attacks,” in Information, Intelligence, Systems and Applications (IISA), 2013 Fourth International Conference on, July 2013, pp. 1–6.

[6] H. Liu and M. S. Kim, “Real-time detection of stealthy ddos attacks using time-series decomposition,” in Communications (ICC), 2010 IEEE International Conference on, May 2010, pp. 1–6.

[7] S. Bhatia, G. Mohay, A. Tickle, and E. Ahmed, “Parametric differences between a real-world distributed denial-of-service attack and a flash event,” in Availability, Reliability and Security (ARES), 2011 Sixth International Conference on, Aug 2011,pp.210–217.

[8] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, “An overview of ip flow-based intrusion detection, ”Communications Surveys Tutorials, IEEE, vol. 12, no. 3, pp. 343–356, Third 2010.

[9] T. Benmusa, D. J. Parish, and M. Sandford, “Detecting and classifying delay data exceptions on communication networks using rule based algorithms,” International Journal of Communication Systems, vol. 18, no. 2, pp. 159–177, 2005. [Online]. Available: http://dx.doi.org/10.1002/dac.694.

[10] W. Ellens, P. uraniewski, A. Sperotto, H. Schotanus, M. Mandjes, and E. Meeuwissen, “Flow-based detection of dns tunnels,” in Emerging Management Mechanisms for the Future Internet, ser. Lecture Notes in Computer Science, G. Doyen, M. Waldburger, P. eleda, A. Sperotto,and B. Stiller, Eds. Springer Berlin Heidelberg, 2013, vol. 7943, pp.124–135.

[11] M. Aiello, E. Cambiaso, M. Mongelli, G. Papaleo, ”An On-Line Intrusion Detection Approach to Identify Low-Rate DoS Attacks” IEEE Security Technology (ICCST), 2014 International Carnahan Conference