International Conference on Cyber Security and Computer Science (ICONCS'18),
Oct 18-20, 2018 Safranbolu, Turkey

Anomaly-Based Detection of Non-Recursive HTTP GET Flood DDoS Attack

Mohammed SALIM[1] and Seçkin ARI[1]

[1] Sakarya University, [1]Sakarya/Turkey, muhammad.salim@ogr.sakarya.edu.tr
[1]Sakarya University, [1]Sakarya/Turkey, ari@sakarya.edu.tr

Distributed Denial of Service (DDoS) attacks are serious threat to any online service on the internet. In contrast to other traditional threats, DDoS HTTP GET flood attack can exploit legitimate HTTP request mechanism to effectively deny any online service by flooding the victim with an overwhelming amount of unused network traffic. This paper introduces a new anomaly-based technique for discriminating DDoS HTTP GET requests and legitimate requests using a combination of behavioral features. The key features are Diversity of the requested objects, requesting rates for all the requested objects, and request rate for the requested object with the most frequency. These features are selected as the key measurements that will be analyzed and processed for developing the proposed detection technique. During the evaluation process, sub set of the UNB ISCX IDS 2012 evaluation dataset representing anomalous traffic, in addition to another sub set extracted from the 98 world cup dataset showing legitimate traffic are used to evaluate the proposed method. The evaluation shows that the proposed mechanism does effective detection due to the subtle behavioral dissimilarity between non-recursive attack and legitimate requests traffic.

Keywords - DDoS attack, Anomaly-based detection, behavioral features, Request Rate, URI Diversity, Machine Learning

I. INTRODUCTION

With increasing reliance on internet services, network-based attacks become a major security concern as online services become more vulnerable to these serious attacks. Intruders attempt to saturate online services and make them unavailable by preventing their legitimate users from accessing these services. This type of network-based threat is called Distributed Denial of service or DDoS. This threat is considered by many organizations specialized in computer networks security as one of the most serious security breaches that can bring any site or service offline for hours, days or even weeks. Large number of internet services owned by governments, organizations and well-known commercial companies were victims to such network-based threats. Service providers are the main target of such attacks, followed by service and network infrastructure during last year [i]. More than two thousands of DDoS Attacks are observed worldwide daily by Arbor Networks. According to Verisign, DDoS attacks are responsible for one third of all downtime incidents for online services.

Perpetrators try to exhaust the victim resources by exploiting multiple infected online machines called a botnet to send overwhelming traffic that can take the targeted site offline for a long time. DDoS attack intensity depends on the botnet size. The size of a particular botnet can varies from tens to hundreds of thousands of bots[1]. An intruder has to create a botnet by sending malicious software through emails, websites and social media. These malicious software tools can be controlled and commanded by the botnet master remotely over the internet to launch the DDoS attack from all the exploited machines at the same time.

A. HTTP GET FLOOD ATTACK

HTTP flooding DDoS attacks are forming more than 80 percent of all nowadays DDoS attacks [ii]. HTTP GET flood attack is a DDoS based threat utilizing HTTP application protocol to apply denial of service for a target victim. HTTP GET flood attack overwhelm the victim with volumetric unwanted HTTP requests to jam the victim resources and make their services unavailable. The same way as any DDoS based attack, HTTP GET flood attack can be initiated by starting a distributed malicious script running remotely from the distributed compromised machines or a prepaid botnet. The malicious script utilize their compromised machine resources and start sending HTTP requests to the victim site. After a period of time and according to the attack intensity, the victim will not be able to respond to any new legitimate request as all its resources are exhausted.

This attack can be considered as one of the serious network-based threats because it is totally compliant with the HTTP protocol. Contrasting with simple network-based threats that attempt to saturate victim links using malformed traffic, this subtle attack perfectly looks like legitimate activities requesting a web page or another available resource. Attacker thoroughly mimic legitimate http request to send flood attack. Therefore, signature based intrusion detection systems may not be able to distinguish this anomalous requests from the legitimate requests.

HTTP GET flooding attacks are implemented using two

[i] Network Infrastructure Security Report VI, Arbor Networks Inc.,Q4 2016

[ii] Holmes, David. The DDoS Threat Spectrum. F5 Networks, Inc. 401 Elliott Avenue West, Seattle, WA 98119, (2013)

different methods. Classification of this attack depends on the content of the HTTP request. [ii]. Simple HTTP GET flooding attack represents the first class of the flooding attacks. It repeats requesting a static set of URI addresses over and over. This type of flooding attacks is a very common threat that can hit layer 7 applications and services. In the other hand, Recursive GET flooding threat is a sophisticated type that firstly iterate through the website to retrieve, fetch or parse every URI address that can be requested and then start flooding requests using the parsed URI addresses. Unlike simple HTTP GET floods, recursive HTTP GET floods require doing some homework to retrieve all or part of the victim URI addresses. Networks security infrastructures may apply specific polices violating or mitigating URI crawling which make parsing URI addresses more complicated. Also, HTTP GET flooding attack can request random generated URI addresses. In this paper, Simple HTTP GET flooding attack will be discussed only due to the limitation found in the available datasets.

In 2010, OWASP provided the public with a free tool called OWASP Switchblade [2]. This tool can simulate three various kinds of behaviors related to DDoS attacks. This tool can be installed and controlled directly to start attacking a particular victim. It was developed to warn the OWASP Community of the DDoS threats and security breaches that can hit application Layer. In default configuration, OWASP Switchblade tool can start an HTTP GET attack. Also, it can be utilized to start a targeted DDoS attack by running and commanding this tool from the distributed mastered machines or bots.

B. Paper Organization

In this paper, a new discrimination method is proposed. This method will be used to develop an anomaly-based intrusion detection system attempting to discriminate between HTTP GET flooding traffic and legitimate traffic. The rest of this paper is organized as follow. Section II discusses the HTTP GET Request method and show how this request method can be utilized to mimic legitimate HHTP traffic by attackers intending to strike a particular victim. Section III describes the related work in the area of DDoS attack detection. In section IV, the proposed approach to detect the HTTP GET flood attack is described. Section VIII listed the evaluation datasets. In section IX, evaluation of the proposed approach is carried out. Finally, conclusion and future work is presented in section 0.

II. HTTP GET Request Format

HTTP is an application layer protocol in OSI network model. It's used to transfer web pages with other objects like scripts over networks [3]. HTTP protocol is installed by default with any client's internet browser. In other words client browser completely rely on HTTP protocol that can send requests for objects like HTML files to dedicated servers and represent responses inside the browser while the client is totally unaware of that process. HTTP protocol is a TCP based protocol which means that both of the connecting, client and server, must successfully pass the three way handshaking process in order to be able to initiate the connection and start sending and receiving application based data. Therefore, all requesting machines weather it is a legitimate or infected must have an online address to be able to send request from any site as it's a prerequisite for the TCP three way handshaking. The constructed channel between any client and the dedicated server is called a session or it can be called a stream which can be used effectively to monitor and control the traffic more precisely without affecting any other active sessions.

There are various types of request method. One of the well-known request types is the request method used to fetch a particular object from a specified web server or site. The host or more specifically, the client should send the complete address of the requested object within the sent request. Furthermore, the network address of the hosting server, where the fetch object is stored, should be send within the sent request. This address is known as the absolute path or URI of the request object or resource. For instance, to fetch a particular representation directly from the hosting server of the object identified as "http://www.example.org/where?q=now". The client attempting to fetch that object should create a new TCP connection with the host "www.example.org". Then, that client should send the URI address of that requested object displayed in figure 1 below through the created TCP connection.

```
GET /where?q=now HTTP/1.1
    Host: www.example.org
```

Figure 1: URI and host fields for the formatted HTTP request.

The absolute path cannot be null. For instance, forward slash or "/" is the simplest address referring the site root [3]. In the simplest http GET flood attack, the anomalous requests do not contain absolute path information and only requesting the root page.

III. Related Work

Layer 7 DDoS attack handling a mitigation is a prosperous research field. Researchers attempt to utilize various techniques and algorithms to effectively discriminate this type of attack. Among the effective techniques utilized for application layer DDoS, statistical and machine learning techniques are remarkably found. Spectral analysis and signal processing techniques also were introduced by many researchers as proposed technique for layer 7 DDoS attack. Moreover, session based or flow based improves the detection process significantly.

Authors in [4] introduce a new technique for intrusion detection based on fast entropy and flow analysis. In this introduced method, the request rate for a particular object is analyzed as the key parameter to detect the anomalous traffic and connections. Another example for session monitoring, authors in [5] proposed a defense mechanism against layer 7 DDoS attack scheme. Active connections or flows are monitored and controlled through analyzing a set of features including instant traffic volume and session behavior. Discrimination is provided based on connection behavior. In [6], authors discriminate between normal and anomalous

traffic by analyzing a set of statistics related to different flow-based features. These statistics are processed and analyzed for developing a new detection method which can be analyzed to distinguish anomalous traffic from legitimate traffic. Entropy of source IPs, variation of source IPs, and packet rate are the key parameters in the proposed method.

In [7] and [8] , the proposed defense mechanism uses a data structure representing IP addresses for storing legitimate clients profile and filtering the anomalous bots at an edge router. The proposed mechanism builds the IP address table from the valid IP addresses that correctly create a TCP connection with the server. The table is frequently updated with the most recent IP addresses. Then, during suspicious activities, the IP address table is used to filter incoming requests and only client with source IP address presents in the table is permitted.

Authors in [9] present a new detection method where the anomalous HTTP GET flood requests is discriminated based on the dissimilarities between the behavior of the bots and legitimate users. The proposed mechanism monitor the requested web objects where these requested objects are hashed to a data structure for advances analysis. In the advanced analysis stage, web object with high request rate are more investigated by monitoring the all the source request IP addresses requesting that object. In this stage source IP addresses attempt to request the hashed web object in an anomalous way will be denied.

Clustering method for layer 7 DDoS detection is proposed in [10]. Users' sessions are clustered in order to detect type of user's activities. Four flow-based features are introduced as the main parameters that are analyzed for clustering user activities. These four parameters include sessions-average size of objects requested in the session, request rate, average popularity of all objects in the session, average transition probability. The selected parameters are used to model the legitimate behaviors or profiles which enable discrimination of malicious traffic.

In [11], the statistic called entropy of HTTP GET requests per source IP address is utilized for developing a new detection technique for layer 7 attacks. The extracted feature is converted into a multidimensional space. Finally, ML-based algorithm is applied to model the classifier from the generated multidimensional data that will be used to identify layer 7 attacks. Also, spectral analysis was utilized in [12]. The paper explores the energy distributions of network traffic in frequency domain. Authors claim that normal TCP traffic can be isolated from malicious traffic according to energy distribution properties.

IV. PROPOSED METHOD

In this work, an anomaly-based detection technique is introduced for discrimination between HTTP GET flood attack traffic and legitimate requests traffic. This implies that the proposed mechanism will concentrate on the traffic behavior rather than the signature or the structure of the traffic generated by the http requests. This is due to the complete similarity between malicious and legitimate HTTP GET requests from structural prospect. Therefore, to differentiate between legitimate and illegitimate http traffic, the HTTP GET requests behavior will be monitored and analyzed by extracting the relevant features from the inbound traffic generated by http requests as shown in figure 2.

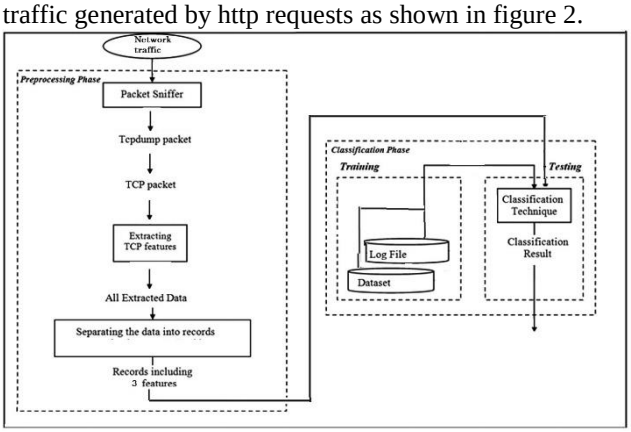

Figure 2: Structure of the introduced system.

Requests generated from HTTP GET flooding attack are sent from the exploited machines. These infected machines are connected and controlled by the same bot net. This means that all requests sent from these machines are sharing common properties and behavior since the same malicious software or tool is used to send such requests from all these machines. As a result, enormous number of requests with high similarity and low diversity among the host and URI addresses fields can be found. The malicious software are uninspired so they do not have the ability or the creativity to choose among the internal retrievable objects or URI addresses from the victim site. Therefore, the overwhelming traffic will often converged to a small static group of URI address(s). On the other hand, the HTTP GET requests generated by the normal or human clients or even search engines will scattered to a large group of URI addresses.

The first and foremost feature is the URI diversity for the incoming requests. Since the malicious tool used to launch the attack from the compromised source machines behaves the same for all the machines mastered by the perpetrator, the request flows originating from the malicious sources tend to be the same. Unlike malicious requests, normal request originating from legitimate users share different behaviors. For example, legitimate requests are distributed and mostly their requested URI addresses have large diversity among them. Second, the request rate for the requested URI address with the maximum frequency among all the requested URI addresses. Normal users tend to request any online resource for limited times during a short period of time, whereas the mastered bots send a massive requests for the same objects over the same TCP connection or stream. This sophisticated parameter can be used to discriminate the anomalous request even having high diversity among the requested URI addresses. Third, request rate per second is also extracted for perfect discrimination.

V. FEATURE EXTRACTION

Figure 3 below describe the steps for extracting the time series data for the selected three parameters from the both offline and online raw network traffic data.

Algorithm 1 Extract input parameters from the online traffic data

Input: *A; // A contains the requested URI addresses*
Output: *http_req_rate, http_req_uri_div, max_requested_uri_per_sec // the selected parameters*
// initialize parameters
http_req_rate ← {};
http_req_uri_div ← {};
max_requested_uri_per_sec ← {};
iterate each sampling interval
// set vector A from the analyzed online traffic data
set A;
// extract number of elements in vector A
set http_req_rate ← {http_req_rate numel(A)};
// extract number of unique elements in vector A
set http_req_uri_div ← {http_req_uri_div numel(unique(A))};
// extract number of occurrences for the most requested URI address element in vector A
ser max_requested_uri ← {max_requested_uri maximum(histcounts(unique(A)))};
end iteration

Figure 3: The selected parameters extracted from network flow.

For traffic discrimination, Linear Support vector machine classifier is used is used to discriminate the traffic based on the selected attributes. The classifier is modeled firstly using the training part of the generated time series dataset as explained in this subsection. Then, the trained model can be used to classify the new or unseen dataset to evaluate the proposed system.

VI. SUPPORT VECTOR MACHINE (SVM)

SVM is a ML-based algorithm. This technique can be learned and trained as a classifier in multi disciplines (e.g. intrusion detection). SVM algorithm was presented by Cortes and Vapnik. For developing and anomaly-based intrusion detection system, SVM algorithm can be utilized for modeling a particular behavior or profile using training data. This model is the core part of the SVM-based classifier which enable the classifier to identify and detect whether a particular instance is an anomalous or normal behavior. The classifier should be provided with the same attributes that where provided in the training phase. The classifier is expected to predict the target class of a particular instance using these attributes. In this algorithm, each feature's value is plotted as a point in k-dimensional space where k is number of generated features in each observation in data space. Value of each feature represents the value of a particular coordinate. Other value(s) of other coordinate(s) are calculated while building the model based on the training data. Then, classification is completed by finding the hyper-plane that differentiate the two classes very well.

For a dataset (xi, yi), xi ∈R, yi ∈ {-1, 1}, i = 1, 2, 3, …, j, …, m. Set "X" represents a particular observation represented in a vector of features. "Y" represent the corresponding class for each observation. SVM learning-based algorithm can be trained to model a separating plane based on the provided data space or vectors of features "X"s. This plane should correctly allocate all observations related to a particular class to one side while allocating the remaining observations on the opposite side of that plan with some margins on both sides. As illustrated in figure 4, in the learning the phase, the algorithm attempts to model a decision boundary that correctly separates all the instances with binary classes.

Let l1 ← wx + b = 1 and l-1 ← : wx + b = -1 be particular planes with all observations in group 1 are located on one side of l1 plane and all observations in group −1 are located on one side of l-1 hyper-plane. For the best separation between the two classes, another plane l ← wx + b = 0 is located in the middle of l1 and l-1 planes. The best separation can be modeled by finding the maximum margin "M" that separates the data from both groups. As M = 2 ||w||-1, to maximize M, ||w|| must be minimized. SVM solves the optimization problem illustrated in (1).

$$minimize(\frac{1}{2}||w||), subject\ to\ y_i(wx_i + b) \geq 1 \quad (1)$$

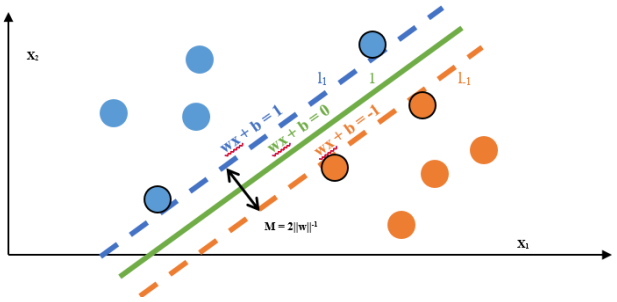

Figure 3: Decision boundary for a linear SVM-based classifier with two different classes.

VII. MATHEMATICAL REPRESENTATION

For time series data generated within a window of size "h", the number of generated samples is represented by the variable "N". For instance, with a window size equals ten seconds and a sampling interval of one second, then output vector will contain ten samples.

A mathematical representation for the set of the requested URI addresses during each sampling interval "Ti" is presented by (2). "A" Set contains all the requested URI addresses during a window "h" whereas "ai" subset contains all the requested URI addresses during the sampling interval "Ti" where ai ⊆ A. The three selected features will be calculated by analyzing the generated subset "ai" for each sampling interval "Ti" in current window "h".

$$A = \{a_i\}_{i\in\{1,2,3,\dots,N\}\ where\ a_i=\{u_j^i\}_{j\in\{1,2,3,\dots,m_i\}}} \quad (3)$$

The variable "u_j^i" represents the jth requested URI address within the ith sampling interval "Ti". It has two indices. The ith index indicates the current sampling interval or "Ti" while the jth index is an indicator for the current requested URI address within the subset "ai". In other words, the variable "u_j^i" represents the jth element within "ai" subset. The other variable "mi" represents number of URI addresses of each subset "ai" during each sampling interval "Ti". Mathematically speaking, "mi" is the cardinality of each subset "ai". The value of this variable is calculated as illustrated in the given algorithm in figure 5 above.

The first parameter, Request rate, is defined in (2) and (3) by the variable "mi" which is the cardinality of each subset "ai" or |ai|.

$$feat1 = |a_i| = m_i \quad (3)$$

Where $1 \leq i \leq N$. A new subset called "q_i" is introduced in (4) representing only unique requested URI addresses within each "ai" subset such that "q_i" is a subset of "a_i".

$$q_i = unique(a_i) = \{r_1^i, r_2^i, r_3^i, \dots, r_l^i, \dots r_{k_i}^i\} \quad (4)$$

Where $q_i \subseteq a_i$, $k_i \leq m_i$ and $1 \leq i \leq N$. The second parameter, URI diversity, is defined in (4) and (5) by the variable "ki" which is the cardinality of each subset "qi" or |qi|. It is generated by counting only unique URI addresses within each subset "ai" as illustrated in the given algorithm in figure 5 above.

$$feat2 = |q_i| = k_i \quad (5)$$

The last feature, the maximum request rate among the request rate for all the requested URI addresses is generated by finding the maximum frequency among all the frequencies of all elements, unique requested URI addresses, in each "qi" subset within each subset "ai" as presented in (6).

$$feat3 = max\left(f(r_1^i),\ f(r_2^i), \ldots, f(r_{k_i}^i)\right) \quad (6)$$

Where $f(\mathrm{r}) \geq 1$, $k_i \leq m_i$ and $1 \leq i \leq N$. The function $f(\mathrm{r})$ indicates frequency or number of occurrences of the current element "r" within the current subset "ai". The variable "r" is contained in current subset "qi" that contains the unique requested URI addresses extracted from "ai" subset shown in (4) above.

During each sampling interval "Ti", only three values will be extracted from the analyzed traffic forming the ith observation {feat1i, feat2i, feat3i} for the next stage or the classification phase as shown in figure 4 in section IV.

VIII. DATASETS

The conducted work is evaluated with two different datasets created from audited network traffic for both of the legitimate and the anomalous requests. For representing a real-world legitimate HTTP GET requests, the "1998 FIFA World Cup" evaluation dataset [13] is used. In the other hand, another evaluation dataset called "UNB ISCX Intrusion Detection System 2012 evaluation dataset [14] is used to represent the anomalous HTTP GET requests.

The 98 world cup traces are a legitimate HTTP GET requests. This evaluation dataset, is built by logging all the requests received by all the web server machines of the 1998 FIFA World Cup domain in a specified period of time. HTTP requests were logged to common log format files by the site logging system where each HTTP Get request logged to a separate record alongside its attributes. Request's attributes set contains client source IP, internal requested URI address, request time formatted to GMT in addition to others. Logging files are organized by a day based numbering system.

The UNB ISCX IDS 2012 evaluation dataset contains audit data for network sessions with labeling information for each session. Also, audit data contains the packet payloads in pcap format. Relevant profiles, labeled as either normal or anomalous, for each network session can be found in xml format. The UNB ISCX IDS 2012 evaluation dataset is publicly available for researchers. It consists of 7 sub datasets containing all network traffic including payload of normal and malicious activities generated during the individual days. The data audited during the fifth day, Tuesday, 15/6/2010, contains the traffic generated by the HTTP GET flood DDoS activities containing more than 23 gigabyte of captured data. The large size of the captured traffic is due to capturing the packet payload alongside their header, rather than only capturing the headers. The attack is started by the bots master and run for 60 minutes. Editcap, a part of Wireshark, is used to split the 24 gigabyte main pcap testbed file to sub pcap files. Each sub pcap file contains the audited data for a sampling interval. Then, tshark tool, also apart of Wireshark software, is used to analyze the splitted sub pcap files and extrate the chosen parameters.

IX. RESULT AND EVALUATION

The proposed features used for discriminating the anomalous requests from legitimate HTTP requests are evaluated only for non-recursive HTTP GET flood due to the limitation in the available datasets for the public. Sampling interval "T" is set to one second that means the proposed features are calculated each second from the analyzed traffic. The HTTP GET request traffic rate for the two datasets is shown in figure 5.

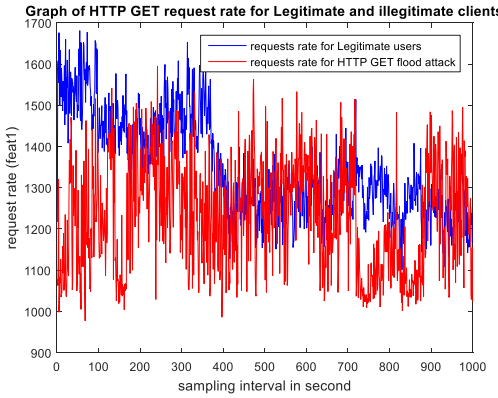

Figure 5: HTTP GET request rate (feat1) for traffic generated by legitimate and illegitimate clients.

As shown in graph 5, anomalous requests rate alone may mislead detection process as both type of traffic can share the same rate. Intruder can flood the victim using hundreds of thousands of bots sending requests at rate similar to legitimate profiles.

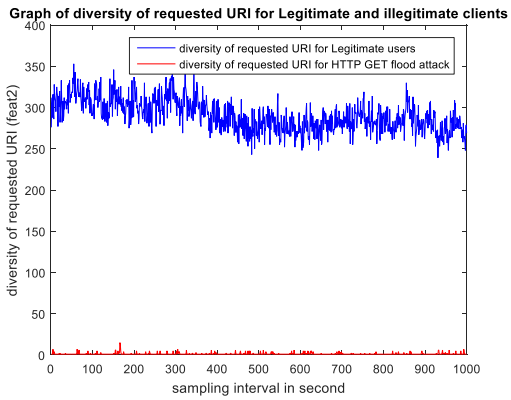

Figure 6: Diversity of requested URI addresses (feat2) for legitimate and illegitimate clients.

Even for spectral analysis, The Author in [15] conclude that using spectral analysis based methods can fail to detect DDoS attacks in a special circumstances when attackers use

random wait times and a sufficiently slow start phase. Therefore, diversity of the requested URI addresses feature is evaluated alongside the request rate. In figure 6, graphs of both the legitimate and HTTP GET flood attack request are plotted. The graphs clearly show the dissimilarity between the two types of traffic.

Also, the diversity parameter can mislead the classification process. The anomalous behavior in the case of non-recursive tends to have low diversity for the requested URI addresses, whereas the legitimate clients often have high values for this parameter. In some situations, especially at low traffic rate, the legitimate users also tend to request a limited number of URIs during the sampling interval. Which means low diversity in requested URI addresses. As a consequence, they will be considered behaving anomalously while they are normal clients. To overcome this situation, the third feature was evaluated as shown below in figure 7. In legitimate profiles, users often request any object or URI address for a limited times during a sampling interval, whereas in anomalous profiles and especially non- recursive profile, bots flood the victim with requests for the same object with the same or URI address. Unfortunately, NAT techniques can lead to conflict with this feature as the site or the server receive many HTTP requests for the same object from the same source, but in fact that source is actually sending these requests on behave of a set of legitimate clients who is serving them simultaneously. In such situation, the first feature or request rate can do the job and remove the conflict.

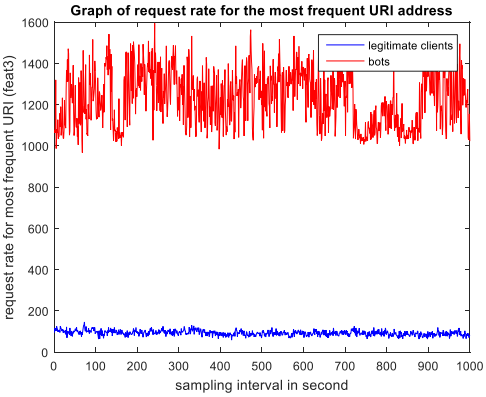

Figure 7: Frequency of max requesting URI (feat3) for legitimate clients and bots.

Due to the clear differences in the previous graphs for the two traffic profiles, the suggested parameters can be used as distinguishing parameters to discriminate a real-world non-recursive HTTP GET flood DDoS attack from legitimate clients. The extracted time series data for the previous illustrated parameters is used to build SVM based classifier using the training part of the generated data. Then, the built model is evaluated using the test part, totally different part, of the generated data. The built model can discriminate the legitimate and the non-recursive flood traffic perfectly. Table 1 shows the actual result and the elapsed time to build and test the SVM based model.

Table 1: Experimental result for SVM based model

Traffic type	# observations	Train time (s)	Test time (ms)	Test Accuracy
legitimate	6019	0.35	2	100
DDoS	3599			

X. CONCLUSION

This proposed work is a simple and clear mechanism to discriminate non-recursive application layer flood traffic by revealing some subtle differences depending on selected behavioral parameters or features. This paper proposes a set of parameters extracted from the inbound traffic including traffic rate, diversity among the requested objects and request rate for the most frequent object which can be used to powerfully differentiate between bots generating non-recursive DDoS HTTP GET request flood and legitimate requests. Moreover, a complete study of two publicly available datasets is presented in section 4. There are noticeable dissimilarities between these two datasets which lead to an efficient classification. In the future work and as a continuation to this research, the proposed methods will be evaluated on different dataset, especially on recursive HTTP GET flood DDoS dataset. Currently, there is no such dataset available for public research presenting huge limitation and restriction on the conducted research in this field. For example, The CAIDA UCSD "DDoS Attack 2007" Dataset is not publically available, since it's restricted only to a limited list of countries. Also, building such dataset from scratch is not an easy task as it requires a complete network infrastructure including at least one online site or service representing a victim with complete monitoring and auditing infrastructures for capturing the desired traffic during the desired activities. Also, budget is one of the most effective factors, for instance, a botnet may be haired during this process which is, botnet, a prepaid service. All these requirements usually are not available to public researches, therefor building such dataset would make a significant improvement in the anomalous traffic detection process.

The future work also includes identifying new additional subtle technique, e.g., for sophisticated detection, each connection or session could be inspected by itself rather than inspecting all the traffic. As a consequence, each session can be handled independently without affecting any other session. This includes modifying the current chosen parameters or features, also it may requires identifying new parameters to distinguish a connection weather it is a legitimate connection or a bot stream.

REFERENCES

[1] Thing, V.L., M. Sloman, and N. Dulay. *A Survey of Bots Used for Distributed Denial of Service Attacks*. in *International Information Security Conference*. 2007. Boston: SpringerLink,DOI: 10.1007/978-0-387-72367-9_20.

[2] Chee, W.O. and T. Brennan *Layer 7 DDoS*. OWASP Project, 2010.

[3] R. Fielding, E. and E. J. Reschke *Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing*. RFC 7230, DOI 10.17487/RFC7230, 2014.

[4] David, J. and C. Thomas. *DDoS Attack Detection using Fast Entropy Approach on Flow-Based Network Traffic*. in *2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)*. 2015. Elsevier.

[5] Liu, H.-I. and K.-C. Chang. *Defending Systems Against Tilt DDoS Attacks*. in *The 6th International Conference on Telecommunication Systems, Services, and Applications*. 2011. ieee.

[6] Hoque, N., D. K Bhattacharyya, and J.K. Kalita. *A novel measure for low-rate and high-rate DDoS attack detection using multivariate data analysis*. in *International Conference on Communication Systems and Networks (COMSNETS)*. 2016. Bangalore, India: ieee, DOI: 10.1109/COMSNETS.2016.7439939.

[7] Tao, P., C. Leckie, and K. Ramamohanarao. *Protection from distributed denial of service attacks using history-based IP filtering*. in *Conference on Communications, 2003. ICC '03. IEEE International*. 2003. Anchorage, AK, USA: ieee, DOI: 10.1109/ICC.2003.1204223.

[8] Ahmed, E., et al. *Use of IP Addresses for High Rate Flooding Attack*. in *25th International Information Security Conference (SEC 2010)*. 2010. Brisbane, Queensland.: ieee.

[9] Jin, J., et al. *Mitigating HTTP GET Flooding Attacks through Modified NetFPGA Reference Router*. in *1st Asia NetFPGA Developers Workshop*. 2010. Daejeon, Korea: ResearchGate, [2009-S-038-01, The Development of Anti-DDoS Technology].

[10 Ye, C., K. Zheng, and C. She. *Application layer DDoS detection using clustering*. in *2012 2nd International Conference on Computer Science and Network Technology*. 2012. CHANGCHUN, CHINA: ieee.

[11] Ni, T., et al. *Real-Time Detection of Application-Layer DDoS Attack Using Time Series Analysis*. in *Journal of Control Science and Engineering, Volume 2013, Article ID 821315, 6 pages*. 2013. Changzhou 213164, China: Hindawi Publishing Corporation.

[12] Chen, Y. and K. Hwang. *Spectral Analysis of TCP Flows for Defense against Reduction-of-Quality Attacks*. in *International Conference on Communications*. 2007. Glasgow, UK: ieee, DOI: 10.1109/ICC.2007.204.

[13] Arlitt, M. and T. Jin, *1998 World Cup Web Site Access Logs*, in *Traces available in the Internet Traffic Archive*. 1998.

[14] Ali, S., et al., *Toward developing a systematic approach to generate benchmark datasets for intrusion detection.* Computers and Security, Volume 31 Issue 3, May, 2012. **31**(3): p. 357-374.

[15] Joel, B. and S. Rishie. *Detectability of Low-Rate HTTP Server DoS Attacks using Spectral Analysis*. in *International Conference on Advances in Social Networks Analysis and Mining*. 2015. Paris, France: IEEE/ACM, DOI: http://dx.doi.org/10.1145/2808797.2808810.