Available online at http://www.journalijdr.com

International Journal of
DEVELOPMENT RESEARCH

IJDR

ISSN: 2230-9926

International Journal of Development Research
Vol. 4, Issue, 12, pp. 2656-2662, December, 2014

Full Length Research Article

DEFENDING INIMITABLE ATTACKING HOSTIN WEB-PROXY BASED TRAFFIC

***Sangeetha, K. and Krishnaveni, S.**

Department of CSE, Apollo Engineering College, Chennai, India

ARTICLE INFO

Article History:
Received 30th September, 2014
Received in revised form
18th October, 2014
Accepted 29th November, 2014
Published online 27th December, 2014

Key words:
HTTP,
DoS,
Resisting,
Complexity,
Legitimate.

ABSTRACT

A novel server side defense system is proposed to resist web proxy based attack. A proxy server is a server that may be a computer system or an application that acts as an intermediary for requests from clients seeking resources from other servers. A client unite with the proxy server, appealing some service, such as a file, connectivity, web page accessing, or other appropriate resources available from a different server and the proxy server evaluates the request as a way to simplify and control its complexity. A web proxy can be used as an attacker tool, by an attacker sends attack requests to a Web proxy and forces it to forward the attack requests to the origin server. Resisting such an attack by the mid Web proxies is not a practical approach, due to the lack of cooperation mechanisms between server and proxies, in particular those uncontrollable private proxies. In the final aggregated proxy-to-server traffic, there is no obvious difference between the normal traffic and the attack traffic except their underlying purposes. In the existing methodology a server denies services for a proxy server as a whole, In the above cases, Here along with an attacking client the legitimate users also need to suffer with DoS. But here we implement an enhanced HTTP protocol in this proxy server. So proxy server doesn’t hide application id from web server. So web server got client identity of each request. So server can group requests based on this application ID's and provides DoS accordingly. So By the revised system a server can serve maximum legitimate users in the way they are meant too.

INTRODUCTION

In computer networks, a proxy server is a server that may be a computer system or an application that acts as an intermediary for requests from clients seeking resources from other servers. A client connects to the proxy server, requesting some service, such as a file, connection, web page, or other resource available from a different server and the proxy server evaluates the request as a way to simplify and control its complexity. Today, most proxies are web proxies, facilitating access to content on the World Wide Web.

A Web proxy may be turned into an attacker by two steps

Attacker sends attack requests to a Web proxy and forces it to forward the attack requests to the origin server;

Step2: Attacker disconnects connections between itself and the proxy.

In Step 1, two methods can be used to penetrate through the Web proxies

Requesting dynamic documents or setting “Caching-Control: none-cache” in the headers of HTTP requests. Repeating these steps, a single host can simultaneously trigger a lot of Web proxies to attack a Web server without the need of invading them. The attraction of such an attack lies in three aspects: It enables the attacking host to break through the client-side restrictions by connecting different Web proxies via HTTP protocols; Resisting such an attack by the mid Web proxies is not a practical approach, due to lack of cooperation mechanisms between server and proxies, in particular those uncontrollable private proxies. Such an attack may confuse most of the existing detection systems designed for the traditional DDoS attacks due to two reasons: first, the origin server cannot directly observe and diagnose the terminal hosts shielded by the hierarchical proxy system; second, the attack traffic is mixed with the regular client-to-proxy traffic by each proxy that forwards the traffic. In the final aggregated proxy-to-server traffic, there is no obvious difference between the normal traffic and the attack traffic except their underlying

***Corresponding author: Sangeetha, K.**
Department of CSE, Apollo Engineering College, Chennai, India

purposes. Thus, the victim server is hard to accurately identify and filter the attack requests. The Web proxy-based HTTP attack is more flexible and covert than most of existing DDoS attacks. The difficulty of detection lies in three aspects:

Real attacking hosts are unobservable to the origin server since they are shielded by the hierarchical Web proxies;

A Web proxy may be passively involved in an attack event and may unconsciously act as an attacker; Observed from the victim server, both legal and illegal traffic comes from the same sources (i.e., Web proxies). Although most of the large-scale official proxies are usually configured to have high security, they cannot avoid being abused for the proxy base attacks. This type of attacks may bring new challenges to existing network security systems. Motivated by these issues, a novel resisting scheme is proposed to protect the origin server from Web proxy-based HTTP attacks in this work. The proposed scheme is based on network behavior analysis. It maps a Web proxy's access behavior to a hidden semi-Markov model (HsMM) which is a typical double stochastic processes model. The output process of an HsMM profiles the observable varying process of a proxy-to-server traffic. The hidden semi-Markov chain of an HsMM describes the transformation of a proxy's internal behavior states, which can be considered as the intrinsic driving mechanism of a proxy to server traffic.

Based on this behavior model, detecting the abnormality of a Web proxy can be achieved by measuring the deviation between an observed behavior and the Web proxy's historical behavior profile. Long-term and short-term behavior assessment methods are proposed. Long-term behavior assessment issues warnings on a large scale, while short-term behavior assessment locates abnormal request sequences embedded in the proxy-to-server traffic. A new "soft-control" scheme is proposed for attack response. The scheme reshapes the suspicious sequences according to the profile of a proxy's historical behavior. It converts a suspicious sequence into a relatively normal one by partly discarding its most likely malicious requests instead of denying the entire sequence. Thus, it can protect the HTTP requests of legitimate users to the greatest extent possible from being discarded.

About HTTP attack

In order to accomplish a denial of service state on systems, flood attacks aim to push limits of system usage to the out of boundaries determined by the normal usage scenarios. There may be a flood attack between the considered normal network traffic and the considered abnormal network traffic. The flood attack name can be determined by the specific protocol that attack is made on. For example, a flood attack on the DNS protocol is called as DNS Flood Attack while a flood attack on the HTTP protocol is called as HTTP Flood Attack. Since every protocol has its own technical architecture and vulnerabilities, flood attacks can differ on the attacking techniques from protocol to protocol. This is a type of application DOS attack.

Application DDoS Attacks

Application DDoS attacks are DDoS attacks targeted at overwhelming Web server, application server or database resources. While application-based attacks still only account for 26% of all DDoS attacks, they are more sophisticated and much more challenging to stop. Application DDoS attacks usually bypass most traditional network security devices because attack traffic often mimic regular traffic and cannot be identified by network layer anomalies. Some application DDoS attacks simply flood a Web application with legitimate requests in an attempt to overwhelm server processing power. Other attacks exploit business logic flaws. For example, a Website's search mechanism may require excessive processing by a back end database server and become a target. An application DDoS attack could exploit this weakness by performing thousands of search requests using wildcard search terms to overwhelm the back end application database."Slowloris" emerged as a perilous Application DDoS attack in 2009. This attack disrupts application service by exhausting web server connection pools. In the Slowloris attack, the attacker sends an incomplete HTTP request and then periodically sends header lines to keep the connection alive, but never sends the full request. Without requiring much bandwidth, an attacker can open numerous connections and overwhelm the targeted Web server. While multiple patches have been created for Apache and other web servers to mitigate this vulnerability, it nonetheless demonstrates the power of more sophisticated DDoS attacks.

Denial-of-Service (DoS) attacks continue to be keythreat to Internet applications. In such attacks, especially distributed DoS attacks, a set of attackers generates a huge amount of traffic, saturating the victims network, and causing significant damage. Overlay networks have been proposed to protect applications against such DoS attacks. These overlay networks are also known as proxy networks. The key idea is to hide the application behind a proxy network, using the proxy network to mediate all communication between users and the application, thereby preventing direct attacks on the application. Realistic study of these approaches should involve large networks, real applications, and real attacks. To date, however, studies of these approaches have been limited to theoretical analysis and small-scale experiments, which cannot capture the complex system dynamics, including packet drops, router queues, temporal and feedback behavior of network and application protocols during DoS attacks. These factors are critical to the application and proxy network performance in the face of DoS attacks. Thus, we still do not have answers to many key questions about the viability and properties of these proxy approaches.

Specifically, with real complex network structures and protocol behavior, can proxy networks tolerate DoS attacks? If so, what are the key parameters to achieve effective and efficient resilience? If we use proxy networks, what are the performance implications for applications? Our main contributions are the following. First, we provide the first large-scale empirical study on the DoS resilience capability of proxy networks using real applications and real attacks; this is a qualitative advance over previous studies based on theoretical models and small scale experiments. Second, we provide the first set of empirical evidence on large-scale network environment to prove that proxy networks have effective and scalable resilience against DoS attacks. Third, we provide a detailed performance analysis of proxy networks in large-scale network environment, and show that, in contrast to intuition, proxy networks can improve user-experienced performance.

Related Work

Proxy Network Approach

Proxy networks have been proposed as a means to protect applications from DoS attacks. Figure 3.2.1 illustrates a generic proxy network encompassing most of the proposed approaches. As shown in Figure 3, an overlay network, known as proxy network, is used to mediate all communication between users and the application. As long as the mediation can be enforced, the proxy network is the only public interface for the application, and the application cannot be directly attacked. Meanwhile a large set of proxies, known as edge proxies, publish their IP addresses, providing application access. The number of edge proxies can be flexibly increased. This allows scalable resilience against DoS attacks on edge proxies, and thereby allows a proxy network to shield the application from DoS attacks. Using this generic proxy network model, we study the fundamental capabilities and limitations of a wide range of proxy networks.

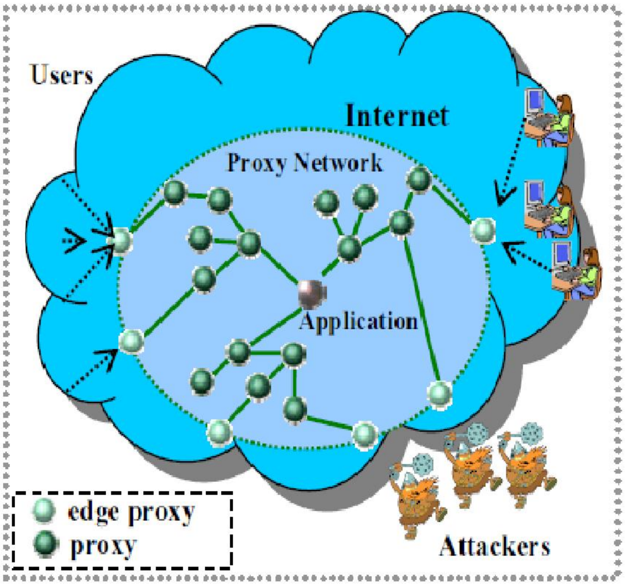


Figure. Prevention of Applications from Direct DoS Attacks

As discussed above, a proxy network must have two key capabilities to successfully protect applications from DoS attacks. First, a proxy network must enforce mediation so that the application can only be reached via the proxy network, thereby preventing direct DoS attacks on the application. Second, a proxy network must provide DoS-resilience mediation so that it can support continued user access to the application under DoS attacks. But most of the Proxy server doesn’t detect or resist these DOS attack and they also hide an application IP address using a proxy network, thereby enforcing proxy network mediation. So web server cannot group request from each application. So most of the HTTP resisting algorithms are grouped requests on proxy server ID based. Our base paper uses this approach. A novel server-side defense scheme is proposed to resist the Web proxy-based distributed denial of service attack. The approach utilizes the temporal and spatial locality to extract the behavior features of the proxy-to-server traffic, which makes the scheme independent of the traffic intensity and frequently varying Web contents. A nonlinear mapping function is introduced to protect weak signals from the interference of infrequent large values. Then, a new hidden semi-Markov model parameterized by Gaussian-mixture and Gamma distributions is proposed to describe the time-varying traffic behavior of Web proxies. The new method reduces the number of parameters to be estimated, and can characterize the dynamic evolution of the proxy-to-server traffic rather than the static statistics. Two diagnosis approaches at different scales are introduced to meet the requirement of both fine-grained and coarse-grained detection. Soft control is a novel attack response method proposed in this work. It converts a suspicious traffic into a relatively normal one by behavior reshaping rather than rudely discarding.

User Activity Analysis

Various features of DDoS attack bounding to a web server includes. Attackers use various genuine as well as fake IP addresses to generate attacking requests towards the Web server. And as a result server get flooded by requests. To defend this first it is needed to distinguish between legitimate user requests and attacking requests. By mixing Pearson correlation coefficient with user activity level we could distinguish between legitimate and attacking user requests. A normal user will always have a pause between his user requests and chances of repeating the pattern are very less. But in the case of an attacker it is in reverse.

$UAR(t1) = UE(IPCount(t1, ipi), r)$

where UAR is user activity record at time t1, UE is the evaluation of count of requests from user IP which is the r’th request to a web server. The Pearson correlation coefficient to relate the legitimate user and attacking host is defined as:

$$PCC_{XY} = \frac{cov(X,Y)}{\sqrt{D(X)}\sqrt{D(Y)}}$$

Where cov (*X*, *Y*) is the covariance of *X* and *Y*. *D(X*), *D(Y)* is the variance of *X* and *Y* respectively. Here the equation is formulated as:

$$PCC_{UAR} \quad (t1, t2) = \frac{cov(UAR(t1), UAR(t2))}{\sqrt{D(UAR(t1))}\sqrt{D(UAR(t2))}}$$

It is known that PCC value ranges [-1, 1]. The value being more closer to 1 the stronger relativity between X and Y. Closer to 0 weaker is the relativity. We make use of this PCC value to identify an attack by setting some threshold value *th.*If the result is above *th* attack occurs else not.

Existing System

Temporal and Spatial Locality

Temporal and spatial locality analysis used to extract the proxy to server behavior. Temporal locality of reference has been widely applied in many fields, for example, program behavior reference pattern of Web access and Web proxy cache replacement strategy. Temporal locality refers to the property that a referencing behavior in the recent past is a good predictor of the referencing behavior to be seen in the near future, whereas the resource popularity metric only represents the frequency of the requests without indicating the correlation between a reference to a document and the time since it was last accessed. Here, we resort to the concept of stack distance. The files are assumed to be placed on a stack such that, whenever a file f is requested, it is either pulled from its position in the stack and placed on the top, or simply added to the stack if the file is not yet in the stack. The stack distance for the request is then the distance of f from the top in the former case or undefined in the latter case. Spatial locality refers to the property that objects neighboring an object frequently accessed in the past are likely to be accessed in the future. For example, when a home page is requested, all its embedded objects are likely to be accessed at the same time. Because spatial locality indicates correlation among a cluster of HTTP requests, capturing spatial locality can help mine the access behavior of Web proxies. Different from, here a new method is used to quantize the spatial locality.

Drawbacks of Existing System

Here this system uses hidden semi-Markov model, model without state information. That is requests are grouped based on proxy server ID. And this model has no idea about original request source. Because proxy server hide this information from web server due to the HTTP protocol limitation. By using this model, we detect both temporal and spatial behavior in this system. And this model also reshape incoming request to remove attacking request instead of blocking proxy server. Requests from attackers and non-attackers through proxy server are reached at web server side in a mixed manner. Web servers have no facility to split this requests. From this request, web server detects spatial and temporal behavior using this Markov model. But this will increase false positive and false negative ratio when number of non-attacking client increases.

Trojan Attacks

Trojans are often used to launch Distributed Denial of Service (DDoS) attacks against targeted systems, but just what is a DDoS attack and how are they performed?

At its most basic level, a Distributed Denial of Service (DDoS) attack overwhelms the target system with data, such that the response from the target system is either slowed or stopped altogether. In order to create the necessary amount of traffic, a network of zombie or bot computers is most often used. Zombies or botnets are computers that have been compromised by attackers, generally through the use of Trojans, allowing these compromised systems to be remotely controlled. Collectively, these systems are manipulated to create the high traffic flow necessary to create a DDoS attack. Use of these botnets are often auctioned and traded among attackers, thus a compromised system may be under the control of multiple criminals – each with a different purpose in mind. Some attackers may use the botnet as a spam-relay, others to act as a download site for malicious code, some to host phishing scams, and others for the aforementioned DDoS attacks. Several techniques can be used to facilitate a Distributed Denial of Service attack. Two of the more common are HTTP GET requests and SYN Floods. One of the most notorious examples of an HTTP GET attack was from the My Doom worm, which targeted the SCO.com website. The GET attack works as its name suggests – it sends a request for a specific page (generally the homepage) to the target server. In the case of the My Doom worm, 64 requests were sent every second from every infected system. With tens of thousands of computers estimated to be infected by My Doom, the attack quickly proved overwhelming to SCO.com, knocking it offline for several days. A SYN Flood is basically an aborted handshake. Internet communications use a three-way handshake. The initiating client initiates with a SYN, the server responds with a SYN-ACK, and the client is then supposed to respond with an ACK. Using spoofed IP addresses, an attacker sends the SYN which results in the SYN-ACK being sent to a non-requesting (and often non-existing) address. The server then waits for the ACK response to no avail. When large numbers of these aborted SYN packets are sent to a target, the server resources are exhausted and the server succumbs to the SYN Flood DDoS. Several other types of DDoS attacks can be launched, including UDP Fragment Attacks, ICMP Floods, and the Ping of Death. For further details on the types of DDoS attacks, visit the The Advanced Networking Management Lab (ANML) and review their Distributed Denial of Service Attacks (DDoS) Resources.

Network Layer DDoS Attacks

TCP and IP properties to discover attack signals, Traditional defense techniques can be focused on the Network layer DDOS attacks. Client can be found based on the trust management mechanism so the application layer DDOS is mitigated by giving priority for users. Zombies can be identified by automatically changing puzzles then the http requests of suspected hosts are blocked. The model defines to profile the normal access behavior of four attributes of web page request sequences. The reconstruction error of a given request sequence is used for detecting DDOS attacks. The flow correlation coefficient was used to measure the similarity among suspicious flows and then the http based DDOS attacks from normal flash crowds.

Proposed Work

To secure the root server from the web proxy-based http attacks to filter the problem. Assuming the attacking traffic from the initial web proxies apart from its real sources. It is applicable but the proxies can only observe the victim and the main intention is to extract the malicious traffic. A web proxy's access behaviors can be of temporal and spatial locality and also the interior driving mechanism i.e., normal and abnormal behavior. The spatial and temporal locality can be observable and it can be controlled by the interior driving mechanism but the origin server cannot able to detect accurately. The estimation can be based on the observation features of proxy-to-server traffic. Web proxy's access nature can be made directly to an hidden state markovian model. The basic HSMM consists of a pair of observation process (αt) and the hsm circumstance process (zt) where t € (1,2,……….n) is said to be the number of events. Where (αt) is bracket together with (zt) by the uncertain allocation depends on the circumstances process that is a "predetermine status semi-markovchain". IN common inappropriate observed value can raise from more than one status. Thus (zt)is not observable relevantly through (αt) but it can be calculated may be distinct or unremitting(continuous or both).Each status represents a driving mechanisms of a classification of proxy-server traffic. Evolution between two status represents the modification of driving mechanisms extent of a particular semi-markov status define the reside time of a relevant driving mechanism. IN the output process the web proxy behavior of temporal and spatial locality can be modeled. The driving mechanisms can be normal and abnormal, conversion. The issues resist in the proxy-based http attack is relevant for surfing the abnormal status of a web proxys access process then the mistrustful request caused by abnormal status needs to be extracted. The common features of the network traffic can be taken for building the modeled output process. The problem described here as: Traffic intensity needs to be independent. The web contents can be varied statically need to realize the early detection.

Steps for layer DDoS

There are numerous network security products in the existing market, but few of them can effectively defence against DDoS attacks. Due to deficiency in design, the common security products such as firewalls, intrusion prevention systems and routers always fail to fully address today's complicated DDoS attacks. Although the fallback policy or system optimization can be taken to cope with low-traffic DDoS attacks, it is not a best option in massive traffic prevention.

Manual Prevention

Generally speaking, there are two ways to prevent DDoS attacks by manual operations:

System optimization? To optimize key parameters of victims for enhancement of their response ability to DDoS attacks. However, this method can low-traffic DDoS attacks only, but not good at mountains of attack traffic prevention.

Source IP tracing? The first response of the system administrator under a DDoS attack would be to consult the uplink network service carriers, which may be the ISP or the IDC, to find out the source of the attack. But if the source IP address of the DDoS attack is forged, the process of finding the attack source often involves many carriers and judicial organizations. Even when the attack source is found out, blocking the traffic from there may cause the loss of normal traffic. Moreover, the prevailing Botnets and newly-emerged DDoS attacks make it impossible to prevent DDoS attacks by network tracing.

Fallback Policy

To prevent DDoS attacks, customers may increase the network bandwidth and system performance, use dynamic IP addresses, add more server cache, and so on. Although those approaches may alleviate the attacks to some extent, the effect of this fallback is not efficient enough because of low performance-price ratio and failed protection of massive traffic. Therefore, this method cannot prevent DDoS attacks essentially.

Router

We can use routers to implement some security measures, for example, setting an ACL, to filter some illegal traffic. ACLs are usually set based on protocols or source addresses. But most of DDoS attacks adopt legal protocols (such as HTTP), thus attack traffic cannot be filtered out by routers. And if DDoS attacks adopt the source address spoofing technology to forge packets, routers cannot prevent these attacks, either. Another DDoS countermeasure based on routers is to adopt Unicast Reverse Path Forwarding (uRPF) to block packets with forged source IP addresses at the network boundary. For today's DDoS attacks, this countermeasure is also useless because, as the basic principle of uRPF, the router blocks or allows a packet to pass the outlet by determining whether its source IP address is from the internal subnet, while attackers can easily forge the address and evade the uRPF prevention policy. Besides, to configure the uRPF policy on each router in front of potential attack sources is hardly achievable in actual environment.

Firewall

Firewalls are the most commonly used security products. But the DDoS attack prevention is not a part of function in its design. In some cases, firewalls even become the target of DDoS attacks and cause denial of service of the entire network.

Deficiency of DDoS detection capability? Firewalls are usually deployed in the network as Layer-3 packet forwarding devices. They not only protect the intranet but also provide access for devices that provide external Internet services for internal needs. If DDoS attacks exploit legal protocols allowed by servers, firewalls will be unable to identify attack traffic from the hybrid traffic precisely. Although some firewalls are equipped embedded modules that can detect attacks, the detection mechanisms are generally based on signatures and firewalls always fail to address the attacks if DDoS attackers change packets slightly. The detection of DDoS attacks must depend on the algorithm of behavior patterns.

Limitation of calculation capability ?Traditional firewalls perform intensive inspection to detect DDoS attacks, which costs a lot of calculation. Massive traffic in DDoS attacks, however, will cause the intense declination of the firewall performance, resulting in the ineffective completion of the packet forwarding tasks. The deployment locations also influence firewalls' capability of preventing DDoS attacks. Traditional firewalls are generally deployed at the network ingress. To some extent, this type of deployment is a good way to protect all resources inside the network, but firewalls in this kind of deployment often become the victims in DDoS attacks, leading to declination of the network performance and failure to response intended users' requests.

IPS/IDS

Currently, the most commonly used tools for attack prevention or detection are the IPS (Intrusion Prevention System) and IDS (Intrusion Detection System). But for DDoS attacks, IPS/IDS products often become incapable. The reason is that although the IDS can detect attacks at the application layer, its most basic level is a signature-based mechanism that needs recovering protocol sessions. But most of today's attacks adopt legal packets to hit the targets, and therefore the IPS/IDS products can hardly detect these attacks. Some IPS/IDS products have the capability of detecting anomaly protocols, but they take effect only after the manual configuration by security experts, complex and inelastic. Algorithm web application level HTTP anti flood system to detect flooder IP addresses and reduce the attack surface against the HTTP flood attacks at the web application level:

- call a global script file/class/library before the every code related to the web application,
- log every IP addresses that sent the request,
- Save all logs to “DBMS:: Flat File” or RAM,
- record the every request time microseconds and counts alongside with the IP addresses,
- compare the counts and times of every requests made by the same IP address,
- record the IP addresses and time values upon a rule breach,
- create exceptions for white listed IP addresses,
- record every IP addresses that breached the rule,
- Stop web application execution to reduce attack surface upon a breach with the psuedo code “EXIT”,
- define a time limit for the suspension of the web application execution,
- Show a CAPTCHA question to user who is accidently blacklisted and doesn’t want to wait for a suspension time,
- send detected IP addresses to the other security components (eg. stateless firewall),
- notify the administrator via an e-mail.

Locality of References

The locality of references can be defined based on the concept of virtual memory. It advanced concepts may be the doubled or more advanced production. While performing the multiprogramming the throughput can be thrashed due to server paging. The concept of locality principle can be used for constructing the generation of compiler codes and also for algorithm replacement. Here the virtual memory can be transformed based on the self adjusted technology without user involvement the throughput can be adjusted. The locality of principle has been applied in vast environment. The virtual memory for organizing the caches is for the purpose of address translation and also for algorithm design. Also the buffer needs to be managed between the main memory and the secondary memory. Temporal locality refers the behavior of the past based on that it can be predicted the future. The temporal locality can also be defined based on the concepts on linked list. The list of files can be stored in the list. The pointer can be used for linking the files. If the single file is requested it can be retrieved from the list. The main contribution of this paper follows: The web server consists of a number of websites. A specific folder can be allocated for each websites. There are two phases trainee and implementation phase

Training Phase

For the webpages stored in the server side, every possible url traverse sequence are found based on linked vector concept and stored as backup. Thus all the different possible request patterns (spatial and temporal) are identified. Based on the user request sequence can be formed. For one or more repeated request possible sequence can be generated. For single request the sequence can be formed as of with length one.

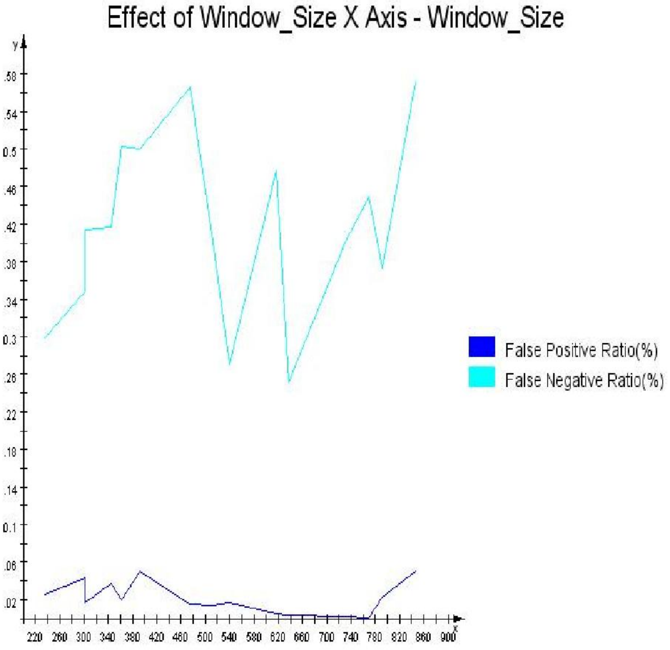
Detection Phase

The threshold based attack detection is implemented here. At first the inbound traffic is captured at the kernel level. After filtering these packets it is submitted for various parameter extractions which include any associative header, IP address, their arrival time etc. Using the training data in the analysis phase repeating request patterns are identified along with their corresponding IP address. Then it checks for the frequency of the repetition of requests from an IP address. And further decides whether to block or provide DoS to the user based on whether the request count or request pattern count exceeds the allowed threshold within the specified time period.

Performance Analysis

The system thus defined has been put to check its performance under various conditions. By changing various parameters like the inter distance of requests, inter sequence distances, the threshold values like temporal which signifies the repetition and spatial which signifies pattern occurrence we found out the FPR and FNR of the developed system. It has been found out that the FNR grows if the legitimate user’s ratio increases. But much better than DOS provided to cent percent users based on proxy.

Performance Evaluation Diagrams



Conclusion

Application layer attack has become a major threat to the internet in today’s world. The focus of this project was to come out with an effective solution for the detection and prevention of clients from inadvertently taking part in such attacks. Accordingly, a threshold based attack detection (TBAD) has been proposed and implemented in windows OS using J2SDK.Experiments were conducted by generating HTTP get attacks and using TBAD for its mitigation. It was evidence that the TBAD suppressed the flooding packet and thus prevented the client system from taking part in such attacks. The proposed system is very efficient in defending inimitable attacking hosts in web proxy based traffic by DOS. This work can be further enhanced by taking measures to optimize FNR values for better service. In addition to this traffic rules specified in this paper, we also implement different mechanisms for detecting attacks like unsupported http method, oversized header and body data size, minimum incoming data etc. But another typical problem is session hijacking for which we can find a mechanism to defend.

REFERENCES

[1] Triukose, S., Z. Al-Qudah and M. Rabinovich, 2009. “Content Delivery Networks: Protection or Threat?” Proc. 14th European Conf. Research in Computer Security (ESORICS), pp. 371-389.

[2] Zhonghua, Z.L.C. and W. Xiaoming, 2009. “Research on Detection Methods of CC Attack,” Telecomm. Science, pp. 62-65.

[3] Garcia-Teodoro, P., J. Diaz-Verdejo, G. Macia-Fernandez, and E. Vazquez, 2009. “Anomaly-Based Network Intrusion Detection: Techniques, Systems and Challenges,” Computers and Security, vol. 28, nos. 1/2, pp. 18-28.

[4] Ferguson, J. ?. “Variable Duration Models for Speech,” Proc. Symp. Application of Hidden Markov Models to Text and Speech,

[5] Yu, S. ?. “Hidden Semi-Markov Models,” Artificial Intelligence,

[6] Xie, Y. and S. Yu, ?. “Measuring the Normality of Web Proxies Behaviour Based on Locality Principles,” Network and Parallel Computing

[7] Xiang, Y., K. Li and W. Zhou, 2011 June. “Low-Rate DDos Attacks Detection and Trace back by Using New Information Metrics, ”IEEE Trans. Information Forensics and Security, vol. 6, no. 2, pp. 426-437.

[8] Yu, Y., K. Li, W. Zhou and P. Li, 2012. “Trust Mechanisms in Wireless Sensor Networks: Attack Analysis and Countermeasures,” J. Network and Computer Applications, vol. 35, no. 3, pp. 867-880.

[9] Jaeyeon, J., Balachander, K., Michael and R. 2002. Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites[C]. WWW pp. 293-304.

[10] Chu-Hsing Lin, Chen-Yu Lee and Jung-Chun Liu, 2010. A detection scheme for flooding attack on application layer based on semantic concept [J]. Computer Symposium (ICS), pp. 385-389.

****