A NOVEL APPROACH FOR DOS (DENIAL-OF-SERVICE) ATTACKS AND DISTRIBUTED DENIAL OF SERVICE (DDOS) USING SOFTWARE PUZZLE

SD.GOUSIYA[#1] and V.PADMAJA[*2]

[#] *M.Tech (CSE), Nimra College of Engineering & Technology, A.P., India.*

[*] *Assistant professor, Dept. of Computer Science & Engineering, Nimra College of Engineering & Technology, A.P., India.*

***Abstract*— Network is a group of nodes that interrelate with each other for switch over the information. This information is necessary for that node is reserved confidentially. Attacker in the system may capture this private information and distorted. So security is the major issue. There are several security attacks in network. One of the major intimidations to internet examine is DDoS attack. It is a malevolent effort to suspending or suspends services to destination node. – Denial of services (DOS) and Distributed Denial of services (DDoS) are the major problem against network security and cyber security that allow a client to perform very expensive and vital operations, before the network services are provided to the respected client. However An attacker may be able to manipulate the DOS and DDOS or built in graphics processing Unit (GPU) and be able to destroy client puzzles. In this paper we study how to preserve DOS and DDOS attacker for being manipulating the puzzle solving techniques. So now we introduce a new client puzzle referred to as Software Puzzle. It is unlike previous puzzle, which generate their puzzle algorithms in advance, a puzzle algorithm in the present software puzzle schemes is randomly generated only after a client request is received from the server side. t the Denial-of-service and distributed DoS attack a client puzzle method is implemented. In order to prevent further attack in network and to enhance the security the request that is provided by the client and the file sent by the server to client is in encrypted form. One drawback of existing system is if the attacker identifies the port, he can intrude or interfere in the communication and flood DOS attack and can hack communicating data. The methodology used is explained as follows. First the client has to solve a puzzle generated by the server. Then the client checks the latency of the file that has to be accessed from server database. The client can test the latency of the server by inputting the corresponding server IP address, number of packets, and the length of data in bytes. After processing the latency checking parameters, ping statistics of the server and the approximate round trip time will be displayed in the result. The client then encrypts the request and sends the request to server. AES Algorithm is used to perform the encryption and decryption. The server upon receiving the request has to decrypt the request using the client port number and IP address. The server sends the requested file by encrypting the file. Finally the client receives the file, decrypts the content and read it. Thus it can be concluded that more reliable communication can be performed between server and clients and active communications remains unaffected even in the presence of DDoS attacks.**

Index Terms*— Denial of Service (DoS), Code Protection, GPU Programming, Distributed Denial Of Service (DDoS), Security, Software Puzzle.

I. INTRODUCTION

The Denial of service attack is one of the types of active attack. The Denial of service attacks which revenue that the attackers can send certain messages which is vulnerable to the system. Sometimes they send packets to the target system which may result in failure [1]. As the remediation of susceptibility and reduction of performance to commerce systems, the harm of common DoS attacks becomes relatively minor. A Distributed Denial of Service attacks is implemented on the source of DoS attack and numerous dispersed attack sources. Usually, the attackers use a huge number of controlled bots dispersed in different locations to start on a great number of denial of service attacks to a lone target or several targets. With the quick growth of botnets in modern years, the attack traffic scale caused by Distributed Denial of Service attacks has been rising, with the target system, including not only industry servers, but also Internet infrastructures such as routers, firewalls and Domain Name Server systems as well as network bandwidth. The attack pressure sphere has also become broader. In computer network they use a protocol for called transmission control protocol .The packets are transferred through TCP. The attacker can send one or more attack packets to the network. This will cause the target servers and network resources and also overloads the server. These are the vital principles of Distributed Denial of Service attacks. The key reason is inflexible avoidance of DDoS attacks deception in the combination up of justifiable traffic and illegitimate traffic. It is difficult to discover the attack packets from the diverse traffic in the avoidance progression, particularly when the harass message packets masquerade to be normal messages. For exemplar, in signature -based pattern corresponding Intrusion Detection system, it is not easy to differentiate illegitimate packets from legitimate messages packets. In universal, according to the uniqueness, DDoS attacks can be divided into the following types:

1. **Volume-based attacks:** Distributed Denial of Service attacks; this type sends huge collection of junk data packets to cause the network devices to be overloaded, which leads to enlarge the networks bandwidth. Hence further more incoming requests are dropped and network will be blocked.

2. **Protocol-based attacks:** The most familiar forms of denial of service attack are traffic flooding attacks. N traffic flooding attack the attackers send a great number of ostensibly legitimate UDP, Transmission Control Protocol/Internet Protocol, ICPM packets in network host. This will cause a more traffic in the networksystem.

3. **Application-based attacks: The** attacks of this type often mail the consequent application-layer; main focus of this system attack is to deny the service of application layer. The low rate of traffic can also lead serious degradation of service.

II. Related Work

Software puzzle can be easily solved by an attacker using Graphical processing unit software. In software puzzle scheme the puzzle function will not be known in advance. Hence the client will use CPU resource only to solve the puzzle challenge. Also the cost of client computation to solve the puzzle will be large when compared to the cost of server computation which includes the puzzle generation and puzzle verification steps. Even if the attacker returns an arbitrary number as solution to the puzzle so as to exhaust the servers time for puzzle verification, the server time is much smaller than the service time or database process time and the returned answer will be rejected with high probability. The existing client puzzle scheme assume that the client solves the puzzle using legacy CPU resource only. But this is not always true. A malicious client may solve the puzzle using GPU (Graphic Processing Unit) component is almost a standard configuration in modern desktop computers, laptop computers, and even smartphones. In the proposed system it is possible to track the individual client behaviour through client's IP address. Nonetheless, if IP tracking is effective to thwart the GPU inflation, IP filtering can be used to defence against DoS attacks directly without utilizing client data. In other words, their defence against GPU-inflated DoS attacks may not be attractive in practice. A new type of client data, called software puzzle, to defend against GPU-inflated DoS and DDoS attacks. Unlike the existing client data schemes which publish a puzzle function in advance, the software puzzle scheme dynamically generates the puzzle function P(•) in the form of a software core C upon receiving a client's request. Specifically, by extending DCG technology which produces machine instructions at runtime, the proposed scheme randomly chooses a set of basic functions, assembles them together into the data core C, constructs a software data C0 x with the data core C and a random challenge x. If the server aims to defeat high-level attackers who are able to reverse-engineer software, it will obfuscate C0 x into an enhanced software puzzle. After receiving the software puzzle sent from the server, a client tries to solve the software puzzle on the host CPU, and replies to the server, as the conventional client data scheme does. However, a malicious client may attempt to offload the data task into its GPU. In this case, the malicious client has to translate the CPU software puzzle into its functionally equivalent GPU version because GPU and CPU have totally different instruction sets designed for different applications. Note that this translation cannot be done in advance since the software puzzle is formed dynamically and randomly. As rewriting/translating a software puzzle is time-consuming, which may take even more time than solving the data on the host CPU directly; software puzzle thwarts the GPU inflated DoSattacks.

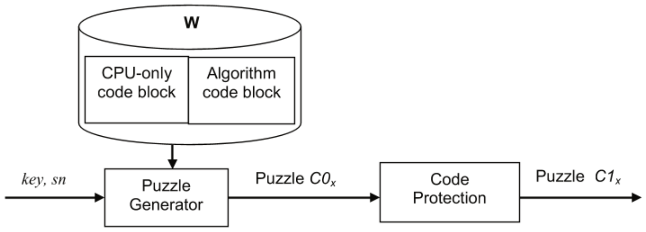

Fig. 1 Overview of System Architecture

III. PROPOSED METHODOLOGY

Here in this segment paper narrates about the methodologies that are incorporated in the experiment as depicted in the figure 1. Step 1: Here in this step requests for a web transaction is received from all the clients by the web server along with the parameter like date, time and client IP to store in the database. Then all this data from the database will be retrieved in a vector for pre-processing, where selected data like IP is fetched in a single dimension vector for clustering process. Step 2: Here Single dimension vector of IP addresses of the client that was fetched in the past step is been set to fuzzyC Means clustering process. Fuzzy C means clustering (FCM) technique which eventually helps to analyse the patterns of the IP through interactive clustering. =1/ where m is any real number whose value should be greater than 1, uij is the degree of membership of xi in the cluster j, xi is the i th of d-dimensional measured data, cj is the d-dimension centre of the cluster.

Then the optimization of the clusters is carried out by the fact of fuzzy portioning which yields fine grained clusters which in turn indicates the abstract patterns of the input client IP.

ALGORITHM 1: FCM Let X = {x1, x2, x3 ..., xn} be the set of data points and V = {v1, v2, v3 ..., vc} be the set of centers.

Step 0: Start
Step 1:Randomly select 'c' cluster centers.
Step 2: Calculate the fuzzy membership 'µij' using: =1/
Step 3: Compute the fuzzy centers 'vj' using: = ( ) / ( ), for

all j=1,2,…c

Step 4: Repeat step 2) and 3) until the minimum 'J' value is achieved or $\|U(k+1) - U(k)\| < \beta$. where, ‘k’ is the iteration step. ‘β’ is the termination criterion between [0, 1]. ‘U = (µij)n*c’ is the fuzzy membership matrix. ‘J’ is the objective function.

Step5:Stop

______________________________________ Step 3: The clustered IP are then considered for their higher priority using the entropy distribution factor of Shannon information gain. Here information gain is used to identify the most important and fluent IP address in the clusters which frequently affecting the web server for its performance. This can be given with the following equations 2. IGR(C) = -∑ (| Ci | / | C |) log (| Ci | / | C |) ....(2) Where Ci is the frequency of the IP address add in Cluster C.

Step 4: Decision trees are generally meant for the decision taking rules which indulge in putting conditions like if - else till to reach a decision. But here in our experiment of identifying DOS attack decision tree takes a two dimensional vector which is loaded with the attributes like IP address and their information gain values. Here each of the indices of the vector is feed to the tree to form the nodes and at every levels of the tree with respect to the Shannon information gain values. Then these values are keep accumulating the at the respective nodes to get the weighted decisions for judging attack level. Then this attack level is normalized in between the range 0 to 100 to get the desired level of software puzzle.

Step 5: This level of attack is been send to proxy server for puzzle generation process with a reference key generated through MD5 algorithm. Once the proxy server receives the attack level it identifies the expression desired to tackle the attack in its raw form. Then the variables in the expression is set to change the variables by assigning random number from 1 to 9. Once the expression is having the real numbers, then this is been evaluated using infix expression evaluation method as mentioned in the algorithm 2.

IV. LITERATURE REVIEW

The paper “A System for Denial-of-Service Attack Detection Based on Multivariate Correlation Analysis”[1],They propose an approach called as MAC which follows an triangular area to extract correlative feature. This uses a threshold-based anomaly detector, which contains a traffic profile that is normal traffic profiles. When new packets are arrives in the network it generate the network traffic profile. This traffic profile is compared with the statistical data of normal traffic profile, by which it detect a DDoS attack.

This paper “DDoS Detection Method Based on Chaos Analysis of Network Traffic Entropy “[2],paper they going to detect a DDoS by Chaos Analysis and Entropy. The entropy has been used in anomaly detection of DDoS attacks. It describes the degree of concentration and dispersal characteristic of traffics .But the entropy depends only on the values computed by each packet field, while the connection information or the relationship between each field has been ignored. In our approach, the volume of network traffic is pre-processed by entropy-based methods. Then, by using chaotic analysis on the entropy of source IPs and destination IPs, DDoS attacks are detected.

The paper “Discriminating DDoS Attacks from Flash Crowds Using Flow Correlation Coefficient” [3],the DDoS attack is detected by using a similarity based algorithm is used. And also they used a flow correlation and coefficient as a metric to find a DDoS attack. Flow correlation which defines a stastiscal relationship between two edge routers .The coefficient defines a specific property of attack. They execute software on every router to count the number of packets for every flow and record this information for a short term at every router. If the packet size is greater than the threshold value it will dropped.

This paper “Adaptive Selective Verification: An Efficient Adaptive Countermeasure to Thwart DoS Attacks” [4],Adaptive Selective Verification (ASV), which is a distributed adaptive mechanism for thwarting attackers efforts to deny service to legitimate clients based on selective Verification. This scheme uses bandwidth as currency, but the level of protection employed by the clients dynamically adjusts to the current level of DDoS attack. At a high level, the clients exponentially ramp up the number of requests they send in consecutive time manner, up to a threshold limit. The server implements a reservoirbased random sampling method to effectively sample from a sequence of incoming packets using bounded space technique. This enables adaptive bandwidth payments with server state whose size remains small and constant regardless of the actions of the attacker.

V. CONCLUSION & FUTURE SCOPE

Software puzzle scheme is used so that the puzzle function used is not known in advance. Hence, malicious client cannot solve the puzzle using GPU software. The proposed system provides even more security using conventional cryptographic techniques. In this scheme it is possible for the client to authenticate the server and vice versa. The communication between client and the server is more reliable in this system. Active communications remains unaffected even in the presence of DoS and DDoS attack. Also, the probability of hacking is also very less in this scheme. This idea can be extended to thwart DoS attackers which exploit other inflation resources such as Cloud Computing. future scope of this paper system can be enhance to generate more complex puzzles for variable operands and operators.

REFERENCES

[1] R. Shankesi, O. Fatemieh, and C. A. Gunter, “Resource inflation threats to denial of service countermeasures,” Dept. Comput. Sci., UIUC, Champaign, IL, USA, Tech. Rep., Oct. 2010. [Online]. Available: http://hdl.handle.net/2142/17372

[2] J. Green, J. Juen, O. Fatemieh, R. Shankesi, D. Jin, and C. A. Gunter, “Reconstructing Hash Reversal based Proof of Work Schemes,” in

Proc. 4th USENIX Workshop Large-Scale Exploits Emergent Threats, 2011.

[3] Y. I. Jerschow and M. Mauve, “Non-parallelizable and non-interactive client puzzles from modular square roots,” in Proc. Int. Conf. Availability, Rel. Secur., Aug. 2011, pp. 135–142.

[4] R. L. Rivest, A. Shamir, and D. A. Wagner, “Time-lock puzzles and timed-release crypto,” Dept. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep. MIT/LCS/TR-684, Feb. 1996. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.110.5709

[5] W.-C. Feng and E. Kaiser, “The case for public work,” in Proc. IEEE Global Internet Symp., May 2007, pp. 43–48.

[6] D. Keppel, S. J. Eggers, and R. R. Henry, “A case for runtime code generation,” Dept. Comput. Sci. Eng., Univ. Washington, Seattle, WA, USA, Tech. Rep. CSE-91-11-04, 1991.

[7] E. Kaiser and W.-C. Feng, “mod_kaPoW: Mitigating DoS with transparent proof-of-work,” in Proc. ACM CoNEXT Conf., 2007, p. 74.

[8] NVIDIA CUDA. (Apr. 4, 2012). NVIDIA CUDA C Programming Guide, Version 4.2. [Online]. Available: http://developer.download.nvidia.com/

[9] X. Wang and M. K. Reiter, “Mitigating bandwidth-exhaustion attacks using congestion puzzles,” in Proc. 11th ACM Conf. Comput. Commun. Secur., 2004, pp. 257–267.

[10] M. Jakobsson and A. Juels, “Proofs of work and bread pudding protocols,” in Proc. IFIP TC6/TC11 Joint Working Conf. Secure Inf. Netw., Commun. Multimedia Secur., 1999, pp. 258–272.

[11] D. Kahn, The Codebreakers: The Story of Secret Writing, 2nd ed. New York, NY, USA: Scribners, 1996, p. 235.

[12] K. Iwai, N. Nishikawa, and T. Kurokawa, “Acceleration of AES encryption on CUDA GPU,” Int. J. Netw. Comput., vol. 2, no. 1, pp. 131–145, 2012.

[13] B. Barak et al., “On the (Im)possibility of obfuscating programs,” in Advances in Cryptology (Lecture Notes in Computer Science), vol. 2139. Berlin, Germany: Springer-Verlag, 2001, pp. 1–18.

[14] H.-Y. Tsai, Y.-L. Huang, and D. Wagner, “A graph approach to quantitative analysis of control-flow obfuscating transformations,” IEEE Trans. Inf. Forensics Security, vol. 4, no. 2, pp. 257–267, Jun. 2009.

[15] S. Wang. (Sep. 18, 2011). How to Create an Applet & C++. [Online]. Available: http://www.ehow.com/how_12074039_createApplet-c.html#ixzz24Lsk0OJQ

[16] J. Bailey. (Oct. 28, 2014). How to Install Java on an iPhone, eHow Contributor. [Online]. Available: http://www.ehow.com/how_5659673_install-java-iphone.html#ixzz24jIAyKiM

[17] J. Ansel et al., “Language-independent sandboxing of just-in-time compilation and self-modifying code,” in Proc. ACM SIGPLAN Conf. Program. Lang. Design Implement., 2011.