ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

IJARCCE

nCORETech

LBS College of Engineering, Kasaragod

Vol. 5, Special Issue 1, Feburary 2016

Prevention of Dos and DDoS Attack Using Cryptographic Techniques

Soumya Suresh[1], Kiran V K[2]

P G Student, Computer Science & Engineering, NSS College of Engineering, Palakkad, India [1]

Assistant Professor, Computer Science & Engineering, NSS College of Engineering, Palakkad, India[2]

Abstract: Network Security is a specialized field in computer science that involves securing a network infrastructure. Denial-of-service (DoS) and distributed DoS (DDoS) are the major threats to cyber-security. In computing, a denial-of-service (DoS) attack is an attempt to make network resource unavailable to its intended users. DDos is short for Distributed Denial of Service. DDoS is a type of DOS attack where multiple systems, which are often infected with a Trojan, are used to target a single system causing a Denial of Service (DoS) attack. To prevent 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. This scheme is mainly used for military applications.

Keywords: Denial of service (Dos) Attack, Distributed Denial of service (DDos) attack, Software Puzzle, AES algorithm.

I. INTRODUCTION

Dos and DDos attack have become a major threat in current computer networks. Known DoS and DDoS attacks in the Internet generally conquer the target by exhausting its resources, that can be anything related to network computing and service performance, such as link bandwidth, TCP connection buffers, application/service buffer, CPU cycles, etc. In a DDoS attack, because the aggregation of the attacking traffic can be tremendous compared to the victim's resource, the attack can force the victim to significantly downgrade its service performance or stop delivering service. Conventional cryptographic tools which can be used to prevent DoS and DDoS attack may also degrade service quality. DoS attack can be prevented either by increasing the computational cost of client or the Computational cost of server. Using the software puzzle method the clients computational cost is increased and by the use of encryption and decryption method it’s possible to increase the computational cost of both the client and the server. The server dynamically generates a software puzzle and sends that puzzle to the client. The client solves the given puzzle and returns the puzzle response. The server verifies the puzzle and the client can send the request to the server only if the puzzle solved by the client is correct. Client puzzle can be solved easily by using Graphic processing unit (GPU) software. The earlier client puzzle scheme assumed that the client solved the software puzzle using legacy CPU resource only. The client puzzle scheme in generates software puzzle dynamically. Hence it is assumed that the puzzle cannot be solved using GPU.

II. BACKGROUND

A. Puzzle-Based Mechanisms

Puzzle-based defence mechanisms correct the imbalance between the cost to the attacker for generating a request and cost to the server for processing a request by demanding a payment, in the form of a puzzle solution, from each client. There are different kinds of schemes that build on this general principle. Earlier puzzle based schemes can be solved

DOI 10.17148/IJARCCE

using Graphical Processing Unit software which consumes less time. The author in paper [] proposed a client puzzle scheme that dynamically generates software puzzles which cannot be solved using GPU software and is solved using legacy CPU resources only.

B. Time-Lock Puzzle Schemes

A time lock puzzle is a mechanism in which a sender publishes whose solution the message to be sent, thus is hiding it until enough time has elapsed for the puzzle to be solved. The original goal was to ensure that a client cannot decrypt a given message until a given period of time into the future has elapsed. To ensure that a client cannot simply throw more computing resources to solve the problem in a shorter period of time, the puzzles were designed to be non-parallelizable. Intuitively, using the solution of a time-lock puzzle as the key to an encryption scheme would force anyone wanting to decrypt the message to perform the computation for the time required to solve the puzzle. By tuning the difficulty of the solution according to the time we would like the message to remain secure, we can ensure that decryption will take at least that amount of time. In the paper [] a time-lock puzzles in the random oracle model is proposed. Due to the non-parallelizability property, this has been suggested as a puzzle-based scheme for DoS defence.

C. Memory-Bound Puzzle schemes

In this scheme the client solves puzzle by successively looking up values from a pre-computed table in the main-memory. Care must be taken to ensure that the table is not too small as it may completely fit in the cache of a higher-end computer. It may also not be too large as it may be more than the memory available to a lower-end device. It is also important that the memory access pattern (while solving the puzzle) is fairly random, otherwise it may lead to higher cache hits.

D. Bandwidth Based Scheme

In a bandwidth-based currency scheme clients use additional bandwidth to get access. It is often assumed that attackers are using all of the bandwidth available to them (or the maximum bandwidth they can afford to use without being detected by other mechanisms) to execute an attack, whereas legitimate clients are using only the resources they require to accomplish their less-demanding objectives. Hence legitimate clients have bandwidth to spare and can use this fact to reduce the attacker's chances of success.

III. OVERVIEW

The client puzzle can be classified into two types as Data puzzle and Software puzzle. In data puzzle, the puzzle scheme is known in advance and is fixed. Hence these 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 theclient 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. Therefore, an attacker can easily utilize the "free" GPUs or integrated CPU-GPU to inflate his computational capacity as shown in the figure below.

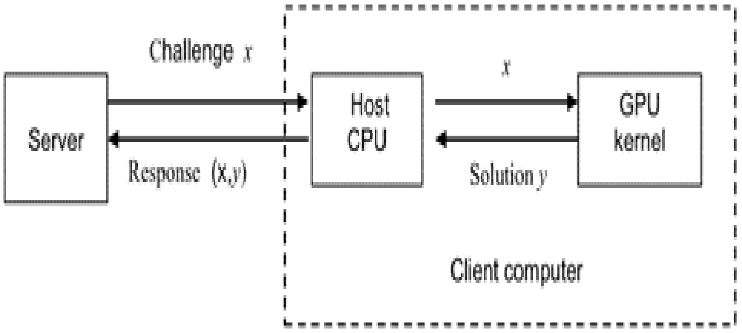

Fig 1: GPU-inflated DoS attack against data puzzle.

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 DoS attacks. To demonstrate the applicability of software puzzle.

IV. NETWORK SERVER MODULE

Network server module comprises of port initiation phase, request processing phase and response forwarding phase. In this server module, the server can receive the incoming requests from the connected clients and can process the request according to the server data and can forward the response to the respective clients. Port initiation phase involves defining the ports on which the server is to be remained open for accepting incoming connections. In request processing phase, the server can access the incoming requests from the connected clients from their respective service module. In response forwarding phase, the server can send response data to the clients for the appropriate requests from them. The server can perform data management service that is, uploading data which can be later viewed and requested by the clients.

V. NETWORK CLIENT MODULE

Network client module comprises of request module and response receiving module. In the requesting module, the network client can send the request, that is, file name, to the server for retrieving the content of the requested file after being processed by the server. In the response receiving module, the network client can receive the response from the network server and the data is stored to the client's allocated memory location.

VI. LATENCY CHECKING MODULE

In the proposed scheme, a latency checking module is also included. This module belongs to network clients. In this module, 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.

VII. COMMUNICATION REINITIALIZATION MODULE

In this module, after establishing communication between server and the network client and after each subsequent request and response transmission, the communication will be terminated and reinitialized, so that, intrusion is greatly prevented via this module, as it requires stable connection to hack or intrude. Along with the software puzzle the server sends the file requested by the client in encrypted form. The client has to solve the puzzle first and send response to the server. The server verifies the puzzle response and only if the response is correct the client will be able to decrypt and save the requested file that is send to the client by the server. Advanced Encryption Standard algorithm is used for the encryption and decryption. Here we use AES algorithm having key length of 256 bits. Hence it is possible to transfer large file size. AES algorithm is used in both client and the server side.

VIII. CONCLUSION

In this paper a software puzzle scheme with cryptographic technique is used to prevent DoS and DDoS attack. A dynamic 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 provide 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.

ACKNOWLEDGEMENT

I hereby first of all thank my guide **Mr. Kiran V.K** for his valuable suggestions and remarks .This work is done as part of the academic project work supported by teaching staff of NSS College of Engineering.I then thank all my friends and colleagues who helped me, my parents and almighty.

REFERENCE

[1] A. Juels and J. Brainard, "Client puzzles: A cryptographic countermeasure against connection depletion attacks," in Proc. Netw. Distrib. Syst. Secur. Symp. 1999

[2] T. J. McNevin, J.-M. Park, and R. Marchany, "A DoS limiting network architecture," Virginia Tech Univ., Dept. Elect. Comput. Eng., Blacksburg, VA, USA, Tech. Rep. TR-ECE-04-10, Oct. 2004.

[3] Sujata Doshi, Fabian Monrose, and Aviel D. Rubin Johns, "Efficient Memory Bound Puzzles Using Pattern Databases" J. Zhou, M. Yung, and F. Bao (Eds.): ACNS 2006, LNCS 3989, pp. 98–113, 2006. c Springer-Verlag Berlin Heidelberg 2006.

[4] J. Green, J. Juen, O. Fatemieh, R. Shankesi, D. Jin, and C. A. Gunter, "A Distributed Approach to Defend Web Service from DDoS Attacks ," in Proc. 4th USENIX Workshop Large-Scale Exploits Emergent Threats, 2011.

[5] Yongdong Wu, Zhigang Zhao, Feng Bao, and Robert H. Deng, " Software Puzzle: A Countermeasure to Resource-Inflated Denial-of-Service Attacks " Ieee Transactions On Information Forensics And Security, Vol. 10, No. 1, January 2015