ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 19, April 2016

Puzzle Security System against DOS and DDOS Attacks

[1]J.Jinu Sophia,[2] B. Jonah Sam Kishore,[3] M. Aswin Sundar,[4] M. Adhithen

[1]Asst. Professor, [2, 3, 4] UG Scholars, Department of Information Technology, Kings Engineering College

[1]jinu42@gmail.com,[2]jonahsamkishore04@gmail.com,[3]mvaswin94@gmail.com,[4]ebi4794@gmail.com

***ABSTRACT*: In this paper, software puzzle scheme is proposed for defeating GPU-inflated DoS attack. It adopts software protection technologies to ensure challenge data confidentiality and code security. A puzzle algorithm in the present software puzzle scheme is randomly generated only after a client request is received at the server side and the algorithm is generated In the present software puzzle, the server has to spend time in constructing the puzzle. In other words, the present puzzle is generated at the server side. An open problem is how to construct the client-side software puzzle so as to save the server time for better defense performance.**

I. INTRODUCTION

DoS and DDoS are effective if attackers spend much less resources than the victim server or are much more powerful than normal users. In this case, conventional cryptographic tools do not enhance the availability of the services; in fact, they may degrade service quality due to expensive cryptographic operations. In the puzzle-solving and verification steps, the client returns a puzzle response, and if the server confirms, the client is able to obtain the service from the server. In this hash-reversal puzzle scheme, a client has to spend a certain amount of time in solving the puzzle, and the server has to spend time in generating the puzzle and verifying the puzzle solution. When a client wants to obtain a service, she sends a request to the server. After receiving the client request, the server responds with a puzzle challenge. If the client is genuine, she will find the puzzle solution by directly on the host CPU and send the response to the server. On this type of puzzle the number of moves are very important, if the number of moves are less puzzle can be solved easily ,if the number of moves are more, then its hard to solve and it is high secure. Secure key establishment between two parties can be addressed when public key infrastructure (PKI) or an online trusted third party (TTP) is available. A software puzzle consists of instruction, and each instruction has a form. It will be delivered to the client who requests for services over an insecure channel such as Internet, and run at the client's side.

Extracting a shared sequence of bits by observing the same physical phenomenon has been addressed in several papers. Received signal power can be used to extract shared secrets between two peers. Multipath fading may introduce asymmetries on the received power but they proposed a few algorithms to recover the errors and eventually agree on a shared secret key. The accelerometer is used to verify if the two peers are carried by the same user. This algorithm involves to put the peers together shake them collecting values from the accelerometer, and finally, make a key agreement on the collected values. DoS and DDoS are effective if attackers spend much less resources than the victim server or are much more powerful than normal users .Each packet carries only one bit of the secret key and the packet source field is hidden; in this way, the adversary knows the value of the secret bit but she does not know the sender of that bit.

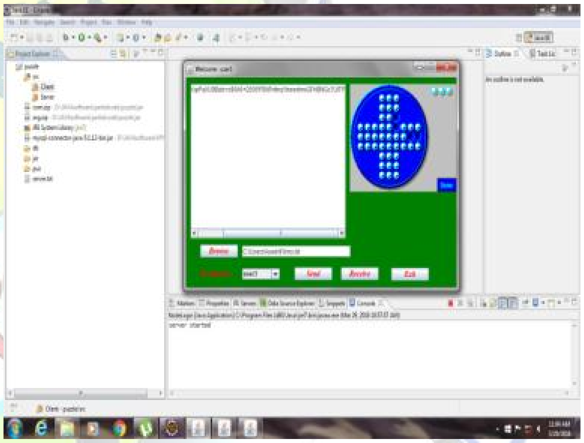
The seriousness of the DoS/DDoS problem and their increased frequency has led to the advent of numerous defense mechanisms. Unlike the existing client puzzle schemes which publish a puzzle function in advance, the software puzzle scheme dynamically generates the puzzle function in the form of a software core upon receiving a client's request. 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 puzzle scheme does. However, a malicious client may attempt to offload the puzzle-solving task. In this case, the malicious client has to translate the CPU software puzzle into its functionally equivalent GPU version because 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 puzzle on the host CPU directly; software puzzle threats the GPU-inflated DoS attacks.

II. STUDY METHODOLOGY

A: DATA PUZZLE

Once a software puzzle is created at the server side and check once before the puzzle was conformed, it will be delivered to the client who requests for services over an insecure channel such as Internet, and run at the client's side. When a software puzzle is built upon a data puzzle, the number of software puzzles is required to be very large such hat an attacker is unable to re-construct the software puzzles in advance and re-use them. Indeed, this requirement can be easily satisfied. For each time the puzzle vary from existing steps. The instructions are given by the sender of the file.

Figure1: puzzle generation



```
server started
```

B: COUNTER MEASURE

In this module, the bits are send by divided into different time slots and in each can perform one transmission. The users will try to send this bit to the other party via a single message based on the randomizing the source and the receiver address of the exchanged packets. Further, to make the slot contention fair, at the beginning of each slot each peer waits for a random time before trying to send its bit. Hence, that bit will be stored by both the peers and will constitute the established one secret bit. If the sender would have been A, the bit would have been complemented and later stored. For the adversary it is difficult to guess how the exchanged bit will be stored by both the peers. Indeed, the value corresponding to the bit has been stored as transmitted (or complemented) uniquely depending on the ID of the sender

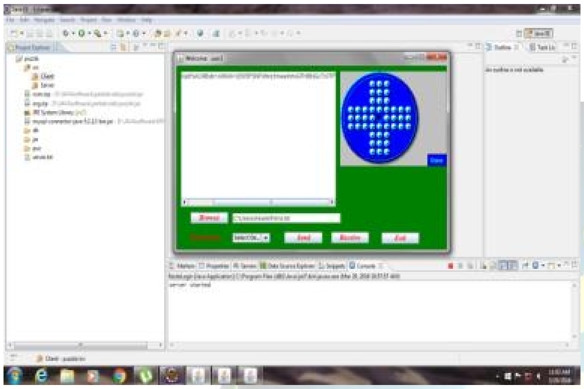
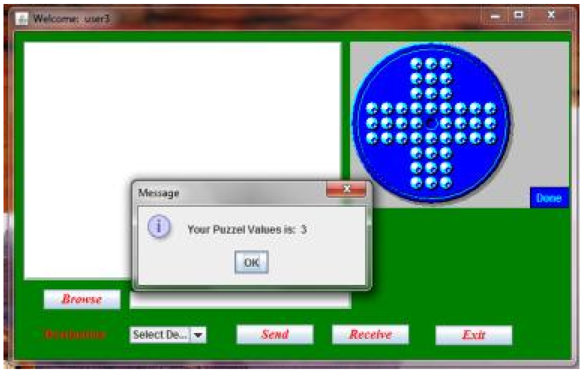


Figure 2: encryption of text

C: ACCESS PRIORITY

All the client puzzle schemes assume that there is no secure channel between the client and the server until puzzle verification completion. Otherwise, the client puzzle scheme is redundant. Thus, an attacker can intercept all the traffic between the client and the server, and start man-in-the-middle attack so sending malicious software puzzles to the client browser so as to launch attacks to the clients. However, an access policy should be defined so as to enable the software puzzle to call some special class generation functions. Hence, the attacker may have extra right to create new classes to make troubles to the clients. After the client successfully solved the puzzle the data will be received to the client side. Without solved the puzzle client cannot get the original file

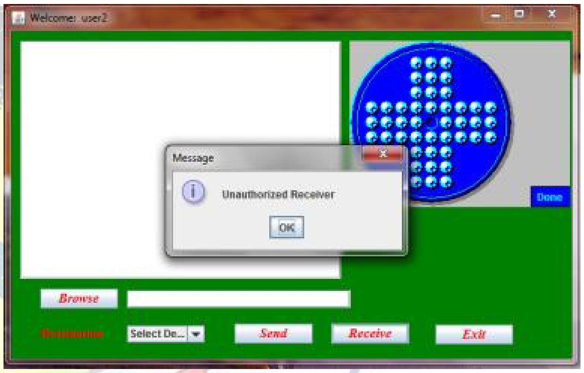
Figure 3: hint for authorized user



Welcome: user3

Message

Your Puzzel Values is: 3



Welcome: user2

Message

Unauthorized Receiver

Figure 4: unauthorized user try to gain file

D: MINIMIZE TRANSMISSION POWER

Our protocol introduces just a small transmission overhead when compared against other key-establishment solutions. Hence, our proposal is also an ideal candidate for those devices where computing capabilities are not a constraint, but energy consumption is, such as smart phones. The communicating peers are able to estimate the minimum transmission power which allows them to communicate each other's. Each node chooses a random power level to transmit the secret bit. The transmission power is chosen at random in the range, i.e., each node chooses a random transmission power between the minimum (guaranteeing the peer communication) and the maximum

III. PROPOSED METHODOLOGY

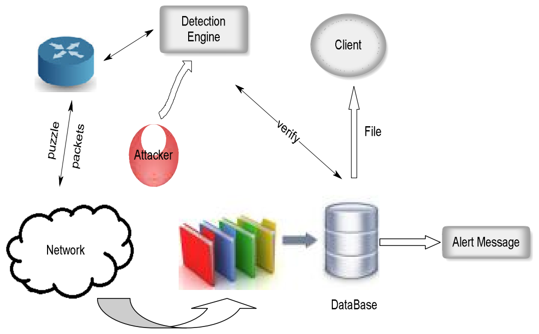
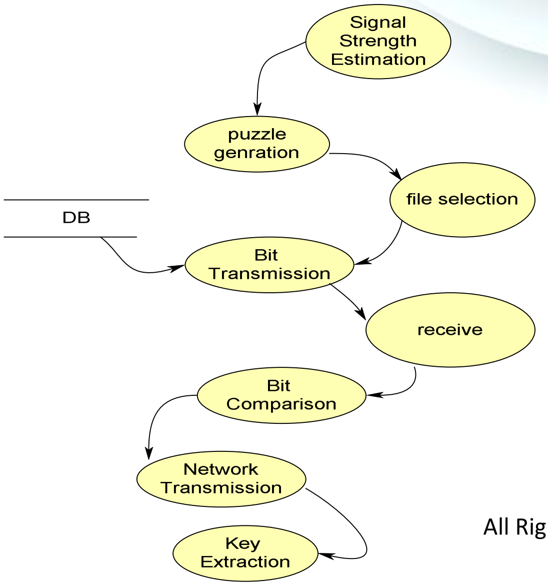


Figure:1 Architecture and mechanism

All the client puzzle schemes assume that there is no secure channel between the client and the server until puzzle verification completion. Otherwise, the client puzzle scheme is redundant. Thus, an attacker can intercept all the traffic between the client and the server, and start man-in-the-middle attack, sending malicious software puzzles to the client browser so as to launch attacks to the clients. The major threat to the security of our proposal is the adversarial distance with respect to the two communicating parties. We can mitigate the adversarial capabilities by tuning the transmission power of cooperating parties.

DFD for the mechanism:



IV. RELATED WORKS

Mitigating Bandwidth-Exhaustion Attacks using Congestion Puzzles was proposed by Xiao Feng Wang Michael K. Reiter present *congestion puzzles* (CP), a new counter measure to bandwidth-exhaustion attacks. Like other defenses based on client puzzles, CP attempts to force attackers to invest vast resources in order to effectively perform denial of-service attacks. Christo Ananth et al. [3] discussed about a method, End-to-end inference to diagnose and repair the data-forwarding failures, our optimization goal to minimize the faults at minimum expected cost of correcting all faulty nodes that cannot properly deliver data. First checking the nodes that has the least checking cost does not minimize the expected costin fault localization. We construct a potential function for identifying the candidate nodes, one of which should be first checked by an optimal strategy. We proposes efficient inferring approach to the node to be checked in large-scale networks.

Defend Dos Attack with Geometric Hashing Function and Software Puzzle was proposed by Kiruthika Rengaraj, Matheshwaran Veerappan, DOS attack can be exhaust the target server's resources have become major threat to today's internet. Even though client problem represent a promising way to defend against certain classes of DOS attack, several questions stand in the way of their use in practice. Existing software puzzle system issues their puzzle algorithm already. The attackers can easily solving the puzzle using puzzle solving software. We proposed a new client puzzle software algorithm that can choose puzzle algorithm randomly during the request form the client. The attacker cannot show progress since server chooses the puzzle algorithm randomly.

We implement an efficient password authentication scheme based on a geometric hashing function in order to improve the confirmation. A single message prepared by the client and sent to the server, and a confirmation performs by the server. It efficiently defends beside DOS, verifier-stolen attack, replay attack, password guessing attack, man-in-the-middle attack.

V. CONCLUSION AND FUTURE ENHANCEMENT

This paper has the step to reduce the DOS/DDOS attacks, the key establishment is done in a secure way, the AES encryption algorithm is used. This paper has a solution for GPU-Inflated DOS attacks the hints are given for the beta client to avoid the complexity in solving the puzzle, the number on moves makes the puzzle so complex. The I.P address is given initially for the destination client. Future enhancement can be made in this area i.e the puzzle security for sending and receiving file in different I.P address.

REFERENCES

[1] J. Larimer. (Oct. 28, 2014). Pushdo SSL DDoS Attacks. [Online]. Available: http://www.iss.net/threats/pushdoSSLDDoS.html

[2] C. Douligeris and A. Mitrokotsa, “DDoS attacks and defense mecha- nisms: Classification and state-of-the-art,” Comput. Netw., vol. 44, no. 5, pp. 643–666, 2004

[3] Christo Ananth, Mary Varsha Peter, Priya.M., Rajalakshmi.R., Muthu Bharathi.R., Pramila.E., “Network Fault Correction in Overlay Network through Optimality”, International Journal of Advanced Research Trends in Engineering and Technology (IJARTET), Volume 2, Issue 8, August 2015, pp: 19-22

[4] T. J. McNevin, J.-M. Park, and R. Marchany, “pTCP: A client puzzle protocol for defending against resource exhaustion denial of service attacks,” Virginia Tech Univ., Dept. Elect. Comput. Eng., Blacksburg, VA, USA, Tech. Rep. TR-ECE-04-10, Oct. 2004.