ISSN (online) : 2277-5390
ISSN (Print): 2277-5382

Enhanced software puzzle with client puzzle technique

B.Rajiv [*1], M. Vasanthi[2]

[1]*Student,* [2]*Assistant Prof., Department of Computer Science, Sri JayendraSaraswathyMahaVidyalaya College of Arts and Science, Coimbatore, Tamil Nadu, India*
rajivmphil2016@gmail.com

Abstract

Objectives: To reduce time consumption and overhead occurred at client side by creating puzzles at client side in Denial of Service (DOS) and Distributed Denial Of Service (DDOS).
Findings: In puzzle solving technique, first the server generates some puzzles and sends to the client. Then the client solve the puzzle and sendsback to the server and finally the server confirms the client to obtain the service from the server. In such type of puzzle solving technique, the attackers can merely reply to the server and it cannot be identified in this puzzle solving technique. Graphics Processing Unit (GPU)-inflation DoS publishes puzzle function in advance. It can be enhanced or overcome by using software puzzle solving technique as it dynamically generates puzzle function. This technique exploits the architectural difference between CPU and GPU. If an attacker tries to move a puzzle transition task into CPU, either they want to translate into functionally equivalent GPU or it should do at dynamic puzzle generation. It is time consuming process to translate or rewriting a software puzzle. This software puzzle has some drawbacks as it cannot carry out puzzle solving process on cloud environment and it doesn't construct client side software puzzle.
Methods: In order to overcome time consuming and overhead problem here we proposed a new client puzzle technique which is based on two puzzles between clients.
Applications/Improvements: To reduce the time consumption and overhead problem new client puzzle technique is proposed.

Keywords: Denial of service, distributed denial of service, software puzzle, puzzle technique.

1. Introduction

In an emerging current computer network, the Denial of Service (DOS) has become a major threat. It is a class of attack instigated by individual or group of individuals which makes use of aspects of the Internet Protocol to reject other users from genuine access to information and systems. Nowadays, most people became aware of the threats of DOS attacks in frequently used websites such Amazon off the air, Yahoo and CNN. There are various techniques are developed to detect and prevent DOS attack in a network [1]. The existing software puzzle technique [2] prevents GPU from being used to accelerate the puzzle-solving process which includes dynamically generated software. It doesn't reveal the puzzle function in advance like data puzzle. Additionally, it used Kerckhoffs's Principle to construct software puzzle. In software puzzle framework, it consists of code block warehouse which contains various software instructions [3] and there are two modules one for puzzle generation and another one for hiding the puzzle for security purpose.

In order to overcome the overhead problem at server side introduced new client puzzle technique, it also prevents DOS attack. In this, every client in a network has to solve two puzzles to contact other client. The first one is Discrete Logarithm Problem (DLP) which is solved by a client and produces a solution based on the trust level and the second puzzle is created based on the solution of DLP as well as a random number which is done by connection initiator. The solution of DLP puzzle can become more difficult or easy depends on the trust level of client. Every client need to follow the same procedure to find a malicious attack.

In [4] proposed a technique to build path table for preserving DDoS attack. The proposed technique created a path ID for both source router and destination router and it is independent of number of attackers in the network and attack duration. The page ID is created based on routing topology information which intimates the attack by notification message generated by the victim and it sends back to the nearer attacker along with the page ID. It is implemented on Click modular router software platform to validate the feasibility of the proposed technique. Thus, this technique found large number of attackers in the network with smaller bandwidth.

In [5] proposed currency based mechanism to reduce Denial of service attack in a network. It is achieved by using resource fairness among challengers. In this resource fairness server allocates its resources or services to the client in quantity to their payment of a resource. The countermeasures for Denial of service attack were the attackers determine ways to maximize their ownership of the payment needed for getting service from the server. The resource inflation attacks proved that an attacker can utilize GPUs, multi-core processors, and cloud computing to inflate its resource payments.

In [6] investigated Hash Reversal Power of Work with different defense mechanism of resource inflation attack in a network. In the hash reversal puzzle scheme the server sends a puzzle to the client. After receiving a puzzle the client found solution for the puzzle by computing hash function through the seed value. The hash function is performed with set of parameters and difficulty level of the puzzle then the server verifies the solution from the client by checking last bit of hash function is zero if it is zero the solution provided by the client is correct. It has a major advantage as data parallel computation of the puzzles.

In [7] introduced a new approach for solving client puzzles. It overcomes the problem of client puzzles scheme of coarse-grained or parallelizable or it can be utilized only inter relatively. The proposed Modular square root puzzles offered polynomial granularity by non parallelizable of client scheme. It can be utilized both interactively and non-interactively to provide the granularity for client puzzle. In this method, server builds a puzzle for a request by allocating a unique quadratic residue modulo a prime and the puzzle solved by a client with the help of modular square root. Then the solution of the puzzle is verified at the server side that needs only few hash function and a single modular operation.

In [8] explained a mechanism for hash based puzzle for web service defending from denial of service attack. This mechanism used client puzzles to resolve the problem of resource imbalance by solving puzzles provided by the client and verified to prove client's genuine intention in requesting services and it also offered DoS mitigation capability which is combined with any web service application without the help of additional components.

In [9] proposed a defense mechanism of denial of service attack in a network through game theory concept. In this mechanism random number generators, difficulty level of puzzle and other parameters involved in puzzle-based defense of game theory were adjusted by the concept of Nash equilibrium. This mechanism provides maximum payoff for the attackers and it is more effective defending mechanism for denial of service attack.

In [10] proposed a new mechanism called leaky bucket rate limiting queue to set the difficulty level of puzzle generation to detect and prevent the denial of service attack in a network. It sets difficulty level of puzzle based on queue delay and it will charge limit the number of request provided by the different clients that prevents the DOS attack. Thus the rate of successful attack is reduced by increasing the difficulty level of puzzle for attackers and they take more time to solve the harder puzzles.

In [11] proposed cryptographic puzzles based on modular exponentiation mechanism for denial of service mitigation. In this mechanism RSA is used to fundamentally the difficulty of calculating a small private exponent when the public key is bigger by various orders of magnitude than the semi-prime modulus. It decreases the cost acquired on the puzzle generator in offered modular exponentiation puzzles.

2. Materials and Methods

Denial of service attack can be detected and prevented by creating and solving puzzles at client sides. It reduces the overhead of server side puzzle generation and verification. In this proposed technique, server acts as a middle man to exchange the information between clients. It creates a decentralized design where connection initiator in the client side is responsible for puzzle generation and puzzle solving as well as verification of puzzle is also carried over at client side.

2.1. Puzzle construction and verification at client side

In this proposed technique, the puzzles are created and handled at client side through game theory which is used for decision making. There are different and various numbers of players involved in game theory represents the number of client and each and every player has set of strategies and a payoff for each combination strategies. It can be explained in Figure 1.

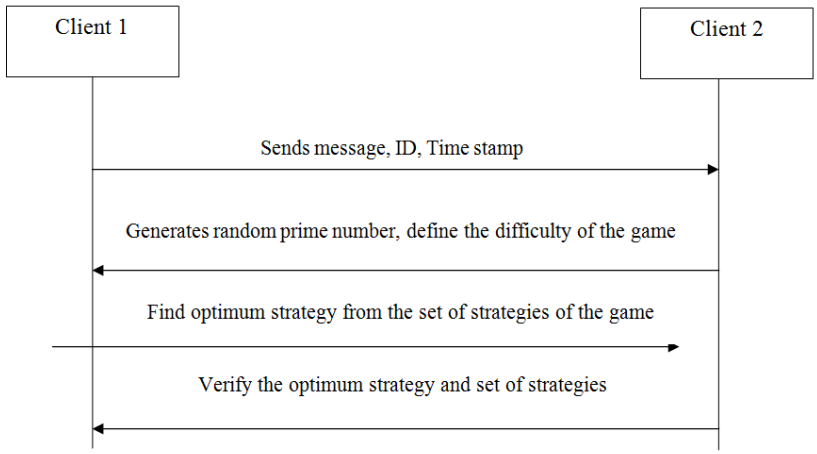
2.2. Puzzle construction

Puzzle construction is the process of creating puzzles at client sides to communicate each other. Figure 1 shows the workflow of puzzle generation and puzzle verification. If client 1 wants to communicate with client 2, client 1

ISSN (online) : 2277-5390
ISSN (Print): 2277-5382

sends three parameters: hello message, its ID and time stamp. After receiving these parameters from client 1, client 2 creates random prime number R, random integer y and a generator h ofY_h^*. Then client 2 set the difficulty level l of the puzzle that client 1 will have to solve the puzzle. In the next step, client 2 determines the value of S and utilize baby step giant step algorithm to find the value of h, overY_h^*. S is calculated by using following equation:

Figure 1. Work flow of puzzle construction and verification



$$S = h^y mod R \qquad (1)$$

Algorithm 1:

Baby step giant step algorithm
Input: A cyclic group h of order m having a generator Y and an element β.
Output: value satisfying $Y^x = \beta$

1. n= ceiling ($\sqrt{m}$)
2. i=0 to n
3. for all i
4. calculate Y^i and store the pair (i,Y^i) in a teble
5. End for
6. Calculate Y^{-n}
7. μ=β
8. for j=0 to n-1
9. Check if μ is the second component (Y^i) of any pair in the table.
10. If $Y^i$is the second component
11. return jn + i.
12. else μ =μ • Y^{-n}
13. end if
14. end for

From the algorithm, the value of $Y_h^*$was found which is simple modification of trial multiplication for computing discrete logarithm. The value of Y^i is calculated and store it in a table and verify μ is the second component return the value jn +i otherwise return μ• Y^{-n}.

The calculated values of S, h, R and Y_h^* send to client 1then it calculates number z that is needed to create the strategies of the puzzle by the integer division of random number and difficulty level. Then the client 1 generates and solves the puzzle after generating z and finds the optimum strategy by calculating all strategies. The strategies of each client are described as the addition of three positive integers (z1, z2, z3) which are in non descending order.

z1+z2+z3=z (2)

Each puzzle will have number of strategies based on the client. The number of strategies in each puzzle is calculated by using equation 3. For all Z ϵ N_+^* the number of strategies is:

$$\Pi(z) = \frac{1}{2} \times \left[\frac{z}{3}\right] \times z + \frac{1}{8} \times (-1)^{z-3\left[\frac{z}{3}\right]} + \frac{1}{8} \times (-1)^{1+z} - \frac{3}{4} \times \left[\frac{z}{3}\right]^2 \quad (3)$$

The client 1 found all possible strategies of the puzzle through the equation 3 and it can able to find the optimal strategy from the set of strategies. It can be found by client 1 with the help of utility function where every strategy of each puzzles plays against all other strategies. The input of utility function is two different strategies where the client 1 compares each strategy by itself. The utility function is performed until all possible combinations have been played. The two different strategies are represented by M= (z1, z2, z3), $M' = (z1', z2', z3')$ and the possible results of utility function are M, DRAW, and M'. It returns M as optimum strategy if strategy M has at least two higher coordinates than the corresponding ones of strategyM' and it returns DRAW when the two input strategies have one equal and two different coordinates. It returns M' only if M' has at least two higher coordinates than the corresponding ones of strategy M. Thus, the client 1 finds the optimum strategy and sends it back to client 2 along with the set of strategies, random number y and its ID. Similarly, the client 2 generate puzzle and set of strategies and found the optimum strategy and it sends back to client 1 to verify the puzzle.

2.3. Puzzle verification

The puzzle verification process creates puzzle, set of strategies and the optimum strategies are verified by another client. If the verification of puzzle is validated then the information is sent through the server otherwise it won't sent the information to requested client i.e it detects attacker in the network by the verification process. Client 2 verifies that equation (3) holds for all possible strategies and used an algorithm 2 to verify whether the client 1 has correctly found set of strategies and optimum strategy or not.

Algorithm 2:
Verifying puzzle at client side
Input: random integer, set of strategies, optimum strategy OPT, ID, X(set of strategies,OPT)
Output: connect the client or disconnect the client

1. if (OPT[0]>$X_i[0]$and (OPT[1]>$X_i[1]$ or OPT[2]>$X_i[2]$))or (OPT[1]>$X_i[1]$and (OPT[0]>$X_i[0]$ or OPT[2]>$X_i[2]$))
2. OPTwins ++
3. end if
4. if $X_{i-1}[1]$<$X_i[1]$
5. sum+=$X_i[0]$
6. end if
7. else
8. exsum+=$(X_i[1] - j).X_i[0]$
9. if exsum==sum & OPTWINS≥ $\frac{\Pi(Z)}{2}$
10. give resources
11. end if
12. else
13. drop connection
14. end else
15. end else
16. end begin

From the algorithm 2, the created puzzles are verified at the end of both clients if it satisfies the condition the connection will be granted between the two clients otherwise the connection will be dropped by the server.

2.4. Information exchange by server

The verification results of both the client sent to server if the verification process is granted then the server allows the clients to exchange the information. The server acts as middle man between clients for information

ISSN (online) : 2277-5390
ISSN (Print): 2277-5382

exchange. Thus the puzzle generation and puzzle verification are carried over at the client. When two clients want to communicate with each other, they generate the puzzle and set of strategies individually and verify the puzzle and strategies generated by the other client thus the overhead at server side was reduced.

Algorithm 4:
Information exchange by server
Input: verification result from the both the clients

Output: connection result

1. Get the verification result from both the clients
2. if result=give resources
3. Connect the clients
4. else
5. Drop the connection
6. End if

Thus the server side overhead is reduced and computational cost for puzzle verification is reduced by the client side puzzle generation and verification.

3. Results and discussion

The results of the existing and the proposed puzzle techniques are performed and in order to prove the effectiveness of the proposed new client side techniques, number of served clients vs number of malicious request per second and time is compared with existing technique.

3.1. Performance evaluation

The performance measure shows the proposed client puzzle can increase the service quality significantly in terms of the percentage of served customers.

Figure 2 shows that the proposed client puzzle can increase the defense capability against time DoS attack at 2 number of malicious request per second server puzzle provide 32% of served clients and at the same number of malicious request per second client puzzle provides 70% of served clients. It is proved that the proposed technique shows high performance than the existing technique.

Figure 2. Performance graph

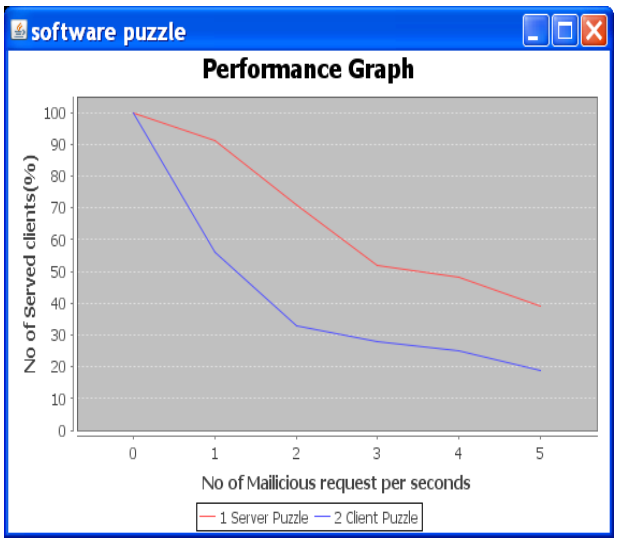
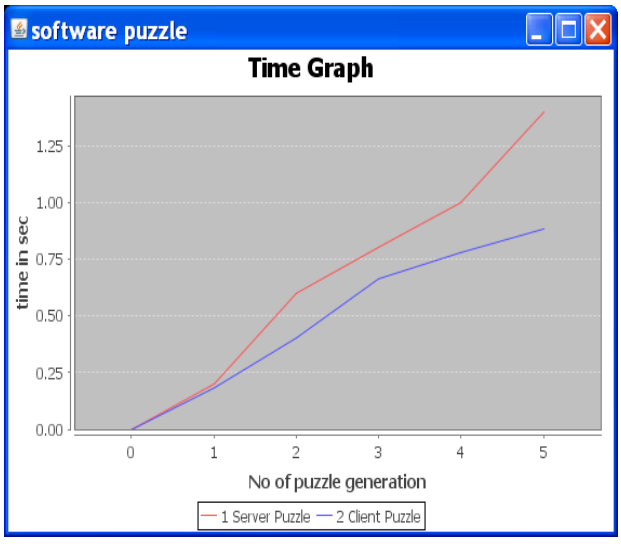


Figure 3. Time graph



ISSN (online) : 2277-5390
ISSN (Print): 2277-5382

3.2. Time

Time measure calculates the time taken for puzzle generation at client side. It measures the time against number of puzzle generation. If the network has more number of attacker's servers takes more time to generate the puzzle with high difficult level.

Figure 3 shows that the proposed technique takes lesser time for number of puzzle generation than the existing technique. At 3 number of puzzle generation server puzzle takes 0.78 secs to generate the puzzle whereas in client puzzle technique 0.68 secs to generate the puzzle. This proved that the proposed detects and prevents the DOS attack more effectively than the existing technique.

4. Conclusion

This paper presented an efficient technique called client puzzle which is used to detect and prevent the malicious threat denial of service attack. This also reduces the overhead occurred at server side by creating and verifying puzzles at client side. The server act as middle man between clients to transmits the information. Based on the verification results of client puzzles, the resources are allocated for the client otherwise it will drop the connection between clients. The experimental results prove that the proposed client puzzle technique works more efficiently than the existing technique in terms of time and the number of served clients.

5. References

1. Subramanirao, Sridhar Rao. Denial of service attacks and mitigation techniques: Real time implementation with detailed analysis. *This paper is from the SANS Institute Reading Room site*. 2011.
2. Y. Wu, Z. Zhao, F. Bao, R.H. Deng. Software puzzle: A countermeasure to resource-inflated denial-of-service attacks. *IEEE Transactions on Information Forensics and Security*. 2015; 10(1), 168-177.
3. M. Maghzi, E. Hadizadeh. Dynamic modeling of the water pumping station in water treatment plant using the bond graph method. *Indian Journal of Innovations and Developments*. 2012; 5(11), 787-794.
4. J. Eun, H. Jung. A technique to make a path table for blocking Distributed Denial-of-Service attacks. In: *2015 9th International Conference on Future Generation Communication and Networking (FGCN),IEEE*. 2015; 13-16.
5. R. Shankesi, O. Fatemieh, C. A. Gunter. Resource inflation threats to denial of service countermeasures. http://seclab.illinois.edu/wp-content/uploads/2011/03/ShankesiFG10.pdf. Date accessed: 2011.
6. J. Green, J. Juen, O. Fatemieh, R. Shankesi, Dong (Kevin) Jin, C. A. Gunter. Reconstructing Hash Reversal based Proof of Work Schemes. In *LEET*. 2011.
7. Y.I. Jerschow, M. Mauve. Non-parallelizable and non-interactive client puzzles from modular square roots. In: *2011 Sixth International Conference on Availability, Reliability and Security, IEEE*. 2011, 135-142.
8. S. Suriadi, D. Stebila, A. Clark, H. Liu. Defending web services against denial of service attacks using client puzzles. In: *2011 IEEE International Conference on Web Services (ICWS), IEEE*. 2011; 25-32.
9. M. Fallah. A puzzle-based defense strategy against flooding attacks using game theory. *IEEE Transactions on Dependable and Secure Computing.* 2010; 7(1), 5-19.
10. J.Y. Koh, J.T.C. Ming, D. Niyato. Rate limiting client puzzle schemes for denial-of-service mitigation. In *2013 IEEE Wireless Communications and Networking Conference (WCNC)*. 2013, 1848-1853.
11. G.O. Karame, S. Čapkun. Low-cost client puzzles based on modular exponentiation. In *European Symposium on Research in Computer Security.* Springer Berlin Heidelberg. 2010; 679-697.

The Publication fee is defrayed by Indian Society for Education and Environment (iSee). www.iseeadyar.org

Citation:

B. Rajiv, M. Vasanthi. Enhanced software puzzle with client puzzle technique. *Indian Journal of Innovations and Developments.* 2016; 5 (8), August.