International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

Available online: www.ijmrset.com e-ISSN: 2394-8752, p-ISSN: 2394-8791

Volume-03, Issue-09, September 2016

Implementation of Different Software Puzzle Methods against DDOS Attack

Smita Miraje [1], Manisha Bharati [2]

PG Student, Dept. of Computer, Savitribai Phule Pune University, Indira College of Engineering and Management, Pune, Maharashtra, India[1]

Professor, Dept. of Computer, Savitribai Phule Pune University, Indira College of Engineering and Management, Pune, Maharashtra, India[2]

ABSTRACT: In today's internet world, Denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks are very harmful and hazardous attacks. These attacks are main intimidation to cyber security. DoS and DDoS attacks diminish an online service resource by crushing the service with bogus requests and prevent genuine clients from accessing a service. Introduce a new defense that submits to as a software puzzle. The fundamental idea is as follows. In a software puzzle defense mechanism, the client needs to solve a complex structure puzzle before they can establish a connection with a server for accessing the services from the server. In this paper, I have proposed a new software puzzle defense mechanism in advance such that: 1) implement software puzzle using three different puzzle formation method 2) an attackers are incapable to done an implementation which is required to solve the puzzle and 3) provide higher security with code obfuscation, an attacker requires considerable effort to understand puzzle in question format. Here, present how to software puzzle mechanism implement partially at both side (i.e. at server side and client side) in detail and present effectiveness of this defense mechanism against DoS and DDoS attack.

KEYWORDS: Software puzzle, code protection, denial of service attacks (DoS), encryption, SPEKE, RC7.

I. INTRODUCTION

In the network denial of service (DoS) and distributed DoS (DDoS) attacks intend to prevent legitimate clients from accessing services are considered a serious hazard to the availability and reliability of the internet services. For example, server receives huge number of junk request from malicious client. For each request, server has to waste extra CPU time for completing process of SSL handshakes .Server cannot handle requests of services from its true customers because it may not have enough resources to handle the request. As a result of this attack is vanished businesses and reputation lost [1].

Represented an advance mechanism that refers as the software puzzle, the aim of this mechanism is to prevent DoS or DDoS attacks and provide services to valid clients. The idea is quite simple. When a client wants to acquire a service from the server, client sends a simple request to the server. After getting the client request, the server sends one puzzle challenge to client. Client must first solve a complex structure puzzle correctly and submit it to the server for accessing services. Server verifies this puzzle solution, if it is correct then server agrees to establish connection with client [3]. To solve this puzzle by every client, prevent vulnerable connection.

A software puzzle is different kinds of methods or complex structure or problem which uses sequence of steps and solving these steps client can access resources. Timestamp, data length, key length and software puzzle complexity these attributes are used for security purpose in puzzle generation process and generates puzzle dynamically. I have used the SPEKE algorithm for key generation; it provides high level security and thwarts man-in-middle attack by password [8]. Implement the RC7 algorithm for encryption purpose. It provides best result in case of throughput and time consumption and provides high level security [6].

II. RELATED WORK

Software puzzle mechanism is introduced here to prevent GPU-inflated DoS attacks. For this purpose, code block warehouse has generated and code obfuscation technique is used. AES (Advanced Encryption Standard) algorithm is flexible and used here for encryption purpose but it provides poor performance in case of throughput and time consumption. Software puzzle implementation is done at server side and server has to take extra time for puzzle verification. So, needs to implement puzzle mechanism partially to save the server time which waste in puzzle verification process. [1].

Available online: www.ijmrset.com e-ISSN: 2394-8752, p-ISSN: 2394-8791

Volume-03, Issue-09, September 2016

The goal of work is to encrypt a message such a way no one can be decrypted, until a specific time amount has passed which is pre-determined. Message is encrypted using RC5 (Rivest's Cipher) encryption algorithm. RC5 provide poor performance as compared to RC7. The CPU time is required for problem solving process and this time depends on the amount and hardware nature which is used to solve this problem. This is the problem of using the time-lock puzzle [2].

In client puzzle defense mechanism, a small cryptographic problem has created. The client puzzle protocol is robust in a stronger attack model and this model capable to prevent attacks increased at very fast speeds. The limitation of this client puzzle protocol approach, it requires that the client already have a program which is capable of solving a client puzzle [3].

A novel customer riddles convention that uses some change of the Extended Tiny Encryption Algorithm. The client puzzle protocol is effective defense mechanism for resource exhaustion DoS attacks within the transport layer. XTEA6 algorithm is very faster and it contains 6 cycles [4].

The currency based defense mechanism is to achieve resource fairness. Two classes of currency-based mechanism: puzzle-based and bandwidth-based. [5].

The Proof of Work Scheme is used which based on hash reversal puzzles. In PoW schemes maintain track of each client behaviour and according to that behaviour increase puzzle difficulty. To increase puzzle difficulty Bloom filter algorithm is used [10].

This paper describes a new scheme for client puzzle which is mainly based on modular square root. To compute the modular square root requires firstly modular exponentiation which is implemented by binary exponentiation method and k-ary method. Further modular square root estimated using Tonelli-Shanks and Cipolla-Lehmer algorithm [11].

III. PROPOSED SYSTEM

The goal of this work is to defense against DoS attacks and provide efficiency. First system architecture is presented and then working of each module is described. Figure 1 demonstrates that different steps.

Software puzzle mechanism contain code block warehouse at the server side. This warehouse stores different types of software instruction blocks. Implementation is done in two parts that is first generate the puzzle C0x and provide high level security to C0x and generate secured puzzle C1x. Puzzles are generated using different mathematical operations with various attributes for security purpose.

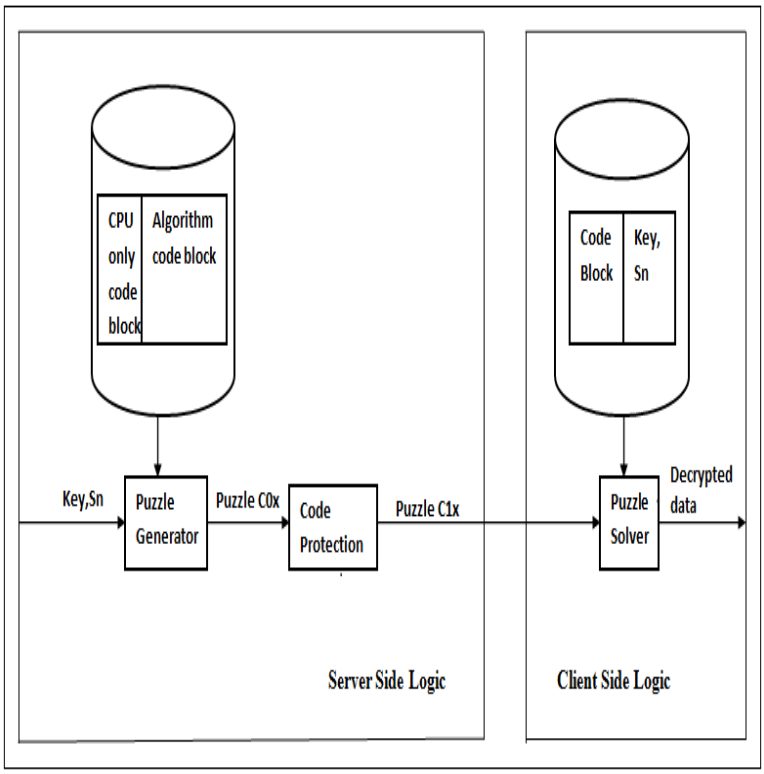


Fig.1. System Architecture

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

Available online: www.ijmrset.com e-ISSN: 2394-8752, p-ISSN: 2394-8791

Volume-03, Issue-09, September 2016

1. Warehouse Block Structure

All compiled instruction blocks are stored in warehouse which is in java byte code format. This instruction blocks save server time otherwise server has to spend lot of time in converting source code into compiled code. In this warehouse, each block is depended on security parameter and size of each puzzle is in fixed size. If the block size is smaller then it provides more security level. Because of smaller block size attacker has to spend extra time to understand puzzle in question.

In warehouse block structure, code block contain two categories: CPU code block and algorithm code block.

1. CPU code block

It contains all instruction set which is required in puzzle generation process. Which different activities are performed during puzzle generation process is presented in CPU code block.

2. Algorithm code block

It contains all operations related to encryption algorithm and stores all mathematical operations.

2. Software Puzzle Generator

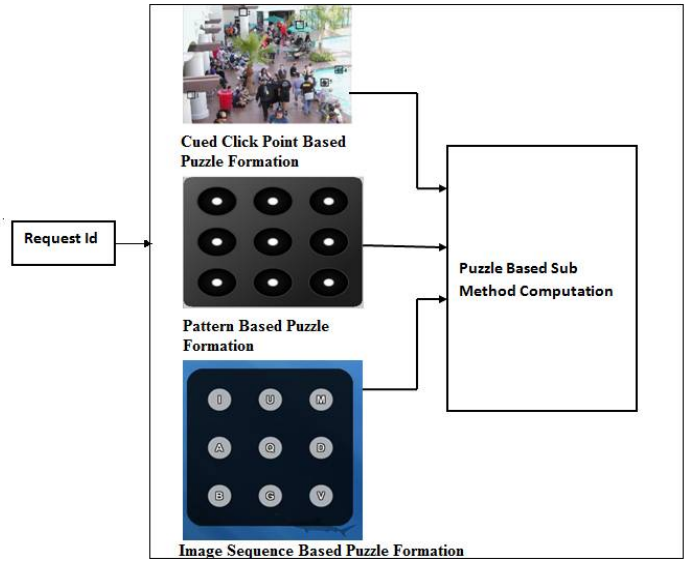


Fig.1. puzzle formation methods.

Puzzle construction is a main part of implementation. Here, puzzle problem has created using different puzzle formation method. In that cued click point, pattern based and alphabets image based methods has used.

3. Code Protection

Provide code protection using code obfuscation technique. Code obfuscation means creating something which is lesser to clear and harder to understand. Here, for code protection RC7 algorithm is used.

4. Puzzle Solver

Here, puzzle solved by client is verified using puzzle solver to save the server time. Puzzle solver is implemented at client side and it will allocate resources to the client if the client finds out correct puzzle solution.

IV. METHOD AND ALGORITHM

1. SPEKE Algorithm

1. Alice and Bob have the same opinion to use p. p is a large and randomly selected prime. Alice and Bob also select hash function H ().
2. Alice and Bob have the same opinion to use shared password π.
3. Alice and Bob both compute a generator by squaring the hash of shared password.
4. Alice computes g^a mod p by selecting a secret random integer a and sends to Bob.
5. Bob computes g^b mod p by selecting a secret random integer b and sends to Alice.
6. The received values are not in the determined range [2, p-2] then Alice and Bob both will terminate to prevent some imprisonment attack.
7. Alice computes K=$(g^b \ mod \ p)^a$ mod p.
8. Bob computes K=$(g^a \ mod \ p)^b$ mod p.

Available online: www.ijmrset.com e-ISSN: 2394-8752, p-ISSN: 2394-8791

Volume-03, Issue-09, September 2016

SPEKE algorithm is a Simple Password Exponential Key Exchange. SPEKE is used for key agreement in password authentication method. In SPEKE algorithm both Alice and Bob computes the shared secret key K. In SPEKE algorithm the communication between Alice and Bob is secured and an attacker cannot learn the shared secret key K.

2. RC7 Algorithm

- Input: Plaintext

  Six w-bit input registers are "H, I, J, K, L and M".
  R is a number of rounds
  W indicates word size in bits.
  r - Number of rounds.
  W -bit round keys "C [0 . . . 2r+1]"

- Output: Cipher text

  Output stored in H, I, J, K, L, M.

- Procedure:

{
\\Declare three global registers and initialize data.
I = I + C [0]
K = K + C [1]
M = M + C [2]
for n = 1 to r do
\\It is no of rounds according to word length
{
x = (I × (2I + 1)) <<<lg w
y = (K × (2K + 1)) <<<lg w
z = (M × (2M + 1)) <<<lg w
\\Addition and left shift operations are performed.
H = ((H ⊕ x) <<<y) + C [2n+1]
J = ((J ⊕ y) <<<x) + C [2n+2]
L = ((L ⊕ z) <<<x) + C [2n+3]
\\Perform XOR with left shift operation.
(H, I, J, K, L, M) = (I, J, K, L, M,H)
}
H= H + C [2r - 1]
J = J + C [2r]
L = L + C [2r + 1]
}
Cipher texts are stored in H, I, J, K, L and M.

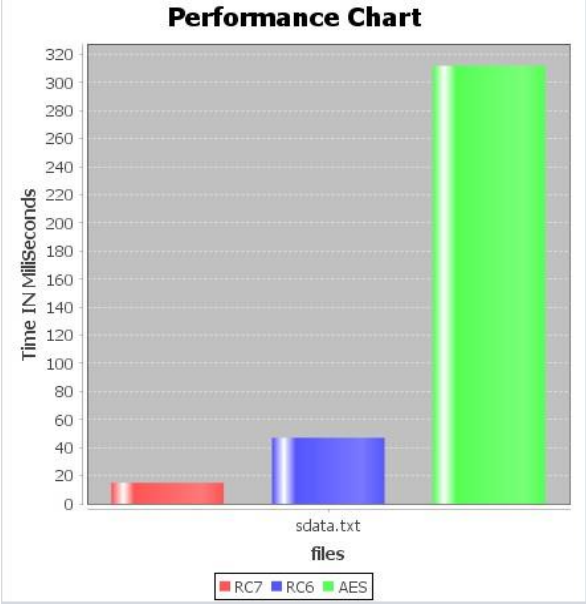
RC7 algorithm is a new extension of RC6 algorithm. As compared to RC6 algorithm RC7 has to takes less compilation time and encryption time.RC7 provides best performance as compared to other algorithm. RC7 algorithm is more flexible and provides efficiency in all application.

V. EXPERIMENTAL RESULTS

AES takes more compilation and encryption time due to maximum rounds as compared to RC6 algorithm as indicated by bellow comparative analysis graph 1. RC6 contains 4 bit working registers therefore it provides simple, fast and secure result.

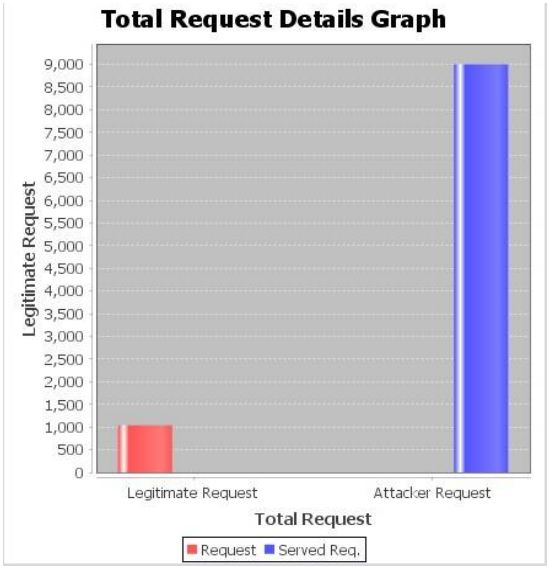
Available online: www.ijmrset.com e-ISSN: 2394-8752, p-ISSN: 2394-8791

Volume-03, Issue-09, September 2016



Graph 1: Algorithm comparison result

RC6 utilizes less compilation and encryption time as compared to AES algorithm. RC7 contain 6 bit working registers and takes less compilation and encryption time as compared to AES and RC6. It is more flexible and contains additional functionality than RC6 hence RC7 algorithm is used for implementation of puzzle mechanism to prevent DDoS attacks. Graph 1 shows sdata.txt file has taken for encryption. RC7 takes 16 milliseconds and RC6 takes 45 milliseconds, AES takes 310 milliseconds. Here, RC7 takes less time for encryption.

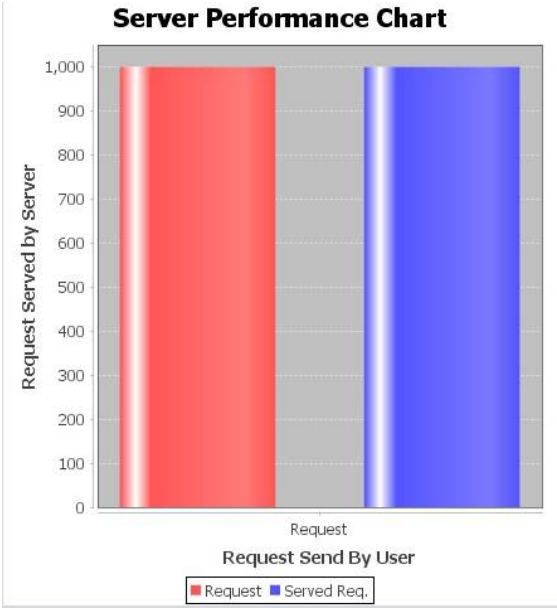


Graph 2: Total legitimate request and attacker request

Graph 2 shows total attacker request as compared to legitimate request which is stored in database. An attacker tool sends thousands request at a time as compared to legitimate client. Graph shows attacker request 9000 in database and legitimate client request 1200 only.

Available online: www.ijmrset.com e-ISSN: 2394-8752, p-ISSN: 2394-8791

Volume-03, Issue-09, September 2016



Graph 3: Request served by server per seconds

Graph 3 shows request served by server per second. Server can handle all client requests without failure. Using different puzzle methods server served all client requests.

VI. CONCLUSION AND FUTURE WORK

In implementation of different software puzzle methods against DDOS attack, software puzzle mechanism is provided to prevent DDoS attack. It reduces servers puzzle verification time by implementing software puzzle mechanism in advance. In this mechanism, puzzle solved by client is verified automatically at client side so no need to send puzzle solution at server side for verification. In existing system, puzzle solved by client sent at server side for verification due to server spent extra time for verification. Sometimes attacker was sending unsolved computation to server and server has to spend some time for verification. If for each verification server takes 2 minutes and an attacker sends 60 unsolved computation then server will take 120 minutes extra.

In this system, three puzzle formation methods are used which are very important for DoS prevention. An attacker unable to solve puzzle because for solving puzzle human presence is required and an attacker is a tool which is not human. This system provides strong authentication, security and efficiency and prevents DoS attacks. Compared to existing system server served all client request so system will not fail due to number of bogus request.

In future scope user can access multiple resources at a time within specific time period by solving only one puzzle.

ACKNOWLEDGEMENT

I wish to thank the researchers as well as publishers for making their resources available and teachers for their precious guidance. I am thankful to the authorities of Savitribai Phule Pune University, Pune for their stable guidelines and support. I am also grateful to reviewer for their valuable suggestions and the college authorities for providing the required infrastructure and support.

REFERENCES

[1] 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.
[2] 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.
[3] A. Juels and J. Brainard, "Client puzzles: A cryptographic countermeasure against connection depletion attacks," in *Proc. Netw. Distrib. Syst.Secur. Symp*, 1999.
[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.
[5] 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.
[6] Rashmi, Vicky Chawla, Rajni Sehgal and Renuka Nagpal "The RC7 Encryption Algorithm" International Journal of Security and Its Applications Vol. 9, No. 5 (2015).

Available online: www.ijmrset.com e-ISSN: 2394-8752, p-ISSN: 2394-8791

Volume-03, Issue-09, September 2016

[7] MILIND MATHUR, AYUSH KESARWANI "COMPARISON BETWEEN DES, 3DES, RC2, RC6, BLOWFISH AND AES" Proceedings of National Conference on New Horizons in IT - NCNHIT 2013.
[8] http://en.wikipedia.org/wiki/SPEKE_ (cryptography).
[9] D. S. Abdul. Elminaam, H. M. Abdul Kader, M. M. Hadhoud," Performance Evaluation of Symmetric Encryption Algorithms" Communications of the IBIMA
Volume 8, 2009.
[10] 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.
[11] 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.
[12] 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.
[13] 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.
[14] C. Douligeris and A. Mitrokotsa,"DDoS attacks and defense mechanisms:Classification andstate-of-the-art,"*Comput. Netw.*, vol. 44, no. 5,pp. 643–666, 2004.

BIOGRAPHY

Ms. Smita Miraje Student at Computer Department, Indira College of Engineering and Management, Pune.

Prof. Manisha Bharati Assistant Professor at Indira College of Engineering and Management, Pune.