GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

SOFTWARE PUZZLE: A COUNTERMEASURE TO RESOURCE-INFLATED DENIAL OF SERVICE ATTACKS WITH OTP

Mr.Ravindra Tahakik[*1], Mr.Prashant Hamane[*1], Mr.Suyog Khatik[*1], Mr.Gorakh Gaikwad[*1] and Prof.Shrikant Dhamdhere[2]

[*1]BE Student, PGMCOE, Department of Information Technology, India

ABSTRACT

Denial-of-service (DoS) and distributed DoS (DDoS) are among the real dangers to cyber-security, and client puzzle, which requests a customer to perform computationally costly operations before being allowed administrations from a server, is a surely understood countermeasure to them. Notwithstanding, an assailant can blow up its capacity of DoS/DDoS assaults with quick confound comprehending programming and/or inherent graphics processing unit (GPU) equipment to fundamentally debilitate the viability of customer riddles. In this paper, we mull over how to anticipate DoS/DDoS aggressors from blowing up their riddle comprehending abilities. To this end, we present another customer riddle alluded to as programming riddle. Dissimilar to the current customer riddle plans, which distribute their riddle calculations ahead of time, a riddle calculation in the present programming riddle plan is haphazardly created when a customer solicitation is gotten at the server side and the calculation is produced such that: 1) an aggressor is not able to set up a usage to explain the riddle ahead of time and 2) the assailant needs impressive exertion in interpreting a focal preparing unit riddle programming to its practically identical GPU form such that the interpretation is impossible continuously. In addition, we demonstrate to execute programming riddle in the nonexclusive server-program model.

Keywords- Software Puzzle,GPU Programming, Code Obfuscation, DDOS.

I. INTRODUCTION

DENIAL of Service (DoS) assaults and Distributed DoS (DDoS) assaults endeavor to exhaust an online administration's assets, for example, system data transfer capacity, memory and overwhelming so as to reckon force the administration with fake requests.1 For instance, a malicious client sends a huge number of refuse solicitations to a HTTPS bank server. As the server needs to invest a considerable measure of CPU energy in finishing SSL handshakes, it might not have adequate assets left to handle administration demands from its clients, bringing about lost organizations.

The reality of the DoS/DDoS issue and their expanded recurrence has prompted the approach of various barrier instruments. In this paper, we are especially intrigued by the countermeasures to DoS/DDoS assaults on server calculation power. Let γ signify the proportion of asset utilization by a customer and a server. Clearly, a countermeasure to DoS and DDoS is to expand the proportion γ, i.e., build the computational expense of the customer or lessening that of the server. Customer riddle is a surely understood way to deal with expansion the expense of customers as it powers the customers to complete substantial operations before being allowed administrations. By and large, a customer riddle plan comprises of three stages: riddle generation, 2 riddle comprehending by the customer and riddle check by the server.

A) MOTIVATION

As there is no detection and prevention framework in a virtual networking environment which motivates me more. Improve accuracy in the attack detection from attackers.

II. EXISTING SYSTEM

Cloud users can install vulnerable software on their VMs, which essentially contributes to loopholes in cloud security. The challenge is to establish an effective vulnerability/attack detection and response system for accurately identifying attacks and minimizing the impact of security breach to cloud users[1]. In a cloud system where the infrastructure is shared by potentially millions of users, abuse and nefarious use of the shared infrastructure benefits attackers to exploit vulnerabilities of the cloud and use its resource to deploy attacks in more efficient ways. Such

attacks are more effective in the cloud environment since cloud users usually share computing resources, e.g., being connected through the same switch, sharing with the same data storage and file systems, even with potential attackers[2]. The similar setup for VMs in the cloud, e.g., virtualization techniques, VM OS, installed vulnerable software, networking, etc., attracts attackers to compromise multiple VMs.

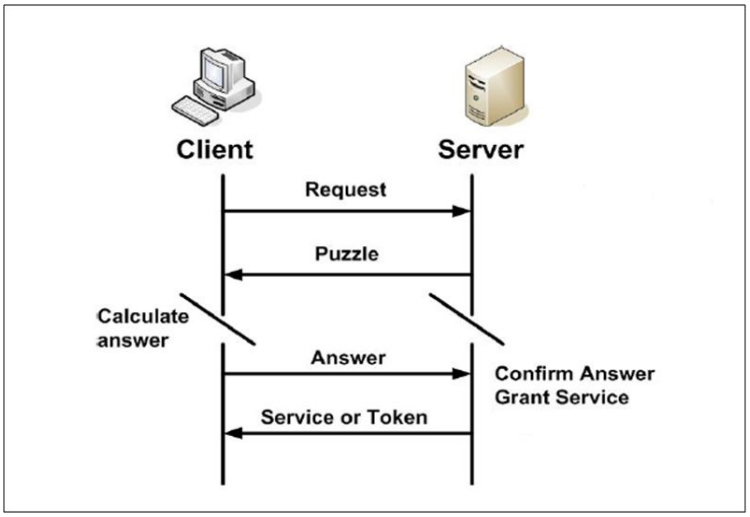


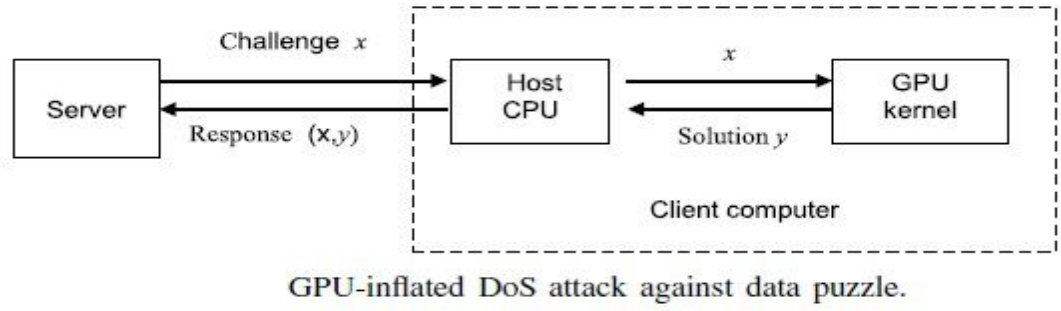
Fig.1.Existing system of puzzle generation

Disadvantages of Existing System:

- No detection and prevention framework in a virtual networking environment.
- Not accuracy in the attack detection from attackers.

III. PROPOSED SYSTEM

Software puzzle scheme is proposed for defeating GPU-inflated DoS attack[4]. It receives software protection technologies to guarantee challenge data confidentiality and code security for an appropriate time period. Hence, it has different security requirement from the conventional cipher which demands long-term confidentiality only, and code protection which focuses on long-term robustness against reverse-engineering only[6]. Since the software puzzle may be built upon a data puzzle, it can be integrated with any existing server-side data puzzle scheme, and easily deployed as the present client puzzle[8] schemes do.



GPU-inflated DoS attack against data puzzle.

Fig.2.Proposed System Dos attack against data puzzle

Advantages of Proposed System:

- Prevent DoS/DDoS attackers from inflating their puzzle-solving capabilities.
- Accuracy in the attack detection from attackers.
- An attacker is unable to prepare an implementation to solve the puzzle.

IV. ALGORITHM

I. AES Algorithm steps:

The AES algorithm is used to encrypt and decrypt files which we are going to upload.

1. Derive the set of round keys from the cipher key.
2. Initialize the state array with the block data (plaintext).
3. Add the initial round key to the starting state array.
4. Perform nine rounds of state manipulation.
5. Perform the tenth and final round of state manipulation.
6. Copy the final state array out as the encrypted data (ciphertext).

II. Key Generation Algorithm Steps:

Random Class is used to random key generation.

1. Generate 6 digit number using Random class.
2. Send this number through email to the user.
3. Verify that number with the generated number.
4. Then give the access to the user for download.

III. Puzzle Generation:

In order to construct a software puzzle, the server has to execute the module: puzzle generation,

- Select 4 random numbers and assign it to A, B, C, D, and select random 3 operators from +, -, *, /.
- Generate the puzzle and display it on user screen.
- Send answer to user through email.

ISSN 2348 – 8034

Impact Factor- 3.155

V. EXPRIMENTAL RESULT

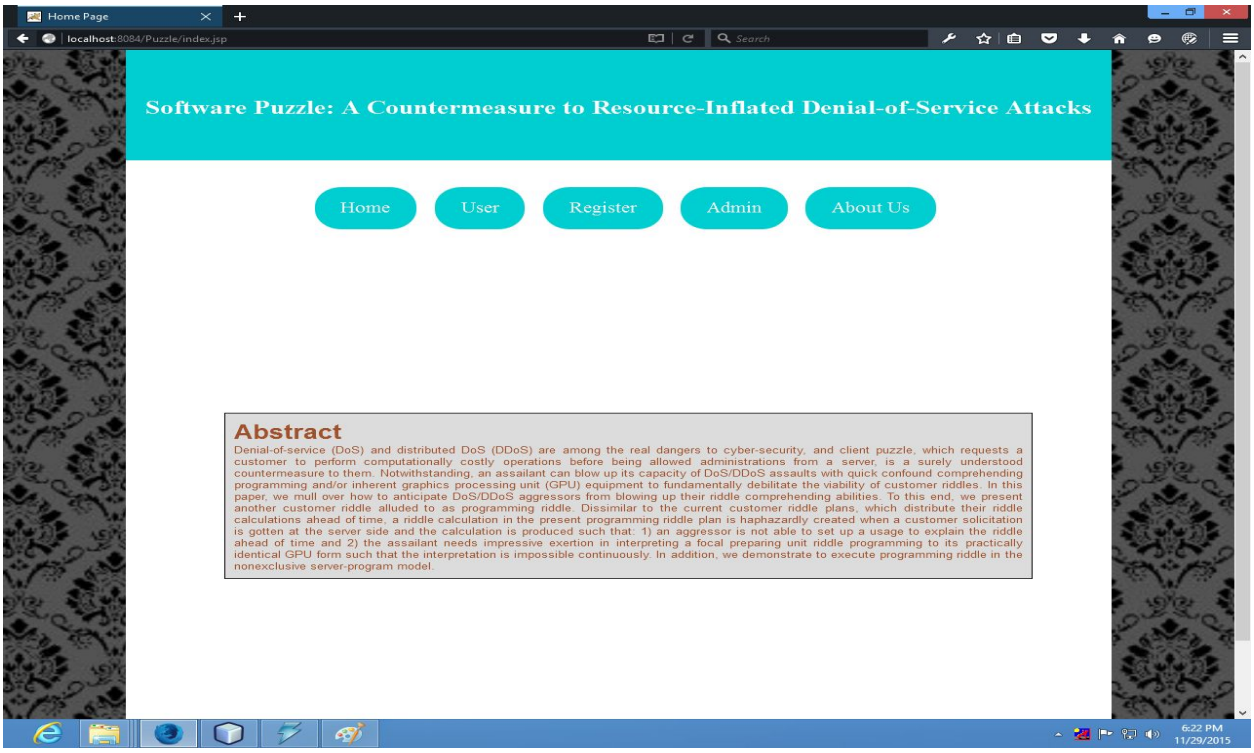


Fig.3.Main page of project

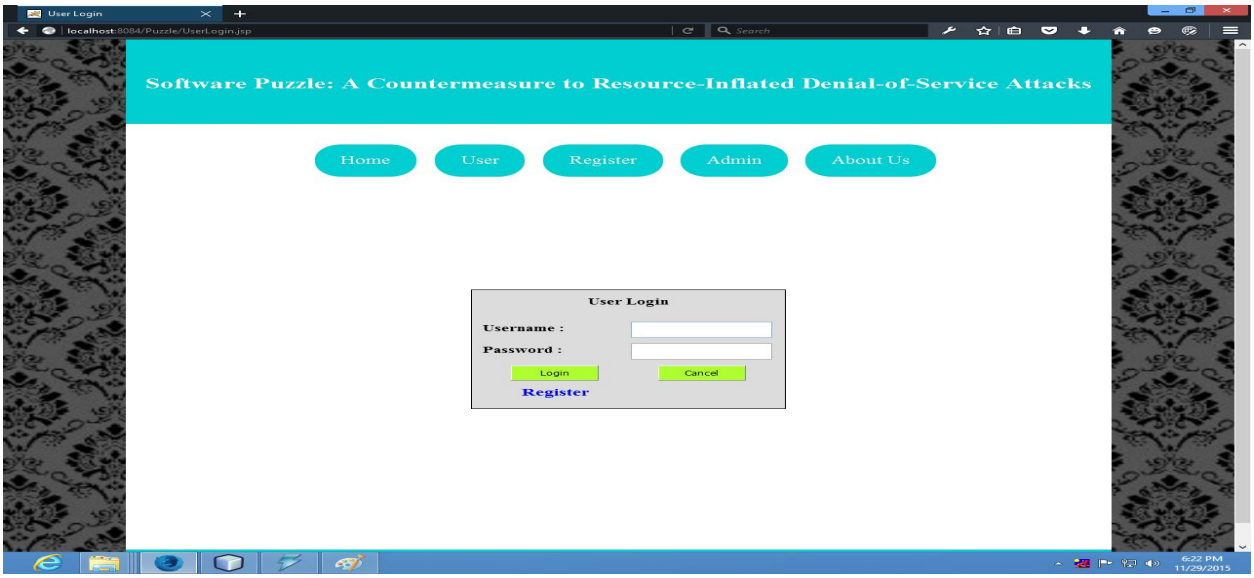


Fig.4.user Login page of project

ISSN 2348 – 8034

Impact Factor- 3.155

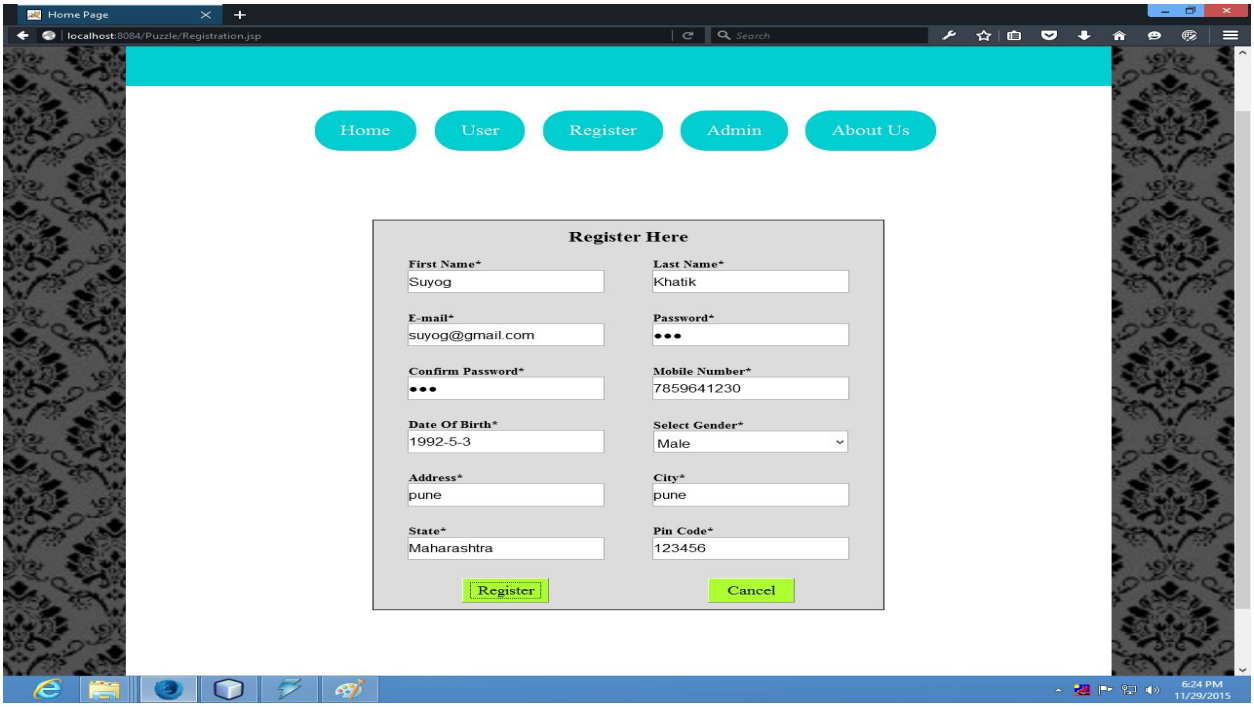


Fig.5.user registration details page of project

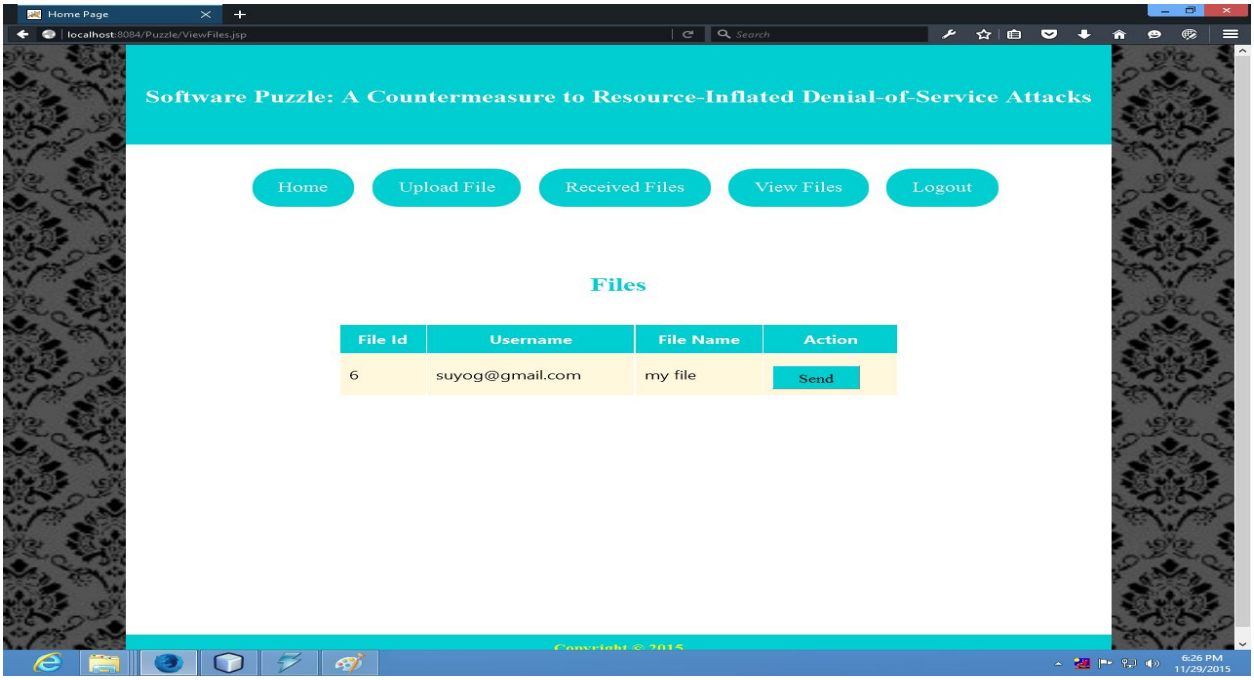


Fig.6.Final details page of project

VI. CONCLUSION

In this project, software puzzle scheme is proposed for defeating GPU-inflated DoS attack. It adopts software protection technologies to ensure challenge data confidentiality and code security for an appropriate time period. Hence, it has different security requirement from the conventional cipher which demands long-term confidentiality only, and code protection which focuses on long-term robustness against reverse-engineering only.

VII. ACKNOWLEDGEMENTS

We are thankful to Mr. Shrikant Dhamdhere Professor, Faculty of Computer Engineering, PGMCOE, Pune for his Guidance and in the successful completion this study. We would also like to thank all our colleagues who have directly or indirectly guided and helped us in the preparation of this report and also for giving me an unending support right From the stage this idea was conceived. I also acknowledge the research work done by all researchers in this field.

REFERENCES

[1] J. Larimer. (Oct. 28, 2014). "Pushdo SSL DDoS Attacks".

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

[3] A. Juels and J. Brainard, "Client puzzles: A cryptographic countermeasure against connection depletion attacks," in Proc. Netw. Distrib. Syst.Secur. Symp., 1999, pp. 151–165.

[4] T. J. McNevin, J.-M. Park, and R. Marchany, "pTCP: A client puzzleprotocol for defending against resource exhaustion denial of serviceattacks," 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 threatsto denial of service countermeasures," Dept. Comput. Sci., UIUC,Champaign, IL, USA, Tech. Rep., Oct. 2010.

[6] 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.

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

[8] R. L. Rivest, A. Shamir, and D. A. Wagner, "Time-lockpuzzles and timed-release crypto," Dept. Comput. Sci.,Massachusetts Inst. Technol., Cambridge, MA, USA, Tech.Rep. MIT/LCS/TR-684, Feb. 1996.

[9] W.-C. Feng and E. Kaiser, "The case for public work," in Proc. IEEEGlobal Internet Symp., May 2007, pp. 43–48.[10] D. Keppel, S. J. Eggers, and R. R. Henry, "A case for runtime codegeneration," Dept. Comput. Sci. Eng., Univ. Washington, Seattle, WA,USA, Tech. Rep. CSE-91-11-04, 1991.