SECURE SOFTWARE PUZZLE AGAIINST DOS ASSAULTS USING A COUNTERMEASURE

[1] Dasari Ashok Kumar, [2] N.Praveena
[1] M. Tech Scholar, [2]Assistant Professor
[1, 2] Department of computer science and technology, Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada, Andhra Pradesh (India) 520007

Abstract— Denial-of-provider (DoS) and distributed DoS (DDoS) are most of the principal threats to cyber-protection, and customer puzzle, which needs a client to carry out arithmetically highly-priced operations earlier than being allowed offerings from a server, is a good countermeasure to them. However, an attacker can inflate its capability of DoS assaults with speedy puzzle solving software and/or integrated Graphics processing unit (GPU) hardware to seriously weaken the effectiveness of Client puzzles. In this, we examine a path to save DoS attackers from inflating their puzzle-solving competencies. To this end, we introduce a new patron puzzle called software puzzle. Unlike the prevailing patron puzzle schemes, which publish their puzzle algorithms in advance, a puzzle set of rules in the current software puzzle scheme is randomly generated most effective after a client request is received at the server aspect and the algorithm is generated such that an attacker is unable to put together an implementation to resolve the puzzle earlier and the attacker needs enormous attempt in translating a Central processing unit puzzle software program to its functionally equivalent GPU version such that the interpretation cannot be achieved in actual time.

Index Terms— DoS, DDoS, Software puzzle.

I. INTRODUCTION

Denial of Service Attack:

DDoS assault is a malicious try to carry down networks, Web-based applications, and/or offerings by means of overwhelming these sources with too much records or impairing them in a few other manner. Unlike a denial-of-Service (DoS) attack wherein the supply is simply a singular pc and connection, a DDoS assault is from more than one sources, and is capable of inflicting fantastic consequences to a business enterprise's emblem, reputation and backside line.

DDoS assaults are intended to focus on any part of a business and its assets, and can without much of a stretch:

- Disable a particular PC, benefit or a whole system
- Target alerts, printers, telephones or portable workstations
- Hit framework assets like transmission capacity, plate space, processor time or directing data.
- Execute malware that influences processors and triggers mistakes in PC microcodes .
- Exploit working framework vulnerabilities to deplete framework assets .
- Crash the working framework.

Yet, DDoS assaults are not all the same. On an abnormal state, a DDoS assault can be first isolated into the accompanying two classes:

- Association based: An assault that happens once an association between a server and a customer has been set up through certain standard conventions.
- Connectionless: An assault that does not require a session to be formally settled before a sender (server) can send "information packets" – an essential unit of correspondence over a computerized network - to a receiver (customer).

Zero Day DDoS Assault:

These assaults happen when a performer misuses a zero-day Vulnerability to do a DDoS assault. A zero-day vulnerability is an application defect previously unknown to the vendor, and has not been settled or fixed. It is known as a "zero-day" on the grounds that once an imperfection is found the vendor has zero days to settle it. Denial of Service (DoS) assaults and Distributed DoS (DDoS) assaults attempt to attack an online Services data such as network bandwidth. Computation power and memory by the malicious requests.For example, an attacker sends a large no of unwanted requests to an HTTPS server. When the server reached these unwanted requests, it has to fill a lot of Processor time in finishing these requests, it may not have to handle service requests from its customers, due to this loss in business and slow processing on that system.

Attribute-Based Encryption:

Attribute based encryption is a public key encryption, this encryption is based upon attributes. These attributes secure the cipher text and the secret key of the user. The decryption of a Cipher text is decrypted when the receiver knows the attributes of the user.

II. LITERATURE SURVEY

In 2010, Resource inflation threats to denial of service countermeasures changed into by R. Shankesi, O. Fatemieh, and C. A. Gunter Currency-based mechanisms were proposed as a way to utilize asset decency among contenders for a transporter to impede Denial of Service (DoS) assaults. Under aid fairness, a server allocates its service to the clients in percentage to their fee of a useful resource, making the aid function a kind of currency. We don't forget the vulnerability of foreign money-based DoS protection mechanisms to numerous resource inflation assaults in which an attacker can notably inflate its possession of the useful resource at low cost and in a way that may be both hard or unwanted for a valid client to do. We offer a easy theoretical analysis of useful resource inflation attacks and look into its utility to a number of fee schemes to rank their in all likelihood vulnerability. We find that the chance of Graphics Processing Units (GPUs) for inflation assaults is specially extreme: we are able to show inflation of up to 630x with common less expensive GPUs. We additionally assessment threats from other competencies, along with multi-center processors, cloud computing, and bandwidth inflation schemes.

In 2011, Reconstructing Hash Reversal based Proof of Work Schemes was by J. Green, J. Juen, O. Fatemieh, R. Shankesi, D. Jin, and C. A. Gunter . Customer puzzles to manipulate limited sources on a server and provide resilience to denial of service attacks. Attacks making use of GPUs to inflate computational capacity, known as useful resource inflation, are a singular and effective hazard that dramatically boom the computational disparity among customers. This disparity renders evidence of labor schemes based on hash reversal ineffective and probably damaging. This paper examines numerous such schemes in view of GPU-based totally attacks and identifies characteristics that allow protection mechanisms to withstand assaults. In unique, we display that, hash-reversal schemes which adapt totally on server load are ineffective underneath attack by using GPU using adversaries; while, hash reversal schemes which adapt based on customer conduct are powerful even below GPU based totally assaults

III. EXISTING SYSTEM

DoS and DDoS are effective if intruders allocate a deal much few sources than the affected server or are a whole lot greater effective than regular clients.The attacker spends less effort in generating a request, however the server has to spend a good deal more computational effort in HTTPS handshake. In this situation, traditional crypto-graphic gear do no longer develop the provision of the services; in truth, they may degrade service satisfactory due to highly-priced cryptographic operations.

The importance of the DoS/DDoS problem and their elevated occurrence has caused the appearance of several defense mechanisms.

As the prevailing browsers which include Microsoft Internet Explorer and Firefox do not explicitly assist patron puzzle schemes, Kaiser and Feng developed a web essentially based client confuse plot which concentrates on straightforwardness and in reverse similarity for incremental organization. The scheme dynamically embeds purchaser-particular demanding situations in web pages, transparently delivers server demanding situations and patron responses.

Cons of Existing System:

Puzzle is designed based totally on purchaser's GPU functionality, the GPU-inflation DoS does now not paintings in any respect. However, we do now not endorse to accomplish that because it's miles difficult for large deployment due to

(1) now not all the clients have GPU-enabled gadgets; and

(2) an extra actual-time surroundings will be mounted in an effort to run GPU kernel.

•However, this scheme is liable to DoS attackers who can put into effect the puzzle characteristic in real-time.

•Existing structures are not dynamic.

IV. PROPOSED SYSTEM

Software puzzle scheme is proposed for defeating GPU-inflated DoS assault. It adopts software safety technology to make sure mission records confidentiality and code protection for the best time period, e.G., 1-2 seconds. Hence, it has exceptional security requirement from the conventional cipher which demands long-time period confidentiality only, and code safety which makes a specialty of long-term robustness against reverse-engineering simplest.

- Since the software puzzle may be built upon a resource puzzle, it may be incorporated with any current server-facet resource puzzle scheme, and without problems deployed as the prevailing client puzzle schemes do. Although this specializes in GPU-inflation assault, its concept may be extended to thwart DoS attackers which exploit other inflation sources along with Cloud Computing.
- By exploiting the architectural distinction between CPU and GPU, this paper presents a brand new sort of patron puzzle, referred to as software puzzle, to guard towards GPU-inflated DoS and DDoS assaults.

Pros of Proposed System:

- SSL/TLS protocol is the maximum famous on line transaction protocol, and an SSL/TLS server plays an luxurious AES decryption operation for every consumer connection request, hence it's miles liable to DoS assault.
- Our objective is to secure SSL/TLS server with software puzzle against computational DoS especially GPU-inflated DoS assault. As an entire SSL/TLS convention incorporates many rounds, we utilize AES decryption step to evaluate the defense effectiveness in terms of the server's time cost for simplicity.
- The software puzzle scheme dynamically generates the puzzle characteristic.

V. BLOCK DIAGRAM

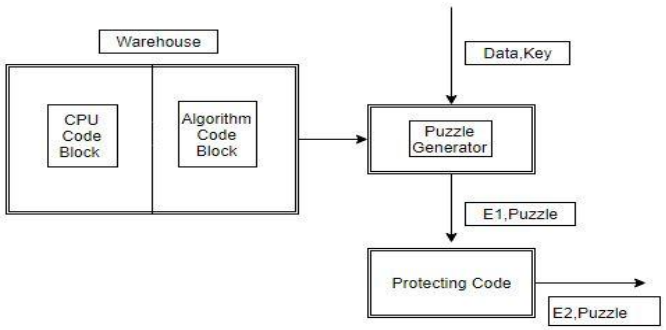

Figure 1: Diagram of software puzzle generated with secret key and data

Warehouse Block Structure All compiled instruction blocks are put away in distribution center. which is in java byte code arrange. This instruction blocks spare server time generally server needs to invest part of energy in changing over source code into compiled code. In this warehouse, each block is depended on safety parameter and length of every puzzle is in constant size. If the block size is smaller then it provides more security stage. Because of smaller block length attacker has to spend more time to understand puzzle in query.

In warehouse block structure, code block contain 2 classes: CPU code block and algorithm code block.

- CPU code block It contains all instruction set that is required in puzzle era system. Which one of a kind activities are executed in the course of puzzle technology process is offered in CPU code block.
- Algorithm code block It includes all operations related to encryption set of rules and stores all mathematical operations.
- Code Protection Provide code safety the usage of code obfuscation method. Code obfuscation method creating something that is lesser to clean and more difficult to understand. Here, for code protection AES algorithm is used.
- Puzzle Solver Here, puzzle solved by means of client is validated the usage of puzzle solver to keep the server time. Puzzle solver is applied at client side and it'll allocate resources to the client if the client unearths out correct puzzle answer.

Process Flow:

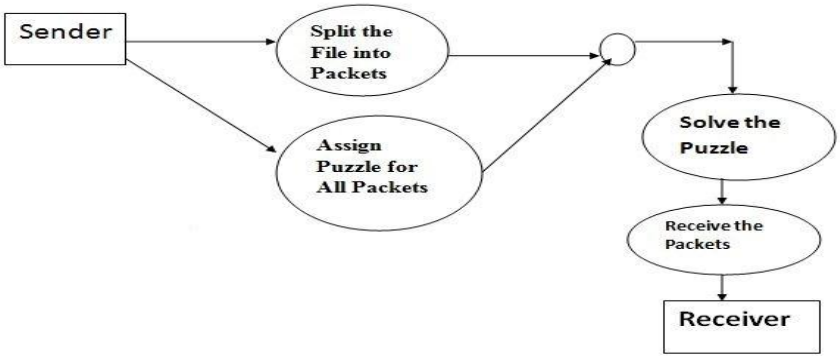

Figure 2: File send from sender to receiver

Client node should have to follow the protocols mention at the time of giving input value like node address, client IP address. And resultant values.

Server node should have to verify the data details along with sender address and file name and Key mention at the of encrypting data.

Puzzle solver should have to check the data decrypted data is valid or not and it should have put it as inflated.

VI. ALGORITHM

AES Algorithm:

AES uses a block period of 128 bits and a key duration that can be 128, 192, or 256 bits.

- We count on a key duration of 128 bits [10].
- The value to the encryption and decryption algorithms is a single 128-bit block.
- The 128-bit secret is depicted as a rectangular matrix of bytes.
- This key is then improved into an array of key schedule phrases: every phrase is four
- bytes and the total key schedule is forty four words for the 128-bit key.

Four distinctive levels are used, one in every of permutation and 3 of substitution:

- Substitute bytes: Uses a table, called an S-field, to perform a byte-through-byte substitution of the block.
- Shift rows: A easy permutation this is done row with the aid of row.
- Mix columns: A substitution that alters every byte in a column as a characteristic of all the bytes within the column.
- Add round key: A simple bitwise XOR of the present block with a portion of the expanded key.

VII. EXPERIMENT RESULTS AND ANALYSIS

Puzzle Generator:

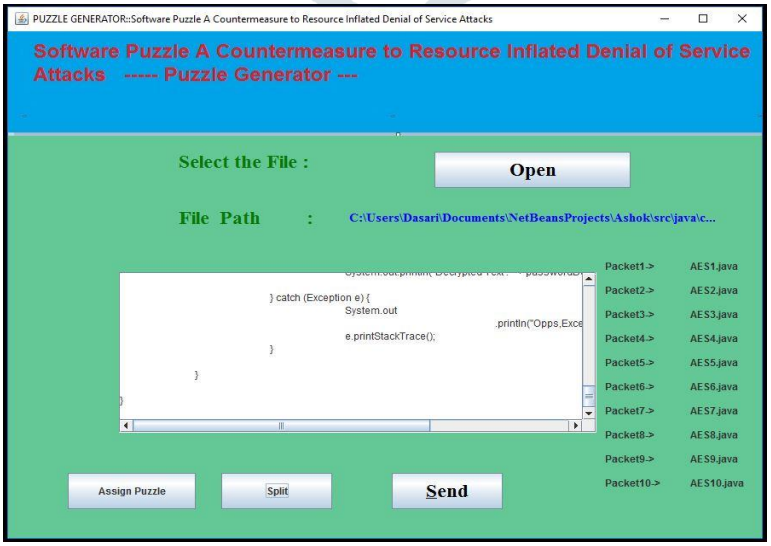

Figure 3: Puzzle generator

Description: Run the puzzle generator. Select the data/file. It shows the file path. Press the Split button to split the data into packets. Assign the puzzle to all packets. Then send the data to receiver.

Figure 4: Authentication Status

- The legitimate user access the data, it treated as Puzzle solved users, When the Puzzle successfully send by the source it shows the check the authentication dialogue will appears.

Middleware Host:

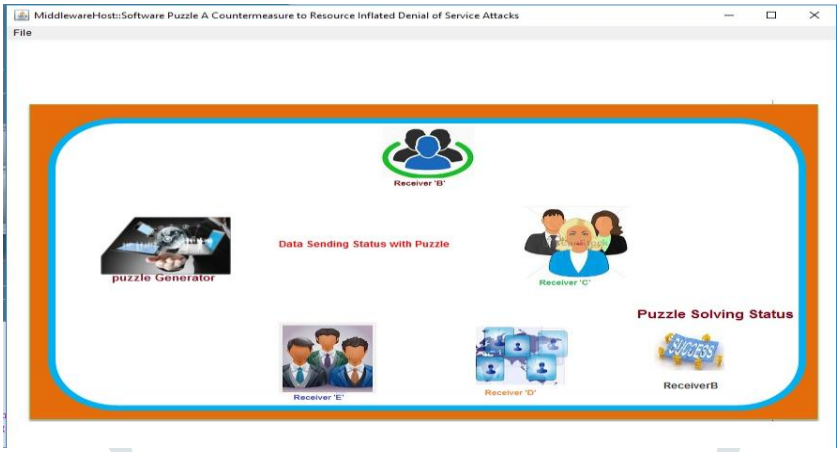

Figure 5: Data sending process Status

- File is send to the receiver you have to assign the puzzle, if the puzzle attributes entered are correct then it shows a dialogue box shown as puzzle solved user.
- File is send to the receiver you have to assign the puzzle, if the puzzle attributes entered are wrong then it shows a dialogue box shown as resource inflated attacker.

Receiver Node B:

- After the authentication, the data received will appear.
- When the data is received to the node it shows the file which is sent by the source.

Figure 6: Receiver node access the data

Analysis:

- Here the puzzle solved users are represented in bar graph.these bar graph shows the solved users that means the legible users.

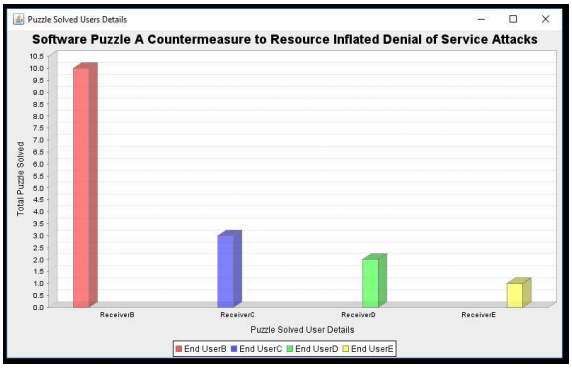

Figure 7: Solved users

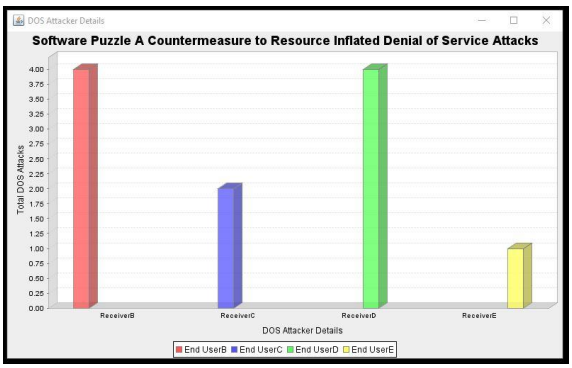

Figure 8: Resource inflated users

VIII. CONCLUSION

In this paper, Software puzzle scheme is for DoS assaults. It is used to secure the data from malicious users and gives the security to the data. To secure the data we r using this Software puzzle, it protect the data from DoS assaults. It provides the long term confidentiality and cant access the server if the user is not legitimate user, if the user is an assault it treated as assault. The puzzle is in the server side and easily deployed in the client side. In this puzzle is generated at server side and it consumes server time because of this server processing is slow, the problem is server how to build the client side puzzle to decrease the server time for good accessing.

REFERENCES

[1] 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.

[2] 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. [Online]. Available: http://hdl.handle.net/2142/17372

[3] 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.

[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] E. Kaiser and W.-C. Feng, "mod_kaPoW: Mitigating DoS with transparent proof-of-work," in *Proc. ACM CoNEXT Conf.*, 2007, p. 74.

[6] X. Wang and M. K. Reiter, "Mitigating bandwidth-exhaustion attacks using congestion puzzles," in *Proc. 11th ACM Conf. Comput. Commun. Secur.*, 2004, pp. 257–267.