ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

Reducing Denial of Service Attacks Using Software Puzzle

Ajinkya Kadam[1], Prateek Mishra[2], Chander Prakash[3], Vikrant Saurabh[4], Prof. F.S.Ghodichor[5]

Student, Sinhgad Institute of Technology, Lonavala, Pune, India[1]

Student, Sinhgad Institute of Technology, Lonavala, Pune, India[2]

Student, Sinhgad Institute of Technology, Lonavala, Pune, India[3]

Student, Sinhgad Institute of Technology, Lonavala, Pune, India[4]

Professor, Sinhgad Institute of Technology, Lonavala, Pune, India[5]

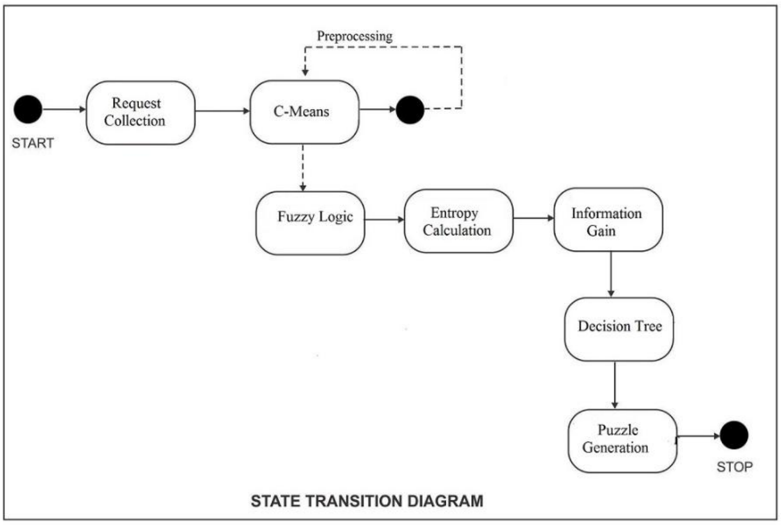
ABSTRACT: DOS attacks are serious threats nowadays. So, it is more circumspect to lead a disaster beforehand than to deal with it after it occurs. There are many systems that can use to prevent DOS attacks but we are using an efficient software puzzle system on websites to protect websites from DOS attacks. Unlike other systems which are being used for preventing DOS attacks, we are designing puzzle algorithms in an advanced way, every time algorithm creates randomly generated puzzles after a request is sent from client to server. The algorithms are designed in such a way that: - 1) an attacker cannot solve a puzzle in advance. 2) The attacker will consume more time in converting CPU puzzle software system to its GPU version.

KEYWORDS: puzzle algorithm; DOS attack; software puzzle algorithm;CPU puzzle; random puzzle generation

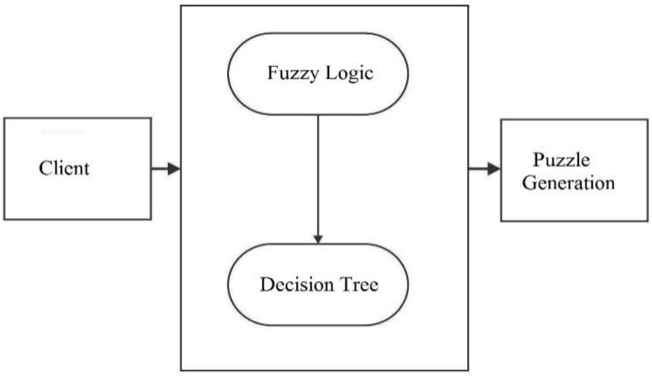
I. Introduction

An efficient system for preventing DOS attacks using software puzzle is designed to prevent websites from attackers and to take action before the situation gets out of control. This system has advantages over all other system methods like captcha, image detection systems, etc. This system uses various algorithms and techniques by which it is preventing DOS attacks on the websites. In particular techniques or algorithms like c means, fuzzy logic, entropy and information gain, etc. are used to make the whole system.Denial of Service (dos) attacks and distributed dos attacks consume machine resources in a way that genuine system usage is confiscated, these attacks are one of the most common security threat to the technical world. DoS attacks also deplete the user bandwidth by consuming server capacity with overflowing requests. GPU hardware's have a faster computing capacity which allows the attacker to infiltrate the system and consume the resources. There are several papers and proposed methodology to end this problem like client puzzle scheme, captcha etc. Along with the newly invoked methods and the built-in GPU hardware's, there is software that can analyse the pattern of an algorithm that generates the puzzle and issues the corresponding string to enter the system for further usage. These software's don't allow the server to authenticate the user as Human or Robot since the puzzle generated is solved efficiently. Due to such problems, there is a need to make an efficient software puzzle that not only dynamically generates the puzzle but also scrutinize the incoming traffic from particular users. Incoming traffic can be flagged usingvalues that decide the further action performed. Our proposed system uses Fuzzy logic and matrix evaluation to categorize users according to the flags. Fuzzy logic is an access to calculating based on "degrees of truth" rather than the usual "true or false" (1 or 0) Boolean logic on which the modern computer is based. This can comparatively reduce the denial of services by generating puzzle as per the flagged value. These flags arecategorized as Very Low, Low, Medium, High and Very High. These flags form the fuzzy member function. An attacker can hence be predetermined using this concept and furtherdynamically creatingmathematical equation depending on the range. Since an attacker may use different machines for different sessions but will be using the same user identification and hence this fuzzy logic helps to keep a track of incoming request from the particular users or IP address.

DOI: 10.15680/IJIRCCE.2017. 0504360



Firstly user request is collected and the c-means algorithm is done. Pre-processing is carried out in c-means. Fuzzy logic is applied to this value to carry ahead the entropy calculations and information gain about the user. On the basis of the collected information and range of the user, the decision tree is applied to categorize the user amongst the ranges. After pooling users according to ranges puzzle generation takes place and the puzzle is finally deployed on the website. The basic concept of reducing denial of service attacks using software puzzle is to reduce errors than other systems and avail more resource. And other systems has issues of producing wrong results, So Proposed system gives an idea of detecting fraudulent users, attackers and their garbage request.



Level 0 DFD

II. Related work

There are several tools for DoS attacks and its defensive tools also exists.Due to increasing use of internet service users the existing techniques of preventing DoS looks insufficient.This proposed paper is a modification to the existing technique where attacker used GPU's computational power to solve puzzle.There are some structural approaches to the DoS problem by developing a classification of DoS attacks and DoS defense mechanisms. These attacks need to be identified and its occurrence pattern must be noted to deal with it. The goal of the paper is to monitor the IP traffic and dynamically generate equations that are complicated for robots to solve.This can be achieved by using web application development for deploying the algorithm that generates the puzzle and client side web pages and other website to prove authenticity of the user.We use several algorithms in a sequence of their precise occurrence that finally lets the system to identify the bogus requester and the legitimate clients. Thus overflowing requests that are generated by client/client side tools can be flagged separately to avail the existing resources to the clients that have lower frequency of generating request.

III. PROPOSED ALGORITHM

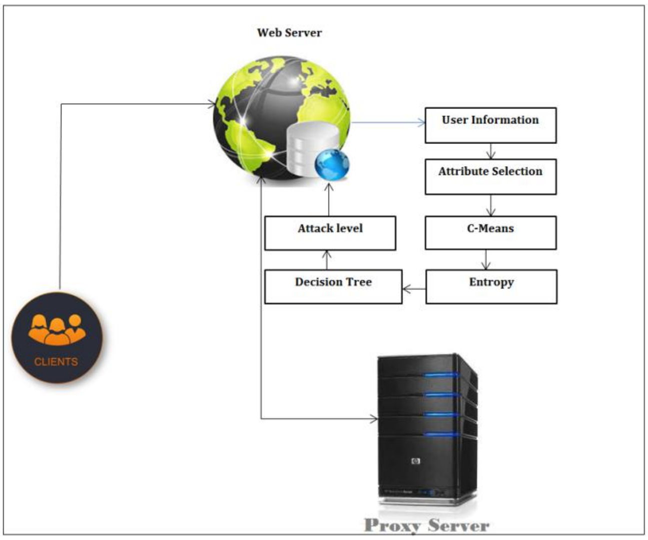


Figure 3: Architecture of proposed system

Phase I: This phase initially information is been collected from web users. User information is been a collection in form of sign in date and time, IP address of Machine used for request and stored in the database for future processing
Phase II: This phase Vector consisting of IP is been generated. C-means algorithm is been initiated. FCM assist in examining IP usage pattern with levels of clustering

$J_m=\sum_{i-1}^{N}\sum_{j-1}^{C}\text{⍰}_{ij}^{m}\left\|X_i - C_j\right\|^2$ 1≤m<∞ -…...………………(1)

In above equation (1)
M= Numerical value >1
uij =degree of membership
Filtration and optimization are been done to yield better clusters indicating clients request and ip usage..

Algorithm: C-Means Clustering
Let Y = {y1, y2, y3 ..., yn} be the set of data points and

W= {w1, w2, w3 ..., wc} be set of centers.

Step I: Start

Step II: Randomly select ‘c’ cluster centers.

Step 2: Calculate the fuzzy membership 'µij' using:
$\text{⍰}_{ij}=1/\sum_{k=1}^{c}(c_{ij}/c_{ik})^{(2/m-1)}$
Step III : Compute fuzzy centers $'v_j'$ using:

$V_j=(\sum_{i=1}^{n}(\text{⍰}_{ij})^m X_i)\,/\,(\sum_{i=1}^{n}(\text{⍰}_{ij})^m)$,

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
(An ISO 3297: 2007 Certified Organization)
Website: www.ijircce.com
Vol. 5, Issue 4, April 2017

for all j=1,2,………………………………..c

Step IV : Repeat step II and III until the minimum *'J'* value is achieved or $||U^{(k+1)} - U^{(k)}|| < \beta$.

Here,

'k' =iteration step

'β' =termination criterion

'U=membership matrix.

'J' is the objective function.

Step V: Stop

Phase III: Entropy Analysis is been done here. IPs from clusters based on priority value are been evaluated with Shannon gain equation (2) as mentioned below. Information gain value assists in finding IP most fluently used and mostly probably affecting system performance.

$IG(C) = -\sum (| C_i | / | C |) \log (| C_i | / | C |)$ ……..............(2)

Here
C_i = frequency of IP address Cluster C

Phase IV: Decision is been generated with simpler IF-Else for Given IP address. In order to identify Dos Attack, the two-dimensional vector is been feed consist of IP and information gain values.
A tree is been generated to evaluate attack levels. These values are been collected and up to a weight of the tree. Based on decision and weight analysis attack level is been generated in between range 0 to 100 to generate a puzzle.

Phase V: Attack information is been sent puzzle generator server and the key is been generated with MD5 algorithm. On attack information receiving puzzle to answer evolution is been done. Random numerical values have been generated to design puzzle. This information is been sent to the web application.
Infix algorithm has been used for expression evolution. Infix algorithm is as described below

Algorithm: Expression Evolution

```
Step 1: initiate Process

Step 2: read character:

Step 3: if operand push on stack

Step 4: else if character is operator

 Step 4a: while the top of the Stack is not of smaller precedence than this character

Step 4b: pop op from Stack

Step 4c: pop two operands op1 and op2 from Stack

Step 4d: store op1 op op2 on Stack back to 4a
```

DOI: 10.15680/IJIRCCE.2017. 0504360

Step 5: else if character is)

Step 6: pop operators till empty

Step 7: pop top 2 operands and push op1 op op2

Step 8: return top from Stack

Step 9: end

IV. Simulation Results

Puzzle system has to been designed in intention that puzzles can be solved by humans only and eliminate attacks like OCR crawler or automated bots. A framework has been designed which generates puzzle and answer for the user. Practical evaluation of system needs to be done for its effectiveness. The system has been deployed on apache tomcat. The system designed here is for metro railway reservation wherein user book cancel tickets. the system has been tested with a tool called WAPT measuring 0.9 loads figur1 4 represents experimental results of the system.As such examination of above system tells that puzzle creation is directly proportional to a number of users as such software over performances in a matter of time. Future system has been tested for existing algorithmic procedures like genetic algorithm

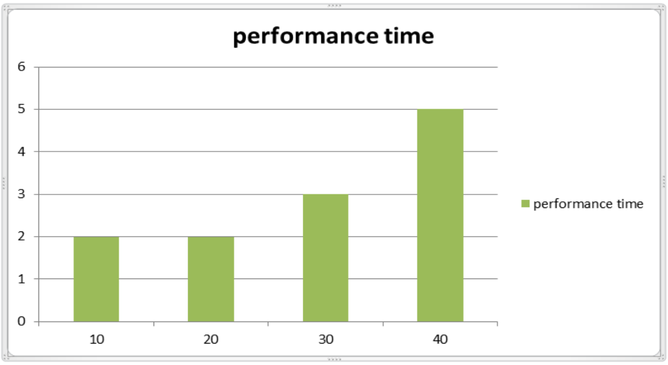
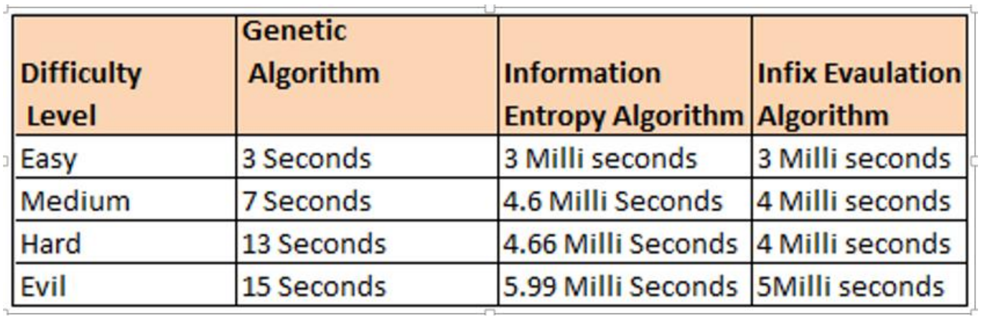


Figure 2: Performance evaluation based on time complexity



Difficulty Level	Genetic Algorithm	Information Entropy Algorithm	Infix Evaulation Algorithm
Easy	3 Seconds	3 Milli seconds	3 Milli seconds
Medium	7 Seconds	4.6 Milli Seconds	4 Milli seconds
Hard	13 Seconds	4.66 Milli Seconds	4 Milli seconds
Evil	15 Seconds	5.99 Milli Seconds	5Milli seconds

Table 1: Comparison table with Different Algorithms

Figure 3 represents a graphical examination of algorithmic procedures. The proposed system has been found to be less complex in terms of time and easy to implement compared to other systems.

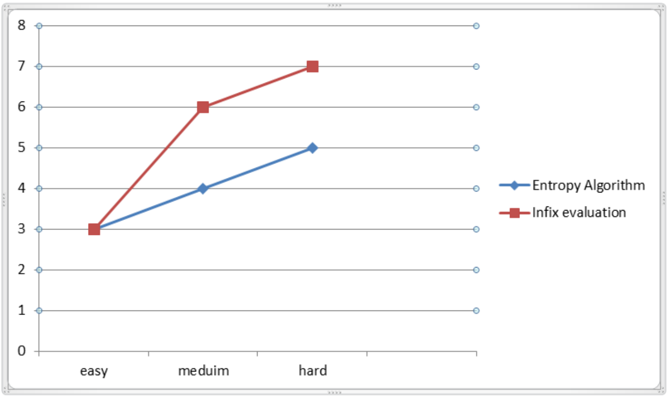


Figure 3: Graphical evaluation of entropy and Infix Algorithm

V. Conclusion and Future Work

The proposed paper describes the consolidation of the DOS assault taking care of for any web applications through utilizing of the intermediary server for confound era rather than captcha or picture number puzzle. Proposed framework effectively distinguishes the assault utilizing Fuzzy C implies grouping and Shannon information for entropy estimation. Decision tree Enhances system performance.

Effective load handling has been achieved with puzzle Proposed system generates a unique puzzle for unique users. The algorithmic design strategy of infix expression algorithm has been found to best. The proposed system is advanced level fight back mechanism against Dos assaults.

The system can be enhanced to generate a more complex puzzle with operands and operator variance and scope of future work.

References

1. von Ahn, Luis; Blum, Manuel; Hopper, Nicholas J.; Langford, John (May 2003). CAPTCHA: Using Hard AI Problems for Security. EUROCRYPT 2003: International Conference on the Theory and Applications of Cryptographic Techniques.
2. "The reCAPTCHA Project - Carnegie Mellon University CyLab". www.cylab.cmu.edu. Retrieved 2017-01-13.
3. "Image Recognition CAPTCHAs" (PDF). Cs.berkeley.edu. Retrieved 2013-09-28.
4. “Software Puzzle: A Countermeasure to Resource-Inflated Denial-of-Service Attacks”, Yongdong Wu, Zhigang Zhao, Feng Bao, and Robert H. Deng, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 10, NO. 1, JANUARY 2015.
5. “DDoS attacks and defense mechanisms: Classification and state-of-the-art,” C. Douligeris and A. Mitrokotsa,*Comput. Netw.*, vol. 44, no.5, pp. 643–666, 2004.
6. “Client puzzles: A cryptographic countermeasure against connection depletion attacks,” A. Juels and J. Brainard, in *Proc.* Netw. Distrib. Syst.Secur. Symp., 1999, pp. 151–165.