North Asian International Research Journal of Sciences, Engineering & I.T.

IRJIF. I.F. : 3.821

ISSN: 2454-7514 Vol. 4, Issue-1 January-2018

Index Copernicus Value: 52.88

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

VAISHNWEE SHREENIWAS NADGAUDA, SUJATA MAHADEV SANAP, SWATI MACHINDRA SANGVIKAR & AKSHATA ASHOK UJGARE

NMIET Alegaon Dabhade, Pune

ABSTRACT

The Software Puzzle: A Countermeasure to Resource-Inflated Denial-of-Service Attacks system is based on networking & cyber Security domain. Cyber security is the body of technologies, processes and practices designed to protect networks, computers, programs and data from attack, damage or unauthorized access. In a computing context, the term security implies cyber security. Ensuring cyber security requires coordinated efforts throughout an information system. An element of cyber security includes network security and Information security.

Denial-of-Service Attacks (Dos) attacks have become a major threat to current computer networks. Denial-of-Service Attacks (Dos) attacks on the internet aim to prevent the authorized user or clients from accessing a service and are considered a serious threat to the availability and reliability of the internet services. Such an attack is easy because the attacker often pays very little for requesting a service, Most of the time only the cost of sending a network packet. In practice the most part of connections of subjects of wide area networks uses the virtual connections as this method is information. Therefore, the interaction without establishing a virtual channel is one of the possible reason for the success of remote attacks such as DoS/DDoS.

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, e.g., 1-2 seconds. Hence, it has different security requirement from the conventional cipher which demands long-term confidentiality.

This DoS/DDoS attack blocks the authorized clients from accessing the services so to overcome or to solve the problem software puzzles are created. These software puzzles are not stored or previously created, they are created at runtime whenever the client requests a particular service the software puzzle is created and the client is supposed to solve the puzzle and give the solution within the required time period. Once the puzzle is solved correctly by the client within the specified time, the client will be provided the service even if the attacker tries to block the service of the client he cannot block it.

The software puzzle scheme is proposed for DoS attack. It adopts software protection technologies to ensure challenge data confidentiality and code security for an appropriate time period. It can be used in many organizations, online banking system.

INTRODUCTION

Denial-of-service (DoS) and distributed Denial of Service (DDoS) are among the major threats to cyber-security, and client puzzle, which demands a client to perform computationally expensive operations before being granted services from a server, is a The author known countermeasure to them. The author over, an attacker can inflate its capability of DoS/DDoS attacks with fast puzzle solving software and/or built-in graphics processing unit (GPU) hardware to significantly The author taken the effectiveness of client puzzles The, study how to prevent DoS/DDoS attackers from inflating their puzzle-solving capabilities. To this end, The author introduce a new client puzzle referred to as software puzzle. an attacker is unable to prepare an implementation to solve the puzzle in advance and the attacker needs considerable effort in translating a central processing unit puzzle software to its functionally equivalent GPU version such that the translation cannot be done in real time. Moreover, The author show how to implement software puzzle in the generic server-browser model. Denial of Service (DoS) and Distributed DoS (DDoS) assaults endeavor to drain online administrations assets, for example, arrange data transmission, memory and calculation control by overauthoring the administration with counterfeit requests1. For instance, a vindictive customer sends an extensive number of trash solicitations to a HTTPS bank server. As the server needs to invest a great deal of CPU energy in finishing SSL handshakes, it might not have adequate assets left to handle benefit demands from its clients, bringing about lost organizations and notoriety. DoS and DDoS assaults are not just hypothetical, be that as it may, likewise sensible, e.g., Push does SSL DDoS Attacks. DoS and DDoS are compelling if aggressors spend substantially less assets than the casualty server or are a great deal more effective than typical clients. In the case above, the assailant spends insignificant exertion in creating a demand, hoTheauthorver the server needs to spend a great deal more computational exertion in HTTPS handshake (e.g., for RSA decoding). For this situation, ordinary cryptographic devices don't upgrade the accessibility of the
Administrations; truth be told, they may debase benefit quality because of costly cryptographic operations. The earnestness of the DoS/DDoS issue and their expanded recurrence has prompted the appearance of various barrier components. In this paper, The author is especially keen on the countermeasures to DoS/DDoS assaults on server calculation control. Let mean the proportion of asset utilization by a customer and a server. Clearly, a countermeasure to DoS and DDoS is to expand the proportion, i.e., increment the computational cost of the customer or decline that of the server. Customer perplex is an outstanding way to deal with increment the cost of customers as it authors the customers to do substantial operations before being authored administrations. By and large, a customer confound conspire comprises of three stages: bewilder generation2, baffle comprehending by the customer also, and perplex confirmation by the server. Hash-inversion is an imperative customer confuse plot which expands a customer cost by driving the customer to split a restricted hash occasion. Actually, in the astound era step, given an open astound work P got from one-way capacities such as SHA-1 or square figure AES, a server haphazardly picks a bewilder challenge x, and sends x to the customer. In the settling and confirmation steps, the customer gives back a confuse reaction (x; y), and if the server affirms x = P(y), the customer can acquire the administration from the server. In this hash-inversion bewilder conspire, a customer needs to spend a specific measure of time in illuminating the astound (i.e., finding the confound arrangement y), what's more, the server needs to invest energy in creating the confound challenge x and confirming the baffle arrangement y. Since the server can pick the test to such an extent that to its for ordinary clients, i.e., 1, an aggressor cannot begin DoS assault effectively by explaining numerous riddles.

MOTIVATION

In previous system attacker attack using D-Dos attack on software puzzle system very easily because there is no time limit and puzzle is already created and stored into database. The existing client puzzle schemes, which publish their puzzle algorithms in advance, a puzzle algorithm in the present software puzzle scheme is randomly generated only after a client request is received at the server side and the algorithm is generated such that: an attacker is unable to prepare an implementation to solve the puzzle in advance.

IMPLEMENTATION DETAILS

System architecture

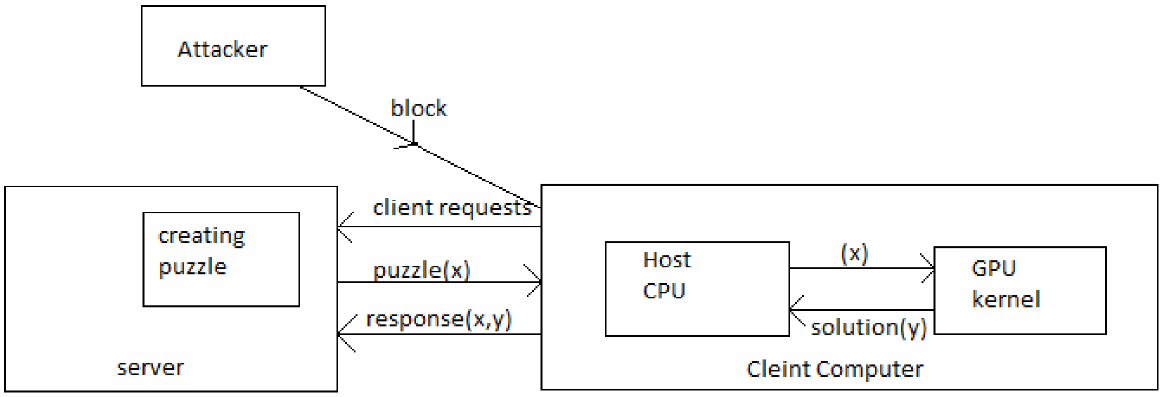


Fig.1:System Architecture

In our project we have identified following objects:-

The user will perform following functions:

1. Client:

    - Register ()

In this we enter the Client information ex. Name, Mac Address, IP Address

    - Login ()

In login page enter User Name, Password for using software puzzle system.

    -Search Query ()

The client request to the Server for Service though search query.

    -solve puzzle ()

Client solve the puzzle in given time and get service from Server.

2. Puzzle:

    -Create ()

The puzzle create at Sever side when client request for service and send it to the client.

    -Processing ()

The puzzle should forwarded to client and

    -Puzzle Answer ()

Client solve the puzzle and give answer to server.

3. Server

-Provide services()

Server check the answer of puzzle if user is authorized then provide service to the user.

-Block IP ()

If client is unauthorized then server will block that user and block the IP.

4. Data Server

-Store data()

Server store IP address, MAC address, User Information

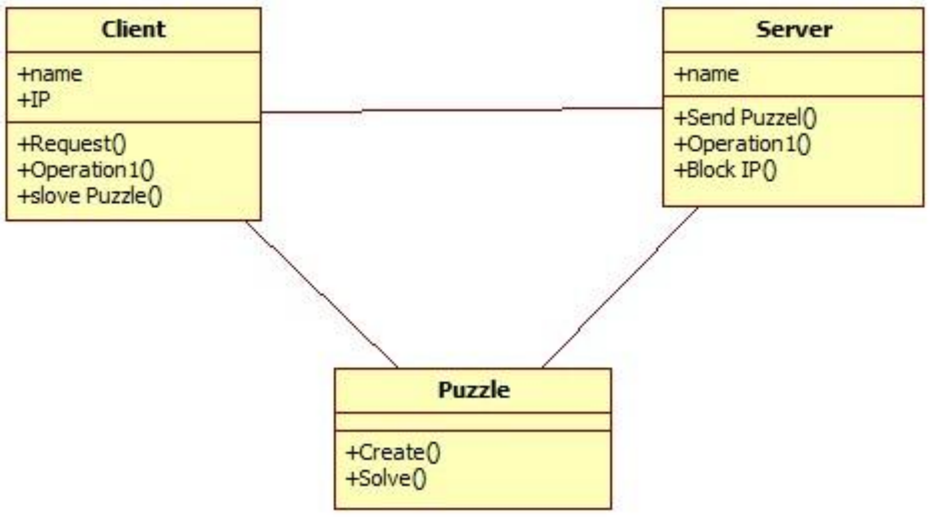
-Send data()

Server sends puzzle to client when client request for service.

-Receive data()

Client get access to service if client solved puzzle in particular time period.

a. Module



Module 1: Client module:

In this module client request to server. Client Solve puzzle which option is present in this module. The purpose of this module is to provide the user interface and view functions for the system. This is the software with which the user directly interacts with server. It communicates with the server to retrieve and modify persistent data when necessary .This module is created to provide the user interface to the system.

Module 2: Puzzle module:

A Puzzle module allows you to insert a puzzle game into a presentation. It is enough to upload an image which is later automatically divided into separate items and put in rows and columns. The required number of rows and columns can be defined in the Properties menu. Puzzle supports the following graphic formats: JPG, PNG, BMP.

Vector formats are not supported.

Module 3: server module:
The purpose of this module is to provide a centralized place where information for the system can be stored, manipulated, and accessed. This module is created to centralize and encapsulate all data storage and retrieval duties on the system. This includes user profiles, success stories, banner ads, pictures, and messages. It also provides some services, such as authentication, network communication and search.

PROPOSED WORK

This DoS/DDoS attack blocks the authorized clients from accessing the services so to overcome or to solve the problem software puzzles are created. These software puzzles are not stored or previously created, they are created at runtime whenever the client requests a particular service the software puzzle is created and the client is supposed to solve the puzzle and give the solution within the required time period. Once the puzzle is solved correctly by the client within the specified time, the client will be provided the service even if the attacker tries to block the service of the client he cannot block it.The software puzzle scheme is proposed for DoS attack. It adopts software protection technologies to ensure challenge data confidentiality and code security for an appropriate time period. It can be used in many organizations, online banking system.

EXPECTED RESULT

In this project built an experimental serve which includes a codeblock warehouse for CPU-only instructions and AES round operations, a module for puzzle generation and a module for instruction-compliant code encryption. Besides,also developed software puzzle package delivery System.

ACKNOWLEDGEMENT

We would like to take this opportunity to thank Prof. GurpreetKaurBhatti for giving us all the help and guidance we needed. We are grateful to her for her kind support. Their valuable suggestions were very helpful.

CONCLUSION

Software puzzle scheme is proposed for defeating GPU-inflated DoS attack. Itadopts softwareprotection technologies ensurechallengedataconfidentiality and code security for an appropriate time period, e.g., 1-2 seconds. 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. Since the software puzzle may be built upon a data puzzle, it can be integrated with anyexistingserver-sidedatapuzzlescheme,andeasilydeployedasthepresentclient puzzle schemes do Forexample,supposetheserverinserts someanti-debuggingcodesfordetecting Cloudplatformintosoftwarepuzzle,whenthepuzzleisrunning, thesoftware puzzle willrejecttocarryonthepuzzle-solvingprocessingonCloud environmentsuchthat the Cloud-inflated DoS attack fails. In the present software puzzle, the server has to spend time in constructing the puzzle. In other words, the present puzzle is generated at the server side. An open problem is how to construct the client-side software puzzle so as to save the server time for better defense performance. Another work is how to evaluate the effect of codede-obfuscation,whichisrelatedtothetechnologyadvanceofcodeobfuscation.

REFERENCE

[1] KeisukeIwai,NaokiNishikawa,TakakazuKurokawaAccelerationofAESencryption on CUDA GPU 2012

[2] Jason Ansel, PetrMarchenko Language-Independent Sandboxing of Just-InTime Compilation and Self-Modifying Code 2011

[3] Hsin-Yi Tsai, Yu-Lun Huang, and David Wagner A Graph Approach to Quantitative Analysis of Control-Flow Obfuscating Transformations 2009

[4] Xiaofeng Wang Mitigating Bandwidth-Exhaustion Attacks using Congestion Puzzles 2012

[5] Jeff Green, Joshua Juen, Omid Fatimah, RavinderShankesi, Dong Jin Reconstructing Hash Reversal based Proof of Work Schemes 2013