IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016

Deployment of Software Puzzle on Server to Offset DoS Attacks

Ms. Rupali Suravase[1], Prof. Pramod Patil[2]

Student, Computer Department, N.M.I.E.T., Pune, India[1]

Assistant Professor, Computer Department, N.M.I.E.T, Pune, India[2]

Abstract: Denial of Service (DoS) attack may be a malevolent decide to create a server or a network resource unobtainable to users, sometimes by briefly interrupting or suspending the services of services of a host connected to the Internet. DOS attacks and Distributed DoS (DDoS) attacks plan to eat up an internet service s resources like network information measure, memory and computation power by overwhelming the service with fake requests. Client puzzle, that demands a consumer to perform computationally expensive operations before being granted services from a server, may be a well-known measure to them. A new client puzzle is generated to countermeasure against DOS and DDoS attacks called as a software puzzle. Unlike the present consumer puzzle schemes, that publish their puzzle algorithms earlier, a puzzle algorithm with this scheme is arbitrarily generated solely 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 and the attacker needs appreciable amount of effort to translate a central processing unit puzzle software to its functionally identical GPU version in such a way that the translation cannot be done in real time.

Keywords: Software Puzzle, Denial of Service, Distributed Denial of Service, Cryptography.

I. INTRODUCTION

A Denial of service (DoS) attack is a hostile effort to make the server or a network resource unavailable to users, generally by temporarily obstructing, or suspending services of a host linked to Internet. A DoS attack typically involves efforts to briefly or indefinitely obstruct or suspend services of the host linked to Internet. Denial-of service threats are common in business and responsible for the website attacks. Resources chosen in an immense DoS attack is a specific pc, a port or a service on the targeted system, a complete network, a element of a given network any system element.

Generally, DoS attacks intend human-system communications (e.g. handicapping an alarm or printer), or a human-response systems (e.g. disabling an fundamental technicians mobile phone or desktop). DoS attacks also can target tangible system resources, like procedure resources (bandwidth, diskspace, processor time); configuration data (routing data, etc ;); state data (for example, uninvited TCP session resetting). In addition, a DoS attack can be made to enforce malware that maximums out a processor, preventing usage; trigger blunders in the machine microcode or sequencing of guidelines, taking the computer to a hazardous state; exploit operating system exposure to deplete system resources; crash the operating system completely. The dominant resemblance in these examples is that, as a result of the winning DoS attack, the system in the question doesn't respond as previous, and repair is either rejected or severely restricted. In computing, a DoS or distributed denial-of- service (DDoS) attack is an attempt to prepare a machine or network resource unavailable to its desired users. In a DoS attack, one laptop and one web affiliation is been employed to overflow a server with packets, with an aim of overburdening the targeted server's information measure and resources. DDoS attack, uses number of gadgets and multiple Internet connections, often spread globally into which is concerned to as a botnet. A DDoS attack is, therefore, much difficult to defect, simply because there is no single attacker to protect from, as the intended resource will be overloaded with requests from many hundreds and thousands of distributed sources. Distributed denial-of-service attacks are sent by two or more people, or bots, and DoS attacks are sent from one person or system. Suspects of DoS attacks generally target sites or services hosted on high-profile internet servers like banks, mastercard payment gateways, and even root nameservers.

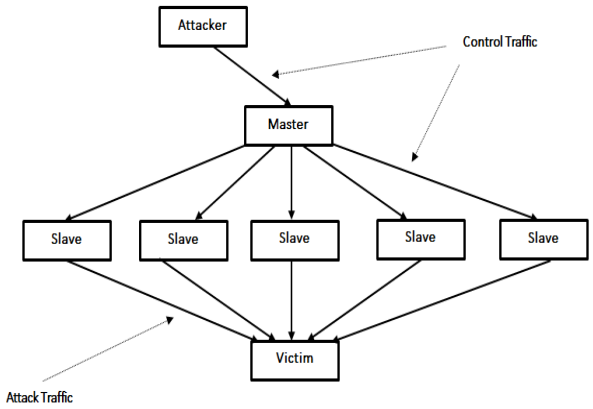

Fig. 1DoS Attack

DOI 10.17148/IJARCCE.2016.57135

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016

II. LITERATURE SURVEY

For proposed method to be better one following the literature is examined for prevailing methodology working and critically assessed on some assessing method to find defects from them.

A. Client Puzzles: A Cryptographic Countermeasure against Connection Depletion

Juels et al. [1] introduced a cryptographically based countermeasure against the connection depletion attack. They initiated a client puzzle protocol. At the point when a server goes under assault, it dispenses little cryptographic puzzles to the clients making service attempts. To finish its demand, a client should solve its puzzle perfectly. A client puzzle is associate quickly calculable cryptographic problem developed using the time, a server secret, and extra client appeal information. A client puzzle protocol doesn't need any guesswork,. It is competent of handling attacks escalated at high speeds also allows for elegant deterioration in service when an attack is performed. This perspective requires that the client previously has a program competent to solve the client puzzle.

B. Client Using Client Puzzles to protect TLS

Drew Dean et al. [2] illustrated anrealization of a simple and backwards compatible client puzzle add-on to TLS. They also confered assesments of CPU load and latency when improved library is used to sheild a secure web server. These approximation signify that client puzzles are realizable technique for shielding SSL servers from SSL based DoS attacks. The TLS protocol divides the underlying TCP stream into a record intended protocol. The TLS specification particularizes that the unknown (to a specific implementation) record type shall be ignored . Therefore, they used new record type for puzzle messages. This permits us to persist backwards compatible with old TLS implementations that does not support the puzzles. Though such implementations may time out connection if they do not reply to a puzzle, they will not observe any protocol violations. This technique is only applicable to TLS and do not work for SSLv3 as SSLv3 doesn't discard unknown record types. When the server is not under attack, no changes in the TLS protocol are needed.

C. New Client Puzzle Outsourcing Techniques for DoS Resistance

Brent Waters et al. [3] had explored new techniques for the utilize of cryptographic puzzles as a countermeasure for DoS attacks. They propound straightforward new techniques that allow the outsourcing of puzzles and their dispensation via a robust extrinsic service that is called a bastion. Numerous servers can depend on puzzles dispensation by a single bastion. Bastion does not need to be well informed of the server's using the system and those resolutions to puzzles can be assessed off-line, causing in negligible user delay. In one of the building, a bastion may comprise solely of a publicly approachable random data source, rather than a special purpose server. This outsourcing techniques help abolish puzzle distribution as a point of compromise. This design has three main benefits over previous approaches. First, it's a lot of proof versus DoS attacks aimed toward the puzzle mechanism itself, defying over 80% attack traffic than existing strategies in their experiments. Second, this scheme is economic enough to apply at the IP level, though it also works at the higher levels of the protocol stack .Third, this technique allows client to unravel puzzles off-line, reducing the requirement for users to hold up for the time their computers solve puzzles.

D. pTCP: A Client Puzzle Protocol for Defending Against Resource Exhaustion Denial of Service Attacks

T. J. McNevin et al. [5] described a defense mechanism for the transport layer, specifically for the Transmission Control Protocol (TCP). TCP is an end-to-end protocol that conveys reliable data transmission in a connection-oriented fashion. Unlike to the distributive filtering schemes for IP layer attacks, security mechanisms the transport layer ought to be incorporated into the end-to-end convention. This paper introduces a novel client puzzle protocol which uses an adjustment of the Extended Tiny Encryption Algorithm. They have discussed the architecture of a client puzzle protocol that calls pTCP. This protocol was been implemented 0n the TCP stack in Linux. pTCP has the capability to sustain a wide of range of attacks that take advantage of the vulnerabilities of the TCP protocol.

E. The Design and Implementation of Network Puzzles

Wu-chang Feng et al. [6] demonstrated network puzzles that are an distinguished mechanism for diminishing the effects of the undesirable network communication. Wu-chang Feng has discussed the design and realization of a network layer puzzle protocol and the algorithm that can be used to effectively lessen the flooding attacks and port scanning activity in this paper. They exhibited the design with an iptables implementation that supports transparent deployment of network puzzles at random locations in the network via proxies and firewalls. The system permits for high-speed executions in the fast path of modern network devices, which can be flexibly positioned, and is resistant against replay and spoofing attacks.

F. BAP: Broadcast Authentication Using Cryptographic Puzzles

Patrick Schaller et al. [7] introduced two broadcast authentication protocols on the ground of delayed key disclosure. These protocols depend on symmetric-key cryptographic primitives and use the cryptographic puzzles to provide an efficient broadcast authentication in different application situations, including those with resource-obliged wireless devices such as sensor nodes. The first protocol (BAP-1) accomplish instant message-origin authentication upon the message reception. The second protocol (BAP- 2) achieves a broadcast authentication using single transmission per authenticated message.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016

G. Toward Non-Parallelizable Client Puzzles

Suratose Tritilanunt et al. [8] inspected how to provide property of nonparallelizability in a real time puzzle. They presented a new puzzle based on subset sum problem. A client puzzle is nonparallelizable if the answer to puzzle can't be computed in parallel. Nonparallelizable client puzzles can be used to protect against DDoS attacks, where a single resister will manage an oversized cluster of compromised machines and launch attacks to targeted server from those machines. If the client puzzle is parallelizable, such an competitor could remit puzzles to different compromised machines to get puzzle solutions faster than the time expected by server.

H. Low-Cost Client Puzzles Based on Modular Exponentiation

Ghassan O. Karame et al. [9] put forth low-cost fixed-exponent and variable exponent cryptographic puzzles based on the modular exponentiation that reduces this overhead .These constructions are based on the reasonable intractability supposition in RSA: essentially in the trouble of computing little private exponent when public key is larger by several orders of the magnitude than semi-prime modulus .They also discussed puzzle construction based on CRT-RSA. Given a semi-prime modulus N, the costs incurred on the verifier in their puzzle are been decreased by factor of —N—/k when compared to the existing modular exponentiation puzzles, where k is the security parameter. They further showed that how puzzle can be integrated in a number of protocols, including those which were utilized for remote verification of calculating performance of devices and for protection against the DoS attacks.

I. Resource Inflation Threats to Denial of Service Countermeasures

R. Shankesi et al. [10] suggested Currency based mechanisms as a way to use the resource fairness among contenders for service to thwart the DoS attacks. They considered the vulnerability of the currency-based DoS defense mechanisms to various resource inflation attack's in which the attacker can substantially inflate its possession of resource at low cost and in a way that may be either difficult or may be undesirable for valid client to do. They provided a simple theoretical analysis of the resource inflation attacks and investigate its application to a number of payment schemes to rank their likely vulnerability .This find that threat of Graphics Processing Units (GPUs) for inflation attacks is especially severe.

J. Non-Parallelizable and Non-interactive Client Puzzles from Modular Square Roots

Y. I. Jerschow and M. Mauve [11] introduced a novel scheme for client puzzles based on the computation of square roots modulo a prime .Modular square root puzzles are non-parallelizable, can be utilized both interactively and non-interactively and supply polynomial granularity. They constructed the puzzle for a particular request request by assigning to it a unique quadratic residue a modulo a prime. Then the client solves the puzzle by extricating the modular square root of a and sends it to the server as evidence of work. Computation is performed by iterate squaring, which is thought to be an intrinsically sequential process. Checking the puzzle on the server side is easy-it needs a single modular squaring operation and a couple hash operations.

III.PROPOSED SYSTEM

A. Module Description

Figure 2 shows the proposed system architecture. Important notations used here are listed below for ease of reference.

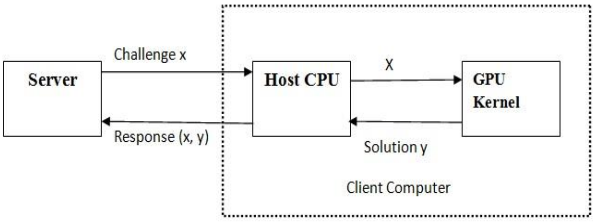

Fig. 2. Proposed System Architecture

- **x:** A challenge chosen by server.
- **m:** A message collected from environment.
- **y:** A solution to the puzzle challenge x.
- **(**(~x; ~y) A puzzle response returned from client.

I. Client Module

Client sends the request to server. When server send the puzzle, the puzzle is extracted by client. Client solves the puzzle and sends answer of puzzle to the server.

II. Server Module

Server handles the puzzle creation scenario. Server generates the puzzle sends puzzle to the client. After receiving the answer of puzzle from the client, server authenticates the client. If client is genuine then server provides the services and resources and if not then server denies the client. If the client is genuine then client solves the puzzle on its host CPU that time it requires the time in milliseconds. If the client is not genuine then, it tries to solve the puzzle by using its GPU capability.

III. Puzzle Generator Module

Puzzle generator takes some input data from server and creates the puzzle. Creation of puzzle consists of following steps.

- **Puzzle Core Generation**

Multiple operations are stored in code block warehouse. From the code block warehouse, the server first chooses mathematical operations to create a puzzle. Server calculates a message m from public data such as their IP addresses, port numbers and cookies. By using the message i.e. IP address of client, time stamp and operation from code block puzzle is generated. Only one operation is chosen at a time from code block.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016

• **Puzzle Challenge Generation:**

Only the core puzzle is encrypted using AES algorithm. Here 256-bit key is used for encryption. Outer part is not encrypted. When puzzle is received at client side , its outer part is directly executed and client has to decrypt the core puzzle and has to solve. After solving the puzzle client will send the answer to the server. The inner layer is used to encrypt the software puzzle. Therefore, after receiving puzzle , the client has to try ~y. If and only if ~y = y, the original software puzzle can be recovered and further used to solve the challenge.

IV. EXPERIMENTAL EVALUATION WITH RESULTS

The system is built by using Java to evaluate the efficiency and effectiveness. The Eclipse IDE used for Building the project. The experiments performed on Core Duo, 2GB RAM under Windows 7. For experiments, MySql is used to store the information in database. Figure. 3 illustrates that the dynamic puzzle can increase the service quality significantly in terms of the percentage of served customers.

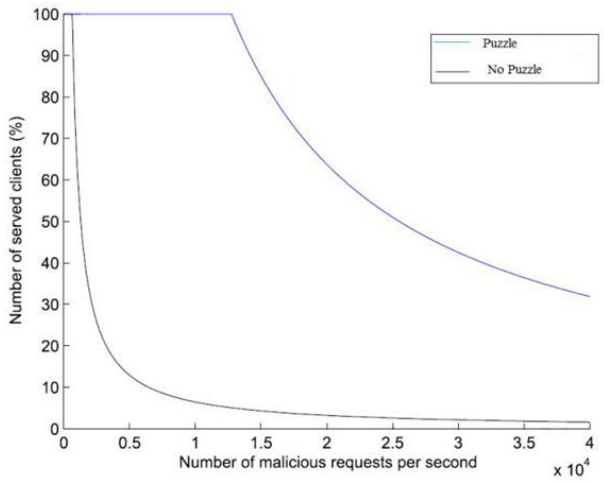

Fig. 3. Service capability comparison of server with/without software puzzle

V. CONCLUSION

Software puzzle scheme is proposed for defeating GPU-inflated Denial of Service attack. It adopts software protection technologies to make sure challenge data confidentiality and code security for an acceptable period of time. Hence, it has distinctive security demand from the traditional cipher which demands long-term confidentiality only, and code protection which focus on long-term robustness against reverse-engineering solely. Since the software puzzle may be built upon a data puzzle, it can be combined with any existing server-side data puzzle scheme, and simply installed as the present client puzzle schemes. 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.

ACKNOWLEDGMENT

We express deepest gratitude to our project guide Prof. Pramod Patil, who modeled us both technically and morally for achieving greater success in life. As a mentor and torchbearer, he guided us to overcome the odds and evens faced during the project work. We would like to extend sincere gratitude towards our PG Coordinator **Prof. L.K. Ahire, Prof. S. B. Ingle**, HOD (Comp dept., NMIET, Talegaon) who have been there for constant guidance and provided support to achieve success in our endeavour.

REFERENCES

[1] Juels, A., &Brainard, J. G. (1999, March). Client Puzzles: A Cryptographic Countermeasure against Connection Depletion Attacks.
[2] Dean, D., & Stubblefield, A. (2001, August). Using Client Puzzles to Protect TLS. In USENIX Security Symposium (Vol. 42).
[3] Waters, B., Juels, A., Halderman, J. A., &Felten, E. W. (2004, October). New client puzzle outsourcing techniques for DoS resistance. InProceedings of the 11th ACM conference on Computer and communications security (pp. 246-256). ACM.
[4] Douligeris, C., &Mitrokotsa, A. (2004). DDoS attacks and defense mechanisms: classification and state-of-the-art. Computer Networks, 44(5), 643-666.
[5] McNevin, T. J., Park, J. M., &Marchany, R. (2004). pTCP: A client puzzle protocol for defending against resource exhaustion denial of service attacks.Department of Electrical and Computer Engineering, Virginia Tech, Technical Report TR-ECE-04-10.
[6] Feng, W., Kaiser, E., &Luu, A. (2005, March). Design and implementation of network puzzles. In INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE (Vol. 4, pp. 2372-2382). IEEE.
[7] Schaller, P., Čapkun, S., & Basin, D. (2007, January). BAP: Broadcast authentication using cryptographic puzzles. In Applied Cryptography and Network Security (pp. 401-419). Springer Berlin Heidelberg.
[8] Tritilanunt, S., Boyd, C., Foo, E., & Nieto, J. M. G. (2007). Toward non-parallelizable client puzzles. In Cryptology and Network Security (pp. 247-264). Springer Berlin Heidelberg.
[9] Karame, G. O., &Čapkun, S. (2010). Low-cost client puzzles based on modular exponentiation. In Computer Security–ESORICS 2010 (pp. 679-697). Springer Berlin Heidelberg.
[10] Shankesi, R., Fatemieh, O., & Gunter, C. A. (2010). Resource inflation threats to denial of service countermeasures.
[11] Jerschow, Y. I., & Mauve, M. (2011, August). Non-parallelizable and non-interactive client puzzles from modular square roots. In Availability, Reliability and Security (ARES), 2011 Sixth International Conference on (pp. 135-142). IEEE.
[12] Wu,Y., Zhao, Z., Bao, F., & Deng, R. H. (2015). Software Puzzle: A Countermeasure to Resource-Inflated Denial-of-Service Attacks. Information Forensics and Security, IEEE Transactions on, 10(1), 168-177.

BIOGRAPHY

Ms. Rupali A. Suravase (P.G. Student) Computer Engineering Department, NMVPMs, Nutan Maharashtra Institute of Engineering and Technology, Talegaon Dabhade, Pune, India.

Prof. P. P. Patil (Assistant Professor), Computer Engineering Department, NMVPMs, Nutan Maharashtra Institute of Engineering and Technology, Talegaon-Dabhade, Pune, India.