International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 4 Issue: 5 413 - 415

Survey on: Software Puzzle for Offsetting DoS Attack

Miss.Rupali Anil Suravase
Department of Computer Engineering
NMIET, Talegaon
Pune, India
rupalisuravase@gmail.com

Mr. Pramod Patil
Department of Computer Engineering
NMIET, Talegaon
Pune, India
pramodkumar.patil@gmail.com

***Abstract*—** A Denial of Service (DoS) attack is a malevolent attempt to make a server or a network resource inaccessible to users, usually by temporarily breaking or suspending the services of a host connected to the Internet. DoS attacks and Distributed DoS (DDoS) attacks attempt to deplete an online service's resource such as network bandwidth, memory and computational power by overwhelming the service with bogus requests. Thus, DoS and DDoS attacks have become a major problem for users of computer systems connected to the Internet. Many state-art of the techniques used for defending the internet from these attacks have been discussed in this paper. After conducting an exhaustive survey on these techniques it has been found that the proposed software puzzle scheme that randomly generates only after a client request is received at the server side gives better performance as compared with previous techniques.

Keywords- *Denial of Service, Distribute Denial of Service, Client Puzzle, Software Puzzle.*

*****

I. INTRODUCTION

A denial of service (DoS) attack is a malicious attempt to make a server or a network resource unavailable to users, usually by temporarily interrupting or suspending the services of a host connected to the Internet. A DoS attack generally consists of efforts to temporarily or indefinitely interrupt or suspend services of a host connected to the Internet. Denial-of-service threats are very common in business, and responsible for website attacks. Resources targeted in a DoS attack can be a specific computer, a port or service on the targeted system, an entire network, a component of a given network any system component. Normally, DoS attacks target human-system communications (e.g. handicapping an alarm or printer), or human-response systems (e.g. disabling an fundamental technician's mobile phone or desktop). DoS attacks can also target tangible system resources, such as computational resources (bandwidth, disk space, processor time); configuration information (routing information, etc.); state information (for example, unsolicited TCP session resetting). In addition, a DoS assault can be intended to execute malware that maximums out the processor, averting utilization; trigger blunders in machine microcode or sequencing of guidelines, driving the computer into a precarious state; exploit operating system vulnerabilities to sap system resources; crash the operating system altogether. The paramount similarity in these examples is that, as a result of the winning DoS attack, the system in question doesn't respond as before, and repair is either denied or severely restricted. In computing, a DoS or distributed denial-of- service (DDoS) attack is an endeavor to form a machine or network resource unavailable to its meant users. In a DoS attack, one computer and one internet connection is used to flood a server with packets, with the aim of overloading the targeted server's bandwidth and resources. DDoS attack, utilizes numerous gadgets and multiple Internet connections, often distributed globally into what is referred to as a botnet. A DDoS attack is, therefore, much harder to defect, simply because there is no single attacker to defend from, as the targeted resource will be flooded with requests from many hundreds and thousands of multiple sources. Distributed denial-of-service attacks are sent by two or more people, or bots, and DoS attacks are sent by one person or system. Perpetrators of DoS attacks typically target sites or services hosted on high-profile web servers such as banks, credit card payment gateways, and even root name-servers. [4]

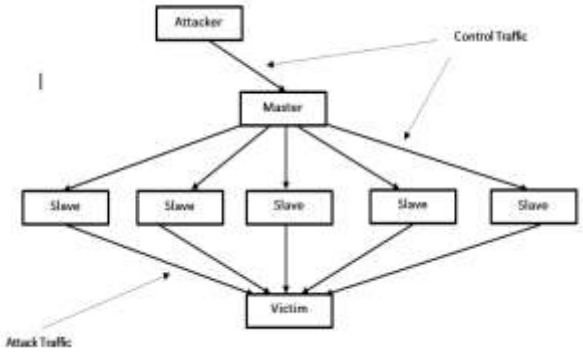

Figure1. DoS Attack

II. RELATED WORK

For proposed methodology to be better one following literature is analyzed for existing methodology working and critically evaluated on some evaluation method to find shortcomings from them.

A. Client Puzzles: A Cryptographic Countermeasure Against Connection Depletion Attacks

Juels et al. [1] introduced a cryptographically based countermeasure against connection depletion attack. They introduced a client puzzle protocol. At the point when a server goes under assault, it distributes little cryptographic puzzles to clients making service requests. To finish its demand, a client should solve its puzzle correctly. A client puzzle is an quickly computable cryptographic problem formulated using the time, a server secret, and additional client request information. A client puzzle protocol doesn't require any assumption. It is capable of handling attacks mounted at very high speeds as well as also allows for graceful degradation in service when an attack is mounted. This approach requires that the client already have a program capable of solving a client puzzle.

B. Client Using Client Puzzles to protect TLS

Drew Dean et al. [2] described an implementation of a simple and backwards compatible client puzzle extension to

TLS. They also presented measurements of CPU load and latency when modified library is used to protect a secure web server. These estimations demonstrate that client puzzles are feasible method for shielding SSL servers from SSL based DoS attacks. The TLS protocol separates the underlying TCP stream into a record oriented protocol. The TLS specification specifies that unknown (to a particular implementation) record type shall be ignored. Therefore, they used a new record type for the puzzle messages. This allows us to remain backwards compatible with old TLS implementations that do not support puzzles. Though such implementations may time out a connection if they do not reply to a puzzle, they will not notice any protocol violations. This technique is only applicable to TLS and does not work for SSLv3 as SSLv3 does not discard unknown record types. When the server is not under attack, no changes in the TLS protocol are required.

C. New Client Puzzle Outsourcing Techniques for DoS Resistance

Brent Waters et al. [3] had explored new techniques for the use of cryptographic puzzles as a countermeasure to DoS attacks. They proposed simple new techniques that permit the outsourcing of puzzles and their distribution via a robust external service that is called a bastion. Numerous servers can depend on puzzles distributed by a single bastion. Bastion does not need to be aware of the server's using the system and those solutions to puzzles can be computed off-line, resulting in minimal user delay. In one of the constructions, a bastion may consist merely of a publicly accessible random data source, rather than a special purpose server. This outsourcing techniques help eliminate puzzle distribution as a point of compromise. This design has three main advantages over prior approaches. First, it's a lot of proof against DoS attacks aimed toward the puzzle mechanism itself, withstanding over 80% attack traffic than previous strategies in their experiments. Second, this scheme is cheap enough to apply at the IP level, though it also works at higher levels of the protocol stack. Third, this technique permits client to unravel puzzles offline, diminishing the requirement for users to hold up while 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, particularly for the Transmission Control Protocol(TCP). TCP is an end-to-end protocol that delivers reliable data transmission in a connection-oriented fashion. Dissimilar to the distributive filtering schemes for IP layer attacks, security mechanisms for the transport layer ought to be incorporated into the end-to-end convention. This paper presents a novel client puzzle protocol that uses an adjustment of the Extended Tiny Encryption Algorithm. They have described the architecture of a client puzzle protocol that calls pTCP. This protocol was implemented into the TCP stack in Linux. pTCP has the capability to combat a wide of range of attacks that take advantage of the vulnerabilities of the TCP protocol. Not only can pTCP defend against synood attacks and help mitigate ash-crowds, it also has the capability to defend against other attacks that may exist at the application layer.

E. The Design and Impletion of Network Puzzles

Wu-chang Feng et al. [6] introduced network puzzles that are an elegant mechanism for mitigating the effects of undesirable network communication. Wu-chang Feng has described the design and implementation of a network layer puzzle protocol and algorithm that can be used to effectively slow down flooding attacks and port scanning activity in this paper. They demonstrated the design with an iptables implementation that supports transparent deployment of network puzzles at arbitrary locations in the network via proxies and firewalls. The system allows for high-speed implementations in the fast path of modern network devices, can be flexibly deployed, and is resistant against replay and spoofing attacks.

F. BAP: Broadcast Authentication Using Cryptographic Puzzles

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

G. Toward Non-Parallelizable Client Puzzles

Suratose Tritilanunt et al. [8] investigated how to provide the property of non-parallelizability in a practical puzzle. They proposed a new puzzle based on the subset sum problem. A client puzzle is non-parallelizable if the answer to the puzzle can't be computed in parallel. Non-parallelizable client puzzles can be used to defend against DDoS attacks , where a single adversary can control a large group of compromised machines and launch attacks to the targeted server from those machines. If the client puzzle is parallelizable, such an adversary could deliver puzzles to different compromised machines to get puzzle solutions quicker than the time expected by the server.

H. Low-Cost Client Puzzles Based on Modular Exponentiation

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

I. Resource Inflation Threats to Denial of Service Countermeasures

R. Shankesi et al. [10] proposed Currency-based mechanisms as a way to use resource fairness among contenders for a service to thwart DoS attacks. They consider the vulnerability of currency-based DoS defense mechanisms to various resource inflation attacks in which an attacker can

substantially inflate its possession of the resource at low cost and in a way that may be either difficult or undesirable for a valid client to do. They provided a simple theoretical analysis of resource inflation attacks and investigate its application to a number of payment schemes to rank their likely vulnerability. This find that the threat of Graphics Processing Units (GPUs) for inflation attacks is especially severe. This can also review threats from other capabilities, including multi-core processors, cloud computing, and bandwidth inflation schemes.

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 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. Puzzle trouble can be tuned by selecting a larger or smaller prime modulus. This can evaluate the performance of modular square root puzzles by benchmarking the verification throughput and the solution time for different levels of difficulty.

K. Software Puzzle: A Countermeasure to Resource Inflated Denial-Of-Service Attacks

A client puzzle, which demands a client to perform computationally expensive operations before being granted services from a server, is a well-known countermeasure to them. However, an attacker can inflate its capability of DoS or DDoS attacks with fast puzzle solving software and built-in graphics processing unit (GPU) hardware to significantly weaken the effectiveness of client puzzles. This paper introduces a new client puzzle is generated to countermeasure against DOS and DDoS attacks called as a software puzzle. Unlike 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:

1) An attacker is unable to prepare an implementation to solve the puzzle in advance and

2) The attacker needs considerable effort in translating central processing unit puzzle software to its functionally equivalent GPU version such that the translation cannot be done in real time.

As rewriting/translating a software puzzle is time-consuming, which may take even more time than solving the puzzle on the host CPU directly; software puzzle thwarts the GPU-inflated DoS attacks. Not at all like the existing client puzzle schemes which publish a puzzle function in advance, the software puzzle scheme that dynamically creates the puzzle function. Software puzzle aims to prevent GPU from being used in the puzzle-solving process based on different instruction sets and real-time environments between GPU and CPU.

III. CONCLUSION

This paper presents a comprehensive survey on puzzles which protects the system from DoS and DDoS attacks. This paper proposes a software puzzle which is capable of defeating GPU inflated DoS attack. This method ensures challenge data confidentiality and code security for an appropriate time period for 1-2 seconds by using software protection techniques.

REFERENCES

[1] Juels, Ari, and John Brainard. "Cryptographic countermeasures against connection depletion attacks." U.S. Patent No. 7,197,639. 27 Mar. 2007.

[2] Dean, Drew, and Adam Stubblefield. "Using Client Puzzles to Protect TLS."*USENIX Security Symposium*. Vol. 42. 2001.

[3] Waters, Brent, et al. "New client puzzle outsourcing techniques for DoS resistance." *Proceedings of the 11th ACM conference on Computer and communications security*. ACM, 2004.

[4] Douligeris, Christos, and Aikaterini Mitrokotsa. "DDoS attacks and defense mechanisms: classification and state-of-the-art." *Computer Networks* 44.5 (2004): 643-666.

[5] McNevin, Timothy J., Jung-Min Park, and Randolph Marchany. "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* (2004).

[6] Feng, Wenjie, E. Kaiser, and Antoine Luu. "Design and implementation of network puzzles." *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*. Vol. 4. IEEE, 2005.

[7] Schaller, Patrick, Srdjan Čapkun, and David Basin. "BAP: Broadcast authentication using cryptographic puzzles." *Applied Cryptography and Network Security*. Springer Berlin Heidelberg, 2007.

[8] Tritilanunt, Suratose, et al. "Toward non-parallelizable client puzzles."*Cryptology and Network Security*. Springer Berlin Heidelberg, 2007. 247-264.

[9] Karame, Ghassan O., and Srdjan Čapkun. "Low-cost client puzzles based on modular exponentiation." *Computer Security–ESORICS 2010*. Springer Berlin Heidelberg, 2010. 679-697.

[10] Shankesi, Ravinder, Omid Fatemieh, and Carl A. Gunter. "Resource inflation threats to denial of service countermeasures."(2010).

[11] Jerschow, Yves Igor, and Martin Mauve. "Non-parallelizable and non-interactive client puzzles from modular square roots." *Availability, Reliability and Security (ARES), 2011 Sixth International Conference on*. IEEE, 2011.

[12] Wu, Yongdong, et al. "Software Puzzle: A Countermeasure to Resource-Inflated Denial-of-Service Attacks." *Information Forensics and Security, IEEE Transactions on* 10.1 (2015): 168-177