International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

A Cumulative Study on Counter Measure Technique for DOS Attacks Using Software Puzzle

Jyoti B. Shende[1], S. V. Todkari[2]

[1]ME Computer (Engineering), Jayawantrao Sawant College of Engineering,
Hadapsar Pune-28, Savitribai Phule Pune University, Pune, India

[2]Head of Department, IEEE Member, Information Technology Engineering, Jayawantrao Sawant College of Engineering,
Hadapsar Pune-28,Savitribai Phule Pune University, Pune, India

Abstract: ***Denial-of-service (DoS) and Distributed Denial of Service (DDoS) are dangerous to cyber-security, and client puzzle, which request a client to perform computationally very high operations before providing services from a server to client, is a well-known countermeasure to them. After all, an attacker can boost its capability of DoS/DDoS attacks with fast puzzle solving software and/or in-built graphics processing unit such like (GPU) hardware to extremely weaken the effectiveness of client puzzles. There are many system exist like Timelock puzzle, Client puzzle are used in this paper. In this paper, we study how to prohibit DoS/DDoS attackers from inflating their puzzle-solving capacities. To this end, we introduce a new client puzzle named as software puzzle. Unlike the existing client puzzle strategy, which publish their puzzle algorithms previously and generate software puzzle for each client request, a puzzle algorithm in the present software puzzle scheme is created only after threshold value of client request exceed which is accepted at the server side by using decision tree and the algorithm is generated such that: 1) an attacker is not able to prepare an implementation to solve the puzzle previously and 2) the attacker needs extensive effort in translating a central processing unit puzzle software to its functionally equal GPU version such that the transformation cannot be done in real time.***

Keywords: Software puzzles generation, information gain, GPU programming, distributed denial of service (DDoS).

1. Introduction

Denial of Service (DoS) attacks and Distributed DoS (DDoS) attacks try to damage an online service's resources such as network bandwidth, memory and computation power by outstanding the service with bogus requests. When client establishing connection with server needs a lot of CPU time to make SSL handshake. It may result an insufficient resources are left to providing services. In this case, conventional cryptographic tools do not enhance the availability of the services; in fact, they may reduce service quality due to expensive cryptographic operations. The seriousness of the DoS/DDoS problem and their increased frequency has led to the advent of numerous defense mechanisms [2]. In this paper, we are particularly excited in the countermeasures to DoS/DDoS attacks on server computation power. Client puzzle [3] is a well-known approach to increase the cost of clients as it pressure the clients to carry out heavy operations before being granted services. Generally, a client puzzle strategy consists of three steps: puzzle generation, puzzle solving by the client and puzzle verification by the server. Many of the system are existed which are using techniques like Timelock puzzle, client puzzle rather than this technology some other techniques also available like mod_kaPoW.

So, this paper presenting an idea of Software puzzle which taking input as request from client, and process the step using software puzzle. Therefore, in either case, a client puzzle can significantly reduce the impact of DoS attack because it permit a server to spend much less time in handling the bulk of malicious requests. Server gives threshold value of client requests, if requests exceeds the threshold value then software puzzle is given to client. Otherwise requested client is a legitimate client operate it's task normally. This paper not only classify the attack is DoS/DDoS and but also request type. Optimizing the puzzle verification mechanism is very important and doing so will undoubtedly improve the server's performance.

For further proceeding of this paper section II is dedicated for related work, section III is for proposed work, section IV gives system architecture and section V is for conclusion.

2. Related Work

1) Client puzzles: A cryptographic countermeasure against connection depletion attacks.
In this paper, introduce a new approach that we refer to as the client puzzle protocol, the aim of which is to fight against connection depletion attacks. The idea is quite simple, when there is no witness of attack, a server accepts connections request normally, that is aimlessly. When a server comes under attack, it accepts connections selectively. In particular, the server gives to each client wishing to make a connection a unique client puzzle. A client puzzle is an quickly computable cryptographic problem formulated using the time, a server secret, and additional client request information. The server resource allocated to it for a connection, the client must submit to itself for a connection, the client must submit to the server a accurate solution to the puzzle it has been given. Client puzzle are deployed in union with conventional time-outs on server resources. Thus, while genuine client will experience only a small degradation in connection time when a server comes under attack, an attacker must have access to large computational resource to create breach in service. Cryptographic puzzles have been used for several task, such as fighting against junk e-mail, creating digital time capsules, and metering Web site usage.

2) Reconstructing Hash Reversal based Proof of Work Schemes

In this paper, elaborated an idea of Proof of Work (PoW) mechanisms, in which a server request that clients prove they have done work previously it commits resources to their requests. Most PoW mechanisms are puzzle-based techniques in which clients solve processing thorough puzzles. For instance, Hash Cashes are puzzle-based mechanisms that aim to prohibit an attacker from sending too much spam. As attacks use more resources, and therefore the puzzle difficulties increase, weaker legitimate clients may experience unacceptable requirements to obtain service. While computationally weaker clients would experience longer latencies during an attack, it would be extremely more functional than a protocol without the PoW based defense. Using Graphical Processing Units (GPUs) provides a powerful technique for launching resource inflation attacks. The attackers can use cheap and widely available GPUs to boost their ability to solve typical hash reversal based puzzles by a factor of more than 600. This paper is the calculation of Hash- Reversal PoW schemes in the presence of resource-inflated attackers. In this show that client-based adaptation is necessary for providing satisfactory service to genuine clients in this situation. Additionally, it show that an robust hash reversal PoW scheme based only on server load will fail to provide service, and can create a novel DoS attack against fair clients. Given these results, hash reversal PoW strategy proposed for DoS protection mechanisms should keep track of client behavior given the developing threat of GPGPU based attacks.

3) Time-lock puzzles and timed-release crypto.

This paper narrate the notion of timed-release crypto where the goal is to encrypt a message so that it can not be decrypted by anyone, not even the sender, until a pre-arranged amount of time has passed. The goal is to send information into the future. We study the problem of creating computational puzzles, called time-lock puzzles that require a precise amount of time to solve. The solution to the puzzle reveals a key that can be used to decrypt the encrypted information. This approach has the obvious problem of trying to make CPU time and real time agree as closely as possible but is nonetheless interesting. The more computational resources might be able to solve the time lock puzzle more quickly, by using large parallel computers. Another approach is the puzzle doesn't automatically become solvable at a given time; slightly, a computer needs work continuously on the puzzle until it is solved.

4) mod_kaPoW: Mitigating DoS with transparent proof-of-work

This paper described a approach of mod_kaPoW system that has the efficiency and human transparency of proof-of-work strategy and also having the software backwards compatibility. There are several disadvantages of using CAPTCHAs. One drawback is the user-interface problem they create; users with visual disabilities are unable to access content legitimately while natural users find it increasingly difficult to solve CAPTCHAs correctly as the images have become less readable in order to thwart sophisticated attacker that have developed automated solvers for simple CAPTCHAs. Another drawback is the static nature of the problems being given out. A proof-of-work scheme alters the operation of a network protocol so that a client must rebound their challenge along with a correct answer before being granted service. The challenge acts as a refine for clients based on their willingness to solve a computational task of varying difficulty. This paper describes the design, performance, and evaluation of a novel web based proof-of-work system that provides the benefit of configurable PoW protocols in a portable manner. Unlike CAPTCHAs, the system is transparent to its users and supports backwards compatibility for traditional clients. The basic approach only requires changes to web servers and is similar to the URL rewriting approach employed by content-distribution networks such as Akamai. In the approach, the web server dynamically rewrites URL references by attaching a computational puzzle to them.

5) Proofs of work and bread pudding protocols

This paper introduces an idea of bread pudding protocol. Bread pudding is a dish that originated with the purpose of reusing bread that has gone stale. In the same manner, a bread pudding protocol to be reused by the verifier to achieve a separate, useful, and verifiable correct computation. In this paper, we deviate from the standard cryptographic aim of proving knowledge of a secret, or the truth of a mathematical statement. POW is a protocol not defined or treated formally, POWs have been defined as a mechanism for a number of security goals, including server access metering, construction of digital time capsules, uncheatable benchmarks and denial of service. This paper contribute bread pudding protocol to be a POW such that the computing effort invested in the proof may be harvested t achieve a separate, useful and verifiably correct computation. These POWs can serve in their own right as mechanisms for security protocols as well as harvested in order to outsource the MicroMint minting operation to a large group of untrusted computational devices.

3. Proposed Work

Existing system mainly concentrated on how to prevent DoS/DDoS attackers from inflating their puzzle-solving capabilities. It doesn't give threshold value for client request handle at server side and never classify types of attack like DoD/DDoS nor kinds of requests. In proposed work, we consider threshold value of requests, types of attack as well as requests.

4. Architectural View

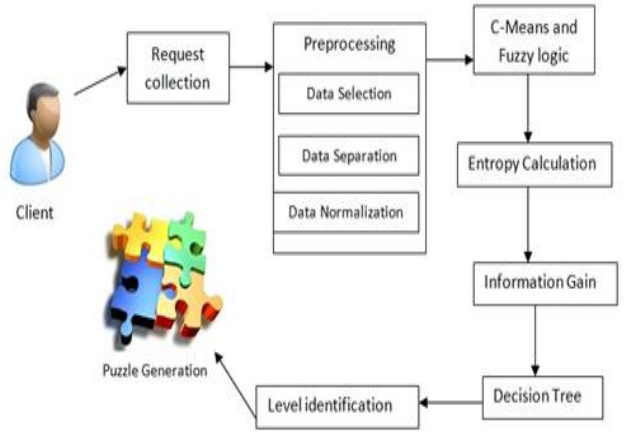

Figure: System Architecture

Sr.No.	Paper	Technique	Advantages	Disadvantages
1	Client puzzles: A cryptographic countermeasure against connection depletion attacks[3].	When server comes under attack , it distribute cryptographic puzzle to client whom want service from server.	1) This model is most robustness in stronger attack, capable of handling attacks mounted at very high speed. 2) This protocol can be built straightforward or can be layered on top.	1) It requires special client side software and client already have a program capable of solving a client puzzle.
2	Reconstructing Hash Reversal based Proof of Work Schemes[4]	This PoW schemes proposed for DoS protection mechanisms which keep track of client behavior given the emerging threat of GPGPU based attacks. A server orders that the clients submit a proof of the work they have performed before processing their request.	This strategy can effectively restrict a resource scaling attacker's capabilities by adjusting puzzle difficulty based on past client behavior.	As attacks use more resources, and the puzzle difficulties increases, weaker legitimate client may experience unnecessary requirements to obtain service.
3	Time-lock puzzles and timed-release crypto[5]	This paper narrates Encrypt a message it can't be decrypted by anyone, not even sender until a pre-arranged time has elapsed.	1) Computational problems that can't be solved without running a computer continuously for at least a certain amount of time, use trusted agents who don't reveal certain information until a specified date.	The CPU time required to solve a problem can depend on the amount and nature of the hardware used to solve the problem and the parallelizability of the computational problem being solved.
4	mod_kaPoW: Mitigating DoS with transparent proof-of-work[6]	This paper present mod_kaPoW a novel system that has the efficiency and human transparency of proof-of-work schemes as well as the software backwards-compatibility of CAPTCHA schemes.	1) This system is transparent to the end users and gives backward compatible to end users. 2) It doesn't require special client software. 3) In the system, a web server dynamically changes URLs.	This technique has a overhead when processing files containing a variable number of URLs.
5	Proofs of work and bread pudding protocols[7]	This paper describes a bread pudding protocol to be a POW such that the computational effort invested in the proof may be reused by the verifier to achieve a helpful, verifiable correct calculations.	Achieve security goal and client pay for access to a resource by offering small amount of its computational power.	The highly computationally intensive operation of minting in the MicroMint strategy.

5. Conclusion

As this complete paper narrate different methodology on software puzzle, but none of the methodology are seems to be perfect. So, this paper as bit introduce an idea of software puzzle which is generated by using fuzzy logic and decision tree, server send query to those client reaching above the threshold value in the warehouse. In this paper also classify the type of request as well as types of attacks that is DoS/DDoS.

References

[1] "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.
[2] "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.

[3] "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.
[4] "Reconstructing Hash Reversal based Proof of Work Schemes," J. Green, J. Juen, O. Fatemieh, R. Shankesi, D. Jin, and C. A. Gunter, in Proc. 4th USENIX Workshop Large-Scale Exploits Emergent Threats, 2011.
[5] "Time-lock puzzles and timed-release crypto," R. L. Rivest, A. Shamir, and D. A. Wagner, Dept. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep. MIT/LCS/TR-684, Feb. 1996.
[6] "mod_kaPoW: Mitigating DoS with transparent proof-of-work," E. Kaiser and W.-C. Feng, in Proc. ACM CoNEXT Conf., 2007.
[7] "Proofs of work and bread pudding protocols," M. Jakobsson and A. Juels, in Proc. IFIP TC6/TC11 Joint Working Conf. Secure Inf. Netw., Commun. Multimedia Secur., 1999.

Author Profile

Ms. Jyoti B. Shende, is currently pursuing M.E (Computer) from Department of Computer Engineering, Jayawantrao Sawant College of Engineering, Pune, India. Savitribai Phule Pune University, Pune, Maharashtra, India -411007. She received her B.E (Computer) Degree from S.B.Patil college of engineering, Indapur, Savitribai Phule Pune University, Pune, Maharashtra, India -411007. Her area of interest is network security.

Assoc Prof. Sachin V. Todkari, received his ME. (IT) Degree from MIT college of engineering Kothrud, Maharashtra, India -411007. He received his B.E (Computer Science) Degree from College of Engineering, Ambajogai, Maharashtra, India. He is currently working as HOD at Department of Information Technology Engineering, in Jayawantrao Sawant College of Engineering, Savitribai Phule Pune University, Pune, Maharashtra, India -411007. His area of interest is Wireless Sensor Network.