Vancha Maheshwar Reddy, B Sandhya Rani, J Vedika, Ch. Veera Reddy / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 3, Issue 1, January -February 2013, pp.751-757

Game Theory based Defense Strategy against Denial of Service Attack using Puzzles

Vancha Maheshwar Reddy[1], B Sandhya Rani[2], J Vedika[3], Ch. Veera Reddy[4]

([1,2,3] Assistant Professor, ACE Engineering College, Hyderabad)
([4]Assistant Professor, SCIENT Institute of Technology, Hyderabad)

ABSTRACT

Security issues have become a major issue in recent years due to the advancement of technology in networking and its use in a destructive way. A number of defense strategies have been devised to overcome the flooding attack which is prominent in the networking industry due to which depletion of resources takes place. But these mechanism are not designed in an optimally and effectively and some of the issues have been unresolved. Hence in this paper we suggest a Game theory based strategy to create a series of defence mechanisms using puzzles. Here the concept of Nash equilibrium is used to handle sophisticated flooding attack to defend distributed attacks from unknown number of sources

General Terms: Computer Networks and Security.
Keywords: Dos Attacks, Game Theory, Puzzles.

I. INTRODUCTION

The pace with which the technology is advancing is amazing. With the advancement in technology there has been a great advancement in networking too. Networking today has become inevitable and is a part and parcel in various aspects of our life. If we consider the present business and political scenario, there has been a rat-race going on which has made individuals not only upgrade their own resources but also degrade their competitor's resources by some malicious activities. Hence in recent years, security concerned issues has received enormous attention in networked system because of availability of services. Networked systems are vulnerable to DoS (Denial of Services) attack. A Denial-of-Service attack (Dos attack) is a type of attack on a network that is designed to bring network to its knees by flooding it with useless traffic. In this area, most researches are based on designing and verifying various defense strategies against denial-of-service (DoS) attacks. A DoS attack characterizes a malicious behavior preventing the legitimate users of a network from using the services provided by that network. Flooding attacks and Logic attacks are the two principal classes of DoS attack. ([1], [2], [3], [4]).

Flooding attacks examples are SYN flood, Smurf, TFN2K which sends a large number of requests to service provided by victim system. SYN flood uses resource starvation to achieve DoS attack [5] whereas Smurf attack uses bandwidth consumption to disable victim system's network resources [6] and TFN2K attacks are launched using spoofed IP addresses, making detecting the sources of the attacks more difficult[7]. These requests reduce or use up some key resources of victim by large amount and so legitimate user's requests for same resources are denied. Capacity of a buffer, CPU time to process requests, available bandwidth of a communication channel are some of the resources of a networked system[4]. The depleted resources revive when the flooding attack stops. Examples of Logical attack are Ping-of-Death, Teardrop .In logical attack, victim's vulnerable software accepts and process a forged fatal message which leads to resource exhaustion. Flooding attack and Logical attack will act as memory eaters, bandwidth loggers, or system crashers. Appropriate remedial actions are to be adopted against logical attacks since effects of attack remain even after attack, whereas it is not the case in flooding attacks. The contents of attack message and legitimate message differ and by making distinction among them, logical attack can be thwarted, which is not possible in flooding attack [4]. As such distinction is not possible in flooding attack; the defence becomes an arduous task against flooding attacks. Here in this paper have solely focused on Flooding Attacks, There is no "magical panacea" for potential threats in computer security. To defend a system in network there is only one way that is to design and employ a number of protections or defence mechanisms that mitigates a specific threat. Large number of defenses against flooding attack have been devised which may be reactive or preventive. Mechanisms such as pushback [8], traceback [9], or filtering [10] are reactive mechanisms which alleviate the impact of flooding attack by detecting the attack on the victim, but they all have significant drawbacks that limit their practical utility in the current scenario. Whereas Preventive strategies make the victim able to tolerate the attack without the legitimate user's request getting denied. Preventive mechanism

enforces restrictive policies such as use of client puzzles that limits the resource consumption. Generally reactive mechanisms have some drawbacks. It suffers from scalability and attack traffic identification problems [4].

Dos can be effectively beaten by utilizing Client Puzzles. In client puzzle approach, the client needs to solve the puzzle produced by the defender (server) for getting services. The server produces computational puzzles to client before committing the resources. Once the sender solves the puzzle he is allocated the requested resources. The attacker who intends to use up the defender's resources by his repeated requests is deterred from perpetrating the attack, as solving a puzzle is resource consuming. To preserve the effectiveness and optimality of this mechanism, the difficulty level of puzzles should be adjusted in timely manner. Network puzzles and puzzle auctions tried to adjust difficulty level of puzzles but they are not much suitable in incorporating this trade-off. In this paper, we show that Puzzle-based mechanism can be effectively studied using game theory. This paper shows Puzzle-based defence mechanism modeled as two player game, one player as attacker who perpetrates a flooding attack and other as defender who counters the attack using client puzzles. Then Nash equilibrium is applied on game which leads to description of player's optimal strategy [4].

II. RELATED WORK:

Burszteinetal [11] presented a model for evaluating the plausibility of successful attacks on a given network with interdependent files and services. This work provided a logic model that accounts for the time needed to attack, crash, or patch network systems. Rather than providing a game theoretic model, the work used the given time and topology constraints to determine if an attack, or defence, would be successful. Sun et al [12] analyzed information security problem in the mobile electronic commerce chain. They claimed that the application of game theory in information safety is based on the hypothesis of player's perfect rationality. Sun et al used game theory to make the analysis and put forward strategy suggestions for defender organization to invest in information security. It is concerned about management and not the technology of the information security. They formulated the problem of two organizations investing in the security, with parameters such as for investment, security risk and disasters. They presented a pay-off matrix. They did the Nash Equilibrium analysis for both pure and mixed strategy and showed them to be consistent. To make the investing a rational option they introduced a penalty parameter associated with not investing. They concluded by presenting an argument for encouraging organizations the investment in information security the original idea of cryptographic puzzles is due to Merkle [13]. However, Merkle used puzzles for key agreement, rather than access control. Client puzzles have been applied to TCP SYN flooding by Juels and Brainard [14]. Aura, Nikander, and Leiwo [15] apply client puzzles to authentication protocols in general [16]. Dwork and Naor presented client puzzles as a general solution to controlling resource usage, and specifically for regulating junk email. Their schemes develop along a different axis, primarily motivated by the desire for the puzzles to have shortcuts if a piece of secret information is known. Xu et al. [17] proposed a game-theoretic model to defend a web service under DoS attack. They used a single bottleneck link to simulate the attacks. The metrics used for the performance of their system are total throughput of the attackers and their legitimate clients, legitimate client's average amount of time to download a web page, number of concurrent attackers and clients, and packet drop probability of the attackers and the clients.

Nonetheless, an attacker who knows the defender's possible actions and their corresponding costs may rationally adopt his own actions to defeat a puzzle-based defence mechanism. For example, if the defender produces difficult puzzles, the attacker responds them at random and with incorrect solutions. In this way, he may be able to exhaust the defender's resources engaged in solution verification. If the defender produces simple puzzles, the mechanism is not effective in the sense that the attacker solves the puzzles and performs an intense attack. Moreover, even if the defender enjoys efficient low-cost techniques for producing puzzles and verifying solutions, he should deploy the effective puzzles of minimum difficulty levels, i.e., the optimum puzzles, to provide the maximum quality of service for the legitimate users. Hence, the difficulty level of puzzles should be accurately adjusted in a timely manner to preserve the effectiveness and optimality of the mechanism. Although some mechanisms such as [19] and [20] have attempted to adjust the difficulty level of puzzles according to the victim's load, they are not based on a suitable formalism incorporating the above trade-offs and, therefore, the effectiveness and optimality of those mechanisms have remained unresolved [4]

III. CLIENT PUZZLE APPROACH

Currently intruders are beginning to more often use legitimate, or expected, protocols and services as the vehicle for packet streams. The resulting attacks are hard to defend against using standard techniques, as the malicious requests differ from the legitimate ones in intent but not in content. Filtering or rate limiting based on anomalous packets are not feasible at all. In fact, filtering or

rate limiting an attack that is using a legitimate and expected type of traffic may in fact complete the intruder's task by causing legitimate services to be denied. Currently, the most feasible way to handle this kind of situation is using the Turing Test mechanism as in Kill-bots. The graphical CAPTCHAs are most widely used today. It consists of a picture with some degraded or distorted image, which will take up a lot of valuable bandwidth especially in the case of the attack. Those graphical CAPTCHA consists of a picture with some degraded or distorted image. In the case of DDoS attack, sending those images from the server to the client for authentication actually consumes quite considerable bandwidth. T. Y. Chan has used the text-to-speech approach to generate the audio-based Turing test, yet it shows that the audio CAPTCHA largely ineffective. And actually audio-based Turing test will also consume remarkable bandwidth. Thus low bandwidth Turing test is very desirable for preventing the DDoS attack. One possible low-bandwidth Turing test is using text-based question answering, since computational linguistics is one of the most prominent research disciplines in artificial intelligence, and at the same time, Turing test in text format normally consume much less bandwidth. Although humans find it easy to understand the natural languages, computers do not [21]. The client puzzle approach means that before engaging in any resource consuming operations, the server first generates a puzzle and sends its description to the client that is requesting service from the server. The client has to solve the puzzle and send the result back to the server. The server continues with processing the request of the client, only if the client's response to the puzzle is correct. This is summarized in the following abstract protocol, where C and S denote the client and the server, respectively: [22]

Step 1 C → S: sending service request
Step 2 S: generation of a puzzle
Step 3 S → C: sending description of the puzzle
Step 4 C: solving the puzzle
Step 5 C → S: sending solution to the puzzle
Step 6 S: verification of the solution
If the solution is correct:
Step 7 S: continue processing service request

One can view the first six steps of the protocol as a preamble preceding the provision of the service, which is subsumed in a single step (step 7) in the above abstract description. The preamble provides a sort of algorithmic protection against DoS attacks. The server can set the complexity level of the puzzle according to the estimated strength (computational resources) of the attacker. If the server manages to set an appropriate complexity level, then solving the puzzle slows down the DoS attacker who will eventually abandon his activity [23]. The idea of puzzle was introduced as early as 1978, and Merkle was the first to incorporate the concept of cryptographic puzzles into authentication protocols. Merkle introduced an idea that, in a given communication, one legitimate participant sends several cryptographic problems that would be broken by the other participant. The security against an eavesdropper is based on the fact that the attacker is forced to solve all the puzzles whereas the legitimate participant only needs to choose and solve one puzzle. Current Client puzzle proposals apply some of Merkle's ideas [24]. Ideal characteristics of a client puzzle protocol [37] First, a puzzle should be easy for the server to create and verify, and should be much more difficult for the client to solve. The level of difficulty can be parameterized, and can be changed if needed. However, if the server is not under an attack, it should be possible that a puzzle would not be generated at all, allowing the client access without solving a puzzle. Second, it should not be possible for an attacker to keep a table of known puzzles and solutions [25] [26]. Third, the client should know that it has the correct answer before submitting it to the server. The puzzle solving process involves a repetitive brute-force task. The client should know when to terminate this process when it has the correct solution. Fourth, the server should know what puzzles it has generated and which ones to verify. There must be some type of mechanism in place that prevents an attacker from fabricating its own puzzle and sending its own solution to the server. The server needs to store a small amount of information so that it can determine which responses from the clients are solutions to valid puzzles Puzzle Characteristics [22] The computational costs employed by the server in generating and verifying the puzzles must be significantly less expensive that the computational costs employed the client in solving the puzzles. The puzzle difficulty, which depends on the server's resources availability, should be easily and dynamically adjusted during attacks.

Clients have a limited amount of time to solve puzzles.

Pre-computing puzzle solutions should be unfeasible.

Having solved previous puzzles does not aid in solving new given puzzles.

Before a correct puzzle solution is submitted, the server does not keep a record of the connection's state.

Initially In the Client Puzzle approach The Tiny Encryption algorithm (TEA) which is a block-cipher encryption algorithm was proposed in 1994 by Wheeler and Needham [27][28]. Both the encryption and decryption algorithms are Feistel type routines that encrypt or decrypt data by addition, subtraction, bit-shifting, and exclusive-OR

operations. The goal of the encryption algorithm is to create as much diffusion2 as possible by incorporating many rounds or iterations of these operations. After TEA was released, certain minor weaknesses were discovered in the encryption algorithm [29]. In response, Wheeler and Needham developed an extension to TEA, called XTEA [30] Later Timothy, Jung-Min & Randolph [31] used variation of XTEA which uses 6 cycles and called it XTEA6.

IV. GAME THEORY BASICS

In this section, we describe game theory concepts used in defence models against Dos/DDoS attacks. Game theory based architectures are now used for network and computer security. In [32], a game theoretic method is presented which analyses security of computer networks. Game theory describes a multi-player decision scenario. There are various games which are used to build-up the Game theory inspired defence architecture. Some of them are given below: Perfect Information Game is a game in which each player is aware of the moves of all other players that have already taken place. Examples are: chess, tic-tac-toe. Imperfect information game is a game where at least one player is not aware of the moves of at least other player that have taken place. Complete Information Game is a game in which every player knows both the structure of the game and the objective functions of all players in the game, but not necessarily the actions. Incomplete information is game in which at least one player is unaware of the structure of the game or the objective function for at least one of the other players. Bayesian Game is a game in which information about the strategies and payoff for other players is incomplete and a player assigns a 'type' to other players at the onset of the game. Such games are labeled Bayesian games due to the use of Bayesian analysis in predicting the outcome. Static/Strategic Game is a one-shot game in which each player chooses his plan of action and all players' decisions are made simultaneously. Dynamic/Extensive Game is a game with more than one stage in each of which the players can consider their action. The sequences of the game can be either finite, or infinite. Stochastic Game is a game that involves probabilistic transitions through several states of the system. The game progresses as a sequence of states. The game begins with a start state; the players choose actions and receives a payoff that depend on the current state of the game, and then the game transitions into a new state with a probability based upon players' actions and the current state. The basic entities in the game are [33]: Player: A basic entity in a game that is tasked with making choices for actions. A player can represent a person, machine, or group of persons within a game.

Action: An action constitutes a move in the given game.

Payoff: The positive or negative reward to a player for a given action within the game.

In this paper, we consider game as the interaction between the attacker and defender as two player game where each player chooses actions which results in the best possible rewards for self.

In this paper, we also study the existence of equilibrium in these games and also show the benefit of using the game-theoretic defence mechanism with puzzles. We use game theoretical concept with Nash Equilibrium which is used to describe puzzle usage. Game theory concepts are and can be used in most of the layers but here we stress on using game theory in application layer.

V. GAME THEORY AND CLIENT-PUZZLES

We have seen increasing activity in denial of service (DoS) attacks against online services and Web applications in order to extort, disable, or impair the competition. In order to extort, disable the competition in online services and web application there are instances of denial of services (DoS) attack. An FBI affidavit [Poulsen 2004] described a case where an e-Commerce Web site, WeaKnees.com, was subject to an organized DoS attack staged by one of its competitors. These attacks were carried out using approximately 5,000 to 10,000 zombie machines at the disposal of the attacker. These attacks began on 6th of Oct 2003, with SYN floods slamming into WeaKnees.com for 12 hours straight, crippling the site, which sells digital video recorders. In response, WeaKnees.com moved to more expensive hosting at RackSpace.com. Rackspace.com could counter the SYN flooding attacks using SYN-cookies and superior bandwidth capabilities. However, the attackers adapted their attack strategy and replaced simple SYN flooding attacks with a HTTP flood, pulling large image files from WeaKnees.com. At its peak, it is believed that this onslaught kept the company offline for a full two weeks, causing a loss of several million dollars in revenue. And so sophisticated DoS attacks are not only increasingly focusing on low-level network flooding, but also on application-level attacks that flood victims with requests. In this paper we are going to use client puzzles which make use of game theory with Nash equilibrium to counter DoS attacks also in discussion we describe how the source of the attacks can be traced and managed to provide more security. If security is provided on three major layers i.e. network layer, transport layer and application layer then we can prevent DoS Attack at good multitude. This we are countering by using client puzzles, before a client can establish a connection with a server, it must first solve a puzzle. In client-puzzle

approach, the defender treats incoming requests similarly and need not differentiate between the attack and legitimate requests. Upon receiving a request, the defender produces a puzzle and sends it to the requester. If it is answered by a correct solution, the corresponding resources are then allocated. By forcing the client to solve this puzzle, we can prevent frivolous and abusive connection attempts by the client.

The common problem among most of the client puzzle schemes that have been proposed is the puzzle verification, which involves the execution of a hash function or an encryption function to verify the client's answer.

This paper uses the concept of Game theory with Nash equilibrium. Nash equilibrium is a solution concept that describes a steady state condition of the game; no player would prefer to change his strategy as that would lower his payoffs given that all other players are adhering to the prescribed strategy. This paper uses the concept of Nash equilibrium in a prescriptive way rather than only in descriptive way. Use of Nash equilibrium in prescriptive way does not lead to exhaustion of defenders resources as the difficulty level of puzzles, random number generators and other parameters are so adjusted to achieve the same.

Fig. 1 Client Puzzle Approach Here we calculate each player's payoff using game theory concepts. We calculate defender's payoff and attacker's payoff. The payoff is considered through actions QT, RA, and CA, which stand for quitting (no answer), random answer to puzzle, and correct answer to the puzzle. It is assumed that a legitimate user always solves the puzzles and returns correct answers. Assume that the defender uses an easy puzzle P1 and a difficult puzzle P2 to defend him.

- αm -> time spend by defender in providing the service.
- αpp -> time taken by defender to produce a puzzle.
- αVp ->time taken by defender to verify the solution.
- αSP1-> expected time of attacker to spend to solve P1.

Defender chooses the puzzles P1 and P2 such that

- αSP1 < αm < αSP2

On receiving a puzzle, the attacker may choose from one among the following actions:

When attacker selects CA for puzzle Pi

Pi: CA = αm + αPP + αV P − αSPi

When attacker selects RA for puzzle Pi

Pi: RA = αm + αPP + αV P

Defender's Time:

Pi: X = -αPP - αV P - αm + αSPi

We are using four Puzzle-based Defence Mechanism based on Nash equilibrium. They are Open-Loop Solutions: Open-loop is history independent solution.PDM1(Puzzle-based Defence Mechanism) is derived from the open-loop solution concept in which the defender chooses his actions regardless of what happened in the game history. The second is Closed-Loop Solutions: Closed loop is history dependent solution.PDM2 resolves PDM1problems by using the closed-loop solution concepts, but it can only defeat a single-source attack. PDM3 extends PDM2 and deals with distributed attacks. This defence is based on the assumption that the defender knows the size of the attack Coalition. PDM4, the ultimate defence mechanism is proposed in which the size of the attack coalition is assumed unknown [34].

VI. DISCUSSION

The defence mechanism proposed in this paper largely depends on the quality of the puzzles i.e. how the PDM levels are used and differentiated using puzzles at application layer. Moreover the security and maintenance of database consisting of puzzles at the defenders side is an important issue which should be considered. In the game theory approach both the attacker and defender will try to increase their pay-off and at the same tries to gain more by reducing the counterpart's pay-off. The attempt of a defender will be considered optimum if the pay-off of a defender; legitimate user is maximum and is minimum for the attacker. Some other important concepts have been discussed below: 6.1 Pushback Let us discuss some issues that may affect the way Pushback [35] could be deployed. First off, it is fairly obvious that the pushback is most effective when an attack is non-isotropic; in other words, there will be routers fairly close to the target where most of the attack traffic will be arriving from a subset of the input links. That is a fairly safe assumption; even the biggest attacks do not involve more than a few thousand compromised machines, and there are many millions of machine on the Internet. It would be particularly hard for an attacker to ensure that the attack slaves are evenly distributed with respect to the target.

Another issue to examine is what fraction of the attack traffic originates from hosts served by the same ISP as the target. The smaller the ISP, the smaller that fraction will be, and even the largest of the top-tier ISPs will have a sizeable fraction of attacks originating from the outside. While an ISP can unilaterally deploy Pushback in its routers, unless agreements with its peering ISPs are made on how to honor pushback requests (an issue fraught with security and policy issues), said ISP will have to take advantage of pushback as best as it can. Now, in general, an ISP's network can be thought of as a cloud where clients attach (on edge routers) and which connects to other ISPs at peering points (private or public). An ISP's network can thus be viewed as a single virtual router, with multiple inputs and multiple outputs. If, in addition to output

rate limiting, we were to implement input rate limiting, then the following variation of pushback could be considered: when an edge router detects an attack toward one of its attached customers, it tries to pushback determine what fractions of the attack traffic are coming through the border routers of the ISP. This could be done with some variation of ITRACE or marking by the border routers that would be caught and examined at the edge routers. Then, using (authenticated) tunnels to the border routers, the edge router would ask them to apply input rate limiting to the requested aggregate. If this is deemed undoable, input rate limiting on the border routers would still be useful in that it would effectively extend pushback by one more hop without the cooperation of the (upstream, belonging to a different ISP) router. 6.2 Password Cracking In a Password Cracking [36] attack an attacker tries to gain unauthorized access to some machine by making repeated guesses at possible usernames and passwords. Password guessing can be done remotely with many services; telnet, ftp, pop, rlogin, and imap are the most prominent services that support authentication using usernames and passwords. Dictionary attack is one such type of attack. A Dictionary attack uses a targeted technique of successively trying all the words in an exhaustive list called a dictionary which is a pre-arranged list of values. In contrast with a brute force attack, where a large proportion key space is searched systematically, a dictionary attack tries only those possibilities which are most likely to succeed typically derived from a list of words for example a dictionary or a bible etc. Dictionary attacks succeed because many people have a tendency to choose passwords which are short (7 characters or fewer), single words found in dictionaries or simple, easily-predicted variations on words, such as appending a digit.

VII. CONCLUSION

Game theory has been used in this paper to provide defence mechanisms for flooding attacks using puzzles. The interaction between the defender and attacker is considered as an infinitely repeated game of discounted payoffs. The mechanism has been divided into different levels. This paper has also described the architecture of a client puzzle protocol. The algorithm selected for the client puzzle can be implemented on almost any platform. For the scenario in which an attacker carries out a DDoS attack, we modelled the actions of the attacker as intensities or data rates employed in carrying out the attack. And to develop a trace back system that can trace a single packet so that the data of the whole message is saved and to reduced eavesdropping risks.

REFERENCES

[1] D. Moore, C. Shannon, D.J. Brown, G.M. Voelker, and S. Savage,"Inferring Internet Denial-of-Service Activity," ACM Trans.Computer Systems, vol. 24, no. 2, pp. 115-139, May 2006.

[2] A. Hussain, J. Heidemann, and C. Papadopoulos, "A Frameworkfor Classifying Denial of Service Attacks," Proc. ACMSIGCOMM '03, pp. 99-110, 2003.

[3] A.R. Sharafat and M.S.Fallah, "A Framework for the Analysisof Denial of Service Attacks," The Computer J., vol. 47, no. 2,pp. 179-192, Mar. 2004.

[4] Mehran S. Fallah, A Puzzle-Based Defence Strategy AgainstFlooding Attacks Using Game Theory, IEEE transactions on dependable and secure computing, vol. 7, no. 1, pg 5-19.

[5] C.L. Schuba, I.V. Krsul, M.G. Kuhn, E.H. Spafford, A. Sundaram,and D. Zamboni, "Analysis of a Denial of Service Attack on TCP,"Proc. 18th IEEE Symp. Security and Privacy, pp. 208-223, 1997.

[6] Smurf IP Denial-of-Service Attacks. CERT Coordination Center,Carnegie Mellon Univ., 1998.

[7] Denial-of-Service Tools. CERT Coordination Center, CarnegieMellon Univ., 1999.

[8] J. Ioannidis and S. Bellovin, "Implementing Pushback: Router- Bssed Defence against DDoS Attacks," Proc. Network andDistributed System Security Symp. (NDSS '02), pp. 6-8, 2002.

[9] D. Song and A. Perrig, "Advanced and Authenticated MarkingSchemes for IP Traceback," Proc. IEEE INFOCOM '01, pp. 878-886,2001.

[10] A. Yaar, D. Song, and A. Perrig, "SIFF: A Stateless Internet FlowFilter to Mitigate DDoS Flooding Attacks," Proc. IEEE Symp.Security and Privacy, pp. 130-146, 2004.

[11] E. Bursztein and J. Goubalt-Larrecq. A logical framework for evaluating network resilience againstfaults and attacks. Lecture Notes in Computer Science; Vol. 4846, 2007

[12] W. Sun, X. Kong, D. He, and X. You. Information security problem research based on game theory. International Symposium on Publication Electronic Commerce and Security, 2008.

[13] R. C. Merkle. "Secure Communications Over Insecure Channels," In Communications of the ACM. April, 1978.

[14] A. Juels and J. Brainard. "Client Puzzles: A cryptographic defence against connection depletion attacks," In Proceedings of NDSS

'99 (Networks and Distributed Systems Security), 1999, pages 151-165.

[15] T. Aura, P. Nikander, and J. Leiwo. "DoS-Resistant Authentication with Client Puzzles," Lecture Notes in Computer Science, vol. 2133, 2001.

[16] C. Dwork and M. Naor. "Pricing via Processing or Combating Junk Mail," In Advances in Cryptology – Crypto '92.Spring-Verlag, LNCS volume 740, pp. 129-147, August 1992.

[17] J. Xu and W. Lee. Sustaining availability of web services under distributed denial of service attacks. IEEE Transactions on Computers, pages 195–208, 2003.

[18] Q. Wu, S. Shiva, S. Roy, C. Ellis, V. Datla, and D. Dasgupta. On Modeling and Simulation of Game Theory-based Defense Mechanisms against DoS and DDoS Attacks. 43rd Annual Simulation Symposium (ANSS10), part of the 2010 Spring Simulation MultiConference, April 11-15, 2010.

[19] W. Feng, E. Kaiser, W. Feng, and A. Luu, "The Design andImplementation of Network Puzzles," Proc. 24th Ann. Joint Conf.IEEE Computer and Comm. Societies, pp. 2372-2382, 2005.

[20] X. Wang and M. Reiter, "Defending Against Denial-of-ServiceAttacks with Puzzle Auctions," Proc. IEEE Security and Privacy,pp. 78-92, 2003.

[21] ShibiaoLin ,Tzi-ckerChiueh A Survey on Solutions to Distributed Denial of Service Attacks

[22] Vicky Laurens, Abdulmotaleb El Saddik, and Amiya Nayak, Requirements for Client Puzzles to Defeat the Denial of Service and the Distributed Denial of Service Attacks

[23] B. Bencsath, I. Vajda, and L. Buttyan, "A Game Based Analysis of the Client Puzzle Approach to Defend Against DoS Attacks," Proc. 11th Int'l Conf. Software, Telecomm., and Computer Networks, pp. 763- 767, 2003.

[24] Merkle R.C., "Secure Communications Over Insecure Channels," Communications of ACM, vol. 21, no.4, pp.294-299, April 1978

[25] A. Juels and J. Brainard. "Client Puzzles: A cryptographic defense against connection depletion attacks," In Proceedings of NDSS '99 (Networks and Distributed Systems Security), 1999, pages 151-165.

[26] T. Aura, P. Nikander, and J. Leiwo. "DoS-Resistant Authentication with Client Puzzles," Lecture Notes in Computer Science, vol. 2133, 2001.

[27] D. Wheeler and R. Needham. "TEA, a Tiny Encryption Algorithm," Unpublished Manuscript. Available at:http://www.ftp.cl.cam.ac.uk/ftp/papers/djw-rmn/djw-rmn-tea.html. November, 1994.

[28] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. "Handbook of Applied Cryptography."CRC Press, 1996.

[29] W. Sun, X. Kong, D. He, and X. You. Information security problem research based on game theory. International Symposium on Publication Electronic Commerce and Security, 2008.

[30] R. C. Merkle. "Secure Communications Over Insecure Channels," In Communications of the ACM. April, 1978.

[31] Timothy J. McNevin, Jung-Min Park, and Randolph Marchany. "pTCP: A Client Puzzle Protocol For Defending Against Resource Exhaustion Denial of Service Attacks"

[32] K. Lye and J.M. Wing. Game strategies in network security.In Proceedings of the 15th IEEE Computer Security Foundations Workshop, 2002.

[33] S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya, and Q. Wu. A survey of game theory as applied to network security. The 43rd Hawaii International Conference on System Sciences, 2010.

[34] Mehran S. Fallah, A Puzzle-Based Defense Strategy Against Flooding Attacks Using Game Theory, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 7, NO. 1, JANUARY-MARCH 2010

[35] John Ioannidis, Steven M. Bellovin, Implementing Pushback: Router-Based Defense Against DDoS Attacks

[36] Tanmay Sanjay Khirwadkar, Defense Against Network Attacks Using Game Theory, University Of Illinois At Urbana-Champaign, 2011

[37] Timothy J. McNevin, Jung-Min Park, and Randolph Marchany, pTCP: A Client Puzzle Protocol For Defending Against Resource Exhaustion Denial of Service Attacks