Game Theory based defence mechanism against Flooding attack using Puzzle

Raju Neyyan,
B.E Computer Engineering,
University of Pune

Ancy Paul,
B.E Computer Engineering,
University of Pune

Mayank Deshwal
B.E Computer Engineering,
University of Pune

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 defence 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 Network and Security.

Keywords

DOS attack, Game theory, Puzzle based Defence.

1. INTRODUCTION

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 defence 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],[19])

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 [4] whereas Smurf attack uses bandwidth consumption to disable victim system's network resources [5] and TFN2K attacks are launched using spoofed IP addresses, making detecting the sources of the attacks more difficult[6]. 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[19]. 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 [19]. 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,

Mechanisms such as pushback [9],traceback [10], orfiltering [11] 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 [19].

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 modelled 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 [19].

2. RELATED WORK:

Burszteinetal [20] 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 [21] analyzed information security problem in the mobile electronic commerce chain. Theyclaimed 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 [22]. However, Merkle used puzzles for key agreement, rather than access control. Client puzzles have been applied to TCP SYN flooding by Juels and Brainard [23]. Aura, Nikander, and Leiwo [24] apply client puzzles to authentication protocols in general [25]. 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. Our goal is much more limited than theirs; we seek only to prevent a denial of service attack on network.

In general, reactive mechanisms suffer from the scalability Problem and difficulty of attack traffic identification. This is not the case in the client-puzzle approach, where 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. As solving a puzzle is resource consuming, the attacker who intends to use up the defender's resources by his repeated requests is deterred from perpetrating the attack.
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 [13] and [14] 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 [19]

3. GAME THEORY

In this section, Game models are presented for DoS/DDoS attacks and their possible countermeasures. We study the existence of equilibrium in these games and the benefit of using the game-theoretic defence mechanisms. We use Network model which describes game strategy. In[19] a model of networked system, which gives description of game, reflecting possible interactions between an attacker and defender in a scenario of flooding attack-defence. Network model is also deployed in specifications of game.

This section identifies the premise of game theory to aid the understanding of the games. Game theory describes multi-person decision scenarios as games where each player chooses actions which result in the best possible rewards for self, while anticipating the rational actions from other players. A player is the basic entity of a game that makes decisions and then performs actions. A game is a precise description of the strategic interaction that includes the constraints of, and payoffs for, actions that the players can take, but says nothing about what actions they actually take. A solution concept is a systematic description of how the game will be played by employing the best possible strategies and what the outcomesmight be. If the plan specifies a probability distribution for all possible actions in a situation then the strategy is referred to as a mixed strategy.

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 solution concept only specifies the steady state but does not specify how that steady state is reached in the game. The Nash equilibrium is the most famous equilibrium. This information will be used to define gamesthat have relevant features for representing network security problems.Payoff is the positive or negative reward to a player for a given action within the game. This means when choosing a plan of action each player is not informed of the plan of action chosen by any other player.

A static game is a one-shot game in which each player chooses his plan of actions and all players' decisions are made simultaneously.Here we use concept of Dynamic/Extensive Game.It 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.

The underlying assumptions of game theory hold in a network that is the reason for using game theory in designing flooding prevention mechanisms. The main assumption is that there are rational players, i.e., their planned actions at any situation and at that time must be optimal .This assumption holds in a network, where players are the active entities. Therefore, a defence mechanism that implements thedefender's strategy obtained from a game-theoretic approach assures the best possible sequence of actions performed against a rational attacker. [19][26]

4. THE GAME OF THE CLIENT-PUZZLE APPROACH

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:

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 accordingto the estimated strength (computational resources) of the attacker.If the server manages to set an appropriate complexitylevel, then solving the puzzle slows down the DoS attacker whowill eventually abandon his activity

A flooding attack-defence scenario is modelled as a two-player infinitely repeated game. Therefore, in the stage-game played at any period t, the defender and the attacker, i.e., the active entities 1, 2 ∈ E, choose from their action spaces γ(1) and γ(2) and cause the game to arrive at period t+1. In the client-puzzle approach, the set of possible actions for the defender is Γ1 = $\{P_1, P_1, \dots P_n\}$, and the one for the attacker is Γ2 = {QT, RA, CA}. The action Pi, 1 ≤ i ≤ n, stands for issuing a puzzle of difficulty level i. It is assumed that the puzzle of level i is less difficult than the one of level j if i < j. The actions QT, RA, and CA stand for quitting the protocol (no answer), answering the puzzle randomly, and answering the puzzle correctly. It is also assumed that a legitimate user always solves the puzzles and returns correct answers. At a period, the attacker knows the action chosen by the defender at that period. Thus, the stage-game is indeed an extensive-form game. In order to convert this game into its equivalent strategic-form game, it is sufficient to consider the action spaces as γ(1)=Γ1 and γ(2) =Γ_2^n, where Γ_2^n is the Cartesian product of Γ1 together itself n times. For example, if the defender can choose between P1 and P2, one of possible actions for the attacker is (CA, QT), which means "chooseCA when the defender chooses P1, and QT when he chooses P2." It is worth noting that a player's strategy for the repeated game is obtained from the functions se and σe, where a player chooses his action according to the history of events he knows. The model of the stage game is completed by the players' payoff functions. The underlying notion of a puzzle-based defence is that the workload of the attacker should be higher than of the defender [12]. In addition, the defender should care about the level of quality of service he provides forlegitimate users. Therefore, an action profile is more preferable for the defender if it results in more cost to the attacker, less cost to the defender, and less cost to legitimate users. Similarly, an action profile is more desirable for the attacker if it causesmore cost to the defenderand less cost to the attacker.Hence, the players' stage-game payoffs are obtained from

g1(a) = -Ψ(1,a) + Ψ(2,a) + ηΨ(u,a), and

g2(a) = -Ψ(2,a) + Ψ(1,a) (1)

where Ψ is the cost function , Ψ (u, a) is the cost to a legitimate user when the action profile a is chosen, and η ϵ [0, 1] is the level of quality of service the defender is willing to provide for legitimate users. As will be seen, a low quality of service is inevitable when the attacker enjoys high capabilities.In the client-puzzle approach, the defender engages two types of resources, one for producing puzzles and verifying solutions, denoted by rp, and the other for providing the requested service. The latter, denoted by rm, is the main resource the defender wishes to protect against flooding attacks. Therefore, δ(1)=$\{rp, rm\}$. Similarly, for the attacker, δ(2) = $\{rs\}$, where rs is the resource he uses to solve a puzzle. Finally, for a legitimate user, the active entity u, δ(u) = {rn}.in which rn is the resource he engages in solving a puzzle.

The repeated game between thedefender and the attacker is of discounted payoffs. Therefore, a discount factor μ ∈ (0, 1)is used as a weighting factor in theweighted sum of payoffs. More precisely, the players'payoff for the repeated game, when the mixed strategyprofile σ=(σ1;σ2) is played, is defined by

$$u_i(\sigma) = \sum_{j=-}^{\infty} \mu^i g_i\left(\sigma^j(h^j)\right) = \sum_{j=0}^{\infty} \mu^i g_i\left(\sigma_1^j(h_1^j); \sigma_2^j(h_2^j)\right) (2)$$

It is also more convenient to transform the repeated gamepayoffs to be on the same scale as the stage-game payoffs.This is done by multiplying the discounted payoff in (2) by1 - μ. Thus, the players' average discounted payoff for therepeated game is as follows as per [19]

$$\bar{u}_i(\sigma) = (16 - \mu)u_i(\sigma) \qquad (3)$$

5. DEFENCE STRATEGIES

In this section we employ the solution concepts of infinitely repeated games with discounting to design the optimum puzzle-based defence strategies against flooding attacks. In general, the strategies prescribed by such solutions are divided into two categories: open loop (history independent) and closed loop (history dependent). The defence strategies proposed in this section are based on the concept of Nash equilibrium. For the ease of reference, this concept is repeated here. Let $\sigma_1^*; \sigma_2^*$ mixed-strategy Nash equilibrium for the two-player infinitely repeated game developed in Section 4. Then, $u_1(\sigma_1^*; \sigma_2^*) \geq u_1(\sigma_1; \sigma_2^*)$ for any $\sigma_1 \in \sum_1$, and function in (2). This means that any unilateral deviation from the strategy profile stated by the Nash equilibrium has no profit for its deviator.

The Nash equilibrium is often used in a descriptive way, where it describes the players' strategies in a game. In this sense, it makes predictions about the behaviors of rational players. In this section, on the contrary, the concept of Nash equilibrium is employed in a prescriptive way in which the defender picks out a specific Nash equilibrium and takes his part in that profile. The attacker may know this, but the best thing for him to do is to be in conformity with the selected equilibrium. If he chooses another strategy, he gains less profit (the attacker's payoff function, defined in (2) and (3), reflects the attacker's profit from a flooding attack). In the defence mechanisms proposed in this section, the defender adopts the Nash equilibrium prescription that brings him the maximum possible repeated game payoff while preventing the attack. In this way, the defence mechanism would be optimal.

5.1 Open-Loop Solutions

In the repeated-game of the client-puzzle approach, in an open-loop strategy, the action profiles adopted at previous periods are not involved in a player's decision at the current period.

5.1.1 PDM1—the puzzle-based defence mechanism

It against flooding attacks derived from the open-loop solution concept of discounted infinitely repeated games.PDM1 treats a distributed attack as a single-source attack, where the attackers re modeled as a single attacker with the capabilities of the corresponding attack coalition. The same approach can be adopted for closed-loop solutions, but some further issues should be considered there. In a distributed attack, the requests come from different machines, and it is no longer reasonable to assume that the defender receives only a small number of requests before receiving the correct or random answer to an issued puzzle. Indeed, a large number of requests are produced by the attack coalition, whereas a small proportion of them are of a single machine. Therefore, in the time a machine is involved in computing the answer, the defender may receive a large number of requests from the other machines in the coalition.

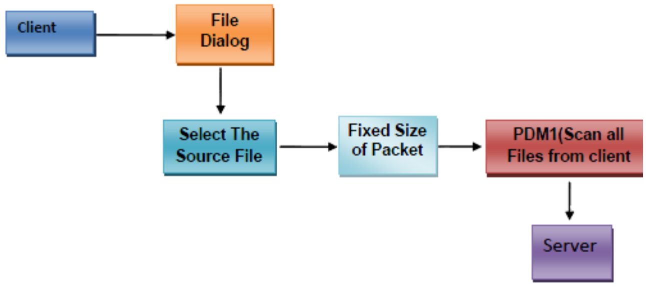

Fig 5.1.1 PDM1

In an open-loop strategy, the action profiles adopted at previous periods are not involved in a player's decision at the current period. More formally, in the repeated-game of the client-puzzle approach, $\sigma_i^t : H_i^t \rightarrow \Delta(\gamma(i))$ is an open-loop strategy for player i if

$$\forall t \in \mathbb{Z}^{\geq 0}\ \forall h_i^t, \tilde{h}_i^t \in H_i^t \left[\sigma_i^t(h_i^t) = \sigma_i^t\left(\tilde{h}_i^t\right)\right],$$

where i = 1, 2, $\gamma(1) = \Gamma_1$, and $\gamma(2) = \Gamma_2^n$.

one of the open-loop solutions to an infinitely repeated game is to play any one of the stage-game Nash equilibria at a period regardless of what actually happened in the corresponding history. In other words, let $(\sigma_1;\sigma_2)$ be an open-loop strategy profile for the infinitely repeated game such that $\sigma_i^t(h_i^t) = \alpha_1^t$ and $\sigma_2^t(h_2^t) = \alpha_2^t$ for all histories $h_1^t \in H_1^t$and$h_2^t \in H_2^t$.

If $(\alpha_1^t;\alpha_2^t)$ is a stage-game Nash equilibrium for any t, then $(\sigma_1;\sigma_2)$ is a sub game-perfect equilibrium for the repeated game [15].

In a flooding attack-defence scenario, the defender may not perfectly know the actions taken by the attacker at previous periods. Thus, adopting an open-loop strategy, as stated above, may be the simplest way he can attain equilibrium. The following theorem identifies the stage-game Nash equilibria for the game of the client-puzzle approach.

Assume that $(x;y)$ represents the class of stage-game pure strategies in which the defender chooses $x \in \Gamma_1 = \{P_1, P_2\}$, and the attacker responds to it by

$y \in \Gamma_2 = \{QT, RA, CA\}$. For example, (P_1, CA) represents the class of strategies $(P_1;b)$, where $b \in \Gamma_2^2$ and $b(1) = CA$. Then, in a strategy profile of the form $\xi_{11} \circ (P_1;QT) \oplus \xi_{12} \circ (P_1;RA) \oplus \xi_{13} \circ (P_1;CA) \oplus$

$$\xi_{21} \circ (P_2;QT) \oplus \xi_{22} \circ (P_2;RA) \oplus \xi_{23} \circ \qquad (4)$$

$(P_i;QT), (P_i;RA)$, and $(P_i;CA), i = 1,2$, are chosen with probabilities ξ_{i1}, ξ_{i2}, and ξ_{i3}, respectively. In the repeated game of the client-puzzle approach, the on-the-equilibrium path can be considered as an infinite sequence of strategy profiles of the form (4). Then, the average discounted on-the equilibrium-path strategy profile is defined by $(1-\mu)(\oplus_{j=0}^{\infty} \mu^j \circ \sigma^j(h^j))$, where $\sigma^j(h^j)$ is the on-the-equilibrium-path strategy profile at period j in the form of (4).

As stated in Section 4, the reference distance of the resources r_p, r_s, and r_n is considered as the maximum time T the allocated amount of the main resource can be kept in use by a request. Moreover, the reference distance of the main resource is the number of requests that can be served by that resource in a period of length T. By adopting such reference distances, the attacker can solve $1/\alpha_{SP}$ puzzles in a time of length T. The defender can either produce $1/\alpha_{PP}$ puzzles or verify $1/\alpha_{VP}$ puzzles in this time, but he cannot do both of them in a single period, because the same resource is engaged in those actions. Finally, the defender can process $1/\alpha_m$ requests using his main resource in such a period. By this modeling, a fair solution can be defined as follows in which N is the number of requests the attacker can produce in a time of length T. Note that only two puzzles are considered, P_1as a simple puzzle and P_2 as a difficult one.

Definition: A strategy profile is a fair solution to the client puzzle approach if the conditions in one of the following cases hold for its average discounted on-the-equilibrium-path strategy profile in (9).

$$Case\ 1.\ \left(N\sum_{i=1}^{2}\xi_{i3}\alpha_{SP_i} \leq 1\right):$$

$$N\sum_{i=1}^{2}\xi_{i3}\alpha_m \leq 1,\ and$$

$$N\left(\sum_{i=1}^{2}\xi_{i1}\alpha_{PP} + \sum_{i=1}^{2}(\xi_{i2}+\xi_{i3})(\alpha_{PP}+\alpha_{VP})\right) \leq 1.$$

$$Case\ 2.\ \left(N\sum_{i=1}^{2}\xi_{i3}\alpha_{SP_i} > 1\right):$$

$$\sum_{i=1}^{2}\left(\xi_{i3}\Big/\sum_{l=1}^{2}\xi_{l3}\right)(\alpha_m/\alpha_{SP_i}) \leq 1,\ and$$

$$N\left(\sum_{i=1}^{2}(\xi_{i1}+\xi_{i3})\alpha_{PP} + \sum_{i=1}^{2}\xi_{i2}(\alpha_{PP}+\alpha_{VP})\right)$$

$$+\sum_{i=1}^{2}\left(\xi_{i3}\Big/\left(\sum_{l=1}^{2}\xi_{l3}\right)\right)(\alpha_{VP}/\alpha_{SP_i}) \leq 1.$$

The attacker can solve all the puzzles that should be correctly answered. The number of such puzzles is $N(\xi_{13} + \xi_{23})$.

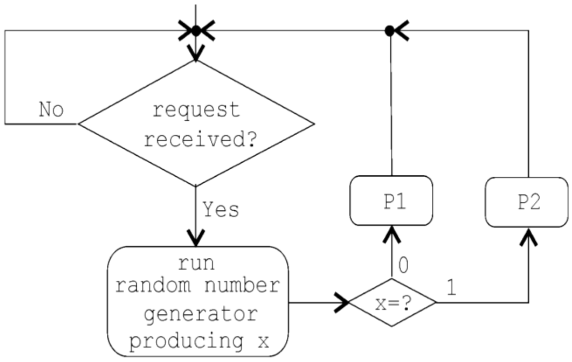

Fig. 2. PDM1—the puzzle-based defence mechanism against flooding attacks derived from the open-loop solution concept of discounted infinitely repeated games.

In order to prevent the main resource from being exhausted by the attacker's requests, this number should be less than or equal to the number of requests that can be served by the main resource in a time of length T, i.e., less than $1/\alpha_m$.

This is the first condition in Case 1. Similarly, the second condition in Case 1 protects the resource r_p used in producing puzzles and verifying solutions. In Case 2, the attacker cannot solve all the puzzles that should be answered by correct solutions. In this case, using the entire resource r_s, he can solve $\xi_{13}/(\xi_{13} + \xi_{23})(1/\alpha_{SA})$ simple puzzles and $\xi_{23}/(\xi_{13} + \xi_{23})(1/\alpha_{SP2})$ difficult puzzles on the average. Again, the first and second conditions are to preserve the main resource and the resource used in producing puzzles and verifying solutions from being exhausted by the attacker's requests.

5.2 Closed-Loop Solutions

The closed loop solutions are history dependent.

5.2.1 IPDM1—the puzzle-based defence mechanism

PDM1 is derived from the open-loop solution concept in which the defender chooses his actions regardless of what happened in the game history. This mechanism is applicable in defeating the single-source and distributed attacks, but it cannot support the higher payoffs being feasible in the game. PDM2 resolves this by using the closed-loop solution concepts, but it can only defeat a single-source attack.

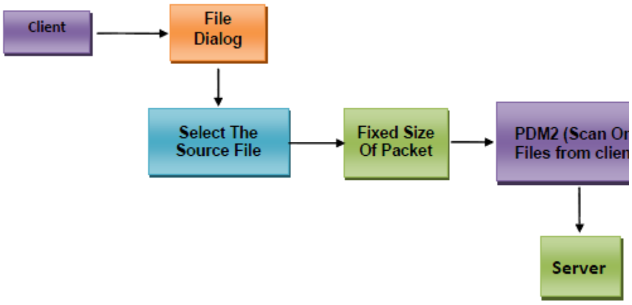

Fig 5.2.1 IPDM1- The puzzle-based defence mechanism

In a fair open-loop solution, the defender's maximum average payoff is$-\alpha_{PP} - \alpha_{VP} - \eta\alpha_{SP2}$. However, there are many payoff vectors in the convex hull with greater payoffs for the defender. Thus, here, a natural question arises: Is there a better fair solution to the game, which results in a greater payoff to the defender? As proven in [16], in the games of perfect information, there is a large subset of the convex hull whose payoff vectors can be supported by perfect Nash equilibria provided that suitable closed-loop strategies are adopted. This subset is denoted by V*, and its elements are called strictly individually rational payoffs (SIRP). In the game of the client-puzzle approach $V* = \{(v_1, v_2) \in \mathrm{conv}(g(\Gamma)) | v_1 > v_1^*, v_2 > v_2^*\}$

Where $\Gamma = \Gamma_1 \times \Gamma_2^n$, and (v_1^*, v_2^*) is the minmax point defined by

$$v_1^* = \min_{\alpha 2 \in \Delta(\Gamma_2^n)} \left(\max_{\alpha 1 \in \Gamma_1 g_1} (\alpha_1; \alpha_2)\right),$$

$$v_2^* = \min_{\alpha 1 \in \Delta(\Gamma_1)} \left(\max_{\alpha 1 \in \Gamma_2^n g_2} (\alpha_1; \alpha_2)\right)$$

In which $\Delta(X)$ is the set of all probability distributions overX.

Furthermore, the mixed strategies resulting in $v_1^*$and, v_2^* are denoted by $M^1 = (M_1^1; M_2^1)$ and $M^2 = (M_1^2; M_2^2)$, respectively. The strategy M_1^2 is the player 1's minmax strategy against the player 2. Similarly, M_2^1 is the player 2's minmax strategy against the player 1.

Fig. 3 shows the convex hull of payoff vectors for the game of the client-puzzle approach when $\alpha_m = 0.2$, $\alpha_{SP1} = 0.15$, $\alpha_{SP2} = 0.23$, $\alpha_{PP} = 0.01$, $\alpha_{VP} = 0.02$, and $\eta = 0.5$. As seen in Fig. 3, the defender's maximum average payoff in PDM1, i.e., $-\alpha_{PP} - \alpha_{VP} - \eta\alpha_{SP2}$, is -0.145, though many payoffs greater than -0.145 can be supported if the game is of perfect information and suitable closed-loop strategies are adopted.The following theorem characterizes the set of payoff vectors that can be supported by perfect Nash equilibrium in an infinitely repeated game of observable actions and complete information where the payoffs are discounted.

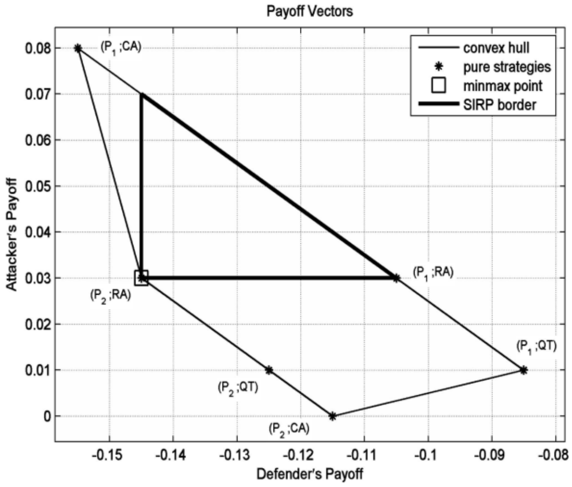

Fig. 3. The convex hull of payoff vectors and SIRP in the game of the client-puzzle approach when α_m = 0.2, α_{SP1} = 0.15, α_{SP2} = 0.23, α_{PP} = 0.01, α_{VP} = 0.02, and $\eta = 0.5$

This reflects those attack-defence circumstances in which the player involved in the defence mechanism knows his opponent's payoff function as well as the actions chosen by his opponent at previous periods. It is worth noting that the puzzles can be designed in such a way that the amounts of resources a machine uses to solve a puzzle are independent of the machine's processing power.[18].Therefore, except for flooding attacks from an unknown number of sources, it is reasonable to assume that the defender knows the attacker's payoff function[19].

6. 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. The present problems of optimality and effectiveness have been solved by this mechanism. It also provides scalability and can be deployed in various environments with requirement of different security levels. Hence by use of game theory we can provide ultimate defence mechanism for flooding attacks.

7. 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] 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.

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

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

[7] Denial-of-Service Attack via Ping. CERT Coordination Center,Carnegie Mellon Univ., 1996.

[8] IP Denial-of-Service Attacks. CERT Coordination Center, CarnegieMellon Univ., 1997.

[9] 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.

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

[11] 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.

[12] J. Leiwo, P. Nikander, and T. Aura, "Towards Network Denial ofService Resistant Protocols," Proc. 15th Int'l Information SecurityConf., pp. 301-310, 2000.

[13] 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.

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

[15] H. Gintis, Game Theory Evolving: A Problem-Centered Introduction toModeling and Strategic Behavior. Princeton Univ. Press, pp. 129-130,2000.

[16] D. Fudenberg and E. Maskin, "The Folk Theorem for RepeatedGames with Discounting and Incomplete Information," Econometrica,vol. 54, no. 3, pp. 533-554, May 1986.

[17] M. Abadi, M. Burrows, M. Manasse, and T. Wobber, "ModeratelyHard, Memory-Bound Functions," Proc. Network and DistributedSystem Security Symp. (NDSS '03), pp. 25-39, 2003.

[18] Boldizs´ar Bencs´ath Istv´an Vajda Levente Butty´an A Game Based Analysis of the Client Puzzle Approach to Defend Against DoS Attacks

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

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

[21] 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.

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

[23] 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.

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

[25] 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.

[26] Quishi Wu, Sajama Shiva, Tankards Roy, Charles Ellis, ViveData, On Modeling and Simulation of Game Theory-based Defence Mechanisms against DoS and DDoS Attack.