

Deception in Honeynets: A Game-Theoretic Analysis

Nandan Garg and Daniel Grosu

Abstract— Recently, honeynets became one of the main tools for understanding the characteristics of malicious attacks and the behavior of the attackers. However the attackers may identify the honeypots and avoid attacking them. Thus the honeynet administrators must be able to deceive the attackers and induce them to attack the honeypots. In this paper we propose a game theoretic framework for modeling deception in honeynets. The framework is based on extensive games of imperfect information. We study the equilibrium solutions of these games and show how they are used to determine the strategies of the attacker and the honeynet system.

Index Terms - deception, game theory, honeynets, security

I. INTRODUCTION

A honeypot as defined by the Honeynet Project [1] is “an information system resource whose value lies in unauthorized access or illicit use of that resource.” In this paper we consider a honeypot to be a host that is left unprotected and available to the attackers. By extension we consider that a honeynet is composed of a set of interconnected honeypots. These systems are carefully monitored in order to learn new vulnerabilities, tools and techniques used to gain access. Honeynets are useful for analyzing large-scale attacks such as worms, viruses and botnets and have little value for analyzing specialized attacks targeted to a particular host.

Recently with the increase in the frequency of Internet attacks several research groups and organizations developed and deployed honeynet systems. The idea behind honeynets is the use of deception, a classical counter intelligence technique which was and is still used extensively by the intelligence organizations. The computer security community began considering this idea only recently, [2] being the first known reference mentioning it.

Depending on the level of interactions between the attacker and the monitored system, honeynets can be classified into two categories: low-interaction honeynets and high-interaction honeynets. *Low-interaction honeynets* simulate a minimal set of system functionalities such as the network stack. The main advantage of low-interaction

honeynets is their scalability, allowing the monitoring of hundreds of thousand of IP addresses. The major disadvantage is that they cannot monitor in detail the attacker’s actions once the host is compromised. *High-interaction honeynets* simulate all the functionalities of a real system thus allowing the gathering of much richer information on the attacker’s actions. The disadvantage of using high-interaction honeynets is their high cost and difficult management. When the honeypot is a real machine with its own IP address we call it a *physical honeypot*. If the honeypot is a simulated host then we have a *virtual honeypot*. The physical honeypots are very expensive to install and maintain and are not scalable, while the virtual honeypots are scalable and inexpensive.

Attackers have certain ways to probe the network attempting to identify the honeypots [3]. Many current honeynet deployments do not use specific strategies to prevent the mapping of honeypots by the attackers. When a host is probed, it gives some response. This response is valuable to the attacker as she gains information about the honeynet. One way a defender can avoid the mapping of honeypots is to cleverly manipulate the response to the probe and deceive the attacker. Our goal in this paper is to provide a game theoretical framework for modeling and characterizing deception in honeynets.

A. Related Work

Low-interaction honeypots usually monitor a large range of IP addresses and they have been very successful in identifying large scale worm outbreaks [4], understanding distributed denial-of-service attacks [5] and creating intrusion detection signatures [6]. “Network telescopes” [7], [8], [9] are examples of successful deployments that passively monitor inbound packets without completing the TCP handshake. iSink [9] is a measurement system deployed to monitor background radiation which employs active responders who emulate the real network behavior an attacker expects. Honeyd [10] is a low-interaction system that simulates the networking stack of different computer systems. In order to simulate real networks it creates virtual networks consisting of arbitrarily routing topologies.

The simplest way to implement high-interaction honeypots is to have individual servers for each monitored IP address. Since this approach is very expensive recently several researchers proposed the use of virtual machine

Manuscript received January 21, 2007

Nandan Garg is with the Dept. of Computer Science, Wayne State University, Detroit, MI 48202 USA (email: nandang@wayne.edu)

Daniel Grosu is with the Dept. of Computer Science, Wayne State University, Detroit, MI 48202 USA (phone: 313-577-5171; fax: 313-577-6868; email: dgrosu@cs.wayne.edu)

environments to implement multiple honeypots on a single server [11], [12]. These systems are easy to manage and have a reduced deployment cost while they provide all functionalities of a real system. Vrabie *et al.* [11] presented Potemkin, a prototype honeynet that uses the virtual machine approach supporting over 64,000 honeypots using only few physical servers.

Deception techniques for defending information systems have been investigated by several researchers. Cohen [13] provides a comprehensive discussion of deception issues for information protection. Among the conclusions of the above study is that deception is a valuable tool for intrusion detection and it has a positive effect on the defense and a negative effect on the attackers. The author stressed the need for rigorous mathematical analysis of deception in information protection. Cohen and Koike [14] shows how deception can be used to control the path of a computer system attack. Deception Toolkit (DTK) [13] is a collection of scripts that allows the emulation of different known vulnerabilities in order to deceive the attackers. Rowe *et al.* [15] suggested the use of fake honeypots to scare away possible attackers. They also discuss the escalation of honeypot anti-honeypot techniques. Rowe [16] presented several tools for evaluating honeypot deceptions.

B. Contributions

The contribution of this paper consists of a game theoretic framework for modeling and analyzing deception in honeynets. We propose and analyze a *strategic game model* for the honeynet system. We determine the equilibrium solutions for this model and analyze the corresponding strategies of the attacker and the defender. We also propose the use of *extensive games of imperfect information* to analyze deception strategies in honeynets. We show how the proposed game can be used to model the attacker-honeynet interactions. We analyze the mixed strategy equilibrium solutions of the proposed game and characterize the deception moves of the defender and the corresponding attacker's actions. To our knowledge, this is the first work proposing a model of the attacker-honeynet interactions based on an extensive game of imperfect information.

C. Organization

The rest of the paper is structured as follows. In Sect. II we describe a prototype strategic form game that models the attacker and the honeynet system. In Sect. III we propose the honeynet deception games in extensive form and discuss their properties. In Sect. IV we draw conclusions and present future directions.

II. A PROTOTYPE HONEYNET GAME IN STRATEGIC FORM

We consider the attacker and the honeynet system as players in a strategic noncooperative game. We denote

the *attacker* by A and the *defender*, which is the honeynet system, by D . The honeynet is assumed to be composed of k honeypot hosts out of n possible hosts within a block of IP addresses. The attacker must choose to attack one of these hosts. The goal of the attacker is to attack a host that is not a honeypot while the goal of the defender is to have a honeypot attacked by the attacker. In this initial model we assume that the attacker and the defender make their decisions independently and without any knowledge of the opponent player's moves. We call the game that models this situation a (n, k) -*honeynet game*, denoted as $HG(n, k)$.

Definition 1: A (n, k) -*honeynet game* consists of:

- (i) the set of players $\mathcal{N} = \{A, D\}$.
- (ii) the set of actions for each player:
 - (a) $A_D = \{(x_1, x_2, \dots, x_n) \mid k \text{ vector components are 1 and } n - k \text{ components are 0}\}$. If $x_i = 1$ then D places a honeypot at position (address) i . If $x_i = 0$ then D places a regular host at position i .
 - (b) $A_A = \{j \mid j = 1, \dots, n\}$. A chooses to attack host j .
- (iii) the payoff functions of each player:

$$u_A((x_1, x_2, \dots, x_n), j) = \begin{cases} -c_1 & \text{if } x_j = 1, \\ c_2 & \text{if } x_j = 0 \end{cases} \quad (1)$$

$$u_D((x_1, x_2, \dots, x_n), j) = \begin{cases} c_1 & \text{if } x_j = 1, \\ -c_2 & \text{if } x_j = 0 \end{cases} \quad (2)$$

where $c_i > 0$, for $i \in \{1, 2\}$.

The parameter c_1 represents the payoff gained by the defender and also the loss incurred by the attacker if the attacker attacks a honeypot. The parameter c_2 represents the payoff gained by the attacker and also the loss incurred by the defender if the attacker attacks a regular host.

As an example we consider a $(2, 1)$ -honeynet game, $HG(2, 1)$. The payoff matrix for this game is given below. In this matrix the row player is A and the column player is D . The first component of the payoff pairs given in the matrix represents the payoff to A while the second component represents the payoff to D .

	(1, 0)	(0, 1)
1	$-c_1, c_1$	$c_2, -c_2$
2	$c_2, -c_2$	$-c_1, c_1$

One solution concept for strategic games is the Nash equilibrium [17], [18]. An *action profile* is a tuple with an entry containing the action for each player. An action profile is a *Nash equilibrium* when no player can do better by altering her action.

Definition 2 (Nash equilibrium [19]) An action profile $a^* = (a_1^*, a_2^*, \dots, a_n^*)$ is a *Nash equilibrium*, if

$$u_i(a^*) \geq u_i(a_i, a_{-i}^*) \text{ for every action } a_i \text{ of player } i, \quad (3)$$

where a_{-i}^* is an action profile without player i , i.e., $a_{-i}^* = (a_1^*, \dots, a_{i-1}^*, a_{i+1}^*, \dots, a_n^*)$.

Any change of action by player i will reduce her payoff and consequently she will choose not to alter her action.

As can be seen from the payoff matrix above, $HG(2, 1)$ does not have a Nash equilibrium in pure strategies. Thus we need to consider mixed strategies since for a finite strategic game we will always be able to find a Nash equilibrium in mixed strategies [19]. A mixed strategy of a player is a probability distribution over the player's pure strategies. In the following we formally define the concept of Nash equilibrium in mixed strategies.

Definition 3 (Mixed strategy Nash equilibrium [19]) The mixed strategy profile α^* is a *mixed strategy Nash equilibrium* if $u_i(\alpha^*) \geq u_i(\alpha_i, \alpha_{-i}^*)$, for every mixed strategy α_i of player i , where α_{-i} is the action profile without player i , i.e., $\alpha_{-i}^* = (\alpha_1^*, \dots, \alpha_{i-1}^*, \alpha_{i+1}^*, \dots, \alpha_n^*)$.

A pure strategy is a mixed strategy with an action that has probability 1.

In the case of $HG(2, 1)$ we denote by a_i , $i \in \{1, 2\}$ the pure strategy of A in which A attacks host i , and by d_i , $i \in \{1, 2\}$ the pure strategy of D in which D places a honeypot at host i (i.e., $d_1 = (1, 0)$ and $d_2 = (0, 1)$). We denote by $\alpha_A(a_i)$ the probability assigned by the mixed strategy α_A of the attacker to its strategy a_i ; and by $\alpha_D(d_i)$ the probability assigned by the mixed strategy α_D of the defender to its strategy d_i . It can be shown that the Nash equilibrium in mixed strategies is given by $(((\alpha_A(a_1), \alpha_A(a_2)), (\alpha_D(d_1), \alpha_D(d_2)))) = ((\frac{1}{2}, \frac{1}{2}), (\frac{1}{2}, \frac{1}{2}))$. The Nash equilibrium in mixed strategies of $HG(2, 1)$ corresponds to the intuitive solution in which the attacker is attacking host 1 and host 2 with equal probability and the defender is placing the honeypot at host 1 or host 2 with equal probability.

The equilibrium analysis of the general game $HG(n, k)$ is as follows. A has n pure strategies $\{a_1, \dots, a_n\}$, where strategy a_i means A attacks host i . D has $m = \binom{n}{k}$ ways to place k honeypots at n hosts, where $\{x_1, x_2, \dots, x_n\}$ is the vector denoting the placement of honeypots. We denote the pure strategies of D by $d_j, j = \{1, \dots, m\}$. Using the payoff function in Definition 1 we see that there is no Nash equilibrium in pure strategies. Intuitively when A attacks node j , D would obtain a higher payoff if she chooses a strategy d_i , in which $x_j = 1$. However in that case A is better off attacking some other node $l \neq j$. Thus there is no pure strategy equilibrium for $HG(n, k)$. To calculate the mixed strategy equilibria of $HG(n, k)$, we consider the probabilities assigned to each pure strategy by the players (as mentioned above in $HG(2, 1)$ example). Let c_{ij}^D denote the payoff of D when A selects strategy a_j and D selects strategy d_i . The expected payoff E_i^D received by D when playing d_i is $E_i^D = \sum_{j=1}^n \alpha_A(a_j) c_{ij}^D$.

The total expected payoff of D will be:

$$E^D = \sum_{i=1}^m \alpha_D(d_i) E_i^D \quad (4)$$

It can be seen that to maximize E^D , D has to play all $\alpha_D(d_i)$ equally, because of the symmetrical payoffs. This means that at the equilibrium in mixed strategies the probability $\alpha_D(d_i)$ with which D selects strategy d_i is equal to $1/m$, i.e., $\alpha_D(d_i) = 1/m, \forall i = 1, \dots, m$. Intuitively, we can see that if D can increase its payoff by increasing the probability of playing strategy d_i and decreasing the probability of playing strategies d'_i then, it will be able to maximize its payoff by setting $d'_i = 0$ and $d_i = 1$, which will be a pure strategy. However, we have already seen that there is no pure strategy equilibrium for this game. Thus the mixed strategy Nash equilibrium for this game is when both players, the attacker and the defender, play all their strategies with equal probabilities.

Since this is a one-shot game the attacker and the defender choose their strategies independently and so the defender cannot influence the strategies of the attacker. The games proposed in this section do not consider deception strategies, but they are the prototype games that will be extended in the next section into games that take into account deception moves.

III. HONEYNET DECEPTION GAMES

We extend the games studied in the previous section to allow deception moves by the defender (honeynet). Examples of deception moves are: make it evident to the attacker that the system is a honeypot but in fact it is a regular host (we call these "fake honeypots"), or hide the honeypot such that the attacker believes it is a regular host. These can be achieved by different techniques some of them being described in [15], [16]. In this section we propose a game theoretic model considering two players, the Attacker and the Defender (honeynet). As in the previous section we assume that the honeynet is composed of k honeypots out of n possible hosts within a block of IP addresses.

As mentioned earlier, the attacker probes the hosts and analyzes their responses to identify honeypots. A clever defender will respond to the probes in a way that the attacker does not gain information on whether the host is a regular one or a honeypot. The response of the host depends on whether the host is a honeypot (different responses by low-interaction and high-interaction honeypots) or a regular host. The response of the honeypot may be generated such that to conceal the identity of the honeypot. However, it is safe to assume that after a few probes an attacker can successfully identify that a host is a honeypot, i.e., the defender cannot infinitely deceive the attacker in believing a host to be regular host, when in fact it is a honeypot.

The goal of the attacker is to identify and attack a host that is not a honeypot while the goal of the defender is to conceal the honeypot (make it appear as a regular host) and

have the honeypot attacked by the attacker. Even if the attacker does not attack the honeypot, the defender gathers crucial information if it can make the attacker probe the honeypot as much as possible.

In the proposed model the attacker and the defender *make their decisions sequentially* by considering the opponent player's moves. We model this scenario as a game with sequential moves. The first move is made by the defender, in which she decides how to place the k honeypots among the n possible hosts. There are $\binom{n}{k}$ different ways in which k honeypots can be placed into n positions. Next the attacker decides what host to probe out of the n possible hosts. The next move is by the defender, who responds to the probe of the attacker. In our model the defender has two types of responses, 'r' and 'h' where response 'r' means that the host being probed is a regular host and the response 'h' means the host being probed is a honeypot. It should be noted that these responses may not represent the actual state of the host, *i.e.*, the defender may lie to deceive the attacker. Depending on the response of the defender, she may choose to probe the same host again or probe some other host. The game continues with alternative moves by the attacker and the defender in the form of probes and responses, until the attacker probes and the defender produces responses γ times. After γ stages (probe-response) the attacker makes the final move in which she chooses to either attack a host or not attack any host. In the intermediate stages of the game the payoffs are computed based on whether the attacker probes a honeypot or a regular host. The payoff of the attacker and the defender in the final move (by the attacker) are based on whether the attacker attacks a honeypot or a regular host or chooses not to attack. The total payoff is the sum of the payoffs of intermediate moves and the final move.

In the above game the final move by the attacker is made after γ probes to the hosts, independent of the host being probed. Another scenario which can be considered is the one in which the attacker makes the final move after she has probed a host at least γ times. In that case the total number of probes made by the attacker will be greater than or equal to γ . In the following, we consider only the first scenario, in which the attacker makes the final move after γ probes, irrespective of the host probed. We call the game that models this situation a (n, k, γ) -honeynet deception game, denoted as $HDG(n, k, \gamma)$.

Since a strategic form game model does not capture the sequential nature of the decision making process, we will model the interactions between the attacker and the system as an extensive form game. An *extensive form game* is a dynamic model that expresses the sequential structure of decision making. In order to have a clear and manageable way to explain the concepts used in our game theoretic modeling we first present a particular version of the general game, $HDG(2, 1, 1)$, in which after one probe the attacker

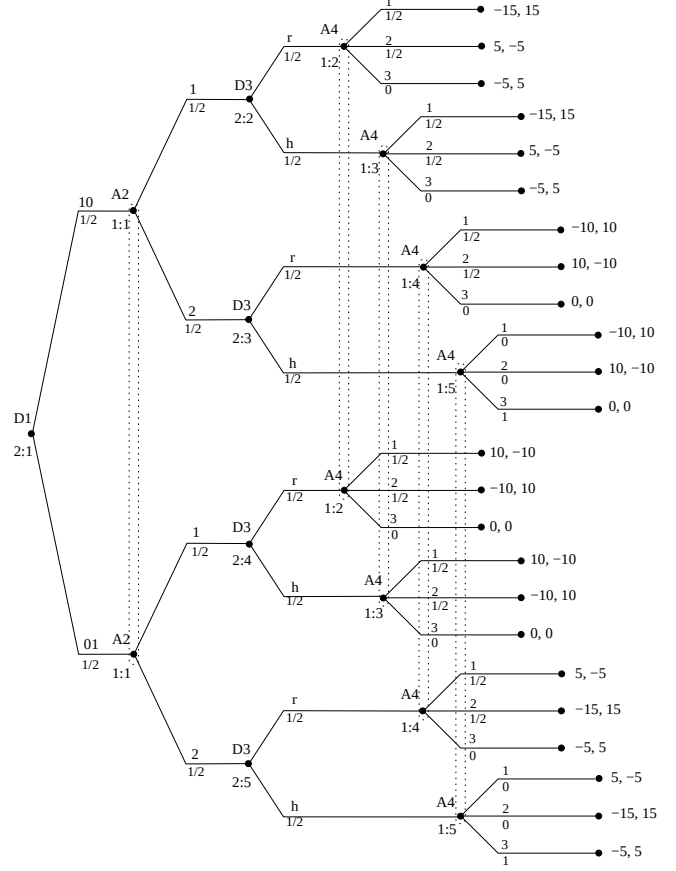


Fig. 1. $HDG(2, 1, 1)$ game

decides to attack one of the hosts or not. Later on we will describe the general game.

Fig. 1 depicts a $HDG(2, 1, 1)$ game in extensive form. In this figure the moves are indicated above the horizontal lines. The game has four stages:

- (i) D1 - The defender (D) decides the position of the honeypot to be at host 1 or host 2 (denoted by actions 10 and 01, respectively). This information is not known to the attacker, thus $HDG(2, 1, 1)$ is a game of imperfect information.
- (ii) A2 - The attacker (A) probes one of the hosts (1 or 2).
- (iii) D3 - The defender produces a response to the probe of the attacker ('r'=regular host, or 'h'=honeypot).
- (iv) A4 - The attacker either attacks one of the hosts (1 or 2) or does not attack at all (3).

The strategies of the defender are as follows:

- (i) *Deception strategies*: When a host is probed, respond with 'r' when the host is an actual honeypot (*i.e.*, conceal the honeypot), or respond with 'h' when the host is an actual regular host (*i.e.*, show a fake honeypot). An example of such strategy is when D places a honeypot at host 1 *i.e.*, chooses move (10), A probes host 1 and D responds with 'r'.

- (ii) *Disclosure strategies*: When a host is probed, respond with the actual status of the host, *i.e.*, with ‘r’ when the host is an actual regular host and with ‘h’ when the host is an actual honeypot.

An example of such strategy is when D places a honeypot at host 1 *i.e.*, chooses move (10), A probes host 1 and D responds with ‘h’.

A series of actions in an extensive game is called a *sequence*. Formally $B = (a_1 \dots a_n)$ is a *sequence* if action a_i was taken during stage i . The game in Fig. 1 has 38 such sequences: (10), (01), (10, 1), (10, 2), (01, 1), (01, 2), (10, 1, r), (10, 1, h), ..., (01, 2, h, 2), (01, 2, h, 3). The sequence (10, 1, r, 3) indicates that D places a honeypot at host 1, A probes host 1, D replies that host 1 is a regular host and then A does not attack any host.

A sequence has *histories*. The sequence (10, 1, r, 3) has five *subhistories*: $\emptyset$, (10), (10, 1), (10, 1, r), and (10, 1, r, 3). The histories $\emptyset$, (10), (10, 1) and (10, 1, r) are *proper subhistories* of (10, 1, r, 3); subhistory (10, 1, r, 3) is a *terminal history*. The terminal histories are annotated with preferences. The *preference* of terminal history l is a tuple of payoffs $u_i(l)$, one for each player. The *preference* tuple indicates the payoff of A and D . The first component of the tuple in Fig. 1 is the payoff to A while the second component is the payoff to D . The *payoff function* $u_i(l)$, $i \in \{A, D\}$ gives the preference of the players given terminal history l . Player i , $i \in \{A, D\}$ prefers terminal history l' to l , if $u_i(l') > u_i(l)$ and is indifferent between histories l' and l if $u_i(l') = u_i(l)$. In the game represented in Fig. 1, D prefers history (10, 1, r, 1) to (10, 1, r, 2) as $u_D(10, 1, r, 1) > u_D(10, 1, r, 2)$.

$HDG(n, k, \gamma)$ is a game with imperfect information since all the players do not have precise information about the previous actions of all the other players at all stages of the game. Specifically, the attacker does not know how the defender has placed the honeypots in the first stage of the game. To represent this lack of information in the model, the notion of *information set* is used. Let us denote by H_i the set of histories after which player i moves. An information set is a partition of H_i . The collection of information sets is called the *information partition* of player i . When making a move, player i is informed about the information set that has occurred but not about the exact history within that set. In Fig. 1 the information set is given below the stage label in the format $x : y$ where x refers to the player number ($x = 1$ is the attacker and $x = 2$ is the defender) and y denotes the number of the information set. In $HDG(2, 1, 1)$ game, the attacker knows the information set $1 : 1 = \{(10), (01)\}$ after stage $D1$, but not which one of the two histories, (10) and (01), has occurred. Formally an extensive-form game can be defined as follows:

Definition 4: (Extensive-Form Game with imperfect Information) [20] An *extensive-form game with imperfect information* consists of the following:

Action	A 's payoff	D 's payoff
A probes a honeypot	-5	5
A probes a regular host	0	0
A attacks a honeypot	-10	10
A attacks a regular host	10	-10
A chooses not to attack	0	0

TABLE I
PAYOFFS FOR $HDG(n, k, \gamma)$ GAME

- (i) The set of players;
- (ii) The set of terminal histories;
- (iii) A function $P(l)$ that assigns a player to the move after subhistory l ;
- (iv) For each player, the information partition (a partition of the set of histories assigned to that player)
- (v) Preferences over the set of lotteries (*i.e.*, probability distributions) over terminal histories, for each player.

We formally define the $HDG(2, 1, 1)$ game in extensive form (Fig. 1) as follows:

- (i) $\{A, D\}$ is the set of players;
- (ii) $\{(10, 1, r, 1), (10, 1, r, 2), (10, 1, r, 3), (10, 1, h, 1), (10, 1, h, 2), (10, 1, h, 3), (10, 2, r, 1), (10, 2, r, 2), (10, 2, r, 3), (10, 2, h, 1), (10, 2, h, 2), (10, 2, h, 3), (01, 1, r, 1), (01, 1, r, 2), (01, 1, r, 3), (01, 1, h, 1), (01, 1, h, 2), (01, 1, h, 3), (01, 2, r, 1), (01, 2, r, 2), (01, 2, r, 3), (01, 2, h, 1), (01, 2, h, 2), (01, 2, h, 3)\}$ is the set of terminal histories;
- (iii) The function $P(h)$ such that $P(D1) = D$, $P(A2) = A$, $P(D3) = D$, and $P(A4) = A$, where $D1$, $A2$, $D3$ and $A4$ are the stages of the game as described earlier;
- (iv) Information partitions: All histories after which the defender moves are contained in individual information sets, which means that the defender observes all actions of the attacker; the attacker's information partition after stage $D1$ consists of the histories $\{(01), (10)\}$ and after stage $D3$ consists of $\{(10, 1, r), (01, 1, r)\}$, $\{(10, 1, h), (01, 1, h)\}$, $\{(10, 2, r), (01, 2, r)\}$ and $\{(10, 2, h), (01, 2, h)\}$, which means that at any point in the game, the attacker cannot distinguish which host is a honeypot and which one is a regular host.
- (v) The preferences over the terminal histories are calculated according to Table I. When the attacker probes a honeypot, it provides some information to the defender so the payoff of the attacker in this case is -5 and that of the defender is 5. If the attacker probes a regular host she does not provide any information to the defender and the defender does not get any benefit, thus both receive a payoff of 0. The payoffs for all the actions of the attacker are summarized in the table. The final preferences are the sums of the intermediate payoffs after stages $A2$ and $A4$. For

example, consider terminal history (10,1,r,2). The defender places the honeypot at host 1 and the attacker probes host 1, thus the attacker obtains an intermediate payoff of -5 and the defender obtains an intermediate payoff of 5 . However, in the final move the attacker attacks host 2 and thus obtains a payoff of 10 in that stage, making her total payoff of $-5 + 10 = 5$. The defender obtains -10 after stage A4 and her total payoff is $5 - 10 = -5$.

As we mentioned earlier, the objective of a player is to maximize her payoff. Players are not isolated; the actions of the defender influence and are influenced by the actions of the attacker. Let $A(I_i)$ denote the actions available to player i at information set I_i . A function which assigns an action $a_i \in A(I_i)$ to each of i 's information sets I_i is called a *pure strategy* of player i in an extensive game. A *mixed strategy* of a player in an extensive game is a probability distribution over the player's pure strategies. We denote the mixed strategy (α_i) of player $i = \{A, D\}$ at information set I_i as a vector of probabilities, where each probability corresponds to the available actions in the order in which the actions are given in the game definition. Thus when the defender has a mixed strategy of playing both the moves ('r' and 'h') at information set $2 : 2$ will be denoted by $\alpha_D(2 : 2)$.

The solution concept for extensive form games that we consider in this study is the mixed strategy Nash equilibrium. This equilibrium is a steady state in which no player can unilaterally increase her payoff. It is formally defined as follows:

Definition 5: (Nash equilibrium of extensive game) [20] The mixed strategy profile α^* in an extensive game is called a *Nash equilibrium in mixed strategies* if, for each player i and every other mixed strategy α_i of player i , player i 's expected payoff to α^* is at least as much as his expected payoff to $(\alpha_i, \alpha_{-i}^*)$ where α_{-i}^* denotes the strategy profile of all players except player i .

The Nash equilibria of an extensive game are often computed by constructing the strategic form of the game and analyzing it as a strategic game [20]. We used Gambit [21], a software tool for solving games, to compute the equilibria of $HDG(2, 1, 1)$. Gambit computed several Nash equilibria in mixed strategies. We will present and discuss only one of these equilibria of the $HDG(2, 1, 1)$ game which is shown in Fig. 1. In the figure the probability of an action being played in this particular mixed strategy equilibrium is shown below the horizontal lines representing the actions of the players. The mixed strategy equilibrium shown in the figure can be represented as $\alpha_D(2 : 1) = \alpha_D(2 : 2) = \alpha_D(2 : 3) = \alpha_D(2 : 4) = \alpha_D(2 : 5) = (1/2, 1/2)$, $\alpha_D(1 : 1) = (1/2, 1/2)$, $\alpha_D(1 : 2) = \alpha_D(1 : 3) = \alpha_D(1 : 4) = (1/2, 1/2, 0)$, $\alpha_D(1 : 5) = (0, 0, 1)$. This strategy implies that at stage D1, the defender chooses one of the configurations with equal probability (both $1/2$),

because she has the same expected payoff for both actions. The attacker probes one of the two hosts with equal probability $1/2$, since she does not have any information about the placement of the honeypots. At stage D3, the defender plays the deception strategy with probability $1/2$. This means that when a regular host is probed, the defender responds with 'h' half of the time to deceive the attacker. Similarly when a honeypot is probed, it responds with 'r' half of the time. This way the attacker can be deceived to attack a honeypot. When the attacker probes host 1, whatever response she gets, she attacks host 1 and host 2 with equal probability. This is safe for the attacker because she does not know which node is a honeypot. When the attacker probes host 2 and receives the response 'h', the attacker chooses not to attack, to avoid the risk of attacking the honeypot. However when she receives the response 'r', she attacks host 1 and host 2 with equal probability. This particular strategy is suitable for an attacker who wants to avoid the risk to a certain degree.

The general $HDG(n, k, \gamma)$ game can be formally defined as follows:

- (i) $\{A, D\}$ is the set of players;
- (ii) The set of terminal histories is denoted by set of vectors $\{(x_1, x_2, \dots, x_m) | m = 2^{\gamma+1}, \text{ where } x_i \text{ is the action taken by the player at stage } i \text{ of the game}\}$. The actions available to the players are as follows:
 - (a) $x_1 = \{(y_1 y_2 \dots y_n) | k \text{ vector components are } 1 \text{ and } n - k \text{ components are } 0\}$. If $y_i = 1$ then D places a honeypot at position (address) i and if $y_i = 0$ then D places a regular host at position i . There are $q = \binom{n}{k}$ such combinations possible. We call this vector a *configuration* and denote configurations by $c_t | t = \{1, \dots, q\}$
 - (b) $x_i = j, j \in \{1, \dots, n\}$, are the actions available to A in intermediate stages (i is even, $i \neq m$) where A chooses to probe host j out of n hosts.
 - (c) $x_i = \{r, h\}$, are the actions available to D in different stages (i is odd, $i \neq 1$) (r denotes D 's response to the probe that x_{i-1} is regular host, h denotes a honeypot (might be a deception move)).
 - (d) $x_m = j, j \in \{1, \dots, n + 1\}$ in which A chooses to attack host j ($j = n + 1$ means no attack).
- (iii) The function $P(l)$ such that $P(\emptyset) = D$, $P(l) = A$, if the last action in subhistory l was taken by D else $P(l) = D$. In other words, D and A play alternatively, starting with D and the last action is taken by A .
- (iv) The information sets in the information partition of D consists of individual histories $\{(x_1, x_2)\}, \{(x_1, x_2, x_3, x_4)\}, \dots, \{(x_1, x_2, \dots, x_{m-2})\}$. The information partition of A has all histories with different values of x_1 but same values of $(x_2, \dots, x_{m-1})$ grouped in the same information set. Formally, the information partition of A is $\{(c_t)\}, \{(c_t, 1, r)\}, \{(c_t, 1, h)\}, \{(c_t, 2, r)\}, \dots, (c_t, n, h, \dots, h) | t =$

- $\{1, \dots, q\}$
- (v) The preferences over the terminal histories are calculated by summing the intermediate payoffs (given in Table I) corresponding to the actions in the terminal history.

Since the information sets in which D moves contain individual histories, the defender observes all actions of the attacker. However each information set of the attacker contains q histories, each history corresponding to different configurations. For example, if $n = 3, k = 2, \gamma = 1$, there are 3 different configurations, $c_1 = 110, c_2 = 101$ and $c_3 = 011$. The information partition of the defender is as follows $\{ \{(\phi)\}, \{(110, 1)\}, \{(110, 2)\}, \{(110, 3)\}, \{(101, 1)\}, \{(101, 2)\}, \{(101, 3)\}, \{(011, 1)\}, \{(011, 2)\}, \{(011, 3)\} \}$

Each information set of the attacker will contain histories with different values of c_t but the same values for the other actions, *i.e.* $\{ \{(110), (101), (011)\}, \{(110, 1, r), (101, 1, r), (011, 1, r)\}, \{(110, 1, h), (101, 1, h), (011, 1, h)\}, \{(110, 2, r), (101, 2, r), (011, 2, r)\}, \{(110, 2, h), (101, 2, h), (011, 2, h)\} \}$. This means that at any point in time, the attacker cannot identify the placement of honeypots and regular hosts in the system.

This general game model allows us to analyze the interactions of the attacker and the defender. Since the equilibrium solutions represent accurate predictions of how the attacker and the defender behave in the honeynet system, they are used to determine the best strategies of the attacker and the defender.

IV. CONCLUSIONS

We proposed a game theoretic framework for modeling deception in honeynets. The framework is based on extensive games of imperfect information. We studied the equilibrium solutions of these games and showed how they are used to determine the strategies of the honeynet system. There are several ideas for future work that stem from this initial study of game theoretic models for honeynets. We plan to investigate and characterize into more details the general deception game proposed in this paper. We also plan to investigate other types of games in which the attacker has partial information on the honeypots placement.

REFERENCES

- [1] Honeynet Project, *Know Your Enemy: Learning about Security Threats*. Boston, MA: Pearson Education Inc., second ed., 2004.
- [2] B. Cheswick, "An evening with berferd, in which a hacker is lured, endured, and studied," in *Proc. of the Winter Usenix Conference*, January 1992.
- [3] J. Bethencourt, J. Franklin, and M. Vernon, "Mapping internet sensors with probe response attacks," in *Proceedings of the USENIX Security Symposium*, pp. 193–208, August 2005.
- [4] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver, "Inside the slammer worm," *IEEE Security and Privacy*, vol. 1, pp. 33–39, July 2003.
- [5] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage, "Inferring internet denial-of-service activity," *ACM Trans. Comput. Syst.*, vol. 24, no. 2, pp. 115–139, 2006.
- [6] C. Kreibich and J. Crowcroft, "Honeycomb - creating intrusion detection signatures using honeypots," *ACM SIGCOMM Computer Communications Review*, vol. 34, no. 1, pp. 51–56, 2004.
- [7] D. Moore, C. Shannon, G. M. Voelker, and S. Savage, "Network telescopes: Technical report," Technical Report CS2004-0795, UCSD, July 2004.
- [8] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson, "Characteristics of internet background radiation," in *Proc. of the Internet Measurement Conference*, pp. 27–40, October 2004.
- [9] V. Yegneswaran, P. Barford, and D. Plonka, "On the design and use of internet sinks for network abuse monitoring," in *Proc. of the Symposium on Recent Advances in Intrusion Detection*, September 2004.
- [10] N. Provos, "A virtual honeypot framework," in *Proc. of the 13th USENIX Security Symposium*, pp. 280–284, August 2004.
- [11] M. Vrabie, J. Ma, J. Chen, D. Moore, E. Vandekieft, A. C. Snoeren, G. M. Voelker, and S. Savage, "Scalability, fidelity, and containment in the potemkin virtual honeyfarm," in *Proc. of the 20th ACM Symposium on Operating Systems Principles*, pp. 148–162, October 2005.
- [12] L. K. Yan, "Virtual honeynets revisited," in *Proc. of the IEEE Workshop on Information Assurance*, pp. 232–239, June 2005.
- [13] F. Cohen, "A note on the role of deception in information protection," *Computers and Security*, vol. 17, no. 6, pp. 483–506, 1998.
- [14] F. Cohen, "Leading attackers through attack graphs with deceptions," *Computers and Security*, vol. 22, no. 5, pp. 402–411, 2003.
- [15] N. C. Rowe, B. T. Duong, and E. J. Custy, "Fake honeypots: A defensive tactic for cyberspace," in *Proc. of the IEEE Workshop on Information Assurance*, pp. 223–230, June 2006.
- [16] N. C. Rowe, "Measuring the effectiveness of honeypot counter-counterdeception," in *Proc. of the 39th Hawaii International Conference on System Sciences*, January 2006.
- [17] J. F. Nash, "Equilibrium point in n -person games," *Proc. Nat'l Academy of Sciences of the United States of Am.*, vol. 36, pp. 48–49, Jan. 1950.
- [18] J. F. Nash, "Non-cooperative games," *Annals of Math.*, vol. 54, no. 2, pp. 286–295, 1951.
- [19] M. Osborne and A. Rubinstein, *A Course in Game Theory*. Cambridge, Mass.: MIT Press, 1994.
- [20] M. Osborne, *An Introduction to Game Theory*. New York, NY: Oxford University Press, 2004.
- [21] R. D. McKelvey, A. McLennan, and T. Turocy, *Gambit: Software Tools for Game Theory*. Version 0.2007.01.30: <http://econweb.tamu.edu/gambit>, 2007.