Concealing Cyber-Decoys using Two-Sided Feature Deception Games

Mohammad Sujan Miah[1], Marcus Gutierrez[1], Oscar Veliz[1], Omkar Thakoor[2], and Christopher Kiekintveld[1]

[1] University of Texas at El Paso, El Paso TX 79968, USA
{msmiah,mgutierrez22}@miners.utep.edu
{osveliz,cdkiekintveld}@utep.edu
[2] University of Southern California, Los Angeles CA 90007, USA
othakoor@usc.edu

Abstract. An increasingly important tool for securing computer networks is the use of deceptive decoy objects (e.g., fake hosts, accounts, or files) to detect, confuse, and distract attackers. One of the well-known challenges in using decoys is that it can be difficult to design effective decoys that are hard to distinguish from real objects, especially against sophisticated attackers who may be aware of the use of decoys. A key issue is that both real and decoy objects may have observable features that may give the attacker the ability to distinguish one from the other. However, a defender deploying decoys may be able to modify some features of either the real or decoy objects (at some cost) making the decoys more effective. We present a game-theoretic model of two-sided deception that models this scenario. We present an empirical analysis of this model to show strategies for effectively concealing decoys, as well as some limitations of decoys for cyber security.

Keywords: Honeypots · Deception · Game Theory

1 Introduction

Both civilian and military computer networks are under increasing threat from cyber attacks, with the greatest threat posed by Advanced Persistent Threat (APT) actors. These attackers use sophisticated methods to compromise networks and remain inside, establishing greater control and remaining for long periods of time to gather valuable data and intelligence. These attackers seek to remain undetected, and estimates from APT attacks show that they are often present in a network for months before they are detected [5].

Cyber deception methods use deceptive decoy objects like fake hosts (honeypots), network traffic, files, and even user accounts to counter attackers in a variety of ways [24, 9, 1]. They can create confusion for attackers, make them more hesitant and less effective in executing further attacks, and can help to gather information about the behavior and tools of various attackers. They can also increase the ability of defenders to detect malicious activity and actors in

the network. This is especially critical in the case of APT attackers, who are often cautious and skilled at evading detection [26]. Broad and effective use of honeypots and other deceptive objects is a promising approach for combating this class of attackers.

However, the effectiveness of honeypots and other deceptive objects depends crucially on whether these fake objects can be designed to look similar enough to real objects that they are not easily identified and avoided. This is especially true for APT threats, which are likely to be aware of the use of such deception technologies and will actively seek to identify and avoid honeypots and other deceptive objects in their reconnaissance [26, 29]. A well known problem with designing effective honeypots is that they often have characteristics that can be observed by an attacker that will reveal the deception [10]. For example, the patterns of network traffic to a honeypot, the response times to queries, or the configuration of services may not be similar to real hosts in the network. However, with some additional effort the deception can be made more effective (e.g., by simulating more realistic traffic to and from honeypots).

We introduce a game-theoretic model of the problem of designing effective decoy objects that can fool even a sophisticated attacker. In our model, real and fake objects may naturally have different distributions of characteristic features than an attacker could use to tell them apart. However, the defender can make some (costly) modifications to *either* the real or the fake objects to make them harder to distinguish. This model captures some key aspects of effective cyber deception that are missing from other game-theoretic models of deception. In particular, we focus on whether the defender can design convincing decoy objects, and what the limitations of deception are if some discriminating features of real and fake objects cannot be easily masked.

We present several analyses of fundamental questions in cyber deception based on our model. We analyze how to measure the informativeness of the signals in our model, and then consider how effectively the defender can modify the features to improve the effectiveness of deception in various settings. We show how different variations in the costs of modifying the features can have a significant impact on the effects of deception. We also consider the differences between modifying only the features of deceptive objects and being able to modify both real and deceptive objects (two-sided deception). While this is not always necessary, in some cases it is essential to enable effective deception. We also consider deception against naïve attackers, and how this compares to the case of sophisticated attackers.

2 Motivating Domain and Related Work

While the model we present can be applied to many different types of deception and deceptive objects, we will focus on honeypots as a specific case to make our discussion more concrete and give an example of how this model captures key features of real world deception problems. Honeypots have had a considerable impact on cyber defense in the 30 years since they were first introduced [25].

Over time, honeypots have been used for many different purposes, and have evolved to be much more sophisticated with greater abilities to mimic real hosts and to capture useful information about attackers [14, 16, 3]. The sophistication of honeypots can vary dramatically, from limited low-interaction honeypots to sophisticated high interaction honeypots [14, 18, 8].

Here, we do not focus on the technological advancements of honeypots, but rather the game-theoretic investigation of honeypot deception. There have been numerous works that emphasize this game-theoretic approach to cyber deception as well. Our work builds upon the Honeypot Selection Game (HSG), described by Píbil et al. [17, 9]. Much like the HSG, we model the game using an extensive form game. We extend the HSG model with the introduction of *features*, which are modifiable tokens in each host that enable more robust deceptions and allow to model more realistic settings. Several game-theoretic models have been established for other cyber defense problems [2, 13, 22, 21], specifically for deception as well [20, 28], however these consider attribute obfuscation as the means of deception rather than use of decoy objects.

[27] notably investigates the use of honeypots in the smart grid to mitigate denial of service attacks through the lens of Bayesian games. [12] also model honeypots mitigating denial of service attacks in similar fashion, but in the Internet-of-Things domain. [7] tackle a similar "honeypots for denial of service attack" problem with Bayesian game modeling in the social networking domain. These works demonstrate the vast amounts of domains honeypots can aid in. This work differs in that we do not model a Bayesian incomplete information game.

A couple of works also consider the notion of two-sided deception, where the defender not only deploy *real*-looking honeypots, but also *fake*-looking real hosts. Rowe et al. demonstrated mathematically that deploying two-sided deception offers an improved defense by scaring off attackers [19]. Caroll and Grosu introduce the signaling deception game where deployed honeypot deception is bolstered through the use of signals [4]. Our work differs in that we define specific features (signals) that can be altered and revealed to the attacker. Shi et al. introduce the mimicry honeypot framework which combines real nodes, honeypots, and *fake*-looking honeypots to derive equilibria strategies to bolster defenses [23]. They validated their work in a simulated network. This notion of two-sided deception is quickly becoming reality; De Gaspari et al. provided a prototype proof-of-concept system where production systems also engaged in active deception [6].

3 Honeypot Feature Selection Game

We now present a formal model of the Honeypot Feature Selection Game (HFSG). This game models the optimal decisions for a player (the defender) who is trying to disguise the identify of real and fake objects so that the other player (the attacker) is not able to reliably distinguish between them. Each object in the game is associated with a vector of observable features (characteristics) that

provides an informative signal that the attacker can use to more reliably detect fake objects. The defender is able to make (limited) changes to these observable features, at a cost. Unlike many models of deception, we consider the possibility that the defender can make changes to both the real and fake objects; we refer this as 2-sided deception.

The original feature vector is modelled as a move by nature in a Bayesian game. Real and fake objects have different probabilities of generative every possible feature vector. How useful the features are to the attacker depends on how similar the distributions for generating the feature vectors are; very similar distributions have little information while very divergent distributions may precisely reveal which objects are real or fake. The defender is able to observe the features, and may choose to pay some cost to modify a subset of the features. The attacker observes this modified set of feature vectors, and makes a choice about which object to attack. The attacker receives a positive payoff if he selects a real object, and a negative one if he selects a fake object.

To keep the initial model simple we focus on binary feature vectors to represent the signals. We will also assume that the defender is able to modify a maximum of one feature. Both of these can be generalized in a straightforward way, at the cost of a larger and more complex model.

3.1 Formal definition of Honeypot Feature Selection Game

We now define the Honeypot Feature Selection Game (HFSG) formally by the tuple $G = (k, n, V, C, P_r, P_h, \tau, \chi, D, A, u)$.

- k is the number of host types
- n is the number of features for any given host
- V is the set of importance values assigned for each host
- C is the set of costs associated with modifying each feature
- $P_r : \{0,1\}^n \rightarrow [0,1]$ is probability distribution over feature vectors for real hosts
- $P_h : \{0,1\}^n \rightarrow [0,1]$ is the probability distribution over feature vectors for honeypots
- The attacker's information set is of the form $\{y^1, y^2\}$ where $y^1, y^2 \in \{0,1\}^n$
- a_I: a binary variable that denotes whether the attacker attacks the lexicographically smaller y^1, $\forall I = \{y^1, y^2\}$. Thus, when $y^1 = y^2$, $a_I = 1$.
- τ: the superset of all attacker information sets ($I \in \tau, I \subseteq 2^{kn}$)
- $\chi : 2^{kn} \rightarrow f(\tau)$ is a function that specifies a set of possible actions D for the defender, who can modify the hosts of a network x and present a modified observation $x' \in 2^{kn}$ to the attacker
- A is set of possible attacker actions $\forall x' \subseteq \tau$
- $u : k \times \tau \times A \rightarrow \mathbf{R}$ is defined if attacker can identify the real system where any feature can be modified by $\chi(x)$

An example of a small HFSG with 1 real host, 1 honeypot, and 1 feature for each host is shown in Figure 1. The feature of a host can be organized in two possible ways ('0' and '1'). The probability distributions $P_r = [0.5, 0.5]$ and $P_h = [0.5, 0.5]$ are randomly generated for each feature combination.

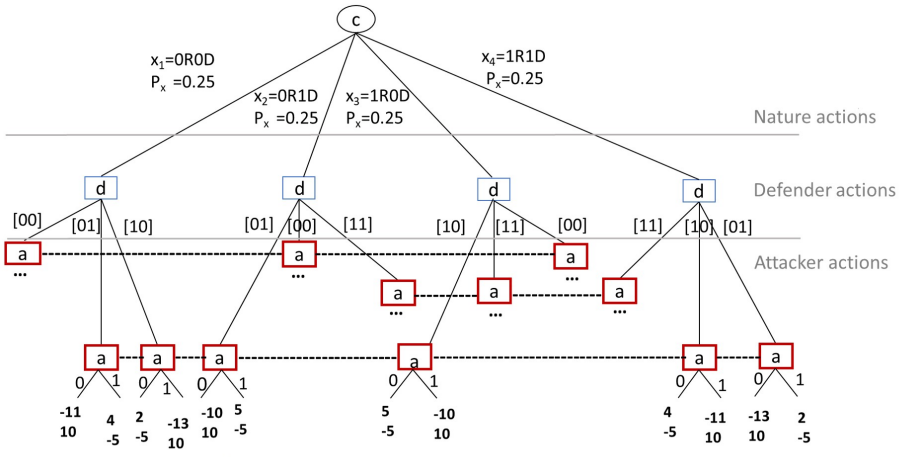

Fig. 1. The extensive form game tree with one real host, one decoy and 1 feature in each host. The importance value of real host is 10 where the modification cost of a feature is 3. The importance value of decoy is 5 where any feature modification cost is 1.

3.2 Nature Player Actions

We assume that both players know the two probability distributions P_r and P_h that define how the feature vectors are selected by nature for real and honeypot hosts, respectively. For each network configuration $x \in X$, all features are stored in a vector M_f. Then, from the number of host types $k \in \mathbf{N}$ and the total number of features $n \in \mathbf{N}$ in each host, we generate all possible network configuration $X \in 2^{kn}$. The nature player selects a network $x \in X$ with probability distribution $P_x = P_r \times P_h$. Both players can compute the value of P_x for the network x. For example, in Figure 1 $P_x = 0.25$ for network "0R1D" is calculated from $P_r('0') = 0.5$ and $P_h('1') = 0.5$.

3.3 Defender Actions

For a network configuration $x \in X$, the defender observes the feature vector M_f and the probability distribution P_x for network x. Then he chooses an appropriate action $d \in D$. Any action d on network $x \in X$ results in a new network $x' \in X$ which serves an information set $I \in \tau$ to the attacker. In the example of Figure 1, the middle action in network "0R0D" leads to a new network "01", the attacker observes this network "01" with different configuring probabilities in the different situation.

3.4 Attacker Actions

The attacker observes the set of feature vectors for each host, but does not directly know which ones are real and which are honeypots. The attacker does

have prior beliefs about how likely each type of host is to generate each feature vector, so he can use this to update his beliefs. Networks belonging to the same information set $I \in \tau$ are indistinguishable to the attacker. Consider two networks networks $\mathbf{y} = (\mathbf{y_1}, \mathbf{y_2})$ and $\mathbf{y'} = (\mathbf{y_1}', \mathbf{y_2}')$, where $(y_{1n}, \ldots, y_{1n})$ is an n-dimensional vector, belonging to same information set $I \in \tau \iff \forall t \in T : y1_t + y2_t = y1'_t + y2'_t$ or $y1_t + y2_t = y2'_t + y1'_t$ holds, t denotes terminal sate of game tree. Networks $y, y' \in I$ have similar feature combinations but differ in the success probabilities in distinguishing the real system and the honeypot. In Figure 1 the networks "01" and "10" will be observed by an attacker as "01".

3.5 Utility Functions

The terminal state $t \in T$ of an extensive form game tree contains the players' utilities $U(x, I, a_i^I)$. A valid sequence from the root node to a terminal state defines rewards for both players, and the rewards depends on a network configuration x in X, the attacker's information set $I \in \tau$ and the attacker's action a_i^I at I. The attacker's utility is computed based on the sum of importance value $\sum_{t \in T}$ V at terminal states - successful identification of a real host gives a positive reward to the attacker otherwise gives negative reward that is equal to the importance value of a honeypot. The attacker's utility at terminal state t is $U_a(x, I, a_i^I) = V_t^k$. We assume that V_t^k is zero-sum component in utility calculation where only feature modification cost C_t of defender makes the game general sum. So, the defender's utility at t is $U_d(x, I, a_i^I) = -V_t^k - C_t^k$.

3.6 Defender's Linear Program

We can solve this extensive form game with imperfect information using a linear program. For solving this game in sequence form [11], we create a path from the root node to the terminal node that is a valid sequence and consists a list of actions for all player. Then we compute defender's behavioral strategies on all valid sequences using a formulated LP as follows, where U_d and U_a are the utilities of the defender and the attacker. To solve the program, we construct a matrix $X[0 : 2^{kn}]$ of all possible network configurations and then the defender chooses a network $x \in X$ to modify. In network x, any action d_x^j of defender leads to an information set $I \in \tau$ for the attacker. Different defender's actions in different networks can lead to the same information set $I \in \tau$. Then, in a particular information set $I \in \tau$, the attacker take an action a_i^I to maximize his expected utility.

$$max \sum_{x \epsilon X} \sum_{j \in D} U_d(x, j, a^{I(x,j)})\ d_j^x P_x \tag{1}$$

$$s.t. \sum_{(x,j) \in I} U_a(x, j, a^I) d_j^x P_x \geq \sum_{(x,j) \in I} U_a(x, j, a)\ d_j^x P_x \quad \forall a \in A \quad \forall I \in \tau \tag{2}$$

$$d_j^x \geq 0 \quad \forall x \in X \quad \forall j \in D \tag{3}$$

$$\sum_{j \in D} d_j^x = 1 \quad \forall x \in X \tag{4}$$

The program's objective is to maximize the defender's expected utility where the attacker also plays his best response. In the above program, the only unknown variables are the defender's actions d_j^x (more precisely the strategies of a defender in a network $x \in X$) and the attacker's actions a^I . The inequality in Equation 2 ensures that the attacker plays his best response in this game. Also Equation 3 defines that the defender strategies in a network x is a standard probability distribution. Finally Equation 4 makes sure that all network configurations by nature are 1.

4 Empirical Study of HFSG

The HFSG game model allows us to study the strategic aspects of cyber deception against a sophisticated adversary who may be able to detect the deception using additional observations and analysis. In particular, we can evaluate the effectiveness of cyber deception under several different realistic assumptions about the costs and benefits of deception, as well as the abilities of the players. We identify cases where deception is highly beneficial, as well as some cases where deception has limited or no value. We also show that in some cases, using two-sided deception is critical to the effectiveness of deception methods.

4.1 Measuring the Similarity of Features

One of the key components of our model is that real and fake hosts generate observable features according to different probability distributions. The similarity of these distributions has a large effect on the strategies in the game, and the outcome of the game. Intuitively, if out-of-the-box honeypot solutions look indistinguishable from existing nodes on the network the deception will be effective without any additional intervention by the defender. However, when the distributions of features are very dissimilar the defender should pay higher costs to modify the features to disguise the honeypots. In some cases this may not be possible, and the attacker will always be able to distinguish the real and fake hosts.

Measuring the similarity of the feature distributions is a somewhat subtle issue, since the defender can make changes to a limited number of features. Standard approaches such as Manhattan distance or Euclidean distance do not provide a good way to compare the similarity due to these constraints. We use a measure based on the Earth Mover's Distance (EMD) [15], which can be seen as the minimum distance required to shift one pile of earth (probability distribution) to look like another. This measure can be constrained by the legal moves,

so probability is only shifted between configurations that are reachable by the defender's ability to change features.

In the experiments, we allow the defender to modify only a single feature in the network. We model the distance from moving the probability of one configuration (e.g., turning $[0, 0]$ into $[0, 1]$) to another by flipping of a single bit at a time with a unit cost of 1. This can be seen visually in Figure 2 where we calculate the EMD of moving the honeypot's initial distribution into that of the real node's initial distribution.

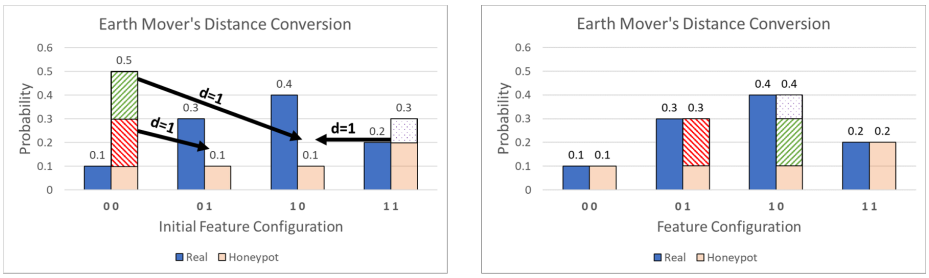

Fig. 2. Earth Mover's Distance process. a) Displays the initial feature configuration probability distributions P_r and P_h and where to move slices of the distribution from P_h and b) Shows the updated P_h after the conversion, resulting in a final EMD of 0.5.

In our experiments we will often show the impact of varying levels of similarity in the feature distributions. We generated 100 different initial distributions for the features using uniform random sampling. We then calculated the similarities using the constrained EMD, which resulted in the distribution of similarities shown in Figure 3. When our experiments vary the similarity, we present the results by aggregating over the similarity intervals of 0.1 and average the results in each interval.

4.2 Deception with Symmetric Costs

Our first experiment investigates the impact of varying the similarity of the feature distributions. We also vary the values of real and fake hosts. As the similarity of the distributions P_r and P_h decreases, we would expect a decrease in overall expected defender utility. We can see this decrease in Figures 4a and 4b as we vary the similarity measured using EMD. In Figures 4a and 4b, we compare the utility differences between an optimal defender that can only modify the features of the honeypot (one-sided deception), an optimal defender that can modify features of *both* the honeypot and real host (two-sided deception), and a baseline defender that cannot make any modifications against a fully rational best response attacker.

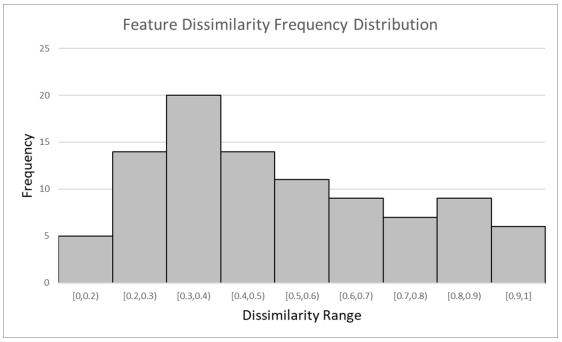

Fig. 3. Frequency distribution of similarity ranges used in the experiments.

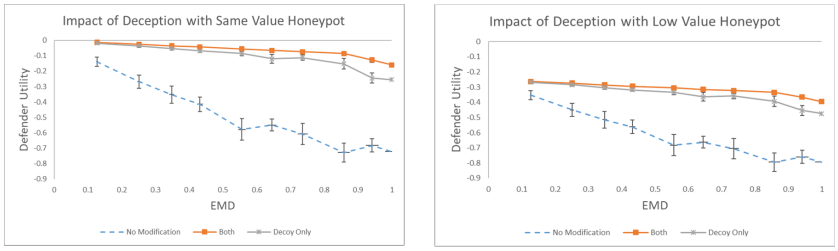

Fig. 4. Comparison of defender utility when the real host's importance value a) doubles that of the honeypot and b) equals that of the honeypot. Here we see one-sided deception provides a comparable defense despite a high initial dissimilarity.

In Figure 4a, the honeypot has the same importance value as the real host, while in Figure 4b, the honeypot value is half of the real host. The first observation is that in both cases the value of deception is high relative to the baseline with no deception, and this value grows dramatically as the feature distributions become more informative (higher EMD). In general, the defender does worse in cases where the hosts have different values. Two-sided deception does have a small advantage in cases with highly informative features, but the effect is small. Here, the costs of modifying the features are symmetric, so there is little advantage in being able to modify the feature on either the honeypot or the real host, since the defender can choose between these options without any penalty.

To further investigate the issue of one-sided and two-sided deception, we fix the honeypot features modification costs and increased real host modification costs as reflected in Table 1. Here, we compare how increasing the real host's feature modification negatively affects the defender's expected utility. As the cost for modifying the real hosts increases relative to the cost of modifying honeypots, the defender must make more changes on honeypots in order to maximize his

Figure	RIV	RMC		HpMC		HpIV
		F 1	F 2	F 1	F 2	
4a	1.0	0.25	0.1	0.1	0.25	0.5
4b	1.0	0.25	0.1	0.2	0.1	1.0
5 (Both (A))	1.0	0.25	0.1	0.1	0.2	0.5
5 (Both (B))	1.0	0.5	0.2	0.1	0.2	0.5
5 (Both (C))	1.0	1.0	0.5	0.1	0.2	0.5
6 (Exp-1)	1.0	0.1	∞	0.1	∞	1.0
6 (Exp-2)	1.0	0.1	∞	∞	0.1	1.0
7 (Exp-1)	1.0	0.2	0.2	0.2	0.2	1.0
7 (Exp-2)	1.0	0.15	0.25	0.25	0.15	1.0
7 (Exp-3)	1.0	0.1	0.3	0.3	0.1	1.0
7 (Exp-4)	1.0	0.05	0.35	0.35	0.05	1.0
7 (Exp-5)	1.0	0.0	0.4	0.4	0.0	1.0
9	1.0	0.25	0.1	0.2	0.1	1.0

Table 1. Parameters used in HFSG experiments. RIV denotes real system's importance value, RMC denotes real system's feature modification cost, HpIV denotes importance value of honeypot and HpMC denotes feature modification cost of honeypot. All numbers are normalized to 1

utility. Altering the real system in this case is not feasible and does not provide a good return on investment.

Traditionally network administrators avoid altering features in their real hosts on the network and simply employ one-sided deception, attempting to alter the honeypot to look like a real host. In the case where modifying a real host to look *less believable* might be be too costly or even impossible, one-sided deception is an obvious choice as demonstrated in Figure 5. However, when these real feature modifications are not too costly, we see that two-sided provides a noticeable increase in defenses when the feature distributions are increasingly dissimilar.

4.3 Deception with Asymmetric Costs

While the results so far have suggested that one-sided deception may be nearly as effective as two-sided deception, they have all focused on settings where the costs of modifying features are *symmetric* for real and fake hosts. We now investigate what happens when the costs of modifying different features are asymmetric. We start with the extreme case where some features may not be possible to modify at all.

In our examples with two features, we can set the unmodifiable features for the real and honeypot hosts to be the same or to be opposite. In Figure 6, we show the results of the game when we set the modification costs of some features to infinity. If the same feature for the real host and honeypot are unmodifiable, then there is little the defender can do to deceive an intelligent attacker when they are highly dissimilar. However, when the features that cannot be modified

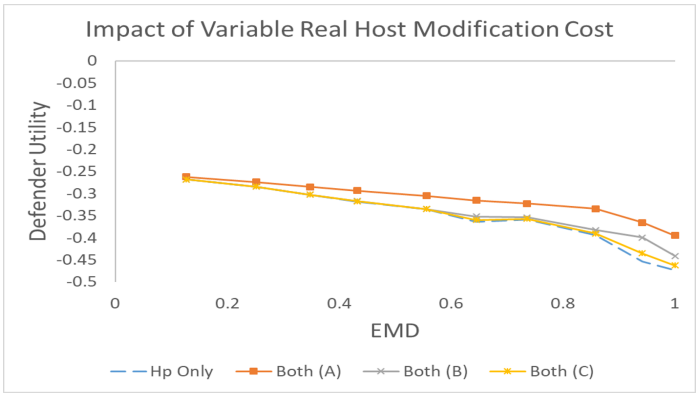

Fig. 5. Comparison of defender utility when the cost of modifying the real host features is different than modifying the honeypot features.

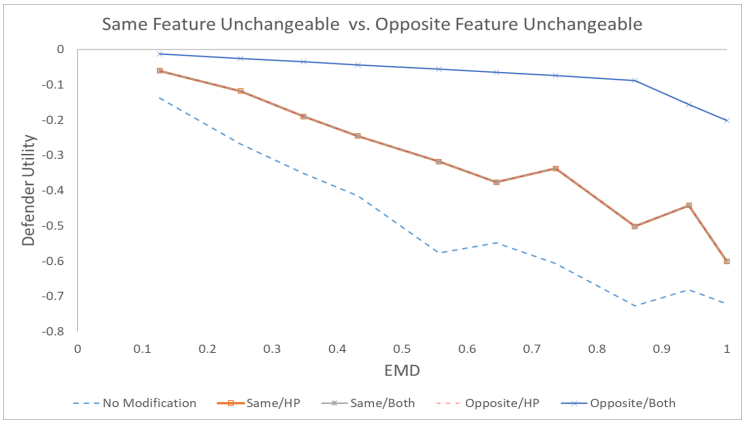

Fig. 6. Comparison of defender utility when some features cannot be modified.

are different for the real and honeypot hosts, we see a very different situation. In this case the defender benefits greatly from being able to use two-sided deception, since he can avoid the constraints by modifying either the real or fake hosts as needed.

In our next experiment, we investigate less extreme differences in the costs of modifying features. We set the costs so that they are increasingly different for real and honeypot hosts, so modifying one feature is cheap for one but expensive for the other, but not impossible. We show the results of using either one or two-sided deception for varying levels of initial feature distribution similarity in Figure 7. The specific costs are given in Table 1. We see that there is very little difference when the initial distributions are similar; this is intuitive since the attacker has little information and deception is not very valuable in these cases. However, we see a large difference when the initial distributions are infor-

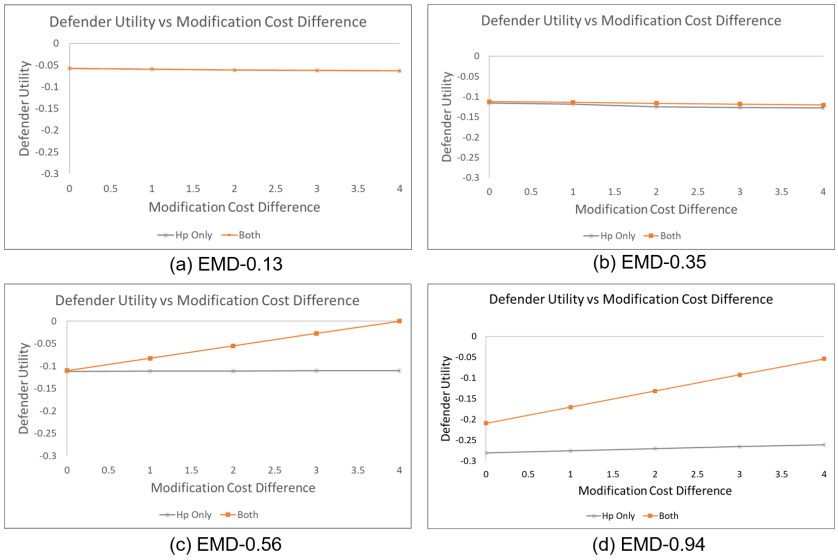

Fig. 7. Impact of modification cost over various initial similarity parameters.

mative. As the difference in the feature modification costs increases, the value of two-sided deception increases, indicating that this asymmetry is crucial to understanding when two-sided deception is necessary to employ effective deception tactics.

We also expect that the number of features available to the players will have a significant impact on the value of deception. While the current optimal solution algorithm does not scale well, we can evaluate the differences between small numbers of features, holding all else equal. Figure 8 presents the results of the modeling HFSG with variable number of features.We found that when the number of features is increased two-sided deception becomes more effective than one-sided deception. The defender in this case has more opportunity to alter the network by changing the features and make it the more confusing network to the attacker. However, the defender payoff decreases with more features due to the constraint on how many features he can modify and the total cost of modifying these features.

4.4 Deception with Naïve Attackers

The previous empirical results all assumed a cautiously rational attacker who actively avoided attacking honeypots. This is a common practice, because fully rational actors present the highest threat. In cybersecurity, these fully rational attackers might be an experienced hacker or APT. However, these are not the only threats faced in cybersecurity and we cannot assume that these attacking agents are always cautious and stealthy.

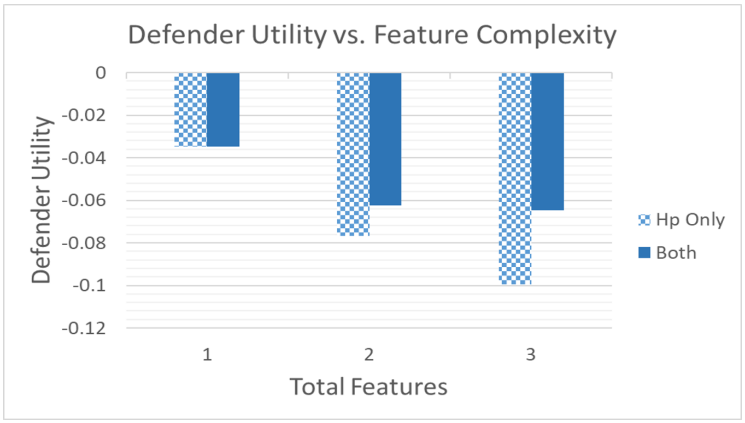

Fig. 8. Comparison of defender utility when increasing the number of features.

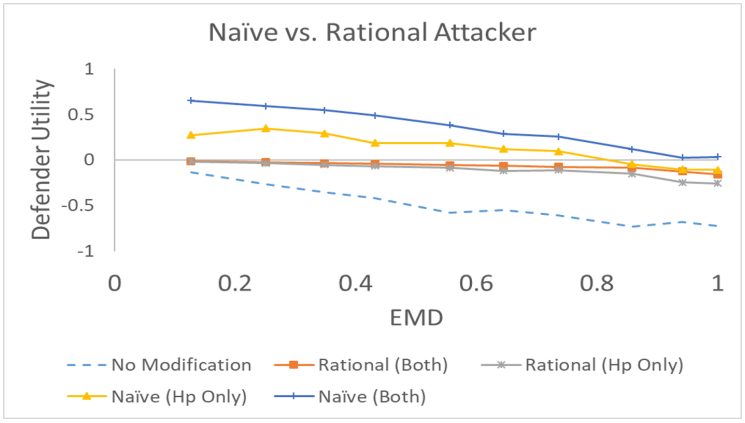

Fig. 9. Comparison of defender utility of a naïve attacker versus a fully rational attacker. Here, the naïve attacker does not consider the defender's utility or strategy at all.

We now consider a more naïve attacker that does not consider the defender's deception. He observes the hosts on the network and assumes no modifications were made. Based on all observations for a particular network he calculates his best response, but does predict the defender's optimal strategy. The results of the experiment are shown in Figure 9 and the costs given in Table 1.

The best case is when the defender can perform two-sided deception against a naïve attacker and the worst case is when the defender performs no deceptive actions against a fully rational attacker. These two cases form an upper- and lower-bound as seen in Figure 9. Two-sided deception is more effective in this case when the feature distributions are similar, while the opposite was true for a rational attacker. Overall, deception strategies are much more effective against naïve attackers.

5 Conclusions

Deception is increasingly becoming a crucial tool for both attackers and defenders in cybersecurity domains, but formal models do not give much guidance on how effective deception can be, and how much effort should be given to disguise deceptive objects such as honeypots. We present a formal game-theoretic model of this problem, capturing the key problem of disguising deceptive objects among real objects when external characteristics may be observed by an attacker.

Our model of HFSG allows us to investigate many aspects of how a defender should make efforts to conceal honeypots. We show that deception can be a highly effective tactic. However, informative signals can limit the effectiveness of deceptive objects. When the costs of modifying these signals are asymmetric, two-sided deception can dramatically improve the effectiveness of the honeypots. In addition, we show that the presence of naive attackers can make the use of deception even more effective.

References

1. Achleitner, S., La Porta, T., McDaniel, P., Sugrim, S., Krishnamurthy, S.V., Chadha, R.: Cyber deception: Virtual networks to defend insider reconnaissance. In: Proceedings of the 8th ACM CCS international workshop on managing insider security threats. pp. 57–68. ACM (2016)
2. Alpcan, T., Başar, T.: Network security: A decision and game-theoretic approach (2010)
3. Bringer, M.L., Chelmecki, C.A., Fujinoki, H.: A survey: Recent advances and future trends in honeypot research. International Journal of Computer Network and Information Security **4**(10), 63 (2012)
4. Carroll, T.E., Grosu, D.: A game theoretic investigation of deception in network security. Security and Communication Networks **4**(10), 1162–1172 (2011)
5. Center, M.I.: Apt1: Exposing one of china's cyber espionage units. Mandiant, Tech. Rep (2013), https://www.fireeye.com/content/dam/fireeye-www/services/pdfs/mandiant-apt1-report.pdf

6. De Gaspari, F., Jajodia, S., Mancini, L.V., Panico, A.: Ahead: A new architecture for active defense. In: Proceedings of the 2016 ACM Workshop on Automated Decision Making for Active Cyber Defense. pp. 11–16. ACM (2016)
7. Du, M., Li, Y., Lu, Q., Wang, K.: Bayesian game based pseudo honeypot model in social networks. In: International Conference on Cloud Computing and Security. pp. 62–71. Springer (2017)
8. Garg, N., Grosu, D.: Deception in honeynets: A game-theoretic analysis. In: 2007 IEEE SMC Information Assurance and Security Workshop. pp. 107–113. IEEE (2007)
9. Kiekintveld, C., Lisy, V., Pibil, R.: Game-theoretic foundations for the strategic use of honeypots in network security. Advances in Information Security **56**, 81–101 (2015)
10. Krawetz, N.: Anti-honeypot technology. IEEE Security & Privacy **2**(1), 76–79 (2004)
11. Kroer, C., Sandholm, T.: Extensive-form game abstraction with bounds. In: Proceedings of the fifteenth ACM conference on Economics and computation. pp. 621–638. ACM (2014)
12. La, Q.D., Quek, T.Q., Lee, J., Jin, S., Zhu, H.: Deceptive attack and defense game in honeypot-enabled networks for the internet of things. IEEE Internet of Things Journal **3**(6), 1025–1035 (2016)
13. Laszka, A., Vorobeychik, Y., Koutsoukos, X.D.: Optimal personalized filtering against spear-phishing attacks. In: AAAI (2015)
14. Mairh, A., Barik, D., Verma, K., Jena, D.: Honeypot in network security: a survey. In: Proceedings of the 2011 international conference on communication, computing & security. pp. 600–605. ACM (2011)
15. Monge, G.: Mémoire sur la théorie des déblais et des remblais. Histoire de l'Académie Royale des Sciences de Paris (1781)
16. Nawrocki, M., Wählisch, M., Schmidt, T.C., Keil, C., Schönfelder, J.: A survey on honeypot software and data analysis. arXiv preprint arXiv:1608.06249 (2016)
17. Píbil, R., Lisỳ, V., Kiekintveld, C., Bošanskỳ, B., Pěchouček, M.: Game theoretic model of strategic honeypot selection in computer networks. In: International Conference on Decision and Game Theory for Security. pp. 201–220. Springer (2012)
18. Provos, N.: Honeyd-a virtual honeypot daemon. In: 10th DFN-CERT Workshop, Hamburg, Germany. vol. 2, p. 4 (2003)
19. Rowe, N.C., Custy, E.J., Duong, B.T.: Defending cyberspace with fake honeypots. JOURNAL OF COMPUTERS **2**(2), 25 (2007)
20. Schlenker, A., Thakoor, O., Xu, H., Fang, F., Tambe, M., Tran-Thanh, L., Vayanos, P., Vorobeychik, Y.: Deceiving cyber adversaries: A game theoretic approach. In: AAMAS (2018), http://dl.acm.org/citation.cfm?id=3237383.3237833
21. Schlenker, A., Xu, H., Guirguis, M., Kiekintveld, C., Sinha, A., Tambe, M., Sonya, S., Balderas, D., Dunstatter, N.: Don't bury your head in warnings: A game-theoretic approach for intelligent allocation of cyber-security alerts (2017)
22. Serra, E., Jajodia, S., Pugliese, A., Rullo, A., Subrahmanian, V.: Pareto-optimal adversarial defense of enterprise systems. ACM Transactions on Information and System Security (TISSEC) **17**(3), 11 (2015)
23. Shi, L., Zhao, J., Jiang, L., Xing, W., Gong, J., Liu, X.: Game theoretic simulation on the mimicry honeypot. Wuhan University Journal of Natural Sciences **21**(1), 69–74 (2016)
24. Spitzner, L.: Honeypots: tracking hackers, vol. 1. Addison-Wesley Boston (2002)
25. Stoll, C.: The cuckoo's egg: tracking a spy through the maze of computer espionage. Doubleday (1989)

26. Virvilis, N., Vanautgaerden, B., Serrano, O.S.: Changing the game: The art of deceiving sophisticated attackers. In: 2014 6th International Conference On Cyber Conflict (CyCon 2014). pp. 87–97. IEEE (2014)
27. Wang, K., Du, M., Maharjan, S., Sun, Y.: Strategic honeypot game model for distributed denial of service attacks in the smart grid. IEEE Transactions on Smart Grid **8**(5), 2474–2482 (2017)
28. Wang, W., Zeng, B.: A two-stage deception game for network defense. In: Decision and Game Theory for Security (2018)
29. Zou, C.C., Cunningham, R.: Honeypot-aware advanced botnet construction and maintenance. In: International Conference on Dependable Systems and Networks (DSN'06). pp. 199–208. IEEE (2006)