Manipulating Adversary's Belief: A Dynamic Game Approach to Deception by Design for Proactive Network Security

Karel Horák[1](✉), Quanyan Zhu[2], and Branislav Bošanský[1]

[1] Department of Computer Science, Faculty of Electrical Engineering,
Czech Technical University in Prague, Prague, Czech Republic
{horak,bosansky}@agents.fel.cvut.cz

[2] Department of Electrical and Computer Engineering, New York University,
New York, USA
quanyan.zhu@nyu.edu

Abstract. Due to the sophisticated nature of current computer systems, traditional defense measures, such as firewalls, malware scanners, and intrusion detection/prevention systems, have been found inadequate. These technological systems suffer from the fact that a sophisticated attacker can study them, identify their weaknesses and thus get an advantage over the defender. To prevent this from happening a proactive cyber defense is a new defense mechanism in which we strategically engage the attacker by using cyber deception techniques, and we influence his actions by creating and reinforcing his view of the computer system. We apply the cyber deception techniques in the field of network security and study the impact of the deception on attacker's beliefs using the quantitative framework of the game theory. We account for the sequential nature of an attack and investigate how attacker's belief evolves and influences his actions. We show how the defender should manipulate this belief to prevent the attacker from achieving his goals and thus minimize the damage inflicted to the network. To design a successful defense based on cyber deception, it is crucial to employ strategic thinking and account explicitly for attacker's belief that he is being exposed to deceptive attempts. By doing so, we can make the deception more believable from the perspective of the attacker.

1 Introduction

As computer systems and devices are becoming increasingly connected and complex in their functionalities, traditional cyber defense technologies (e.g. firewalls, malware scanners, and intrusion detection/prevention systems) have been found inadequate to defend critical cyber infrastructures [23]. Moreover, sophisticated adversaries such as the advanced persistent threats (APTs), can use a combination of social engineering and software exploits to infiltrate the network and inflict cyber and/or physical damages of the defended systems. Therefore, to defend against a sophisticated adversary, we have to accept that the adversary

© Springer International Publishing AG 2017

S. Rass et al. (Eds.): GameSec 2017, LNCS 10575, pp. 273–294, 2017.
DOI: 10.1007/978-3-319-68711-7_15

can study and evade technology-based defenses [20,25]. To move away from the defense paradigm where the attacker has the advantage to the one of defender's advantage, proactive cyber defense is a new defense mechanism in which systems strategically engage the attacker and learn and influence his behaviors.

Cyber deception is a key component of the proactive cyber defense that can create and reinforce attacker's view of the network by revealing or concealing artifacts to the attacker. The attacker needs to pay attention to identifying deceptive artifacts in order to devise the right attack sequence. This becomes challenging in an adversarial environment and the attacker's progress thus becomes slower and less effective. Deception mechanisms, such as honeypots [22,32], honeytokens [3,17], camouflaging [21,28] and moving target defense [12,13,29] are methods that have been used to manipulate the attacker's belief on system parameters and increase their cost of information acquisition.

Understanding deception in a quantitative framework is pivotal to provide rigor, predictability, and design principles. To this end, we analyze deception through a game-theoretic framework [2,16,19,30]. This framework allows making quantitative, credible predictions, and enables the study of situations involving free choice (the option to deceive or not to deceive) and well-defined incentives. Specifically, the class of dynamic games of incomplete information allows modeling the multi-round interactions between an attacker and a defender as well as the information asymmetry that forms the essential part of deception.

In this work, we focus on the applications of cyber deception techniques in the field of network security. Strong proactive incident response strategies can only be devised if we understand the impact of deceptive operations on the attacker's beliefs. To this end, we employ the framework of competitive Markov models with imperfect information, or *partially observable stochastic games* [10,11], to reason about the uncertainties of the two sides of the cyber warfare—the defender of the network and the attacker—and understand how this uncertainty influences their behavior. This framework provides a mathematical formalism of the attacker's belief state to capture his level of engagement and allows the defender to take defensive actions based upon attacker's state of mind.

When the presence of the attacker in the network environment is detected by the sensing systems, the defender can attempt to engage the attacker and start *actively* deceiving him by taking proactive deceptive (and defensive) actions aimed to combat the upcoming attack scenario. He can use the sensing systems to track attacker's further progress and often, by inspecting the log records and/or analyzing the past communication with attacker's command and control servers [5,9], he can also reconstruct a significant part of the history of the attack – thus getting a near-perfect information about the attacker's point of view. We assume that the defender can reconstruct this view *perfectly* which allows us to apply the framework of *one-sided partially observable stochastic games* [11].

To make the deception effective in the long run, we need to make it difficult for the attacker to identify that the deception is employed. An attacker will try to reason about and recognize our deceptive attempts and will adapt his attack plan accordingly—and thus mitigate the impact of the deception. We provide a model which explicitly reasons about attacker's belief about the deception state and we show how important it is for the defender to carefully manipulate this belief to maximize the defensive impact of the cyber deception. We conduct a case study to illustrate the consequences of strategic deception on the security level of the network. Namely, we make the following important observations about cyber deception. First, we observe that the standard incident-response approach which relies on excluding the attacker from the network immediately is inefficient from the perspective of the deception. In fact, it may render the network more vulnerable as it does not take attacker's beliefs into account (we term this phenomenon as the *curse of exclusion*). Second, we observe that it is easier to deceive the attacker when he had already dedicated significant effort to accomplish his goals as he is more greedy about realizing his intents (we term this phenomenon as the *demise of the greedy*).

The rest of the paper is organized as follows. In Sect. 2, we introduce related work on cyber deception and introduce the game-theoretic framework we use. In Sect. 3, we provide a generic approach for reasoning about the deception which accounts for the necessary aspects of the deception, i.e. the informational asymmetry, sequential nature of deception problems and which accounts for the strategic nature of the deception. In Sect. 4, we state the problem from the perspective of cyber deception in network security. Next, we provide a case study illustrating the impact of cyber deception on attacker's beliefs and his ability to inflict damage in Sect. 5. Finally, in Sect. 6 we summarize our main results.

2 Related Work

Typical attacks conducted by advanced attackers consist of multiple stages [24] that can be broadly summarized as *reconnaissance* and realization of attacker's primary goals, e.g. data exfiltration. Underbrink [26] classifies deception techniques into two broad categories – *passive* and *active deception*. The passive deception is targeted against attacker's reconnaissance efforts and relies on a proactively deployed *static* infrastructure of decoy systems, e.g. honeypots [14,22] or fake documents [4]. Unlike the legitimate users, the attacker does not know about their deceptive nature and may thus reveal his presence by inadvertently interacting with them. The active deception, on the other hand, attempts to *interactively* engage the attacker who has been already detected by the sensing systems. The defender attempts to anticipate probable future actions of the attacker and takes proactive countermeasures against them to prevent the attacker from achieving his goals.

A lot of work has been dedicated to understanding both technological [1,27] and strategical [6,18,31,32] aspects of passive deception techniques and decoy infrastructures. Considerably less attention has been, however, paid to the active deception. To the best of our knowledge, very few works have focused on the strategical aspects of active deception. [26] has introduced the concept of active deception and the *Legerdemain* approach to active deception was described. The Legerdemain approach secretly manipulates critical assets in the network (such as data files or access credentials) to confuse the attacker and prevent him from getting access to critical resources. A dynamic game model, based on two coupled Markov decision processes, is used to assist the defender in designing the actively deceptive strategy. The model, however, assumes that the attacker will never realize that mechanisms of active deception are applied against him – which simplifies solving the game but makes the model not realistic. In fact, we show that accounting for attacker's belief about the deception is critical for designing strong deceptive strategies.

Our approach reasons explicitly about the belief the attacker has and thus avoids the drawback of the Legerdemain approach. To this end, we use the framework of *one-sided partially observable stochastic games* (one-sided POSGs) [11]. In this class of games, one of the players is assumed to be perfectly informed about the course of the game, which is not the case for the other player. This game-theoretic model has been originally devised to reason about robust defensive strategies by assuming that attacker is able to get a perfect picture of the game. In this work, we provide a novel application of this model to reason about the active deception by assuming that the defender (or deceiver) has already detected the attacker (and thus is able to track his progress) while the attacker (or deceivee) lacks some information about the game (and thus is vulnerable to defender's deceptive attempts). We discuss the way we use this class of games to reason about deception in Sects. 3 and 4 in greater detail.

3 Deception Game Framework

The asymmetry of information plays a major role in many conflicts seen in the real world, starting from the warfare and ending with conflicts as innocent as card games. The success in these operations typically depends on the way we handle the information and in particular on the way we protect our informational advantage. Deception has even evolved to be vital for the survival of many wildlife species, such as chameleons, and has been adopted by armies worldwide.

We cannot, however, expect that a simple presence and naïve use of the deceptive techniques is sufficient to guarantee success – the way we employ them is important to explore. As an example, consider that we have two colored balls, red and blue, and we do not want others to know which one of them we are carrying. To this end, it may seem reasonable to paint each of these balls to the opposite color beforehand and pretend that the red one is, in fact, blue (and vice versa). In such a case, however, other actors will soon discover the principle we use to manipulate the truth and realize that the ball we are carrying is in fact of the opposite color – and hence our attempt to disguise others becomes unsuccessful.

When deciding on the use of deceptive techniques we have to think in a strategic way. We need to understand what impact our *deception strategy* σ_D has on the beliefs of other actors as they will learn and eventually understand the way we misrepresent the truth. The deceived players will derive a *counter-deception strategy* σ_A with the aim to understand the signals they receive and reconstruct the truth (or at least reconstruct how likely each possibility is to be true). Both of these strategies have to account for the beliefs of the players and are thus essentially functions of these beliefs.

We focus on the deception problems where there are two sides of the conflict (or two players). We assume that one side of the conflict, the *deceiver*, knows the truth (i.e. state s of the system), while the other side, the *deceivee*, aims to recognize that. This type of knowledge is often seen in reality. For example, in security problems, the defender usually knows the parameters of the system he is about to defend, e.g., he knows the plans of the facility or the topology of the computer network, and he knows where the important assets are located. In addition, he is equipped with monitoring facilities which allow him to monitor attacker's actions (or, at least, allow him to analyze these actions retrospectively). On the other hand, the attacker is uninformed about the true system parameters and he has to recognize these parameters to plan his activities properly. This setting underlies the need for reasoning about the information and beliefs of the uninformed player as the information is the only advantage we have.

3.1 Deception in a Sequential Setting

We study the deception in a sequential setting, where both the players take sequences of actions to either deceive the adversary, or attempt to recognize the truth, respectively. In each step $t \geq 1$, both the deceiver and the deceivee take an action (a_D and a_A). As a matter of result, the deceivee gets an observation about the true state of the system (e.g. that the ball is painted red) and the state of the system may change (which is then known only to the deceiver again). Moreover the deceiver has to pay a cost associated with his deceptive action and possibly other costs associated with the choice of actions a_D and a_A, denoted $l^{(t)}$. We characterize these costs using a loss function $\mathcal{L}_D$.

The goal of the deceiver is to keep the losses $l^{(t)}$ as low as possible – or at least mitigate them by delaying them in time. This is characterized by the *discounted-sum* objective when the aggregated loss of the deceiver is

$$L = \sum_{t=1}^{\infty} \gamma^{t-1} \cdot l^{(t)}, \tag{1}$$

where $0 < \gamma < 1$ is a constant termed the *discount factor*. In our case, the deceiver is the defender of the system and we aim to devise robust deceptive strategies that account for the worst case scenario, hence we assume that the goal of the deceivee is to maximize the loss L. We also term such games as *zero-sum*.

We aim to understand the *value of deception* $\overline{V}$ and the *value of counter-deception* $\underline{V}$ – and the strategies that induce these values. We define $\overline{V}$ as the expected loss of the deceiver when he is forced to commit himself to a deception strategy σ_D which is then observed by the deceivee who tries to identify the weaknesses of σ_D, i.e.

$$\overline{V} = \inf_{\sigma_D} \sup_{\sigma_A} L(\sigma_D, \sigma_A) \tag{2}$$

where $L(\sigma_D, \sigma_A)$ stands for the expected discounted loss when strategies σ_D and σ_A are followed by the players. Similarly, we define the value of counter-deception $\underline{V}$ as the value where the deceivee is forced to commit himself first to a counter-deceptive strategy σ_A he uses to combat the deception and then the deceiver decides what deceptive techniques he uses, i.e.

$$\underline{V} = \sup_{\sigma_A} \inf_{\sigma_D} L(\sigma_D, \sigma_A). \tag{3}$$

Note that the deceiver can guarantee that the loss will be no higher than $\overline{V}$, while $\underline{V}$ is the minimum loss the deceivee can enforce.

3.2 Game-Theoretic Model

We propose to formulate deception as a partially observable stochastic game with one-sided information (one-sided POSG) [11]. This model has been originally devised to reason about robust strategies of the defender by assuming that the adversary is perfectly informed. The asymmetric nature of the information present in the model, however, makes it convenient to reason about the deception. A deception game based on the model of one-sided POSGs is a tuple $\langle \mathcal{S}, \mathcal{A}_A, \mathcal{A}_D, \mathcal{T}, \mathcal{L}_D, \mathcal{O}_A, b^0 \rangle$, where

- $\mathcal{S}$ is a finite set of states of the system (recall that the true state of the system is known to the deceiver, while the deceivee does not know it). A state may for example represent where both the players have deployed their units in a warfare.
- $\mathcal{A}_D$ is a finite set of actions the deceiver can use to deceive the adversary.
- $\mathcal{A}_A$ is a finite set of actions the adversary, the deceivee, can use to learn more about the system, or potentially in security problems to inflict damage.
- $\mathcal{T} : (\mathcal{S} \times \mathcal{A}_A \times \mathcal{A}_D) \to \Delta(\mathcal{O} \times \mathcal{S})$ is a transition function representing possible changes to the system (e.g. movements of the units) and observations the deceivee can receive in a probabilistic way.
- $\mathcal{L}_D : (\mathcal{S} \times \mathcal{A}_A \times \mathcal{A}_D) \to \mathbb{R}$ is defender's loss function and describes how much the defender loses in each step of the deception game.
- $\mathcal{O}_A$ is a finite set of observations the attacker can get about the state of the system.
- $b^0 \in \Delta(\mathcal{S})$ (where $\Delta(\mathcal{S})$ is a probability distribution over $\mathcal{S}$) is the initial belief of the deceivee, where $b_0(s)$ denotes the probability that the initial state of the deception game is s. As an example, the deceivee may know where his units are located, but he may lack the information about the position of deceiver's units. Thus he forms a belief over possible positions of the deceiver in the form of a probability distribution over states that match the current (known) position of units of the deceivee.

A play in the deception game proceeds as follows. First, an initial state of the game s^0 is drawn from b^0. Then, in each step t, players decide simultaneously their actions $(a^t_D, a^t_A) \in \mathcal{A}_D \times \mathcal{A}_A$. Based on their choice, the deceiver loses $l^{(t)} = \mathcal{L}_D(s^{t-1}, a^t_A, a^t_D)$. Then the deceivee receives an observation o^t and the game state changes to s^t with probability $\mathcal{T}(s^{t-1}, a^t_A, a^t_D)(o^t, s^t)$.

Deceiver observes the course of the deception game perfectly, hence he knows what the past states, actions and observations were. He can use all this information to make an informed decision about his next action. He makes this decision based on his deception strategy $\sigma_D : (\mathcal{S}\mathcal{A}_D\mathcal{A}_A\mathcal{O}_A)^*\mathcal{S} \to \Delta(\mathcal{A})$, where $\sigma_D(\omega, a_D)$ denotes the probability that the deceiver chooses an action $a_D \in \mathcal{A}_D$ when the current history is ω.

The deceivee only observes the observations o^t and remembers the actions a^t_A he made. He cannot thus make use of the complete information available to the deceiver. The attacker thus proceeds according to a counter-deception strategy $\sigma_A : (\mathcal{A}_A\mathcal{O})^* \to \Delta(\mathcal{A}_A)$, when $\sigma_A(\omega, a_A)$ stands for the probability that the deceivee uses action a_A given that $\omega \in (\mathcal{A}_A\mathcal{O})^*$ are the actions and observations he has used and seen previously.

The results in [11] show that the players need not remember the histories of the play to make decisions. Instead, they can just keep track of the *belief* $b \in \Delta(\mathcal{S})$ over the states $\mathcal{S}$ of the deception game and play according to one-step strategies $\pi^{(b)}_D : \mathcal{S} \to \Delta(\mathcal{A}_D)$ and $\pi^{(b)}_A \in \Delta(\mathcal{A}_A)$ which are directly functions of beliefs. This emphasizes the fact that the deceivee forms a belief which then directly drives his decisions. The players keep track of the belief using a Bayesian update rule characterized by the following equation:

$$\tau(b, a_A, o, \pi_D)(s') = \frac{1}{K} \sum_{s a_D \in \mathcal{S}\mathcal{A}_D} b(s) \cdot \pi_D(s, a_D) \cdot \mathcal{T}(s, a_A, a_D, o, s') \qquad (4)$$

where $\tau(b, a_A, o, \pi_D)$ stands for the updated belief of the deceivee given that the previous belief was b, he played action a_A and received observation o, and the deceiver followed a deception strategy π_D. K stands for the normalization constant.

In the case of zero-sum deception game, the *value of deception* $\overline{V}$ and the *value of counter-deception* $\underline{V}$ have been shown to be equal [11], i.e.

$$\inf_{\sigma_D} \sup_{\sigma'_A} L(\sigma'_A, \sigma_D) = \sup_{\sigma_A} \inf_{\sigma'_D} L(\sigma_A, \sigma'_D). \qquad (5)$$

We represent the values of deception (or counter-deception) using a convex value function $v^* : \Delta(\mathcal{S}) \to \mathbb{R}$ which maps beliefs over the system states to the expected value of deception for that belief. This value function satisfies the following fixpoint equation

$$v^*(b) = \min_{\pi_D : \mathcal{S} \to \Delta(\mathcal{A}_D)} \max_{\pi_A \in \Delta(\mathcal{A}_A)} \Big[\sum_{s a_A a_D} b(s) \cdot \pi_A(a_A) \cdot \pi_D(s, a_D) \cdot \mathcal{L}_D(s, a_A, a_D) + \gamma \sum_{a_A o} \Pr[a_A o \,|\, b, \pi_A, \pi_D] \cdot v^*(\tau(b, a_A, o, \pi_D)) \Big]. \qquad (6)$$

One of the ways to reason about the value of the deception and the associated optimal strategies of the players is to approximate the value function v^* using an approximate value iteration algorithm presented in [11]. We can then derive the optimal strategy for the deceiver by considering the maximizing π_D of Eq. (6) in each step of the interaction.

Remark 1. The convexity of the value function v^* supports our intuition that the deceivee never gets satisfied with being deceived. The value of his counter-deception would never get lower, had he got additional information. For example, assume that the deceivee recognized the true state of the system before he is about to act (i.e. his belief changes from b to b_s, where b_s is a belief where the attacker *knows* the true state). Then, since $b = \sum_{s\in\mathcal{S}} b(s) \cdot b_s$ and due to the convexity of v^*, we get

$$\sum_{s\in\mathcal{S}} b(s) \cdot v^*(b_s) \geq v^* \left(\sum_{s\in\mathcal{S}} b(s) \cdot b_s \right), \tag{7}$$

i.e. if the attacker recognizes the true state (i.e. with probability $b(s)$ he recognizes that the true state is s) and plays accordingly, the loss he is able to cause is greater or equal than in the situation where he has to reason about the state he is in (i.e. his belief is b).

4 Game-Theoretic Approach to Cyber Deception

The ideas we have presented so far are general enough to be applied to reason about the deception in a wide range of scenarios. We are going, however, to focus on the use of the deception in the context of computer networks to improve the security of networked systems. The deception over the networks possesses certain features which allow us to make the model of deception game more specific. Namely, the attacker who is going to be deceived does not know two key properties of the networked system. First, he does not know the topology of the network which he needs to understand to target his attack properly. Second, he does not know whether the defender, the deceiver, already knows about his presence in the network. Understanding both of these aspect is critical from attacker's perspective – and concealing this information from attacker's view is important for the defender to devise strong defensive strategies.

In this section, we describe a general idea how we can use one-sided partially observable stochastic games to reason about active deception in network security, where the defender interactively decides about the actions to mislead the attacker in the course of an attack and mitigate the possible damage to the network. Our model accounts for the uncertainties of the attacker about the topology of the network, whether he has been detected and about defender's actions – both past and upcoming ones. To this end, we represent the states of the game as $\mathcal{S} = \mathcal{N} \times \mathcal{X}_A \times \mathcal{X}_D \times \mathcal{D}$ where

- $\mathcal{N}$ is a set of possible network topologies the defender can choose from based on a fixed distribution $\xi \in \Delta(\mathcal{N})$

- $\mathcal{X}_A$ is a set of possible attack vectors representing the state of an attack (e.g. privileges the attacker has already acquired); $\emptyset \in \mathcal{X}_A$ denotes that the attack has not started yet
- $\mathcal{X}_D$ is a set of possible defense vectors representing the state of defense resources (e.g. dynamic decoy systems deployed in the network); $\emptyset \in \mathcal{X}_D$ denotes that the defender has not deployed any dynamic resources yet
- $\mathcal{D}$ is a set of possible detection states; we assume $\mathcal{D} = \{\texttt{true}, \texttt{false}\}$ denoting whether the attacker has been detected or not by the sensing systems

We denote a state of the game as (n, x_A, x_D, d).

The defender initially chooses a network topology he is going to defend according to a probability distribution $\xi \in \Delta(\mathcal{N})$. We then derive the initial belief of the underlying one-sided POSG $b_0 \in \Delta(\mathcal{S})$ as $b(n, \emptyset, \emptyset, \texttt{false}) = \xi(n)$ and $b_0(\cdot) = 0$ otherwise. This means that we draw the initial network topology from $\xi(n)$ and make both the attack and defense vectors empty, and the attacker is initially undetected.

Once the attacker gets detected by the sensing systems (i.e. $d = \texttt{true}$), the defender may start taking actively actions $a_D \in \mathcal{A}_D$ to combat the attacker's presence in the network. His actions may manipulate the defense vector (e.g. by deploying new defense resources), interfere with actions of the attacker, or they may restrict attacker's access to the network (defense action $\texttt{block} \in \mathcal{A}_D$). We assume that in such case, the attacker is able to change his identity and attack the network again (therefore x_A is set to $\emptyset$ and d to $\texttt{false}$ as we lost track of the attacker when he changed his identity, and the game continues). If the attacker has not been detected yet, however, the defender cannot take any active counteraction (i.e. active deception techniques are not available to him) and he is forced to use action $a_D = \texttt{noop}$. The fact that the defender cannot use any action other than $\texttt{noop}$ when the attacker has not been detected yet allows us to assume a perfect information of the defender, i.e. make the defender be the perfectly informed player in the one-sided partially observable stochastic game. The defender cannot leverage the extra information about the attacker (he would not have in reality) up to the point when the attacker gets detected.

The attacker can choose from attempting to acquire new privileges (and thus manipulating the attack vector x_A), changing his identity (i.e. making $x_A = \emptyset$ and $d = \texttt{false}$) and leveraging his current privileges to cause damage — or combination of any of these. Each action of the attacker is associated with the risk of alerting the defender, we denote the probability of triggering an alert when using action a_A in network n by $p_{trig}(n, a_a)$.

The transition function $\mathcal{T}$ respects the actions the players have taken, i.e. describes possible changes to vectors x_A, x_D and the detection state d in a probabilistic way. Furthermore the attacker receives an observation $(x'_A, o) \in \mathcal{O}_A$. The attacker is always aware of his current attack vector, i.e. for any $x'_A \neq x''_A$ the following holds

$$\mathcal{T}((n, x_A, x_D, d), a_A, a_D)((x'_A, o), (n, x''_A, x'_D, d')) = 0. \tag{8}$$

Moreover, once the network topology is chosen it never be changes, i.e. for any $n \neq n'$

$$\mathcal{T}((n, x_A, x_D, d), a_A, a_D)((x'_A, o), (n', x'_A, x'_D, d')) = 0. \tag{9}$$

The detection probabilities (i.e. the probability of transitioning from $d = \texttt{false}$ to $d' = \texttt{true}$) are independent of action effects, i.e.

$$\textstyle\sum_{(x'_A,o)} \mathcal{T}((n, x_A, \emptyset, \texttt{false}), a_A, \texttt{noop})((x'_A, o), (n, x'_A, \emptyset, \texttt{true})) = p_{trig}(n, a_A) \tag{10}$$

The losses $\mathcal{L}_D$ for individual transitions can be set arbitrarily to match the costs (and eventually possible gains if we succeed in exploiting attacker's actions) in the real network and the costs of the deception. We only require that $\mathcal{L}_D((n, x_A, \emptyset, \texttt{false}), a_A, a_D) = M$ for every $a_D \neq \texttt{noop}$ where M is a large constant to ensure that the defender does not use active deception techniques when the attacker has not yet been detected.

5 Manipulating Attacker's Belief Using Active Deception

The use of the active deception can significantly improve the security level of the network. In this section, we provide a case study based on a simple game with sets $\mathcal{N}$ and $\mathcal{X}_D$ containing only one element (i.e., $\mathcal{N} = \{n\}$ and $\mathcal{X}_D = \{\emptyset\}$) to illustrate the concept of active deception. In the case of this game, we use the deception only to manipulate attacker's belief over being detected (i.e. the $\mathcal{D}$ part of the state) and we try to make him uncertain about the progress of the attack and eventually take a wrong action. We show that we cannot, however, rely solely on the deceptive actions if we want to maximize the effectiveness of the deceptive operation. The deception is the most effective if it is *stealthy* and the attacker remains unaware that we are trying to deceive him, or, at least, if we make him uncertain about the state of deception.

As soon as the attacker realizes that we are trying to deceive him, his behavior changes significantly. He will attempt to take evasive actions in attempt to lose defender's attention (e.g. by changing his network identity), or, as a matter of last resort, he may opt to inflict severe damage based only on the information he collected so far. These decisions of the attacker make the defender's attempts to contain the attack substantially harder and should be averted (if possible).

To preserve the stealthy nature of the deception, it is crucial that the attacker thinks that the signals he receives are not *too good to be true*. The defender has to manipulate attacker's belief about the deception state carefully if he wants to make the attacker believe that no deceptive operation is taking place and keep him *engaged* in the network.

5.1 Network Topology and the Anatomy of an Attack

We illustrate the concept of active deception using a network topology $n \in \mathcal{N}$ depicted in Fig. 1. We use it as an abstraction of a multilayer network which is commonly adopted in critical network operations, such as power plants or production facilities [15]. Our example network consists of three layers. The outermost layer of the network (Layer 1) is directly exposed to the Internet via demilitarized zones (DMZs) and provides less sensitive services that are used to communicate with the customers and business partners, such as web or email servers.

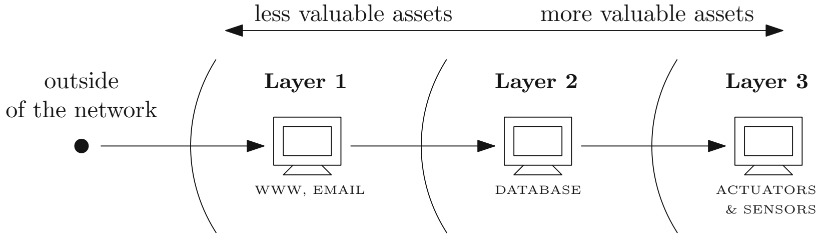

Fig. 1. Network topology (attacker starts outside of the network and attempts to gain access to the most valuable assets in the network)

More critical assets are located in the deeper layers of the network. In our case, the second layer consists of data stores containing confidential data the loss of which may have a severe impact on the company. The third layer is the most critical one since it provides an access to physical devices, such as actuators and sensors, the integrity of which is absolutely essential for the secure operation of the facility. Breach of assets in the Layer 3 of the network may even pose a risk of physical damage, such as in the case of the Stuxnet attack [7,8].

Attack Options. We assume that an attack is initiated from a computer outside of the network ($x_A = \emptyset$). In this section we describe attacker's actions (set of actions $\mathcal{A}_A$) which he can use to acquire new privileges and penetrate deep into the network and to cause damage to it. The attacker attempts to take control of a system in Layer 1 ($x_A = \texttt{layer}_1$) and then escalates his privileges to take control of the computers located deeper in the network (i.e. acquiring $x_A = \texttt{layer}_i$) by compromising them (hence we refer to this action of the attacker as `compromise`). At any point, the attacker can either `wait` or leverage the current access. Apart from attempting to compromise a host in the next layer, he has two options:

The first option is to cause significant immediate damage, such as eliminating a physical device in Layer 3 (having the attacker had access to it) – we refer to this action as `takedown`. Such an action surely attracts the attention of the defender and will lead to the detection of the attacker's presence. Therefore, the attacker is forced to quit the network and possibly repeat his attack later (hence $x_A = \emptyset$ and $d = \texttt{false}$ as a result).

The second option is to cause smaller amount of damage while attempting not to attract defender's attention. The actions the attacker can use to this purpose include, e.g., a stealthy exfiltration of data or a manipulation of the records in the database – for simplicity we refer to them collectively using the `exfiltrate` action. Nevertheless, even these careful options run into a small risk of being detected. Moreover these options run into the risk that the defender will avert the damage resulting from them by means of active deception and possibly even use the fact that the attacker uses the `exfiltrate` action for his benefit (e.g., to collect evidence; see discussion in Sect. 5.2). This makes it critical for the attacker to understand whether he is deceived or not.

Detection System. An intrusion detection system (IDS) is deployed in the network and can identify malicious actions of the attacker. This detection is not reliable. We assume that the attacker's presence is detected with probability $p_{trig}(n, \texttt{compromise}) = 0.2$, if he escalates his privileges and penetrates deeper in the network using the `compromise` action. If the attacker performs stealthy exfiltration of the data (`exfiltrate` action), we detect him with probability $p_{trig}(n, \texttt{exfiltrate}) = 0.1$. We have chosen these probabilities based on a discussion with an expert, however, the model is general enough to account for any choice of these parameters.

Active Deception. We assume that the passive defensive systems, such as IDS and honeypots, are already in place and we focus on the way the defender can *actively* deceive the attacker when his presence has been detected. We take an abstracted view on defender's actions (set $\mathscr{A}_D$) to focus on the main idea of deception, however, our model is general and these actions can be refined to account for *any* actions the defender can use. In our example, he can either use a stealthy deceptive action and attempt to `engage` the attacker in the network, or he can attempt to exclude the attacker from the network (non-deceptive `block` action). We assume that the `block` action really achieves its goal and all the privileges of the attacker get revoked, and the attacker thus has to start his attack from scratch (i.e. x_A becomes $\emptyset$, and $d = \texttt{false}$). If it were not the case and the `block` action was less powerful, blocking the attacker would have been less tempting and hence the use of deception we are advocating would have been even more desirable. By engaging the attacker we attempt to anticipate the action of the attacker and minimize (or even eliminate) the damage caused by his stealthy damaging action of `exfiltrate`. We cannot, however, contain the more damaging `takedown` action by engaging the attacker – the only way to prevent that kind of damage is to `block` the attacker in time. Note that both of these actions of the defender can only be used once the attacker got detected – otherwise, the defender has to rely on the infrastructure of passive defensive systems (i.e., use `noop` action) as the attacker has to be detected first.

5.2 Game Model

We analyze the active deception in the context of the network presented in Sect. 5.1 using a game-theoretic model of one-sided partially observable stochastic games (see Sect. 3) and we capture the interaction between the defender and the attacker using a transition system depicted in Fig. 2. The state space is divided into two parts. In the upper half, the presence of the attacker in the network has not yet been revealed by the IDS ($d = \texttt{false}$), therefore, the defender cannot take active countermeasures yet. Triggering an IDS alert switches the game states into the bottom part ($d = \texttt{true}$) and thus gives the defender an opportunity to decide between `engage` and `block` actions.

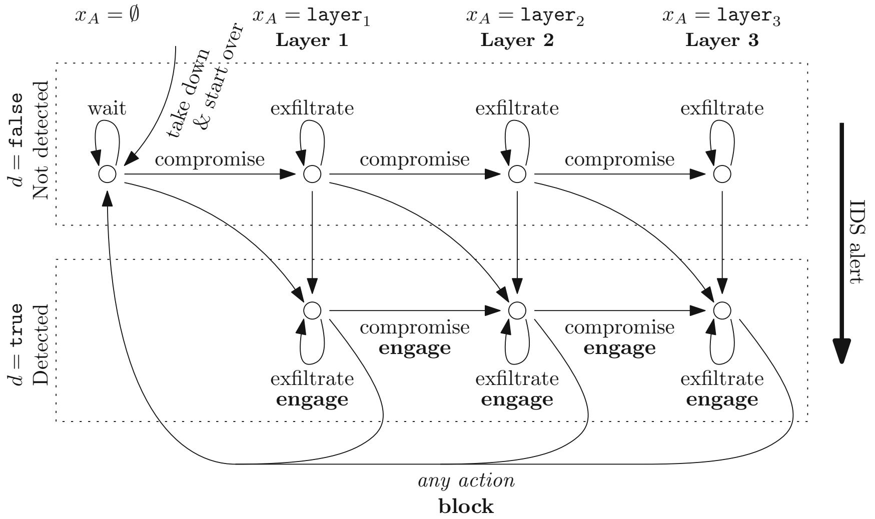

Fig. 2. Transition system of a partially observable stochastic game representing attack on the network from Fig. 1. The attacker can use the `takedown` action in every layer. The `wait` action of the attacker has been omitted for clarity and is always applicable.

The arrows in the diagram represent individual transitions in the game (i.e. represent the transition function $\mathcal{T}$). We assume that the transitions in the game are deterministic, except for the transitions between $d = \texttt{false}$ and $d = \texttt{true}$ that are defined using p_{trig}. The attacker never receives an observation that would reveal him some information about the detection state d (i.e. he only gets to know the new attack vector x_A).

If the attacker uses `compromise` action, he penetrates deeper in the network. If he opts for `exfiltrate`, he stays in the current layer of the network while possibly gaining access to confidential information. And finally, he can decide to do the immediate damage by the `takedown` action at any time. In such a case he gets detected and thus returns to the initial state, outside of the network. The defender can stop all this from happening by taking the `block` action (had he detected the attacker) when the defender is pushed out of the network as well.

The attacker knows his current attack vector x_A and can identify the layer he has penetrated (i.e. he knows the "column" of the transition system where he is located), but he does not know whether he has been detected or not (i.e. whether the game is in the upper or lower half). The defender also does not have perfect information about the state of the attack in reality – namely, he does not know anything about the attacker until the IDS generates an alert. After the alert is generated, however, we assume that he can get a close to perfect information about the attacker by studying the traces he has created in the system. Since the defender cannot make use of the information about the attacker in states where $d = \texttt{false}$ (he cannot take any active countermeasures), we can safely assume that the defender has a *perfect* information in the whole game, which results in a type of information asymmetry we discussed in Sect. 3.

Game Utilities. We associate a loss (or cost) of the defender to each action the attacker performs (i.e. each transition in Fig. 2). Since the attacker takes his actions sequentially, a sequence of costs $l^{(1)}, l^{(2)}, \ldots$ is generated, and we use discounting to obtain the aggregated loss of the defender using the formula

$$L = \sum_{t=1}^{\infty} \gamma^{t-1} \cdot l^{(t)}. \tag{11}$$

The use of discounting (in our case, we use $\gamma = 0.95$) reflects the attacker's impatience during an attack as he does not want to wait forever to achieve his goals as the value of information he can steal diminishes.

Each of the costs $l^{(t)}$ depends on the current state of the attack (what layer the attacker has penetrated and whether he has been detected), the action the attacker performs and the counteraction of the defender (if applicable). Note that in our case, we have just one network n and one defense vector x_D so we do not account for these explicitly. This utility model is general enough to capture any kind of preferences of the defender. The costs we use in our case study are based on a discussion with an expert and are summarized in Table 1. Recall that the players take their action simultaneously and the costs thus depend on their joint action.

The `compromise` action does not cause any immediate harm to the defender and only leaks information to the defender (e.g. about an exploit used) so the loss of the defender is negative ($L_1 = L_7 = -2$). Note that a negative loss is in fact a gain. Moreover, if the defender is already aware of attacker's presence and engages him in the network, he can better understand the techniques used by the attacker and thus his loss is ($L_4 = -4$).

The `exfiltrate` action is already harmful to the defender. If the defender does not take any active countermeasures, the attacker accesses confidential data which implies a significant damage to the defender. Since the assets located deeper in the network are more valuable, we account for this by defining the cost for the defender of $L_2^i = 15i$ for losing data located in the i-th layer.

If the defender realizes that he is dealing with a malicious user, he can minimize or eliminate the risk of losing sensitive data, e.g. by presenting (partly) falsified data to the attacker, using the `engage` action. The attacker then receives useless data and only provides the defender with time to collect the forensic evidence. The loss of the defender is, therefore, negative ($L_5 = -2$) if the attacker

Table 1. Game costs for the game represented in Fig. 2. In each time step, the players take their actions simultaneously and the loss of the defender in the current time step is determined according to their joint action.

State (s^{t-1})		Action		Defender's loss
Attacker's position (x_A)	Detected (d)	Attacker (a_A^t)	Defender (a_D^t)	$\mathcal{L}_D(s^{t-1}, a_A^t, a_D^t)$
any	no	`compromise`	—	-2 $(= L_1)$
`layer`$_i$	no	`exfiltrate`	—	$15i$ $(= L_2^i)$
`layer`$_i$	no	`takedown`	—	$25i$ $(= L_3^i)$
any	yes	`compromise`	`engage`	-4 $(= L_4)$
`layer`$_i$	yes	`exfiltrate`	`engage`	-2 $(= L_5)$
`layer`$_i$	yes	`takedown`	`engage`	$25i$ $(= L_6^i)$
any	yes	`compromise`	`block`	-2 $(= L_7)$
any	yes	`exfiltrate`	`block`	0 $(= L_8)$
any	yes	`takedown`	`block`	0 $(= L_9)$

exfiltrates data while being engaged. The defender can also prevent the data exfiltration by restricting attacker's access to the network (action `block`), however, by doing so, he loses the option to collect the evidence and hence the reward is $L_8 = 0$.

If the attacker decides to cause significant immediate damage by the `takedown` action, the only option of the defender to prevent this from happening is to `block` the attacker (if applicable) when the loss is $L_9 = 0$. Otherwise, the cost for the defender is $L_3^i = L_6^i = 25i$ (when i represents the layer the attacker is in).

5.3 Optimal Defense Strategy

Once the defender succeeds in detecting the presence of the attacker, he can investigate log records to analyze past attacker's actions and estimate his belief about being detected. The defender can make use of this belief to reason about the defensive measures he should apply and to design an optimal defense strategy. We are aware that in real world deployments, accurate tracking of attacker's belief need not be possible and we discuss this in Sect. 5.5.

The optimal defense strategy incurs expected long-term discounted loss of the defender of 282.154. This is a significant improvement over the common practice nowadays of attempting to block the attacker immediately after he is detected. The *always-block* strategy where the defender is restricted to play only `block` action once he detects the attacker leads to an expected loss of 429.375. It is also, however, not good to keep the attacker engaged in the network forever (and try to deceive him by never blocking him, and always use the `engage` action – we refer to this strategy as *always-engage*). Such an approach would not make the deception believable, and the attacker would rather cause the damage and forfeit his current attack attempt, than battle the deception.

We represent the optimal defensive strategy as a mapping from the current position of the attacker (i.e. the layer of the network he penetrated) and his *belief* about being detected (and thus being deceived). Since the defender has only two actions available, we express the probability of playing the `engage` action only (had he succeeded in detecting the attacker), $\sigma_D(i,b)$, where $i \in \{1,2,3\}$ is the current layer and $b \in [0,1]$ is the attacker's belief about the detection state. Note that $\sigma_D(i,b)$ corresponds to $\pi_D((n, \texttt{layer}_i, x_D, \texttt{true}), \texttt{engage})$, where π_D is the minimax solution of Eq. (6) evaluated for $v^*(\hat{b})$, $\hat{b}(n, \texttt{layer}_i, x_D, \texttt{true}) = b$, $\hat{b}(n, \texttt{layer}_i, x_D, \texttt{false}) = 1 - b$. The optimal defense strategy $\sigma_D(i,b)$ for each of the layers is depicted in Fig. 3.

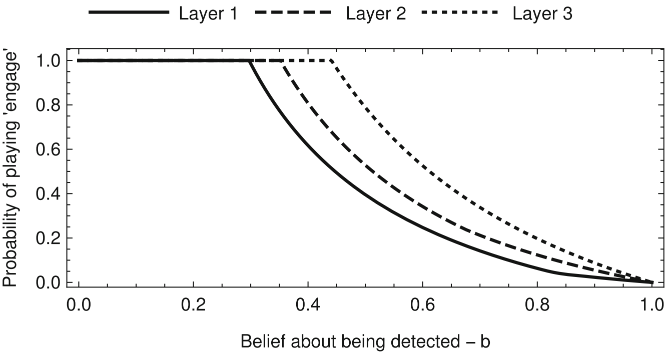

Fig. 3. Optimal defense strategy σ_D for the network from Fig. 1. The optimal strategy of the defender is randomized and depends on the current position of the attacker (the layer he penetrated) and his belief about the detection state.

The optimal defense strategy prescribes the defender to always keep the attacker in the network when the attacker is highly confident that he has not been detected yet. In such a situation, the attacker will opt for data exfiltration, which we can prevent, e.g. by providing him with fake data. At a certain point, however, the attacker starts being worried about being detected and starts considering to cause immediate damage, incur a high loss to the defender and leave the network (i.e. use the `takedown` action). The defender has to react to this development and think about blocking the attacker by decreasing the probability of keeping the attacker in the network.

Remark 2 (Demise of the greedy). We can observe that the closer the attacker is to his primary goals (or at least the closer he thinks to be), the less concerned he is about the fact that he might be detected and the more greedy he is about realizing his intents. It is thus easier for the defender to deceive the attacker in such a situation. This is caused by the fact that the attacker must have put more effort to get into deeper layers of the network and the damage he can possibly cause now is more significant—thus he is willing to take a greater risk of being detected. This in turn allows the defender to deceive him more efficiently. While in the Layer 1, the attacker starts considering the `takedown` action when he thinks that he is detected with probability 0.298 (and the defender has to react accordingly), in the Layer 3 he delays this decision up to the point when his belief about the detection state is 0.442. We conjecture that this type of behavior of the deceivee can be seen in a wide range of deception problems and the deceiver can capitalize on that.

To better understand the implications of the optimal defense strategy and the need for precise randomization between `engage` and `block` actions, we simulate an attack on the network and depict attacker's belief about being detected when applying the optimal, *always-block* and *always-engage* strategies.

After performing an action and getting feedback from the network, the attacker updates his belief about the detection state from b to b'. Assume that the attacker was in Layer i and he used action $a_A \in \{\texttt{compromise}, \texttt{exfiltrate}\}$ in the last step and he didn't get blocked. In order to be detected at the current time step, the attacker could have either triggered an alert using his last action (which happens with probability $(1-b)p_{trig}(n, a_A)$), or he must have been already detected and the defender must have decided not to block the attacker (the probability of which is $b\sigma(i,b)$). The probability of not getting blocked equals to $1 - b(1-\sigma(i,b))$. We can thus derive a belief update formula (see Eq. (4)) specifically for this game when the updated belief of the attacker b' is the probability of being detected in the next time step:

$$b' = \frac{(1-b)p_{trig}(n, a_A) + b\sigma(i,b)}{1 - b \cdot [1 - \sigma(i,b)]}. \tag{12}$$

We assume that the attacker conducts an attack that consists of penetrating to the deepest layer of the network using three consecutive `compromise` actions and then the attacker exfiltrates data forever. The comparison of the evolution of attacker's belief while the defender uses the optimal strategy with the always-block and always-engage strategies is shown in Fig. 4.

First of all, we explain why the current best practice in incident response represented by the *always-block* strategy is inferior. Whenever the attacker realizes that he has not been blocked and his access to the network has not been restricted (or limited), he *knows* that he cannot have been detected in the previous time step (since otherwise, the defender would have blocked him according to his *always-block* strategy). His belief about being detected thus depends solely on the detection rate of IDS – which in our experiments is $p_{trig}(n, \texttt{compromise}) = 0.2$ when the attacker uses `compromise` action to penetrate deeper into the network (first 3 steps) and $p_{trig}(n, \texttt{exfiltrate}) = 0.1$ afterward. Since the attacker remains highly confident that he is not detected at each time step, he can cause a lot of harm by a long-term data exfiltration.

The *always-engage* strategy suffers from playing `engage` action even at times when the attacker becomes highly confident that he has been detected and thus realizes that the data he exfiltrates may be useless. At that point, the attacker

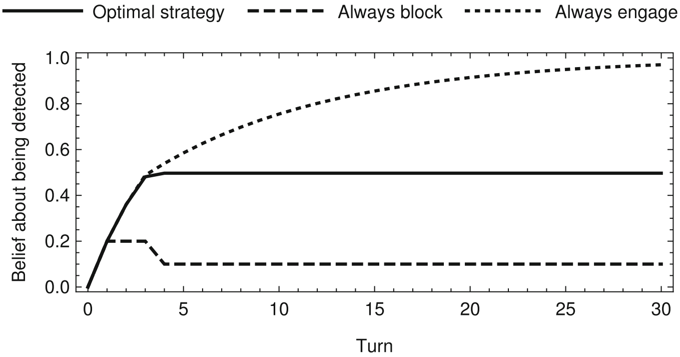

Fig. 4. Evolution of attacker's belief over time. If we block the attacker immediately after detection, he remains highly confident that we cannot employ deceptive actions which allows him to perform long-term data exfiltration. If we always attempt to deceive the attacker by engaging him, he realizes that he likely faces a deception and decides to cause immediate damage – which cannot be prevented by the deceptive `engage` action.

deviates from the assumed attack plan and opts for causing immediate damage and leaving the network temporarily (before launching a new attack).

The optimal defense strategy, on the other hand, stabilizes attacker's belief about being detected at the value of $b = 0.4968$. This is the right belief where the attacker still thinks that it is worth attempting to cause a long term damage by data exfiltration, despite being vulnerable to defender's deceptive attempts.

Remark 3 (Curse of exclusion). This result draws one important conclusion about the use of deception to manipulate attacker's belief. The decision to exclude the attacker from the network (or even more importantly the decision *not* to block him) leaks a valuable piece of information to the attacker. If we do not think about blocking the attacker in a strategic way, the attacker can capitalize on getting this information to devise a powerful attack plan. We have to weigh the use of stealthy and non-stealthy defensive actions carefully not to alert the attacker to the use of deception. The optimal defensive strategy (unlike the *always-block* and *always-engage* strategies) achieves a belief point where no further information leaks to the attacker and the malicious effects of attacker's actions are minimized.

5.4 Engaging the Attacker

In Sect. 5.3 we have shown that the common practice in incident response deployments of blocking the attacker immediately after detection is susceptible to severe drawbacks. We proposed an alternative strategy, based on a game-theoretic model, that postpones the decision to block the attacker to minimize the long-term damage to the network. The key motivation for using this strategy is that by anticipating malicious actions of the attacker, we can minimize negative impacts of his actions and delay his progress. On the other hand, excluding the attacker from the network is only temporary. The attacker is potentially able to reenter the network and cause significant damage before we manage to detect him again.

Our strategy has, however, one more significant advantage since it can be leveraged to decrease false positive rates of the IDS. False detections can have a considerable negative impact on the network operations. By engaging a suspicious user in the network, we can make use of the extra time given by our deceptive strategy to identify the user, infer their objectives and take proper defense actions to reduce the impact of the network defense system on legitimate users. To this end, we can use various types of deceptive signals that do not influence legitimate users considerably, but make the progress of an attacker difficult. These signals are not explicitly captured in our example, but the model is general enough to account for them.

We conducted an experimental evaluation of our game-theoretic strategy to determine the average time between the first IDS alert and the time we decide to block the user. We evaluated our strategy against an advanced attacker who plays a best response to strategy σ_D and we considered only the attacks where the attacker does not decide to quit the network himself. We found out that the average time between detection and the time we decide to restrict attacker's access is in our case 4.577 time steps. In this time window, the defender gets additional alerts from the IDS which may help him to decide about the credibility of the alert better and thus assure that he is about to block a malicious user.

5.5 Robustness of the Model

In real world setups, it need not be possible for the defender to keep track of attacker's belief accurately as a result of failing to reconstruct the exact history of the attacker and/or deficiencies in the model of the network. In this section, we focus on the impact of not knowing the exact IDS detection

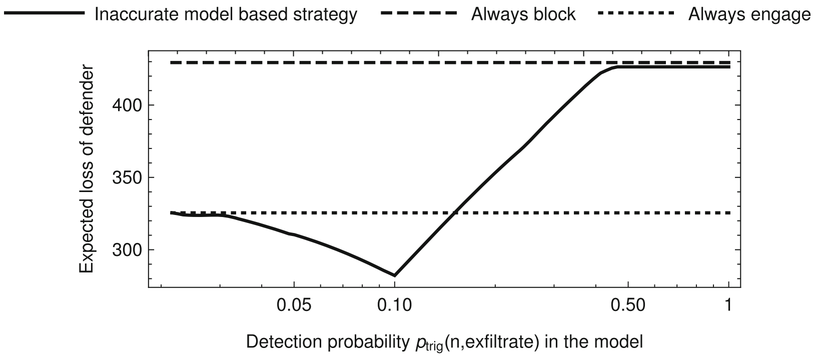

Fig. 5. Expected loss of the defender when using a strategy originating from an inaccurate model. Strategy is computed while assuming detection probability $p'_{trig}(n, \texttt{exfiltrate})$ and this strategy is evaluated in a network with the detection probability of $p_{trig}(n, \texttt{exfiltrate}) = 0.1$.

probabilities. We compute the optimal strategy of the defender based on a model where $p'_{trig}(n, \texttt{exfiltrate})$ does not match the detection probability in the real network. We then evaluate the resulting strategy in the network where the detection probability is $p_{trig}(n, \texttt{exfiltrate}) = 0.1$. Since the model is no longer accurate, the resulting strategies need not be optimal. The experimental evaluation of these strategies is shown in Fig. 5. The experimental results show that our strategy provides significant room for the error in the design of the model, especially if we are pessimistic about the detection rates.

6 Conclusions

We have provided a principled analysis of cyber deception in network security based on game-theoretic foundations. First, we have introduced a generic game-theoretic model for strategic reasoning about deception, then we applied this model to the network security, and we illustrated the impact of active deception on the security level of the network in a case study. Our results have shown that the use of cyber deception techniques can reduce the risks associated with network operations and minimize the damage a sophisticated attacker can inflict to the network. The deceptive operation, however, achieves the maximum efficiency if the attacker is unaware of being deceived. While this result is not surprising, our analysis provides theory supporting this result.

Our work serves as a proof of concept to motivate the interest in thinking about active cyber deception in a strategic way. We used a simplified example to introduce main ideas and discuss the need for reasoning about the belief and adaptation process of the adversary. In the future work, however, we plan to address computational challenges introduced by large networks by leveraging the structure and symmetries found in the problem. An interesting, and also natural, continuation of our work is to relax the assumption that the defender can reconstruct the view of the attacker perfectly. In general, the *two-sided* imperfect information presents significant theoretical and computational challenges, however, we believe that it is possible to identify significant subclasses relevant for the network security that allow for efficient solution techniques.

Acknowledgments. This research was supported by the Czech Science Foundation (grant no. 15-23235S), NSF grants CNS-1544782 and CNS-1720230, the DOE grant DE-NE0008571, by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes not with standing any copyright notation here on.

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 2016 International Workshop on Managing Insider Security Threats, pp. 57–68. ACM (2016)
2. Başar, T., Olsder, G.J.: Dynamic Noncooperative Game Theory. SIAM, Philadelphia (1998)
3. Bercovitch, M., Renford, M., Hasson, L., Shabtai, A., Rokach, L., Elovici, Y.: HoneyGen: an automated honeytokens generator. In: IEEE International Conference on Intelligence and Security Informatics, ISI 2011, pp. 131–136. IEEE (2011)
4. Bowen, B.M., Hershkop, S., Keromytis, A.D., Stolfo, S.J.: Baiting inside attackers using decoy documents. In: Chen, Y., Dimitriou, T.D., Zhou, J. (eds.) SecureComm 2009. LNICSSITE, vol. 19, pp. 51–70. Springer, Heidelberg (2009). doi:10.1007/978-3-642-05284-2_4
5. Dagon, D., Qin, X., Gu, G., Lee, W., Grizzard, J., Levine, J., Owen, H.: HoneyStat: local worm detection using honeypots. In: Jonsson, E., Valdes, A., Almgren, M. (eds.) RAID 2004. LNCS, vol. 3224, pp. 39–58. Springer, Heidelberg (2004). doi:10.1007/978-3-540-30143-1_3
6. Durkota, K., Lisý, V., Bošanský, B., Kiekintveld, C.: Approximate solutions for attack graph games with imperfect information. In: Khouzani, M.H.R., Panaousis, E., Theodorakopoulos, G. (eds.) GameSec 2015. LNCS, vol. 9406, pp. 228–249. Springer, Cham (2015). doi:10.1007/978-3-319-25594-1_13
7. Falliere, N., Murchu, L.O., Chien, E.: W32. stuxnet dossier. White Paper Symantec Corp. Secur. Response **5**(6), 2–3 (2011). https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf
8. Gostev, A., Soumenkov, I.: Stuxnet/Duqu: The evolution of drivers (2011). http://www.securelist.com/en/analysis/204792208/Stuxnet_Duqu
9. Gu, G., Zhang, J., Lee, W.: BotSniffer: detecting botnet command and control channels in network traffic. In: Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS 2008) (2008)
10. Hansen, E.A., Bernstein, D.S., Zilberstein, S.: Dynamic programming for partially observable stochastic games. AAAI **4**, 709–715 (2004)
11. Horák, K., Bošanský, B., Pěchouček, M.: Heuristic search value iteration for one-sided partially observable stochastic games. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI 2017) (2017)
12. Jajodia, S., Ghosh, A.K., Subrahmanian, V., Swarup, V., Wang, C., Wang, X.S. (eds.): Moving Target Defense II - Application of Game Theory and Adversarial Modeling. Advances in Information Security, vol. 100. Springer, New York (2013)
13. Jajodia, S., Ghosh, A.K., Swarup, V., Wang, C., Wang, X.S. (eds.): Moving Target Defense - Creating Asymmetric Uncertainty for Cyber Threats. Advances in Information Security, vol. 54. Springer, New York (2011)
14. Kreibich, C., Crowcroft, J.: Honeycomb: creating intrusion detection signatures using honeypots. ACM SIGCOMM Comput. Commun. Rev. **34**(1), 51–56 (2004)
15. Kuipers, D., Fabro, M.: Control systems cyber security: Defense in depth strategies. United States, Department of Energy (2006)
16. Manshaei, M.H., Zhu, Q., Alpcan, T., Bacşar, T., Hubaux, J.P.: Game theory meets network security and privacy. ACM Comput. Surv. (CSUR) **45**(3), 25 (2013)

17. McRae, C.M., Vaughn, R.B.: Phighting the phisher: using web bugs and honeytokens to investigate the source of phishing attacks. In: 40th Annual Hawaii International Conference on System Sciences 2007, HICSS 2007, p. 270c. IEEE (2007)
18. Mohammadi, A., Manshaei, M.H., Moghaddam, M.M., Zhu, Q.: A game-theoretic analysis of deception over social networks using fake avatars. In: Zhu, Q., Alpcan, T., Panaousis, E., Tambe, M., Casey, W. (eds.) GameSec 2016. LNCS, vol. 9996, pp. 382–394. Springer, Cham (2016). doi:10.1007/978-3-319-47413-7_22
19. Osborne, M.J., Rubinstein, A.: A Course in Game Theory. MIT Press, Cambridge (1994)
20. Pawlick, J., Farhang, S., Zhu, Q.: Flip the cloud: cyber-physical signaling games in the presence of advanced persistent threats. In: Khouzani, M.H.R., Panaousis, E., Theodorakopoulos, G. (eds.) GameSec 2015. LNCS, vol. 9406, pp. 289–308. Springer, Cham (2015). doi:10.1007/978-3-319-25594-1_16
21. Rostami, M., Koushanfar, F., Rajendran, J., Karri, R.: Hardware security: threat models and metrics. In: Proceedings of the International Conference on Computer-Aided Design, pp. 819–823. IEEE Press (2013)
22. Spitzner, L.: Honeypots: Tracking Hackers, vol. 1. Addison-Wesley Reading, Boston (2003)
23. Stech, F.J., Heckman, K.E., Strom, B.E.: Integrating cyber-D&D into adversary modeling for active cyber defense. In: Jajodia, S., Subrahmanian, V., Swarup, V., Wang, C. (eds.) Cyber Deception, pp. 1–22. Springer, Cham (2016). doi:10.1007/978-3-319-32699-3_1
24. Symantec: Preparing for a cyber attack (2013). http://www.symantec.com/content/en/us/enterprise/other_resources/b-preparing-for-a-cyber-attack-interactive-SYM285k_050913.pdf. Accessed 17 Apr 2017
25. Tankard, C.: Advanced persistent threats and how to monitor and deter them. Netw. Secur. **2011**(8), 16–19 (2011)
26. Underbrink, A.: Effective cyber deception. In: Jajodia, S., Subrahmanian, V., Swarup, V., Wang, C. (eds.) Cyber Deception, pp. 115–147. Springer, Cham (2016). doi:10.1007/978-3-319-32699-3_6
27. Vollmer, T., Manic, M.: Cyber-physical system security with deceptive virtual hosts for industrial control networks. IEEE Trans. Industr. Inf. **10**(2), 1337–1347 (2014)
28. Weinstein, W., Lepanto, J.: Camouflage of network traffic to resist attack (CONTRA). In: DARPA Information Survivability Conference and Exposition 2003. Proceedings, vol. 2, pp. 126–127. IEEE (2003)
29. Zhu, Q., Başar, T.: Game-theoretic approach to feedback-driven multi-stage moving target defense. In: Das, S.K., Nita-Rotaru, C., Kantarcioglu, M. (eds.) GameSec 2013. LNCS, vol. 8252, pp. 246–263. Springer, Cham (2013). doi:10.1007/978-3-319-02786-9_15
30. Zhu, Q., Basar, T.: Game-theoretic methods for robustness, security, and resilience of cyberphysical control systems: games-in-games principle for optimal cross-layer resilient control systems. IEEE Control Syst. **35**(1), 46–65 (2015)
31. Zhu, Q., Clark, A., Poovendran, R., Başar, T.: Deceptive routing games. In: IEEE 52nd Annual Conference on Decision and Control (CDC), pp. 2704–2711. IEEE (2012)
32. Zhu, Q., Clark, A., Poovendran, R., Basar, T.: Deployment and exploitation of deceptive honeybots in social networks. In: IEEE 52nd Annual Conference on Decision and Control (CDC), pp. 212–219. IEEE (2013)