Strategic Cyber Camouflage

Christopher Kiekintveld, Aron Laszka, Mohammad Sujan Miah, Shanto Roy, Nazia Sharmin

University of Texas at El Paso, University of Houston

Abstract. One of the most fundamental tasks for an AICA agent will be to manipulate information that an adversary can observe, either about a network or the AICA agent itself. This includes taking actions to conceal or camouflage the agent or specific network assets and taking actions to deceive or otherwise affect the beliefs of an adversary conducting reconnaissance activities. In this chapter we provide an overview of tactics that have been proposed in the literature for implementing cyber camouflage and deception actions, as well as some foundational models in AI from game theory and machine learning that have been used to deploy these tactics strategically. We go into detail on three particular models; the first uses game theory to optimize the use of decoys or modified signals, the second uses game theory to consider the modification of features for both real and fake objects to confuse attackers, and the third applies machine learning methods to scale up feature modifications to create more effective deceptive objects at scale. All of these models can be customized to different types of strategic questions around effectively deploying camouflage to affect an adversary, and they serve as a starting point for implementing autonomous strategies that use camouflage tactics. We end by discussing some of the different ways that camouflage and deception have been evaluated so far in the literature, noting that more work is needed to assess AICA agents using these strategies in realistic environments.

1 Introduction

From the earliest history of conflict, stealth and deception tactics have been a critical way to gain strategic advantage on the battlefield. While the details vary, the goal is always to control the information space, preventing the adversary from gaining useful information and creating false or misleading beliefs in some cases. Camouflage is one example of this; it has a long history of use in physical environments as a method to make the presence or actions of an entity difficult to detect against the backdrop of the environment. In cyber warfare the control of information is even more central, and the ability to perform (or hinder) effective reconnaissance will likely be decisive in many engagements. Therefore, developing effective methods for implementing and strategically deploying camouflage in a cyber context is an important research objective for cyber operations.

In the particular context of an Autonomous Intelligent Cyber-defense Agent (AICA), we identify three primary reasons why cyber-camouflage techniques are important:

- An AICA may be tasked with implementing and deploying camouflage actions for a network or individual host to make reconnaissance more difficult for the adversary
- An AICA may need to conceal its own presence or actions from the adversary to evade detection
- An AICA may need to detect and identify threats that are using camouflage tactics to conceal their own activities, so the agent would need to be able to mitigate camouflage tactics of the opponent

In this chapter we present an overview of some common methods for implementing camouflage tactics in the cyber environment. We then present some basic mathematical frameworks based in game theory that have been developed to model the strategic aspects of how to use and optimize camouflage in the cyber environment. We go into detail on three particular models. The first two use game theory to formulate specific optimization problems, and the last one shows how we can extend these models using machine learning to implement more effective decoy objects (e.g., honeypots or fake traffic) that are difficult for adversaries to detect. Finally, we review some of the ways in which camouflage (and more generally, deception methods) have been evaluated so far in the research literature. While we

Published as Kiekintveld, C., Laszka, A., Miah, M.S., Roy, S., Sharmin, N. (2023). Strategic Cyber Camouflage. In: Kott, A. (eds) Autonomous Intelligent Cyber Defense Agent (AICA). Advances in Information Security, vol 87. Springer, Cham. https://doi.org/10.1007/978-3-031-29269-9_9

cannot cover all of the important topics on cyber-camouflage here, we present some fundamental concepts and models that can be adapted to address many key challenges for AICA and provide references for additional study.

2 Implementing Camouflage

The goal of cyber camouflage is to take actions that make the presence, actions, and intentions of systems or artificial agents more difficult for an adversary to correctly perceive. This can be achieved using a wide variety of specific techniques for manipulating information depending on the context and objectives. We begin by introducing some representative methods from the literature for implementing camouflage at a technical level to give a sense of what types of actions can potentially be used to manipulate the information space. We focus our discussion on two broad categories of actions: obfuscation (hiding information) and deploying decoys (a form of deception). More thorough coverage and discussion can be found in related survey articles (e.g., Han et al. [2018], Fraunholz et al. [2018])

2.1 Obfuscation Techniques

One of the most basic goals for cyber camouflage for AICA agents is to conceal the presence of an agent in the first place, or to conceal specific actions or objectives. Cyber attackers use a wide variety of methods to conceal their activities, such as stealthy scanning, obfuscated malware, obfuscated command and control communications, and specific actions to cover the tracks of an attack. Many of these are also relevant for cyber defense to make it more difficult for attackers to perform basic system reconnaissance as well as to conceal the nature of cyber defenses. However, cyber defense has typically placed less emphasis on effective concealment and obfuscation of information for several reasons, including potential impacts on legitimate users, the complexity of implementing such strategies broadly on a network, and the desire not to rely entirely on obfuscation for security. However, as automated agents for both attack and defense become more sophisticated, it is both possible and necessary to focus more attention on defensive obfuscation to gain advantages early in the cyber kill chain by hindering attacker reconnaissance and planning efforts Hosseinzadeh et al. [2015]. We now briefly introduce some existing methods for defensive obfuscation at different levels.

Network Layer: Basic properties of the network topology and configuration can be obscured by manipulating the data plane in various ways to limit the accuracy of passive and active network scanning methods. This can include intercepting and modifying path tracing probes directly Meier et al. [2018], route obfuscation utilizing ranking-based route mutation Bin-Yahya and Shen [2022], utilizing honey links and hiding important links in a large network Liu et al. [2021], delaying identified probe packets to hinder Network Topology Inference Hou et al. [2020], etc. New methods for obfuscation are actively being developed that make use of AI techniques such as adversarial machine learning to more effectively obfuscate the characteristics of network traffic Verma et al. [2018], Datta et al. [2018].

System Layer: Attackers also use fingerprinting methods to identify specific software or configuration details for individual systems, such as operating system versions. Information is often leaked by protocols and services, but information can be either redacted or modified to limit or mislead fingerprinting attempts [Anderson and McGrew, 2017, Hosseinzadeh et al., 2015].

Application Layer: The application layer encompasses many different applications that could be running on a host, as well as their configurations, associated data, and user activities. This includes security applications, including AICA agents. Examples of application-level obfuscation include the framework proposed by Perez et al. that identifies and obfuscates user data using metadata and related obfuscation strategies [Perez et al., 2018]. Software or application data can also be obfuscated level by level to achieve layered security [Xu et al., 2020]. Other examples have used adversarial learning to obscure data without compromising semantic attributes [Bertran et al., 2019].

2.2 Decoy Technologies

Moving beyond obfuscation, deception methods aim to explicitly create false beliefs, rather than just hiding or changing the characteristics of existing systems or data. One of the most common forms of deception is using decoy objects (e.g., host, files, tokens, etc.) that can be used to distract and confuse attackers, as well as to improve detection and monitoring of malicious activities [Rauti and Lepp¨anen, 2017]. Han et al. provides a layered categorization for different deception techniques[**?**]. Effectively deploying decoys may be an important task for AICA agents, who may also do this dynamically in response to detected attacker activities (e.g., deploying a new honeynet in response to specific scanning activities). AICA agents may also use decoys to try to distract opponents from their own presence or activities.

Network Layer: Network traffic sniffing, scanning, and fingerprinting major attacks in the network layer, which involve capturing and analyzing existing network packets, or generating malicious and fingerprinting packets. The system can deceive the attacker by redirecting attack traffic (e.g., ICMP/TCP packets) to a honey network or fake virtual machines [Sharma and Kaul, 2018], generating vulnerability-driven honey traffic to prevent optimal fingerprinting or packet analysis [Anjum et al., 2020], etc. These methods may also lead the attacker to form incorrect beliefs and plan ineffective attacks or target fake systems rather than real ones.

System Layer: Attackers typically want to compromise internal systems one after another and plan for the next attacks. To deceive adversaries from attacking a real system, honeypots are widely used in the industry. A honeypot is a fake system that may or may not resemble the original hosts. Recent applications of honeypot include VANET Cloud [Sharma and Kaul, 2018], industrial cyberphysical systems [Sun et al., 2020], real-time intrusion detection [Baykara and Das, 2018], defending IoT based botnet DDoS attacks [Vishwakarma and Jain, 2019, Du and Wang, 2019], capturing CPE and IoT zero days [Vetterl and Clayton, 2019], and classifying botnet attacks [Lee et al., 2021].

Application Layer: Application layer reconnaissance includes software and application vulnerability scanning. Both native and web-based applications are targeted by the attackers. Several fake entities of software and applications can be utilized to detect and monitor malicious activity [Rauti and Lepp¨anen, 2017]. Software decoys are widely used to prevent counter-intelligence [FergusonWalter et al., 2021]. Other application-level decoys include honeytokens [Ferguson-Walter et al., 2019], honeypermissions for insider threat detection [Kaghazgaran and Takabi, 2015], and honeyfiles such as automated decoy documents [Voris et al., 2015].

3 Optimizing Camouflage Strategies

We have given some examples of specific actions and tactics that can be used to achieve the broad goals of cyber camouflage. Now we turn to the question of how to deploy these camouflage techniques and actions *effectively*, taking into account the costs and possible impacts on resource utilization, activities of real users, etc. The details of these decisions will vary depending on the purpose of the camouflage, the techniques being used, specific costs and constraints, and assumptions about the adversary. However, the literature provides a set of fundamental principles, models, and algorithms that are abstract enough that they can be used for decision support and automation across a broad range of cyber camouflage and deception situations. We now introduce some basic models for optimizing cyber camouflage decisions using game theoretic models and provide references for further reading on more advanced models.

3.1 Optimizing Decoy Resource Allocation

One area where game-theoretic approaches have been very successful in finding optimal strategies for allocating limited deceptive resources to detect and distract attackers [Carroll and Grosu, 2011, Kiekintveld et al., 2015]. One example is the *Honeypot Selection Game* (HSG) [P´ıbil et al., 2012, Kiekintveld et al., 2015] that models the problem of allocating honeypots to a network. In a real network, not all systems are equally important. A database server may be much more valuable than a user laptop or mobile device. A strategy for deploying honeypots should take this into account when deciding what kinds of systems to create as decoys. The HFG model uses a zero-sum game to optimize the importance values of honeypots to deploy to increase the likelihood that an attacker will target a honeypot rather than a real system. Durkota et al. [Durkota et al., 2015] extends this model by using attack graphs to determine the attacker's optimal attack plans against the defender strategy, where the defender strategy modifies the attack graph by adding honeypots to interdict attacker actions. The attack graphs allow the attacker to attack sequentially, with costs and probabilities of success or failure associated with each attempt. [Anwar et al., 2021] also determine the optimal strategy for deploying honeypots on the attack graphs in a dynamic environment where the attacker and defender interact and can make changes based on observations of the other player. [Wang et al., 2017] uses Bayesian games to explore honeypot strategies in the smart grid to prevent denial of service attacks. La et al. [La et al., 2016] also optimizes honeypot deployment for mitigating denial-of-service attacks in the Internet-of-Things domain. [Du et al., 2017] uses Bayesian game modeling to solve a similar problem for honeypots in the social networking domain. Anjum et al. [Anjum et al., 2020] use a Stackelberg game to deploy honey flows (fake network traffic) optimally to confuse the attacker in distinguishing real and fake vulnerabilities.

3.2 Optimizing Feature Obfuscation

In addition to deploying deceptive objects, there is also a significant body of work in optimizing strategies to obfuscate features of particular objects. This can be used both to make more effective decoys (by making them look more realistic) and to disguise or camouflage real objects (by hiding information or making them look fake). For example, and AICA agent may want to disguise the features of a binary or network traffic to make it look like a normal application.

The Cyber Deception Game (CDG) [Schlenker et al., 2018] computes an optimal deception strategy for concealing specific characteristics of network hosts. This game focuses on invalidating an attacker's information in the reconnaissance phase by deciding what signals the defender wants to send about the type of the host. The defender can respond with obfuscated messages when the attacker probes network hosts, but the model is limited to zero-sum settings. The Cyber Camouflage Games (CCG) [Thakoor et al., 2019] extends the CDG model by considering a general-sum setting. This model also considers uncertainties in the defender's knowledge of the attacker's valuations of different network hosts. [Miah et al., 2020] present a Bayesian game model to find the optimal strategy for obfuscating the observable characteristics of either real or fake objects, making it difficult to distinguish between them. [Guan et al., 2001] camouflages payload traffic components, such as the communication system, location, diversity of hosts, network topology, etc., such that their pattern is unrelated to the operational status of applications to an observer. However, this method is inefficient and can result in significant network overhead.

While there are various methods for obfuscating network traffic, Ciftcioglu et al. [Ciftcioglu et al., 2017] use a game model to obfuscate network traffic, considering that defender has limited resources and obfuscation has network overhead. The water-filling algorithm is another efficient method for finding traffic obfuscation strategies for a given budget [Ciftcioglu et al., 2018]. Machine learning methods have also been used for optimizing feature obfuscation, making use of the gradient of the loss function for generating a perturbation [Carlini and Wagner, 2017, Szegedy et al., 2013]. Verma et al. [Verma et al., 2018] present an adversarial machine learning approach that uses a post-processing procedure on the resulting distributions to manipulate network traffic. However, the proposed method sometimes generates incorrect perturbations and does not correspond to real-world scenarios. [Granados et al., 2020] impose more generalized constraints for obfuscating traffic samples and generate valid perturbation and distribution.

4 Example Methods for Optimizing Camouflage

This section presents three examples of cyber camouflage optimization techniques from the literature in more detail. The first model determines an optimal strategy for disguising network configurations using a game-theoretic model for optimizing signaling strategies. The second model uses game theory to determine how to modify individual features of both real and deceptive objects to make them more effective as decoys, or confusing real objects. The final approach brings in a different set of techniques in machine learning to address similar questions of how to modify features in a more scalable way. All of these models can be generalized to different cyber camouflage problems by considering different action spaces and objective functions.

4.1 Disguising Network Attributes

A network topology comprises multiple systems, each with its own set of attributes such as the operating system, running services, antivirus protection measures, etc. A system's true configuration (TC) could be any combination of these attributes, and systems have different TCs. An attacker can employ network scanning to learn about each system's characteristics before attempting to exploit a target. This reconnaissance reveals potential weak points, such as open ports, operating services, subnetworks, user information, etc. Then the attacker uses specific vulnerability information to find a strategy for system exploitation. However, if the network defender obfuscates the information collected by an attacker, the likelihood of a successful attack decreases. A defender can benefit from using a combination of truthful, false, and obscured responses to the attacker's network probes. For example, consider a network with one system running NGINX web serber and two systems running a Tomcat proxy server. The attacker has a specific NGINX exploit and examines all systems using NGINX before deploying the exploit. If the defender can deceive the attacker about the webserver, the attacker needs to exploit all systems to infiltrate the network. The attacker's network infiltration is delayed by this deception strategy, giving the defender more time to detect an attack. The defender might also use deception techniques to reveal parts of a system's observable attributes that are not true configuration, such as changing the TCP/IP stack or spoofing a running service on a port. Determining deception strategies to alter an attacker's perception is challenging for the defender and also associated with cost.

Cyber Deception Game The Cyber Deception Game (CDG) [Schlenker et al., 2018] addresses this problem and determines the optimal strategies to optimize the defender's deception strategy. The CDG is a two-player zero-sum Stackelberg game between a defender and an attacker. The defender moves as a leader and determines how to respond to the attacker's scanning activity. The attacker moves as a follower and chooses a system to attack based on the observations. The model assumes that when an attacker probes a system, the defender controls the attacker's perception of observed configuration (OC). Masking a true configuration TC with an OC incurs a cost for the defender. The true configuration TC of a system is associated with a utility that is the attacker's reward and an equal loss for the defender. Therefore, the defender's objective is to determine optimal strategies to mask TCs with OCs to minimize the attacker's expected utility while considering deception costs. The following is a formal description of the game notation:

- χ and $\bar{\chi}$ represents all possible TCs and OCs respectively.
- The true state of the network (TSN) is a vector $\upsilon = (\upsilon_x)_{x \in \chi}$, where υ_x is the number of systems on the network with a TC $x \in \chi$.

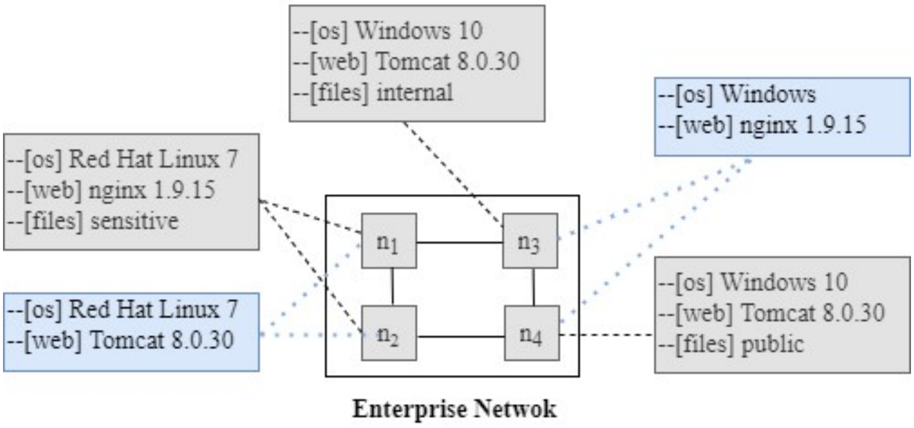


Fig.1: Example of an enterprise network.

- Similarly, the attacker's observed state of the network (OSN) is a vector $\bar{\upsilon} = (\bar{\upsilon}_{\bar{x}})_{\bar{x} \in \bar{\chi}}$. Here, two systems with the same $\bar{x}$ as their OC are indistinguishable from the attacker's perspective.
- Λ is the feasibility constraint as a (0,1)-matrix that defines whether or not x can be masked with $\bar{x}$, with 1 denoting feasibility.
- $\zeta(x,\bar{x})$ denotes the defender cost of masking a TC x with an OC $\bar{x}$

Defender Strategies: The CDG considers that the defender knows the TSN, all possible TCs and OCs, costs, total budget and feasibility constraints. The defender strategy Θ is to determine how many of the υ_x systems having TC x , should be assigned to the OC $\bar{x}$. Therefore, all possible strategies are a $|\chi| \times |\bar{\chi}|$ matrix where $\Theta_{x,\bar{x}}$ representing the number of systems having TC x is masked with OC $\bar{x}$. Θ must satisfy the following constraints:

- Any entry $\Theta_{x,\bar{x}}$ of $|\chi| \times |\bar{\chi}|$ matrix must be a non-negative integer
- The total number of systems having any TC x and OC $\bar{x}$ must be equal to υ_x since the CDG assumes that the TSN υ_x is fixed.
- The Θ must satisfy feasibility constraints. The defender is not allowed to mask any TC x with any OC $\bar{x}$ if the entry $\Lambda_{x,\bar{x}}$ of (0,1)-matrix Λ is 0.
- Finally, the total masking cost must be less than or equal to the cost budget.

Attacker's Strategies: Following the defender's move, the attacker observes the OSN $\bar{\upsilon}$ and tries to attack the OC $\bar{x}$ that gives the highest expected utility. The attacker is indifferent in attempting an attack against all such $\bar{\upsilon}_{\bar{x}}$ because all the systems with the same OC $\bar{x}$, are indistinguishable. Therefore, when he selects an OC $\bar{x}$, he means that he attacks all systems with an OC $\bar{x}$ with the same probability.

Utility The defender aims to protect a set of systems *Ns* from potential exploits where each system is associated with a utility that is the attacker's reward for attacking it. This utility depends on the TC of a system where Ψ_x denotes the utility of each $x \in \chi$. The Ψ_x might be negative when a system's security level is high, or the attacker receives incorrect information. If the defender's strategy is Θ, the attacker's expected utility $\bar{\Psi}_{\bar{x}}$ for attacking an OC $\bar{x}$ with $\bar{\upsilon}_{x>}\bar{}_0$ is defined by:

$$\bar{\Psi}_{\bar{x}} = \sum_{x \in \chi} \frac{\Theta_{x,\bar{x}}}{\bar{\upsilon}_{\bar{x}}} \Psi_x$$

The equation denotes that $\bar{\upsilon}_{\bar{x}}$ systems having an OC $\bar{x}$, $\Theta_{x,\bar{x}}$ have a TC x. When the attacker attacks $\bar{x}$, the defender's expected utility is $-\bar{\Psi}_{\bar{x}}$ since the game is zero-sum. Here, the attacker is restricted to attack an OC $\bar{x}$ when $\bar{\upsilon}_{\bar{x}} = 0$ because it leads his expected utility to $-\infty$.

Small Example Figure (1) shows a example of a small network that comprises a set of systems $N = \{n_1,n_2,n_3,n_4\}$, a set of TCs $\chi = \{x_1,x_2,x_3\}$ (Shown in Figure (1) as the gray boxes) and set of OCs $\bar{\chi} = \{\bar{x}_1,\bar{x}_2\}$ (Shown in Figure (1))as the blue boxes). Let $\chi_{\bar{x}_1} = \{x_1,x_2\}$ and $\chi_{\bar{x}_2} = \{x_2,x_3\}$ be feasibility constraints sets. According to Figure (1), the following are the TCs:

$x_1 = [[os]L,[web]N,[files]S]$ $x_2 =$

$[[os]W,[web]T,[files]I]$ $x_3 =$

$[[os]W,[web]T,[files]P]$

Also, the following are the OCs:

$\bar{x}_1 = [[os]L,[web]T]$ $\bar{x}_2 = [[os]W,[web]N]$

Let the utilities be $\Psi_{x1} = 5, \Psi_{x2} = 0$, and $\Psi_{x3} = 6$. For simplicity, let all the costs $\zeta(x,\bar{x})$ be 0 with no budget constraint. According to Figure (1), the true state of the network $(\upsilon_x)_{x \in \chi}$ is (2, 1 ,1), and the defender strategy Θ is given by

$\bar{x}_1 \bar{x}_2$

x_1" 2 0 # x_2 0

1 x_3 0 1

Now, if the attacker attempts to attack $\bar{x}_1$, his expected utility is $\bar{\Psi}_{\bar{x}_1} = (2 * 5)/2 = 5$. On the other hand, the expected utility of attacking $\bar{x}_2$ is $\bar{\Psi}_{\bar{x}_2} = (0 + 6)/2 = 3$. Therefore, attacking $\bar{x}_1$ is the best response for the attacker and the defender loses an equal amount.

4.2 Feature Selection Game

The Feature Selection Game (FSG)[Miah et al., 2020] addresses a different aspect of the camouflage problem, deciding how exactly to modify the features of real or fake objects to achieve a specific goal (e.g., making fake objects appear more realistic). The FSG is modeled as a general-sum two-player extensive form imperfect information game between an attacker and defender. The defender's goal is to strategically modify both real and fake objects so that the attacker can't tell the difference. Objects are associated with observable feature vectors that can provide useful information to the attackers, allowing them to distinguish objects more accurately. To make classification difficult, the defender changes the observable features of real and fake samples, which we call 2-sided deception. The FSG can be formally defined by the tuple $FSG = (K,v^r,v^h,P^r,P^f,\tau,\chi)$. Here, K represents the complete set of real and fake samples. P^r and P^f are the probability distributions over feature vectors of real and fake samples where the nature player generates the configurations based on these distributions. Samples $x = (x_1,...,x_k)$ are generated according to the joint distribution P^x where $P^{x}(x) = \prod_{i=1}^{r} P^{r}(x_i) \times \prod_{i=r+1}^{k} P^{f}(x_i)$. The defender examines a sample $x \in X$, where X represents all possible samples, and then takes steps to change each object's features. An action $d \in D$ results a new configuration $x' \in X$, which the attacker observes and uses as an information set $I \in \tau$. In each I, the attacker perceives any permutation of configurations in the same way. Therefore, he cannot reliably detect real and fake objects in a feature vector. The data set for the attacker is the set of all possible combinations of object configurations where attacker's action a^I is to detect real and fake objects in each information set I.

The utility functions in this game are calculated based on the importance values of the objects and the cost of changing the features. The attacker gets positive rewards when he correctly detects real and fake objects, but he receives a penalty for misclassifying. In particular, if an attacker's action a in the information set I corresponds to a real object, then the utility function $U(x,j,a) = v^r$, whereas, if it corresponds to a fake object, then $U(x,j,a) = -v^f$ where v^r and v^f are the importance values for real and fake objects, respectively. The defender loses the same amount as the attacker's positive reward, but the situation is reversed when the attacker misclassifies. This part of the utility function represents the zero-sum component of the game. However, the defender needs to pay additional cost to change the characteristics, which makes the game model non-zerosum. The defender's action in a sample x that produces an information set I, where different actions in different network samples can result in the same I. Then, in each $I \in \tau$, the attacker plays the best response where the defender's objective is to maximize his utility, considering feature modification costs. In principle, the FSG game allows us to determine optimal camouflage strategies for the defender to modify the appearance of different objects. However, in practice this model has limited scalability due the exponential growth in the strategy spaces as the number of features grows. This leads us to consider a machine learning variation that can approximate this type of strategy in the next section.

4.3 Two-Sided Feature Deception Using Adversarial Learning

The Two-Sided Generative Adversarial Network (TS-GAN) solves the two-sided feature deception problem in a complex and large feature space by using adversarial learning techniques. It generates fake samples that look like real samples and real samples that look like fake samples. This model consists of two parts: the attacker and the defender. The defender contains two modules, and both are neural networks. One of the networks is a generator that generates fake data, which is represented as G_θ with θ parameters. The G_θ uses a latent space z from an l-dimensional spherical Gaussian distribution P_g to create a fake sample $x' = G_\theta(z)$. It learns to estimate the distribution from which the real training data is drawn to generate fake samples. The objective of G_θ is to minimize the probability of a generated sample being detected as fake by the attacker. The defender's second neural network is the Obfuscator, which refers to O_θ with θ parameters. The O_θ takes the original instance x as input and generates a perturbation $O_\theta(x)$. The dimensions of the input data and output perturbed data of this network are identical. Then $x+O_\theta(x)$ will be passed to the attacker. The learning goal of O_θ is to create a perturbed adversarial example that is indistinguishable from a fake sample.

The attacker or Discriminator (D_θ) is also a neural network and learns to detect as well as possible between the real and fake samples. The problem can be formulated as follows: Let $(x_1,.......,x_n)$ represent the training instances and (x_i,y_i) is the i^{th} instance in the training set, which is made up of feature vectors $x_i \in \chi$ where $\chi \subseteq R^n$ represents the feature space and y_i corresponding real class label (1). Also, let $G_\theta(z_1),....,G_\theta(z_r)$ be a collection of r examples from the generated distribution Pg that are corresponding fake class label (0) and represented by $(x_1^{'}, \cdots, x_n^{'})$ where $x_i^{'} \in \chi$. Similarly, assume, $O_\theta(x_1),....,O_\theta(x_n)$ is a set of perturbation generated from $(x_1,.......,x_n)$ where $x_i+O_\theta(x_i) = x^{adv}{}_i \in \chi$ is the i^th adversarial example, such that $D_\theta(x^{adv_i}) = t$ (target attack) where t is the target class (0). The attacker's learning goal is to learn a classifier $D_\theta : \chi \rightarrow Y$ from the domain χ to the set of classification outputs $Y \in \{0,1\}$, where $|Y|$ represents the number of classification outputs. Figure(2) shows the basic architecture of TS-GAN.

The TS-GAN model can be considered as a game between a defender and an attacker where the defender uses two networks G_θ and O_θ to minimize the detection success of the attacker and forms a minimax game between the attacker and the defender.

5 Evaluating Camouflage

We now discuss some general strategies for measuring and evaluating the effectiveness of cyber camouflage and deception. There are several frameworks for cyber camouflage that have evaluated their work based on effectiveness (e.g., optimal defender utility in game-theoretic models [Anwar et al., 2020, Miah et al., 2020], expected number of attacks deceived [Rawat et al., 2019]) and cost (e.g., reducing defender's cost [Anwar et al., 2020], deceived attacks with respect to deception deployment time [Rawat et al., 2019]). We divide the evaluation of camouflage models primarily based on two approaches: theoretical and experimental and discuss some metrics that have been used to evaluate existing models.

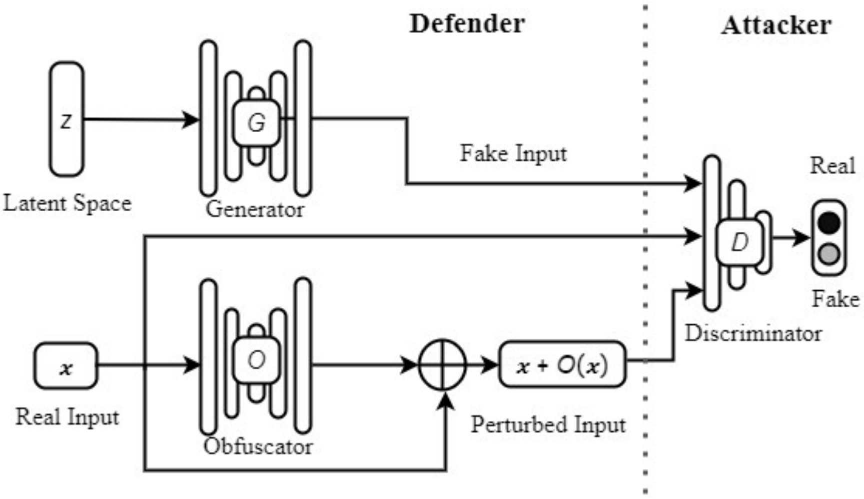


Fig.2: Two-sided generative adversarial network architecture.

5.1 Theoretical Evaluation

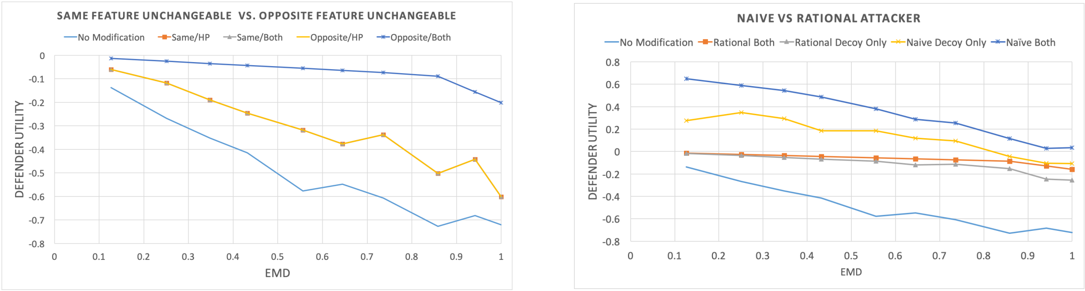
Theoretical evaluation is one of the first steps in assessing the potential benefits of camouflage strategies. These evaluations assess performance within the context and assumptions of a particular model, and usually present an optimistic view of the potential impact in a realistic setting. They are relatively easy to do, and a useful first step in evaluating different approaches. We present some examples of these types of evaluation from the literature.

Non-Game-Theoretic Evaluation: Jajodia et al. argued that attackers could map system configurations (e.g., type of operating systems, applications, or services) for a particular node in the network [Jajodia et al., 2017]. The authors propose a belief state model that considers an interval of probabilities for specific configurations and then tightens the interval over time. The authors proposed two algorithms (Naive-PLD and Fast-PLD) to keep the attacker away from the valuable nodes by answering a scan query that minimizes the damage. They estimated the average damage against the attacker's steps when applying these algorithms. Sugrim et al. utilize Bayesian inference to update the attacker's belief for an individual node property (e.g., IP address) [Sugrim et al., 2018]. The authors quantified the attacker's updated belief over the increasing number of operations. They also measured the attacker's belief error, yield, and footprints.

Game-theoretic Evaluation: Game theoretic models evaluate each player's (attacker and defender) strategies, and current works focus on optimizing strategies for the defender. In a typical Cyber Camouflage Game, computing the optimal defender strategy is NP-hard [Thakoor et al., 2019, Milani et al., 2020], where the first model masks each machine with different observable configurations in a zero-sum game setting, and the second model alters the perceived structure of the attack graph, respectively. The authors proposed approximation algorithms (e.g., MILP, NAS, etc.) to calculate optimal defender strategies. Additionally, there are several honeypot allocation games over the attack graphs in cyber deception or camouflage games [Anwar et al., 2020].

Milani et al. quantified average defender utility for different proposed algorithms achieved against the number of nodes in the network [Milani et al., 2020]. The authors compared the performances of these algorithms by calculating the average defender utility over time. One of the essential evaluation metrics is the run time of proposed algorithms to approximate an optimal solution and how these algorithms handle the scaling of a network. For example, a typical experiment could be quantifying the algorithm run time against the increasing size of the network [Anwar et al., 2020]. Similarly, Thakoor et al. calculated the run time of the proposed MILP with cuts against the strategy space size. Another metric is each player's cost estimation. The goal is always to increase the attack cost or maintain the defender's cost as low as possible. For example, Anwar et al. estimated the defender reward at Nash Equilibrium (optimal allocation) and random allocation at different attack costs [Anwar et al., 2020].

Here, we show an example game-theoretic measurement presented by Miah et al. [Miah et al., 2020], where the authors calculated defender's utility in different scenarios 3. The authors showed that the defender can benefit significantly through utilizing the two-sided feature deception model when the unmodifiable features are different in real and honeypot hosts. Figure 3a considers two-sided feature deception while calculating the defender's utility. Figure 3b presents a comparison between a rational and naïve attacker. The author confirms that the best case is when the defender can perform two-sided deception against a naïve attacker and the worst case is when the defender ignores deception against a fully rational attacker.



(a) When some features cannot be modified.

(b) Na¨ıve attacker versus a fully rational attacker.

Fig.3: Game-theoretic model evaluation (comparison of defender utility) Miah et al. [2020].

5.2 Experimental Evaluation:

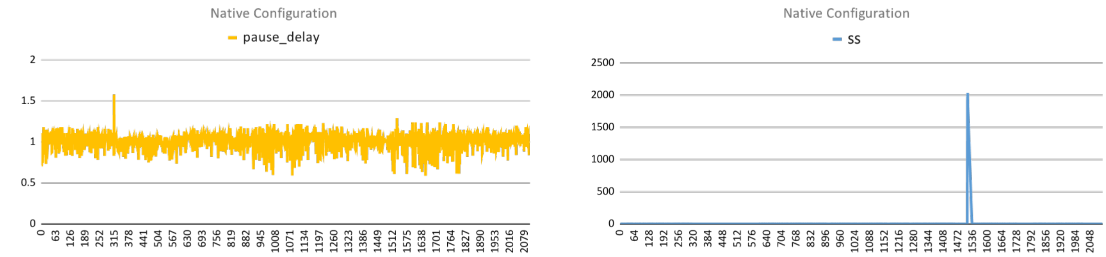
Theoretical models may not always correspond to the results obtained from a real-world scenario for a variety of reasons. Therefore, it is important to also conduct evaluations using experiments in more realistic settings, ideally using real-world architectures, data, etc. as much as possible.

Automated Adversaries: Automated evaluation depends on particular objectives in a predefined scenario, such as a particular type of attacker or a typical vulnerability/exploit choice. Simulation can be used to evaluate strategies based on a pre-defined automated attacker, which has the advantage of consistency and speed. For example, Rawat et al. evaluated performance of deception system for deceiving cyber adversaries in adaptive virtualized wireless networks [Rawat et al., 2019]. The authors quantified the expected number of attacks and deceived attacks with respect to deception deployment time. They also plotted the successful attack time with respect to the deception deployment time.

Human Adversaries: In many cases the ideal evaluation is done using humans, including penetration testers, read teams, or ethical hackers to evaluate the impact of strategies in a realistic scenario. Evaluation using humans can account for how humans may really make decisions (including imperfect ones), which could vary considerably from perfect models in cyber deception scenarios. However, human data is also limited and expensive, and humans can exhibit a wide variety of behaviors and their responses may depend heavily on background knowledge and expertise, especially in very technical domains.

Shade et al. performed an experimental evaluation of host-based deception that involved 30 participants in choosing any host to attack [Shade et al., 2020]. The authors measured the ratio of successful task completion, the proportion of successful commands, and time to task completion. They also estimated the total time to complete, time wasted on decoys, reported surprises, etc.

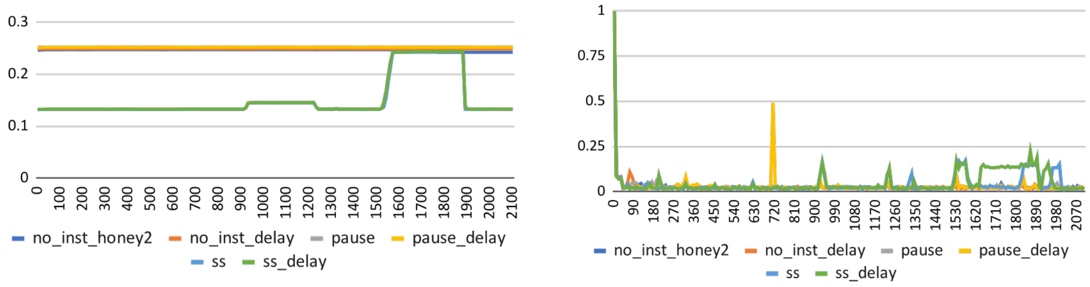
Acosta et al. designed a cyber deception experimentation system (CDES) where the authors proposed an on-demand honeypot instantiation approach [Acosta et al., 2021]. Here the honeypots are dynamically instantiated and presented before an identified attacker. They proposed three types of configurations: *no inst* configuration where the honey VM is instantiated beforehand, *pause resume* configuration where VMs are usually in a suspended state and activated only when resumed, and *save state* case where the VMs are offloaded, but their state is saved and restored based on the requirement. Figure 4 compares the ping delays using CDES in *pause resume* and *save state* configurations. Here, the the *Native* setup uses a separate laptop to run CDES. The authors also experimented with the *In-VM* setup, which uses the CORE emulator within a virtual machine. Later, the authors estimated the CPU and memory usage (Figure 5) while executing these frameworks.



(a) pause resume configuration

(b) save state configuration

Fig.4: Ping delays in Native configuration Acosta et al. [2021]



(a) Memory Utilization

(b) CPU Utilization

Fig.5: Utilization during the execution of the three configurations [Acosta et al., 2021]

The primary goals of evaluating camouflage frameworks are to estimate optimal defender's strategy and cost while minimizing the affect in network or system performance. Even though there are several theoretical models, it is necessary to test the effectiveness of the models or frameworks with experimental setups and human evaluation to evaluate outcomes in more realistic settings.

6 Summary and Conclusions

This chapter has discussed several different aspects of strategic cyber camouflage, including implementation, modeling, optimization, and evaluation. All of these are key considerations for an Autonomous Intelligent Cyber-defense Agent (AICA), both for taking actions to disguise a network and to conceal the activities of the AICA agent. In the basic form, cyber camouflage is about hiding information from an adversary, making their reconnaissance less effective. However, more advanced forms can also use deception tactics to introduce false information and beliefs, such as the use of decoy objects (hosts, traffic, etc.) into a network. These tactics can all achieve goals including confusing the attacker and increasing uncertainty, delaying attacks, creating additional opportunities for detection, etc. An AICA can implement decoy and obfuscation technologies at different layers (network, system, and application) and can choose the best strategies based on an optimal solution. The game theory and machine learning models presented here are examples that can be used as the basis for implementing AI strategies for using camouflage, but they are only a starting point, and many additional factors can be taken into account in developing more advanced strategies. In addition, we have presented some initial evaluations but there is much work to be done to evaluate different cyber camouflage tactics deployed by real AICA agents in realistic networks, particularly in the presence of adversarial agents.

Bibliography

Jaime C Acosta, Anjon Basak, Christopher Kiekintveld, and Charles A Kamhoua. Lightweight ondemand honeypot deployment for cyber deception. In *The 12th EAI International Conference on Digital Forensics & Cyber Crime (EAI ICDF2C), Singapore*, 2021.

Blake Anderson and David McGrew. Os fingerprinting: New techniques and a study of information gain and obfuscation. In *2017 IEEE Conference on Communications and Network Security (CNS)*, pages 1–9. IEEE, 2017.

Iffat Anjum, Mohammad Sujan Miah, Mu Zhu, Nazia Sharmin, Christopher Kiekintveld, William Enck, and Munindar P Singh. Optimizing vulnerability-driven honey traffic using game theory. *arXiv preprint arXiv:2002.09069*, 2020.

Ahmed H Anwar, Charles Kamhoua, and Nandi Leslie. Honeypot allocation over attack graphs in cyber deception games. In *2020 International Conference on Computing, Networking and Communications (ICNC)*, pages 502–506. IEEE, 2020.

Ahmed H Anwar, Charles A Kamhoua, Nandi Leslie, and Christopher D Kiekintveld. Honeypot allocation games over attack graphs for cyber deception. *Game Theory and Machine Learning for Cyber Security*, pages 62–76, 2021.

Muhammet Baykara and Resul Das. A novel honeypot based security approach for real-time intrusion detection and prevention systems. *Journal of Information Security and Applications*, 41:103–116, 2018.

Martin Bertran, Natalia Martinez, Afroditi Papadaki, Qiang Qiu, Miguel Rodrigues, Galen Reeves, and Guillermo Sapiro. Adversarially learned representations for information obfuscation and inference. In *International Conference on Machine Learning*, pages 614–623. PMLR, 2019.

Manaf Bin-Yahya and Xuemin Shen. Secure and energy-efficient network topology obfuscation for software-defined wsns. *IEEE Internet of Things Journal*, 2022.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

Thomas E Carroll and Daniel Grosu. A game theoretic investigation of deception in network security. *Security and Communication Networks*, 4(10):1162–1172, 2011.

Ertugrul Ciftcioglu, Rommie Hardy, Kevin Chan, Lisa Scott, Diego Oliveira, and Gunjan Verma. Chaff allocation and performance for network traffic obfuscation. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, 2018.

Ertugrul N Ciftcioglu, Rommie L Hardy, Lisa M Scott, and Kevin S Chan. Efficient chaff-aided obfuscation in resource constrained environments. In *MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM)*, pages 97–102. IEEE, 2017.

Trisha Datta, Noah Apthorpe, and Nick Feamster. A developer-friendly library for smart home iot privacy-preserving traffic obfuscation. In *Proceedings of the 2018 Workshop on IoT Security and Privacy*, pages 43–48, 2018.

Miao Du and Kun Wang. An sdn-enabled pseudo-honeypot strategy for distributed denial of service attacks in industrial internet of things. *IEEE Transactions on Industrial Informatics*, 16 (1):648–657, 2019.

Miao Du, Yongzhong Li, Qing Lu, and Kun Wang. Bayesian game based pseudo honeypot model in social networks. In *International Conference on Cloud Computing and Security*, pages 62–71. Springer, 2017.

Karel Durkota, Viliam Lisy`, Christopher Kiekintveld, and Branislav Boˇsansky`. Game-theoretic algorithms for optimal network security hardening using attack graphs. *Database*, 20:4xPC, 2015.

Kimberly Ferguson-Walter, Sunny Fugate, Justin Mauger, and Maxine Major. Game theory for adaptive defensive cyber deception. In *Proceedings of the 6th Annual Symposium on Hot Topics in the Science of Security*, page 4, 2019.

Kimberly J Ferguson-Walter, Maxine M Major, Chelsea K Johnson, and Daniel H Muhleman. Examining the efficacy of decoy-based and psychological cyber deception. In *30th USENIX*

Security Symposium (USENIX Security 21), pages 1127–1144, 2021.

Daniel Fraunholz, Simon Duque Anton, Christoph Lipps, Daniel Reti, Daniel Krohmer, Frederic Pohl, Matthias Tammen, and Hans Dieter Schotten. Demystifying deception technology: A survey. *arXiv preprint arXiv:1804.06196*, 2018.

Alonso Granados, Mohammad Sujan Miah, Anthony Ortiz, and Christopher Kiekintveld. A realistic approach for network traffic obfuscation using adversarial machine learning. In *International Conference on Decision and Game Theory for Security*, pages 45–57. Springer, 2020.

Yong Guan, Xinwen Fu, Dong Xuan, Prashanth Umesh Shenoy, Riccardo Bettati, and Wei Zhao. Netcamo: camouflaging network traffic for qos-guaranteed mission critical applications. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 31(4):253–265, 2001.

Xiao Han, Nizar Kheir, and Davide Balzarotti. Deception techniques in computer security: A research perspective. *ACM Computing Surveys (CSUR)*, 51(4):1–36, 2018.

Shohreh Hosseinzadeh, Sampsa Rauti, Sami Hyrynsalmi, and Ville Lepp¨anen. Security in the internet of things through obfuscation and diversification. In *2015 International Conference on Computing, Communication and Security (ICCCS)*, pages 1–5. IEEE, 2015.

Tao Hou, Zhe Qu, Tao Wang, Zhuo Lu, and Yao Liu. Proto: Proactive topology obfuscation against adversarial network topology inference. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 1598–1607. IEEE, 2020.

Sushil Jajodia, Noseong Park, Fabio Pierazzi, Andrea Pugliese, Edoardo Serra, Gerardo I Simari, and VS Subrahmanian. A probabilistic logic of cyber deception. *IEEE Transactions on Information Forensics and Security*, 12(11):2532–2544, 2017.

Parisa Kaghazgaran and Hassan Takabi. Toward an insider threat detection framework using honey permissions. *J. Internet Serv. Inf. Secur.*, 5(3):19–36, 2015.

Christopher Kiekintveld, Viliam Lisy`, and Radek P´ıbil. Game-theoretic foundations for the strategic use of honeypots in network security. In *Cyber warfare*, pages 81–101. 2015.

Quang Duy La, Tony QS Quek, Jemin Lee, Shi Jin, and Hongbo Zhu. Deceptive attack and defense game in honeypot-enabled networks for the internet of things. *IEEE Internet of Things Journal*, 3(6):1025–1035, 2016.

Seungjin Lee, Azween Abdullah, Nz Jhanjhi, and Sh Kok. Classification of botnet attacks in iot smart factory using honeypot combined with machine learning. *PeerJ Computer Science*, 7:e350, 2021.

Yaqun Liu, Jinlong Zhao, Guomin Zhang, and Changyou Xing. Netobfu: A lightweight and efficient network topology obfuscation defense scheme. *Computers & Security*, 110:102447, 2021.

Roland Meier, Petar Tsankov, Vincent Lenders, Laurent Vanbever, and Martin Vechev. {NetHide}: Secure and practical network topology obfuscation. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 693–709, 2018.

Mohammad Sujan Miah, Marcus Gutierrez, Oscar Veliz, Omkar Thakoor, and Christopher Kiekintveld. Concealing cyber-decoys using two-sided feature deception games. In *HICSS*, pages 1–10, 2020.

Stephanie Milani, Weiran Shen, Kevin S Chan, Sridhar Venkatesan, Nandi O Leslie, Charles Kamhoua, and Fei Fang. Harnessing the power of deception in attack graph-based security games. In *International Conference on Decision and Game Theory for Security*, 2020.

Beatrice Perez, Mirco Musolesi, and Gianluca Stringhini. You are your metadata: Identification and obfuscation of social media users using metadata information. In *Twelfth International AAAI Conference on Web and Social Media*, 2018.

Radek P´ıbil, Viliam Lisy´, Christopher Kiekintveld, Branislav Boˇsansky´, and Michal Pˇechouˇcek. Game Theoretic Model of Strategic Honeypot Selection in Computer Networks. (1):201–220, 2012. https://doi.org/10.1007/978-3-642-34266-0$_1$2*. URLhttp* : *//link.springer.com/*10*.*1007*/*978 − 3 − 642 − 34266 − 0 12*.*

Sampsa Rauti and Ville Lepp¨anen. A survey on fake entities as a method to detect and monitor malicious activity. In *2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, pages 386–390. IEEE, 2017.

Danda B Rawat, Naveen Sapavath, and Min Song. Performance evaluation of deception system for deceiving cyber adversaries in adaptive virtualized wireless networks. In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, pages 401–406, 2019.

Aaron Schlenker, Omkar Thakoor, Haifeng Xu, Fei Fang, Milind Tambe, Long Tran-Thanh, Phebe Vayanos, and Yevgeniy Vorobeychik. Deceiving cyber adversaries: A game theoretic approach. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 892–900. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

Temmie Shade, Andrew Rogers, Kimberly Ferguson-Walter, Sara Beth Elsen, Daniel Fayette, and Kristin E Heckman. The moonraker study: An experimental evaluation of host-based deception. In *HICSS*, pages 1–10, 2020.

Sparsh Sharma and Ajay Kaul. A survey on intrusion detection systems and honeypot based proactive security mechanisms in vanets and vanet cloud. *Vehicular communications*, 12:138–164, 2018.

Shridatt Sugrim, Sridhar Venkatesan, Jason A Youzwak, Cho-Yu J Chiang, Ritu Chadha, Massimiliano Albanese, and Hasan Cam. Measuring the effectiveness of network deception. In *2018 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 142–147. IEEE, 2018.

Yanbin Sun, Zhihong Tian, Mohan Li, Shen Su, Xiaojiang Du, and Mohsen Guizani. Honeypot identification in softwarized industrial cyber–physical systems. *IEEE Transactions on Industrial Informatics*, 17(8):5542–5551, 2020.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Omkar Thakoor, Milind Tambe, Phebe Vayanos, Haifeng Xu, Christopher Kiekintveld, and Fei Fang. Cyber camouflage games for strategic deception. In *International Conference on Decision and Game Theory for Security*, pages 525–541, 2019.

Gunjan Verma, Ertugrul Ciftcioglu, Ryan Sheatsley, Kevin Chan, and Lisa Scott. Network traffic obfuscation: An adversarial machine learning approach. In *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*, pages 1–6. IEEE, 2018.

Alexander Vetterl and Richard Clayton. Honware: A virtual honeypot framework for capturing cpe and iot zero days. In *2019 APWG Symposium on Electronic Crime Research (eCrime)*, pages 1–13. IEEE, 2019.

Ruchi Vishwakarma and Ankit Kumar Jain. A honeypot with machine learning based detection framework for defending iot based botnet ddos attacks. In *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 1019–1024. IEEE, 2019.

Jonathan Voris, Jill Jermyn, Nathaniel Boggs, and Salvatore Stolfo. Fox in the trap: Thwarting masqueraders via automated decoy document deployment. In *Proceedings of the Eighth European Workshop on System Security*, pages 1–7, 2015.

Kun Wang, Miao Du, Sabita Maharjan, and Yanfei Sun. Strategic honeypot game model for distributed denial of service attacks in the smart grid. *IEEE Transactions on Smart Grid*, 8(5): 2474–2482, 2017.

Hui Xu, Yangfan Zhou, Jiang Ming, and Michael Lyu. Layered obfuscation: a taxonomy of software obfuscation techniques for layered security. *Cybersecurity*, 3(1):1–18, 2020.

Biography

Christopher Kiekintveld is an associate professor at the University of Texas at El Paso (UTEP). His research is in the area of intelligent systems, focusing on multi-agent systems and computational decision making. He is also interested in applications of artificial intelligence to security, trading agents, and other areas with the potential to benefit society. He received his Ph.D in 2008 from the University of Michigan for thesis work on strategic reasoning, including applications in designing a champion trading agent for the TAC SCM competition. He has authored more than 80 papers in peer-reviewed conferences and journals (e.g., AAMAS, IJCAI, AAAI, JAIR, JAAMAS, ECRA). He has received several best paper awards, the David Rist Prize, and an NSF CAREER award.

Aron Laszka is an Assistant Professor in the Department of Computer Science at the University of Houston. His research interests revolve around the applications of artificial intelligence and machine learning to cybersecurity and societal-scale cyber-physical systems. His recent work has been funded by the National Science Foundation, the Department of Energy, and the Department of Transportation. Previously, he was a Research Assistant Professor at Vanderbilt University from 2016 to 2017, and a Postdoctoral Scholar at the University of California, Berkeley from 2015 to 2016. He graduated summa cum laude with a Ph.D. in Computer Science from the Budapest University of Technology and Economics in 2014.

Mohammad Sujan Miah is a Ph.D. student in Computer Science at the University of Texas at El Paso. His research interest lies between game theory and adversarial machine learning areas. His main research focuses on the application of game theory in cyber defense. Particularly, he has been exploring deceptive strategies using the game-theoretic model. He received a Bachelor of Science in Computer Science and Engineering from the University of Dhaka, Bangladesh. After graduation, he worked as a software engineer at a Samsung Electronic research center for a few years. He also had professional experience working as a software engineer at other leading software companies in Bangladesh.

Shanto Roy joined the Resilient Networks and Systems Lab as a Ph.D. student in the fall 2019 semester. His research interest is in Cyber Security, Privacy, Data Analytics, Machine Learning, Cloud-IoT ecosystem, etc. Earlier, he earned his BSc and MSc in Information Technology from Jahangirnagar University, Bangladesh in 2015 and 2016 respectively. Later, he served as a lecturer in the Dept. of Computer Science at the Green University of Bangladesh from 2016 to 2019. In the meantime, he contributed to several cloud-IoT ecosystem security and optimization projects. Currently, his research work focuses on Cyber Deception, an approach to minimize significant damages in an enterprise network by deceiving and misdirecting adversaries.

Nazia Sharmin is a computer science Ph.D. student at the University of Texas at El Paso. She received an MS and a bachelor's degree in physics from the University of Texas, El Paso, and the University of Dhaka, respectively. Her research interest includes cyber security and machine learning, particularly in the application of machine learning to the existing threat in the network.