Learning to Play an Adaptive Cyber Deception Game

Yinuo Du
Carnegie Mellon University
Pittsburgh, Pennsylvania, United States
yinuod@andrew.cmu.edu

Zimeng Song
Tsinghua university
Beijing, China
songzm19@mails.tsinghua.edu.cn

Stephanie Milani
Carnegie Mellon University
Pittsburgh, Pennsylvania, United States
smilani@andrew.cmu.edu

Cleotilde Gonzalez
Carnegie Mellon University
Pittsburgh, Pennsylvania, United States
coty@cmu.edu

Fei Fang
Carnegie Mellon University
Pittsburgh, Pennsylvania, United States
feifang@cmu.edu

ABSTRACT

Due to the increase in cyber threats and the dynamic nature of cyber attacks, it is important to use adaptive strategies to decide when and where to deploy the defense resources and engage the attackers through deceptive cyber artifacts to reduce attack effectiveness. However, most existing work that uses game-theoretic framework to analyze cyber deception does not consider defender and attacker's ability to adapt to real-time observations. In this paper, we propose an Adaptive Cyber Deception Game, a two-player Markov game model that accounts for sequential moves between defender and attacker in a cyber deception scenario on an attack graph. We also study the use of a reinforcement learning algorithm – Proximal Policy Optimization (PPO) – with self-play in this game. Preliminary experimental results show that the defender policies found by PPO are significantly better than a heuristic policy.

ACM Reference Format:
Yinuo Du, Zimeng Song, Stephanie Milani, Cleotilde Gonzalez, and Fei Fang. 2022. Learning to Play an Adaptive Cyber Deception Game. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Auckland, New Zealand, May 9–13, 2022*, IFAAMAS, 6 pages.

1 INTRODUCTION

Cyber space is continuously threatened [11, 18] by various forms of attacks, which are difficult to detect and mitigate, given the increasing number of devices in the network and the ever-growing amount of valuable data assets. Research efforts from various perspectives have been devoted to tackling cyber threats [16, 24, 47]. In recent years, deception has received significant attention as a promising measure the defender can take to slow down the attackers and waste their resources [2, 8, 9, 14, 17]. Through deception, defenders could confuse attackers during the reconnaissance phase or induce them to pursue a dead end. More concretely, cyber deception could be achieved by information dissimulation or information simulation. Dissimulation strategy consists of partially or fully concealing the true state of reality. For example, masking [1, 7] attempts to hide or erase crucial information. On the other hand, the simulation strategies exhibit false information. One example is decoying [30, 49], where network defenders deploy honeypots or other kinds of decoys to lure attackers. In the real world, decoying techniques like honeypots may be created via port hardening or by putting fake content in computer systems [4].

There have been some existing works that model the cyber deception problem as a two-player game between the defender and the attacker [12, 32, 37, 41, 43]. For example, [25] models vulnerable systems as an attack graph, with nodes and edges representing the abstract states and their transition relations. However, most of these works assume the players' decision-making to be one-shot, i.e., the attacker chooses one target to attack or choose a path in the attack graph, and the defender chooses a deceptive and protective strategy ahead of time, which is not in alignment with the reality [21]. Attackers usually have limited knowledge about the network topology at the early stage regardless of their level of capability unless they are insiders. Therefore, they need to keep collecting information as they proceed in the attacking process. It is also a waste of resources from the defender's perspective to deploy defenses all at once since the attacker will only go through one route in the graph. With the assistance of intrusion detection and network forensics tools [36, 45], it is possible for the defender to have partial observations of attacker's status and take actions accordingly. Thus, the defender needs to use a strategy that is adaptive to the attacker's actions.

To address this gap, we propose a new game model – Adaptive Cyber Deception Game. This is a two-player Markov game, and it is the first game model for cyber deception that accounts for the sequential moves in the defender-attacker interaction. The game is based on an attack graph [39] where each node can represent an abstract state as in [25] or a computer system in a network. Each node is associated with a value. In each time step of the game, the defender can take deceptive actions and allocate protective defense resources. The deceptive actions include adding a fake edge, hiding an existing edge, or changing the appeared value of a node. The attacker can move from one node to a neighboring node in each step. We also study the use of reinforcement learning (RL) algorithms in solving the game. Specifically, we use Proximal Policy Optimization (PPO) [38] with self-play to learn the policies for the players. Since the players have partial observability of the state (e.g., the location of the attacker), we use Long-Short Term Memory (LSTM) architecture in the policy network to leverage historical observations. We evaluate the performance of PPO-derived policies and show that they outperform heuristic policies significantly.

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Auckland, New Zealand.

2 RELATED WORK

Attack graph. We build our game model on an attack graph. Attack graph [39] is a commonly-used, versatile modeling tool for security challenges in different domains. As networks of hosts continue to grow in size and complexity, attack graphs become handy to model the attack surface [28, 44], and an attacker's arsenal of atomic attacks, expose the effects of individual host vulnerabilities and uncover multi-stage vulnerabilities introduced by their interactions. In practice, an attack graph is typically generated automatically by model checking [5]. Each node in the attack graph represents an attack status, while each edge represents an atomic attack that the intruder can use to change the network status or collect knowledge about it.

Cyber deception. A recent review of the literature regarding deception in cybersecurity proposes a game-theoretic taxonomy that categorizes cyber deception into six types: perturbation, moving target defense, obfuscation, mixing, honey-x, and attacker engagement [3, 15, 17, 19, 31, 34]. Cyber deception has been proved to be effective in automated and dynamic systems [26, 35] to orchestrate decoy deployment. It is also modeled with game-theoretic approaches [40] which aim to formally reason about strategic behavior and find optimal deception strategies. The most relevant work to ours is [13], which proposed a game model to analyze adaptive cyber deception. However, their model considers an extensive-form game with high-level defense actions such as disabling login, setting up a decoy, and blocking the attacker. In contrast, our model is a Markov game on an arbitrary attack graph, with a detailed set of deceptive and protective actions. Besides, their work does not consider computational aspects, and the framework can only be applied to very small scenarios where the entire game tree can be enumerated and analyzed by hand. We not only provide a game model but also present a RL-based approach for solving the game.

Reinforcement learning (RL) for cyber security. RL can be used to train an agent to take sequential actions optimally with possibly limited prior knowledge about the environment and has been proved to perform well in highly dynamic cyber security environments [22, 33]. The incorporation of deep learning enabled an even larger number of applications to various aspects of cyber security [27]. Some attacking scenarios involving multiple agents (attacker and defenders) are modeled using game-theoretic frameworks and solved using RL [10] or multi-agent RL [29, 46]. Our work also uses a RL-based approach to solve the proposed game.

3 ADAPTIVE CYBER DECEPTION GAME

In cyber security in the real world, one network is protected by multiple security analysts and attacked by multiple attackers with diverse motivations and different levels of attack capabilities. The capability of defenders to detect and respond to network status updates also varies. Depending on the network topology, there can be one or more hosts valuable to the adversaries. In the proposed *Adaptive Cyber Deception Game*, only one attacker is playing against one defender protecting an attack graph. The defender updates its defense deployment at a fixed frequency. The attacker propagates through the graph at the same speed. This setting is chosen for the sake of modeling simplicity. It is possible to extend the game from two-player to two-team and relax the frequency of defense deployment.

In each round, every agent takes action in turn. The defender chooses his strategy first; the uncaught attacker selects his strategy after surveillance. The attacker is assumed to take the presented information as truth. In this section, we describe the attack graph, the players' strategy space, and the players' payoff.

Attack Graph: Attack graphs $G = (N, E, V, P)$ represent attackers' possible attack paths. The nodes N represent attack status and can be interpreted at various granularity, from the tokens collected to the foothold in a network. The yellow nodes represent the entering node in the attack graph (with in degree equals zero) and the possible initial status of the attacker in the beginning. Each node is assigned a value v_e in resemblance to the value of digital assets. Each edge is assigned with a probability p_e to indicate the success probability of transiting from the source node to the destination node. The edges $E = E_{real} \cup E_{fake} = E_{visible} \cup E_{invisible}$ represent actions that attackers can take to transit between nodes. E_{real} represents the real edges in the attack graph. E_{fake} represents the available fake edges that the defender can choose to add. $|E_{fake}| = N(N-1)$ since there's one fake edge quota between each pair of nodes for a specific transit direction. $E_{fake} \cap E_{visible} = \emptyset$ when no fake edges are added. The model assumes that there's no more than one action that can bring the attacker from one state to the same successor state, and only unidirectional movement is allowed. Thus, there's no more than one real edge between each pair of nodes.

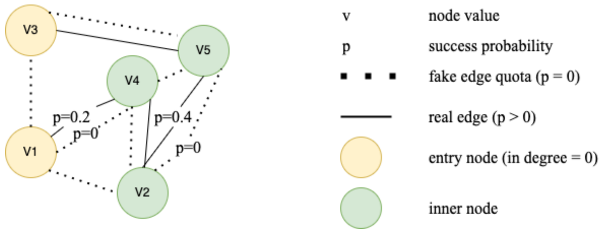

Figure 1: Attack Graph

Attacker Action Space A_a: The attacker can choose one outgoing edge of the current node as the next move to advance its status. He can also choose not to take action and stay at the same location for another time step.

Defender Action Space A_d $= \{(V', e_{hide}, e_{add}, e_{monitor}) | e_{hide} \in E_{real}, e_{monitor} \in E_{real}, e_{add} \in E_{fake}\}$*:* The defender can take two types of action: deceptive and protective actions. By deception, the defender can (i) hide a real edge, (ii) add a fake edge, (iii) modify the perceived values of a set of nodes. By protection, the defender can monitor an edge e to interrupt the attacker's movement. The hidden real edge will become invisible to the attacker. The added fake edge is monitored by nature. An attacker attempting to go through a fake edge will get caught by the defender.

Defenders are given a budget for defense. The particular action space A_d^tat timestamp t is updated based on the attack graph topology. If the location of the attacker is known to the defender (determined by threshold h), he will be able to deploy precisely targeted defenses through only defending the reachable nodes and edges starting from the attacker's location. No action will be taken when the attacker's location is known and there's no precise defense available. If the attacker's location is unknown, the defender will then have no choice but to make decisions out of a larger space. The defender is allowed to manipulate the perceived value of any number of unoccupied nodes, monitor a real edge, hide a real edge, and add a fake edge simultaneously at one timestamp. The defender can choose not to change the values of modifiable nodes by keeping them the same as previous round. The defender is also allowed to opt out for edge hiding, adding, or monitoring. All above actions are temporary and can last for one single timestamp. That is, the fake edge will disappear if not chosen to be added in the next step, and the hidden real edge will reveal itself if not chosen to be hidden in the next step. The defender needs to choose different real edges to hide or monitor if he decides to use both techniques.

Attacker's Observation $O_a = (loc, E^{'}, V^{'})$: The observation space of the attacker consists of three parts: the location of himself, the visible edges, the perceived node values. The observation is updated at each timestamp determined by the state transition of the attacker himself and the action of the defender. $E^{'} = E_{visible} \cap e_{ij,i=loc}$, meaning that the attacker observe only the forward edges within the horizon of one hop. $V^{'}_t = V^{'}_{t-1} \cup \{v_i | i \in \{n | e_{nj} \in E^{'}\}\}$, meaning that the attacker keeps accumulating knowledge while pivoting through the network.

Defender's Observation $O_d = (loc|h, V, V^{'}, E_{visible}, budget)$: The observation space of the defender consists of five parts: the attacker's location (at probability h), the real node values, the manipulated node values, visible edges, and the available budget. The defender might be able to know the location of the attacker at probability h, to reflect the difficulty of detecting and keeping track of the attackers.

Reward Scheme: The attacker gets immediate reward for each successful establishment of foothold on a node. If finally get caught, he will get a 10 points penalty but get to keep the values collected. The assumption is that the attacker is able to get the benefit out of the reached nodes and cause irreversible loss to the defender.

$$R_a(S_t) = \begin{cases} v_{n_t}, & \text{if t } \leq \text{ episode length \& } n_t \neq n_{t-1} \text{ \& safe} \\ 0, & \text{if t } \leq \text{ episode length \& } n_t = n_{t-1} \text{ \& safe} \\ -10, & \text{if caught} \end{cases} \tag{1}$$

The defender in the contrast gets his reward at the end of each episode to capture his oblivious about attacker's status.

$$R_d(S_t) = \begin{cases} 0, & \text{if t < episode length} \\ \sum_{n \in N_{exploited}} v_n, & \text{if t = episode length\& didn't catch} \\ 10 - \sum_{n \in N_{exploited}} v_n, & \text{if t = episode length \& caught} \end{cases} \tag{2}$$

This reward mechanism is designed to incentivize the attacker to pivot through the most valuable route in the network and try to avoid the defender in the meanwhile. While the defender is motivated to catch the attacker as fast as possible before losing too much information to the attacker.

Powerful Attacker: Attackers in the real world can possess different levels of skills and knowledge. We also consider the possibility of being confronted with powerful attackers who can see through the deceptions in the attack graph and evade the traps. Specifically, they might be able to learn the ground truth values of nodes, identify hidden real edges and differentiate fake edges.

4 LEARNING TO PLAY ADAPTIVE CYBER DECEPTION GAME

To solve the adaptive cyber deception game, we explore the use of an RL algorithm PPO with self-play. PPO is a policy gradient-based algorithm targeting at optimizing the policy directly. It has shown great success in various single-agent and multi-agent problems such as Atari games [48], real-time strategy games [6] and robotic control [42]. The policy network of PPO takes a vector describing the state as input and outputs a probability distribution over available actions.

Given that the set of valid actions varies per step and the number of valid actions is large, an invalid action masking scheme is utilized to prevent the agents from selecting actions that are invalid or cannot be performed in the current state. The policy network of PPO is customized to take a binary action mask vector in companion with the observation vector. The customized network outputs the sum of action logits and the logarithm of the action mask in which the probabilities of invalid actions are zero.

The set of valid actions of the attacker is conditioned on its location on the attack graph and the visibility of real and fake edges. An attacker is only allowed to move along visible outgoing edges from its foothold. The available actions of the defender are dependent on the occupied status of nodes, the budget remained, and the knowledge about attacker's location. The defender can not manipulate the value of nodes occupied by the attacker. Actions with costs higher than his available budget are also disabled. When the attacker's location is known, only defenses on nodes at the next layer of the attacker's foothold are considered valid.

The game state is partially observable for both defender and attacker. Nodes $n \in N$ in attack graph $G = (N, E, V, P)$ are indexed from 1 to N. Real edges $e_{real} \in E_{real}$ are indexed from 0 to $|E_{real}| - 1$, while fake edges $e_{fake} \in E_{fake}$ are indexed from $|E_{real}|$ to $|E_{real}| + N(N-1)$. Defender's observation vector is the concatenation of the one-hot encoding of attacker's location, the real node value integer vector, the manipulated node value integer vector, a binary vector indicating the visibility of edges, and an integer representing remained defense budget. The encoding of attacker's location is a zero vector when it is unknown. Attacker's observation is the concatenation of the one-hot encoding of his location, the manipulated node value integer vector, and a binary vector indicating the visibility of edges.

Since it is possible for the attacker to stay at his original location because of a failure to make a state transition or a deliberate decision to stay still, the observation vector is further encoded by an LSTM [20] network for the attacker to observe conflicting node values and different visible edges in the history.

PPO is initially designed to be a single-agent RL algorithm. To solve our game, we follow other works that use PPO for multi-agent setting, e.g., [6] and run PPO with self-play. Concretely, the defender and attacker will each maintain a policy network. In each training step, we collect experiences based on the players' current policies, and then run one-step update for the defender's and attacker's network parameters using PPO separately.

5 EXPERIMENT

We compare the performance of PPO-based policies against heuristic policies. We use the RLlib [23] implementation of the PPO policy network and LSTM network. The heuristic attacker always chooses to move to the observable node with the highest value. The heuristic defender will first try to change the perceived value of nodes to have the highest-valued node non-differentiable from others. If the budget is not exhausted after masking, the defender will then randomly hide a real edge, add a fake edge, or defend a real edge in the following steps.

The number of nodes and real edges in the attack graph is fixed as 4 and 5, respectively. The graph topology is randomly sampled in each episode. It is guaranteed that there exists only one real edge between each pair of nodes. The value of each node is an integer randomly sampled from $[0, 2]$. The defender's budget is also randomly sampled from the range of $[1, U + 1]$. The upper bound $U = \min_{v^* \in \mathbb{N}} \sum_{i \in N} |v_i - v^*|$ is the minimum cost it takes to make the value of all nodes the same. The attacker is assigned to one of the entry nodes randomly at the beginning.

We run 100000 episodes for training the PPO defender and the PPO attacker. After training, we evaluate the policies by two performance metrics: 1) *exploitability* and 2) *defender's utility* against a diverse set of opponents.

The exploitability of a policy pair (π_a, π_d) from the attacker a and the defender d is calculated by $Exp(\pi_a, \pi_d) = Exp_a(\pi_a, \pi_d) + Exp_d(\pi_a, \pi_d)$, where

$$\begin{cases} Exp_a(\pi_a, \pi_d) = U_a(\pi_a, \pi_d) - U_a(\pi_a, BR_d(\pi_a)) \\ Exp_d(\pi_a, \pi_d) = U_d(\pi_a, \pi_d) - U_d(BR_a(\pi_d), \pi_d). \end{cases} \quad (3)$$

U_a and U_d are attacker and defender's utilities. An agent's utility of a policy pair is evaluated by running the policy against another policy for 1000 episodes and calculating the average accumulative reward of this agent. $BR_d(\pi_a)$ is the defender's best response against an attacker's policy π_a. $BR_a(\pi_d)$ is the attacker's best response against an defender's policy π_d. Specifically, in our zero-sum game,

$$Exp(\pi_a, \pi_d) = -\left(U_a(\pi_a, BR_d(\pi_a)) + U_d(BR_a(\pi_d), \pi_d)\right). \quad (4)$$

Finding the best response to a policy is difficult in our game model. To calculate one agent's best response against another agent's policy, we train a PPO policy for this agent by running 1000000 episodes with another agent's fixing policy.

The exploitability of a pair of policies measures the distance from the Nash equilibrium. Fig 2 shows the exploitability of different combinations of policies. The exploitability of the PPO defender evaluated against the PPO attacker is relatively small, which means that this pair of policies is closer to a Nash equilibrium. Also, with much smaller exploitability, the PPO attacker is much better than the heuristic attacker.

We also compare the PPO defender with the heuristic defender by showing their utilities against multiple attackers. The attackers we use are 1) the heuristic attacker; 2) the PPO attacker trained against the heuristic defender; 3) the PPO attacker trained against a PPO defender in the PPO self-play.

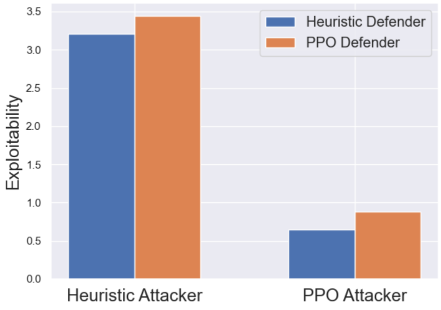

Figure 2: Exploitability of the heuristic defender and the PPO defender evaluated against the heuristic attacker and the PPO attacker

To test the robustness of our PPO defender policy, we further evaluate the PPO defender against multiple powerful attackers who can see through the deceptions in the attack graph. The powerful attackers we use are 4) the powerful heuristic attacker who always chooses to move through the real edge to the node with the real highest value; 5) a powerful attacker trained with PPO policy against the heuristic defender; 6) a powerful attacker trained with PPO policy against a PPO defender in the PPO self-play.

We run 1000 episodes to calculate the defender's utility for each pair of defenders and attackers. Fig 3 shows the evaluated results. With a higher utility against each attacker, the PPO Defender always outperforms the heuristic defender. Meanwhile, each defender's utility decreases from left to right, which verifies that the powerful attacker is more aggressive and shows that the PPO attacker causes more loss to the defender.

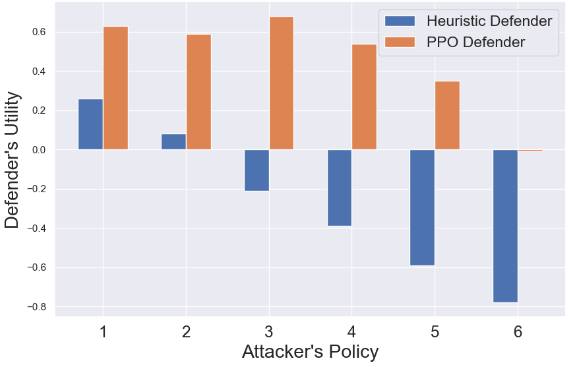

Figure 3: Defenders' utilities against multiple attackers

6 DISCUSSION AND FUTURE WORK

The preliminary results of our experiments to some extent show promise for using RL to learn deceptive defense policies on a cyber attack graph game. However, the scale of the attack graph and node values are limited. To determine whether the RL algorithm is indeed more beneficial compare to heuristic models, we are planning to run more experiments on attack graphs of various sizes and node values comparing the performance of PPO policy against heuristic policy.

We also plan to explore the effectiveness of multi-agent RL algorithms rather than utilizing single-agent RL algorithm directly. A further step is to extend the model to a multi-attacker multi-defender setting to investigate the interaction between cooperative attackers and cooperative defenders.

ACKNOWLEDGMENTS

This research was sponsored by the U.S. Army Combat Capabilities Development Command 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 Combat Capabilities Development Command Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

[1] Palvi Aggarwal, Omkar Thakoor, Shahin Jabbari, Edward A Cranford, Christian Lebiere, Milind Tambe, and Cleotilde Gonzalez. 2022. Designing Effective Masking Strategies for Cyberdefense through Human Experimentation and Cognitive Models. *Computers & Security* (2022), 102671.

[2] Ehab Al-Shaer, Jinpeng Wei, W Kevin, and Cliff Wang. 2019. Autonomous cyber deception. *Springer* (2019).

[3] Mohammed H Almeshekah and Eugene H Spafford. 2016. Cyber security deception. In *Cyber Deception*. 25–52.

[4] Mohammed H Almeshekah, Eugene H Spafford, and Mikhail J Atallah. 2013. Improving security using deception. *Center for Education and Research Information Assurance and Security, Purdue University, Tech. Rep. CERIAS Tech Report* 13 (2013), 2013.

[5] Mridul Sankar Barik, Anirban Sengupta, and Chandan Mazumdar. 2016. Attack Graph Generation and Analysis Techniques. *Defence science journal* 66, 6 (2016).

[6] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. 2019. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680* (2019).

[7] Warren Cabral, Craig Valli, Leslie Sikos, and Samuel Wakeling. 2019. Review and analysis of cowrie artefacts and their potential to be used deceptively. In *2019 International Conference on computational science and computational intelligence (CSCI)*. IEEE, 166–171.

[8] Edward Cranford, Cleotilde Gonzalez, Palvi Aggarwal, Sarah Cooney, Milind Tambe, and Christian Lebiere. 2020. Adaptive cyber deception: Cognitively informed signaling for cyber defense. In *Proceedings of the 53rd Hawaii International Conference on System Sciences*.

[9] Fabio De Gaspari, Sushil Jajodia, Luigi V Mancini, and Agostino Panico. 2016. Ahead: A new architecture for active defense. In *Proceedings of the 2016 ACM Workshop on Automated Decision Making for Active Cyber Defense*. 11–16.

[10] Richard Elderman, Leon JJ Pater, Albert S Thie, Madalina M Drugan, and Marco A Wiering. 2017. Adversarial Reinforcement Learning in a Cyber Security Simulation.. In *ICAART (2)*. 559–566.

[11] Jose Esteves, Elisabeth Ramalho, and Guillermo De Haro. 2017. To improve cybersecurity, think like a hacker. *MIT Sloan Management Review* 58, 3 (2017), 71.

[12] Fei Fang. 2021. Game Theoretic Models for Cyber Deception. In *Proceedings of the 8th ACM Workshop on Moving Target Defense*. 23–24.

[13] Kimberly Ferguson-Walter, Sunny Fugate, Justin Mauger, and Maxine Major. 2019. Game theory for adaptive defensive cyber deception. In *Proceedings of the 6th Annual Symposium on Hot Topics in the Science of Security*. 1–8.

[14] Kimberly J Ferguson-Walter, Dana S LaFon, and TB Shade. 2017. Friend or faux: deception for cyber defense. *Journal of Information Warfare* 16, 2 (2017), 28–42.

[15] Nandan Garg and Daniel Grosu. 2007. Deception in honeynets: A game-theoretic analysis. In *2007 IEEE SMC Information Assurance and Security Workshop*. IEEE, 107–113.

[16] Manuel Gil Pérez, Alberto Huertas Celdrán, Pietro G Giardina, Giacomo Bernini, Simone Pizzimenti, Félix J García Clemente, Gregorio Martínez Perez, Giovanni Festa, and Fabio Paglianti. 2021. Mitigation of cyber threats: Protection mechanisms in federated SDN/NFV infrastructures for 5G within FIRE+. *Concurrency and Computation: Practice and Experience* 33, 7 (2021), 1–1.

[17] Cleotilde Gonzalez, Palvi Aggarwal, Christian Lebiere, and Edward Cranford. 2020. Design of dynamic and personalized deception: A research framework and new insights. (2020).

[18] Clement Guitton. 2013. Cyber insecurity as a national threat: overreaction from Germany, France and the UK? *European Security* 22, 1 (2013), 21–35.

[19] Kristin E Heckman, Michael J Walsh, Frank J Stech, Todd A O'boyle, Stephen R DiCato, and Audra F Herber. 2013. Active cyber defense with denial and deception: A cyber-wargame experiment. *computers & security* 37 (2013), 72–77.

[20] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[21] Sushil Jajodia, George Cybenko, Peng Liu, Cliff Wang, and Michael Wellman. 2019. *Adversarial and uncertain reasoning for adaptive cyber defense: control-and game-theoretic approaches to cyber security*. Vol. 11830. Springer Nature.

[22] Chao Li, Junjuan Xia, Fagui Liu, Dong Li, Lisheng Fan, George K Karagiannidis, and Arumugam Nallanathan. 2021. Dynamic offloading for multiuser muti-CAP MEC networks: A deep reinforcement learning approach. *IEEE Transactions on Vehicular Technology* 70, 3 (2021), 2922–2927.

[23] Eric Liang, Richard Liaw, Robert Nishihara, Philipp Moritz, Roy Fox, Ken Goldberg, Joseph Gonzalez, Michael Jordan, and Ion Stoica. 2018. RLlib: Abstractions for Distributed Reinforcement Learning. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, 3053–3062. https://proceedings.mlr.press/v80/liang18b.html

[24] Leandros Maglaras, Mohamed Amine Ferrag, Abdelouahid Derhab, Mithun Mukherjee, and Helge Janicke. 2019. Cyber security: From regulations and policies to practice. In *Strategic innovative marketing and tourism*. Springer, 763–770.

[25] Stephanie Milani, Weiran Shen, Kevin S Chan, Sridhar Venkatesan, Nandi O Leslie, Charles Kamhoua, and Fei Fang. 2020. Harnessing the power of deception in attack graph-based security games. In *International Conference on Decision and Game Theory for Security*. Springer, 147–167.

[26] David Nguyen, David Liebowitz, Surya Nepal, and Salil Kanhere. 2021. HoneyCode: Automating Deceptive Software Repositories with Deep Generative Models. (2021).

[27] Thanh Thi Nguyen and Vijay Janapa Reddi. 2019. Deep reinforcement learning for cyber security. *IEEE Transactions on Neural Networks and Learning Systems* (2019).

[28] Eric Osterweil, Danny McPherson, and Lixia Zhang. 2014. The shape and size of threats: Defining a networked system's attack surface. In *2014 IEEE 22nd International Conference on Network Protocols*. IEEE, 636–641.

[29] Martina Panfili, Alessandro Giuseppi, Andrea Fiaschetti, Homoud B Al-Jibreen, Antonio Pietrabissa, and Franchisco Delli Priscoli. 2018. A game-theoretical approach to cyber-security of critical infrastructures based on multi-agent reinforcement learning. In *2018 26th Mediterranean Conference on Control and Automation (MED)*. IEEE, 460–465.

[30] Kyungmin Park, Samuel Woo, Daesung Moon, and Hoon Choi. 2018. Secure cyber deception architecture and decoy injection to mitigate the insider threat. *Symmetry* 10, 1 (2018), 14.

[31] Jeffrey Pawlick, Edward Colbert, and Quanyan Zhu. 2019. A game-theoretic taxonomy and survey of defensive deception for cybersecurity and privacy. *ACM Computing Surveys (CSUR)* 52, 4 (2019), 1–28.

[32] Jeffrey Pawlick and Quanyan Zhu. 2021. *Game Theory for Cyber Deception*. Springer.

[33] Ratih Hikmah Puspita, Syed Danial Ali Shah, Gyu-min Lee, Byeong-hee Roh, Jimyeong Oh, and Sungjin Kang. 2019. Reinforcement learning based 5G enabled cognitive radio networks. In *2019 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 555–558.

[34] Neil C Rowe and E John Custy. 2007. Deception in cyber attacks. In *Cyber warfare and cyber terrorism*. IGI Global, 91–96.

[35] Md Sajidul Islam Sajid, Jinpeng Wei, Basel Abdeen, Ehab Al-Shaer, Md Mazharul Islam, Walter Diong, and Latifur Khan. 2021. SODA: A System for Cyber Deception Orchestration and Automation. In *Annual Computer Security Applications Conference*. 675–689.

[36] Brian Salazar. [n.d.]. Survey on Real Time Security Mechanisms in Network Forensics. *International Journal of Computer Applications* 975 ([n. d.]), 8887.

[37] Aaron Schlenker, Omkar Thakoor, Haifeng Xu, Milind Tambe, Phebe Vayanos, Fei Fang, Long Tran-Thanh, and Yevgeniy Vorobeychik. 2018. Deceiving cyber adversaries: A game theoretic approach. In *International Conference on Autonomous Agents and Multiagent Systems*.

[38] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[39] Oleg Sheyner, Joshua Haines, Somesh Jha, Richard Lippmann, and Jeannette M Wing. 2002. Automated generation and analysis of attack graphs. In *Proceedings 2002 IEEE Symposium on Security and Privacy*. IEEE, 273–284.

[40] Zheyuan Ryan Shi, Ariel D Procaccia, Kevin S Chan, Sridhar Venkatesan, Noam Ben-Asher, Nandi O Leslie, Charles Kamhoua, and Fei Fang. 2020. Learning and planning in the feature deception problem. In *International Conference on Decision and Game Theory for Security*. Springer, 23–44.

[41] Arunesh Sinha, Fei Fang, Bo An, Christopher Kiekintveld, and Milind Tambe. 2018. Stackelberg security games: Looking beyond a decade of success. IJCAI.

[42] Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. 2018. Sim-to-real: Learning agile locomotion for quadruped robots. *arXiv preprint arXiv:1804.10332* (2018).

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

[44] Christopher Theisen, Nuthan Munaiah, Mahran Al-Zyoud, Jeffrey C Carver, Andrew Meneely, and Laurie Williams. 2018. Attack surface definitions: A systematic literature review. *Information and Software Technology* 104 (2018), 94–103.

[45] Ednard T Toivo, Alicia W Kambrude, and Attlee M Gamundani. 2021. Packet Forensic Analysis in Intrusion Detection Systems. In *2021 3rd International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*. IEEE, 1–4.

[46] Mason Wright, Yongzhao Wang, and Michael P Wellman. 2019. Iterated Deep Reinforcement Learning in Games: History-Aware Training for Improved Stability. In *Proceedings of the 2019 ACM Conference on Economics and Computation*. 617–636.

[47] Qichao Xu, Zhou Su, and Rongxing Lu. 2020. Game theory and reinforcement learning based secure edge caching in mobile social networks. *IEEE Transactions on Information Forensics and Security* 15 (2020), 3415–3429.

[48] Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, and Yi Wu. 2021. The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games. *arXiv preprint arXiv:2103.01955* (2021).

[49] Li Zhang and Vrizlynn LL Thing. 2021. Three decades of deception techniques in active cyber defense-retrospect and outlook. *Computers & Security* 106 (2021), 102288.