Human and AI Decision-Making in Cybersecurity: A Multiagent Modeling Perspective

Yinuo Du

June 2024

Software and Societal Systems Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:

Fei Fang	(Co-chair)	Carnegie Mellon University
Cleotilde Gonzalez	(Co-chair)	Carnegie Mellon University
Christian Lebiere		Carnegie Mellon University
Prashanth Rajivan		University of Washington
Tiffany Bao		Arizona State University

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.*

Abstract

Decision-making in cyber defense is a complex challenge arising from multiple factors. First, it involves strategic interaction among multiple decision-makers: attackers, defenders, and end-users. Second, the diversity of adversaries, ranging from sophisticated nation-state actors to opportunistic script kiddies, adds another layer of difficulty. Third, the network security status constantly evolves due to the progressive nature of attacks, making it even more challenging to maintain robust defenses. This challenge is exacerbated when integrating human and AI elements into decision-making. Limited research has been done to address this multifaceted challenge comprehensively. This thesis aims to integrate human and AI defense research to counteract the diverse and dynamic threats in network environments. Specifically, the thesis addresses four key challenges: reasoning about diverse adversary strategies, developing adaptive defenses in an evolving security landscape, exploring human-AI defense teams, and fostering cross-organizational collaboration. My completed, in-progress, and proposed work tackles these challenges using game theory, reinforcement learning, human behavior modeling, and human experimentation.

Contents

Chapter 1

Introduction

Cybersecurity problems are increasingly imminent. The Identity Theft Resource Center reported a 78% increase in data compromises in the US, with 3,205 incidents affecting 353 million individuals [12]. The sheer size of enterprise networks is immense, encompassing thousands to tens of thousands of devices. The attack surface is even larger, considering the diversity of devices and their inherent vulnerabilities. For example, a large enterprise network might include 10,000 devices, ranging from computers and servers to mobile devices and IoT gadgets [1]. This vast array of devices significantly increases potential entry points for cyberattacks. This problem is exemplified by delays in identifying and patching vulnerabilities. For instance, in 2017, the WannaCry ransomware attack [22] exploited unpatched vulnerabilities, affecting over 230,000 computers across 150 countries and causing damages estimated to be in the billions of dollars. Therefore, it is imperative to detect and mitigate cyber-attacks to ensure the secure operation of society's critical systems.

In response to this situation, a variety of cybersecurity technologies have been developed for system hardening through improved software security engineering [4] (to reduce vulnerabilities and attack surface) and layering security through defend-in-depth [43] (by adding encryption, access controls, firewalls, intrusion detection systems, and malware scanners, for example). In recent years, researchers have also started to investigate technologies to make network systems less homogeneous and less predictable [40]. With increased security comes increased maintenance overhead for system management and more trade-offs among security properties like Confidentiality, Integrity, and Availability (CIA). For example, having N-diverse servers mirroring the same content can increase availability because the attacker has to bring down all N variants. It also makes for a larger attack surface because a breach of any of them can compromise confidentiality. This is but one example of decision-making in cyber defense. In fact, virtually all techniques involve parameter choices both as individual standalone techniques and especially so when used in combinations [3].

Decision-making in cyber defense is a multiagent problem. It involves strategic interaction among multiple decision-makers, attackers, defenders, and end-users, with different objectives and information about the network and each other. At each time instant, the cyber network's security status and each agent's information depend, in general, on exogenous random events and all the agent's strategies; such strategies are not common knowledge among all agents. Furthermore, the degree to which each agent achieves his objectives depends on his strategy and all the other agents' unknown strategies. Adversaries can use these features to their advantage by taking undetectable actions or detectable actions that do not reveal their full intent. Diverse adversaries ranging from sophisticated nation-state actors to opportunistic script kiddies can differ from each other across several dimensions, including motivation, resource availability, target selection, persistence, and many other aspects. Under this condition, the configuration of defense strategies is a formidable problem.

This challenge is exacerbated when integrating human and AI elements into decision-making. Deciding the level of autonomy to grant AI systems in making decisions is complex, especially when it comes to difficult decisions that involve trade-offs between short-term loss and long-term benefit, which often require human oversight. Establishing effective collaboration between human analysts and AI systems requires clear communication and a good understanding of each other's capabilities and limitations. AI tools need to be reliable, interpretable, and cooperative. Cybersecurity professionals need to be trained to understand and effectively use AI tools. The complexity is further heightened when collaboration is extended across different organizations. Cross-organizational collaboration for cyber defense is crucial for creating a robust and comprehensive security posture against increasingly sophisticated cyber threats. However, given the various objectives, priorities, structures, and cultures in different organizations, it is challenging to establish trust and maintain cooperation.

Limited research has been done to address this multifaceted challenge comprehensively. In this thesis, the goal is to integrate human and AI defense research to counteract the diverse and dynamic threats in network environments. Specifically, the thesis addresses four key challenges: reasoning about diverse adversary strategies, developing adaptive defenses in an evolving security landscape, exploring human-AI defense teams, and fostering cross-organizational collaboration. My completed, in-progress, and proposed work tackles these challenges using game theory, reinforcement learning, human behavior modeling, and human experimentation.

The thesis proposal will be structured as follows: I will first review related work on human and AI decision-making in cyber defense in Chapter 2. Next, I will describe projects I have initiated (Chapter 3) or planned (Chapter 4). Among my completed and in-progress work, my contributions include:

- A human-like adversary emulation method, an interactive defense game, and a human-subject experiment show that training against a human-like adversary is necessary to be prepared against diverse adversary strategies.
- A two-player Markov game model that accounts for sequential moves between defender and attacker and simulation experiments demonstrating the potential to use reinforcement learning for adaptive cyber defense.
- A semi-supervisory human-AI teamwork paradigm, a team defense game, and a human-subject experiment show that different skill levels and decision-making styles impact the teamwork process and outcome.
- An empirical evaluation of how to foster cross-organizational cooperation through manipulating the incentive structure and mutual information and a cognitive model to explain the cooperative behavior in a multi-player prisoner's dilemma.

Lastly, in Chapter 5, I will outline a timeline to complete the work in Chapter 4 over the next year.

Chapter 2

Challenges and Related Work

2.1 Reasoning About Diverse Adversary Strategies

Attackers can be script kiddies, state hackers, organized crime groups, insider attackers, hobbyists, hacktivists, legitimate penetration testers, or terrorists. Their role in cyberspace can differ regarding skill, knowledge, resources, access, and motives or SKRAM [54]. In industry, enterprises conduct attack mimicry through red teaming, where the security teams try to break into an organization's network, identifying vulnerabilities along the way. Red teams, unfortunately, can be difficult to employ – conducting a red team assessment is a largely manual task, and the personnel costs alone are fairly significant.

Computational adversary simulation aims to automate red team actions and reduce time costs. To this end, several simulations have been designed to explore network intrusion and other forms of cyberattacks. For example, Kotenko [38] modeled a DDoS attack, and Razak et al. [58] simulated network intrusions. However, these simulations do not specifically portray the attacker. Early models contained pre-scripted, static patterns for attacker agents to follow [27]. These models eventually gave way to graph-based [36] and state-based [2] attack simulation methods, which provide a useful characterization of the attacker's profile, such as goals, starting points, and available time. Similar to MulVal [52], this group of simulation methods models and stores generic attack patterns with pre-conditions and post-conditions in a knowledge base. Additional attack pattern attributes include the cost of attempts, execution time, base success probability, and maximum attempts. Together with structural models that declare the host and its installed components, the knowledge base is merged into an attack graph that serves as the base for selecting and scheduling the start and end of the attacker's actions.

Despite the technical fidelity, most automated adversary simulation methods ignored the social context and lacked a dynamic behavior component [34]. Human attackers have varying levels of risk tolerance, which might affect their choice of target and attack methods [73]. Human attackers can also learn from their experiences [39], adapt to defenses they encounter, and modify their strategies accordingly, which makes them more dangerous over time as they become more adept at evading detection and exploiting vulnerabilities. In this thesis, I aim to design a human-like adversary simulation that can capture the behavioral complexities of the human adversary.

2.2 Developing Adaptive Defenses in Evolving Security Status

The network security status is constantly evolving due to the progressive nature of attacks. Attackers can use their capabilities to attack and capture computers/hosts. Each attack unfolds in multiple stages [48], from reconnaissance to exploitation, privilege escalation, to data exfiltration [70]. Exploiting one vulnerability creates additional opportunities for the attacker [33]. Advanced persistent attackers that continually refine their techniques and adapt to defensive measures can maintain long-term access for years [63]. Through consistency checks and fingerprinting techniques, attackers can detect the existence of cyber deception techniques, which might trigger further investigation, validation, and counter-deception [21]. Defenders must continuously monitor the network and the adversary and adapt deception techniques accordingly.

Game theory provides a foundational set of mathematical tools for modeling the strategic interaction between the defender and the attacker. A special class of games with wide security applications is the Stackelberg Security Game (SSG) [20]. Several works have used the SSG to formulate problems in cybersecurity. Thakoor et al. [65] model the cyber deception problem as an SSG between the defender and the attacker, where the defender allocates defense resources, i.e., honeypots, in the network, and makes them appear attractive and valuable. The attacker observes the defender's strategy by scanning the network. If the deception techniques are effective, the attacker will be induced into the honeypots rather than the real assets. [45] 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 chooses 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 [32]. Reinforcement learning (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 [41, 56]. Incorporating deep learning enabled an even larger number of applications to various aspects of cyber security [50]. Some attacking scenarios involving multiple agents (attacker and defenders) are modeled using game-theoretic frameworks and solved using RL [19] or multi-agent RL [68, 53]. In this thesis, I aim to design a game model to capture the sequential moves in the defender-attacker interaction and solve the game through RL.

2.3 Exploring Human-AI Defense Teams

Despite the successful applications of AI for cybersecurity in intrusion detection, malware analysis, and many other areas, AI technologies are mostly used as decision-support tools rather than equal teammates of human analysts. The drastic growth of the attack space has made using such highly capable autonomous agents, both physical and software-oriented, in cyber security a necessity [14]. Empirical research showed that team performance can differ depending on whether the AI is viewed more as a tool or a legitimate teammate [74]. Various factors, including predictability, directability, and common ground, could influence how human teammates perceive AI teammates [37]. A panelist of cyber security experts discussed the challenges of incorporating AI into cyber security teams [11]. One key challenge is balancing AI systems' autonomy with human oversight [29]. Too much autonomy might lead to undesirable outcomes if the AI acts in ways not anticipated or desired by its human counterparts. Conversely, too much control might negate the benefits of using AI, such as its ability to operate at high speeds and handle large datasets. Another challenge is to determine the appropriate adaptivity of the AI teammate. According to a survey with cyber incident response experts [28], dynamic, human-like adaptivity is vital for the success of a human-AI team. However, they also emphasized that it is important for humans to train with AI and have a better understanding of their AI teammates. Finally, it is unclear how to integrate AI into the existing workflows. Cyber protection teams (CPT) possess a variety of roles that are tasked with considering how mission, policy, organization, process, and technology can and are affected by computer security incidents [51]. Most CPTs follow the National Institute of Standards and Technology (NIST) phases of incident response: preparation, identification, containment, eradication, and recovery [46]. It is unclear how to integrate an AI agent seamlessly into such work cycles. In this thesis, I aim to explore the potential paradigms for human AI team defense and contribute insights on the design of AI teammates in cyber defense.

2.4 Fostering Cross-Organizational Collaboration

Cyber attacks are well-organized crimes. Adversaries have been exchanging experience and knowledge and acting collectively to launch coordinated attacks [13, 75]. Their coordinated strategies help them succeed and extend the time during which they remain undetected in the systems [75]. However, defenders across organizations are less well coordinated. Therefore, it is important to investigate the factors contributing to sharing cyber-threat information among defenders. In recent times, several initiatives have been taken to improve trusted partnerships, such as the Cyber Information Sharing and Collaboration Program (CISCP), the European Cybercrime Center (EC3), and the Cyber Defense Alliance (CDA) for the exchange of threat and vulnerability information to improve the collective capabilities of cyber defense [59, 60, 9]. Despite these initiatives and efforts, the trend is that organizations do not share cyber-threat information on a large scale[49]. Organizations hesitate to share cyber threat data because of the possibility that this information would be misused for attacks or hinder their reputation in the market. Shared information can also be delayed and irrelevant when received. For example, in a recent Colonial Pipeline Company attack, the company did not disclose the incident or sensitive technical information about the attack on time to the Cybersecurity and Infrastructure Security Agency (CISA), and timely disclosure of the information would have been helpful in preventing similar attacks against other targets[31]. Finally, a ”free-rider problem” can discourage organizations from sharing their attacks and vulnerabilities, as some would only consume information to their own advantage but do not contribute [67]. Thus, the current situation creates a dilemma for collaboration among defenders: while they realize the benefits of sharing cyber attack information, they also realize the potential costs and risks. Research communities have focused mainly on developing game theory formulations of information sharing and developing game theory equilibrium-based solutions to learn their effects; another main focus has been on technical solutions such as secure information exchange and trusted computing [75]. In this thesis, I empirically evaluate the theories about fostering collaboration among cross-organizations through human experimentation and design cognitive models to explain their behavior.

Chapter 3

Completed and In-Progress Work

3.1 Training Defenders Against Human-Like Adversary

3.1.1 Interactive Defense Game [55]

We created a new Interactive Defense Game (IDG), a Django-based web application. IDG offers a web-based graphical user interface to allow human participants to perform the task proposed in our adapted CAGE scenario [55].

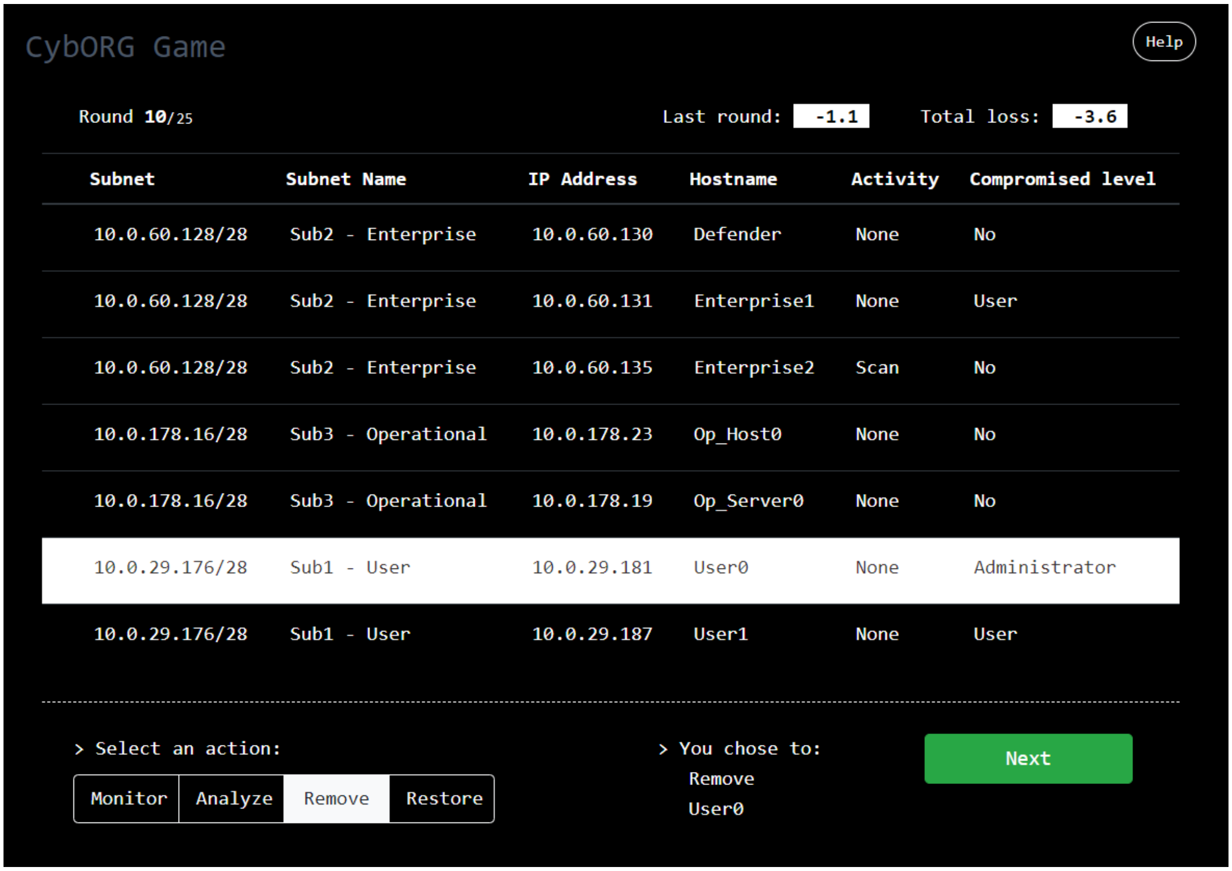
The IDG interface shown in Figure 3.1 consists of a central interactive table representation of the network and the related information on each host or server: IP Address, name, subnet, last detected activity, and compromised level. In this task, a human defender can select from a set of actions represented in the buttons on the bottom right of the screen: *Monitor, Analyze, Remove, Restore*.

Human defenders can select a host by clicking on its row in the table and then choose one of the four actions to perform on that particular host. Then, by clicking on the ”Next” button, the action selected takes effect, and the defender can see the result (i.e., points lost) from the execution of that action in the ”Last round” value. A new and updated version of the environment is presented to the human defender, demonstrating the network elements’ status (activity and compromised levels). The ”Last round” outcome provides immediate feedback regarding the effectiveness of the past action, and the ”Total loss” presents the human defender with a cumulative account of the loss during the episode.

3.1.2 Human-Like Attacker [15, 18]

We used three types of red agents: Two strategic attackers: (1) a highly efficient deterministic agent, $Beeline_{Red}$ and (2) a stochastic agent $Meander_{Red}$; and (3) a dynamic cognitive agent, IBL_{Red}. All red agents start at the host *User0* as their network entry point.

$Beeline_{Red}$ and $Meander_{Red}$ were proposed in the Cage Challenge scenario [62]. $Beeline_{Red}$ assumes that the attacker has prior knowledge of the network topology and moves directly to the operational server following the red path ($User0 \rightarrow User1 \rightarrow Enterprise1 \rightarrow Enterprise2 \rightarrow Op\_Server0$) (see Fig. 3.2) in a predictive and deterministic way. $Meander_{Red}$ assumes no prior



Round **10**/25   Last round: -1.1   Total loss: -3.6

Subnet	Subnet Name	IP Address	Hostname	Activity	Compromised level
10.0.60.128/28	Sub2 - Enterprise	10.0.60.130	Defender	None	No
10.0.60.128/28	Sub2 - Enterprise	10.0.60.131	Enterprise1	None	User
10.0.60.128/28	Sub2 - Enterprise	10.0.60.135	Enterprise2	Scan	No
10.0.178.16/28	Sub3 - Operational	10.0.178.23	Op_Host0	None	No
10.0.178.16/28	Sub3 - Operational	10.0.178.19	Op_Server0	None	No
10.0.29.176/28	Sub1 - User	10.0.29.181	User0	None	Administrator
10.0.29.176/28	Sub1 - User	10.0.29.187	User1	None	User

> Select an action:

> You chose to:
Remove
User0

Figure 3.1: Interactive Defense Game user interface.

knowledge about the network structure and behaves stochastic by choosing a random target to move forward.

In contrast, the cognitive agent, IBL_{Red}, is a novel contribution to this research, and it is a dynamic agent that learns from experience, as described in the IBLT section above. IBL_{Red} intends to represent cognitive memory-based decisions that can adapt their actions dynamically according to the conditions of the environment and the actions of the blue agent. The instances represent each decision and are structured with three elements.

State, s_a: The state of the instances of the IBL_{Red} agent is composed of features, f, constructed using the concept of Attack Models and Attack Graphs to model the security vulnerabilities of a network and their exploitation from the perspective of an attacker. Specifically, contextual characteristics include the success status of the previous action of the IBL_{Red} agent and the resources it occupied. A slot is dedicated to each resource type in various states, as shown in Fig. 3.3.

Specifically, a subnet can be newly *Detected* or already *Scanned*, while hosts are classified as *Detected*, *Scanned*, *Exploited (User)*, *Exploited (Root)*, *Impacted*.

The starting status denotes when the IBL_{Red} agent has just successfully established its foothold on the network on *User0*. At that point, only the *User* subnet is detected in addition to

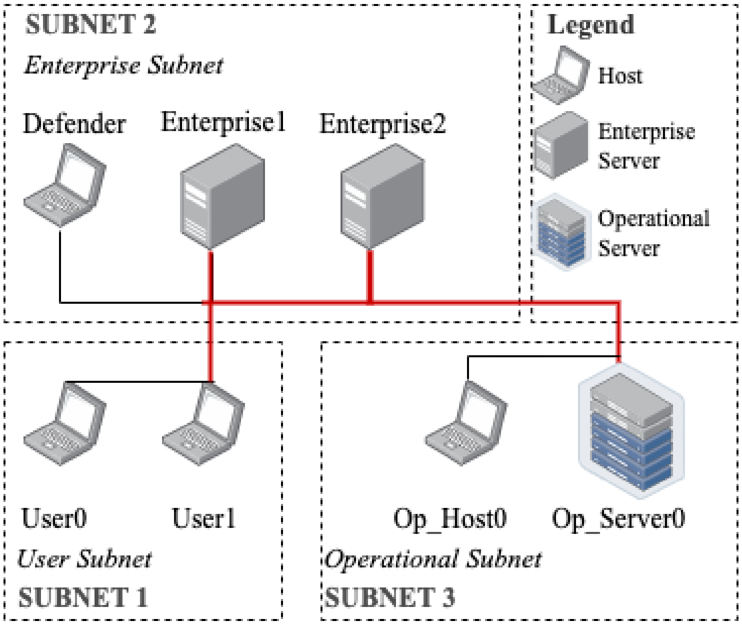


Figure 3.2: Adaptation of the Cage Challenge Network

its entry point *User0*, while the rest of the slots are empty. The most successful final state for the IBL_{Red} agent is where all hosts and servers are exploited at the *Root* level and when critical *Op_Server0* is impacted.

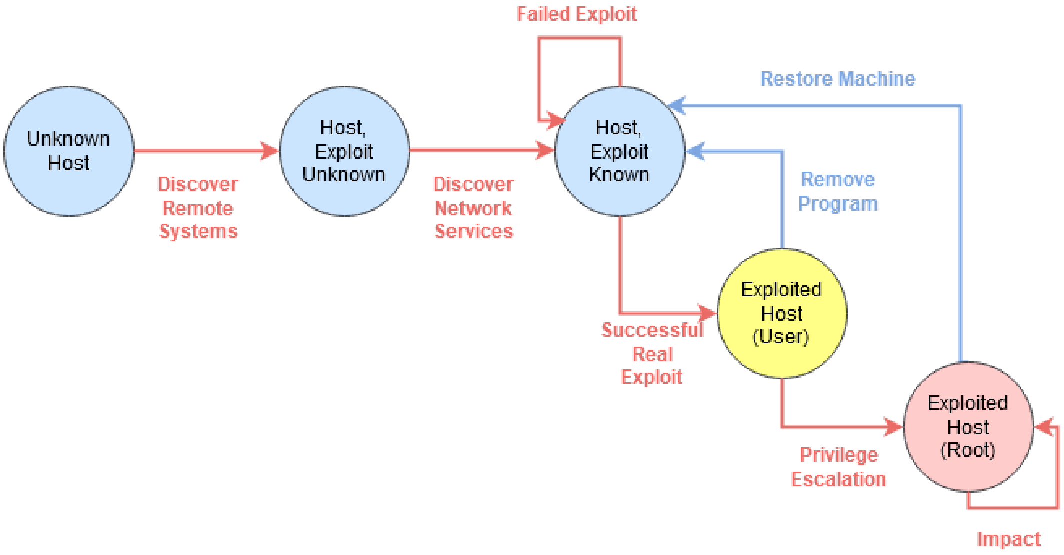


Figure 3.3: Effect of actions on the host state (diagram from [61])

Action Space, a_a: The action space for the IBL_{Red} agent is dynamically constructed at each step based on the status of each host on the network. Each action consists of a target host and an applicable command. As shown in Fig. 3.3, IBL_{Red} can choose to collect more information about hosts in the network or advance the attack status of known hosts.

Utility, z_a: A reward is calculated at each step, based on the attack status, as shown in Table 3.1. Higher rewards are assigned when the IBL_{Red} agent can access more significant systems. Only root access to the systems and successful impact on the operational server are rewarded. The IBL_{Red} agent receives a reward of 0 for any other action.

Event or Action	**Reward**
Administrator access on a Host	0.1
Administrator access on a Server	1
Successfully *Impact Op_Server0*	10

Table 3.1: Utility in the IBL model: Events and actions costs

Metrics for the Red Agent: The performance of the Red agent was evaluated for each episode, using the following metrics: **(1) Reward:** the cumulative rewards received during the execution of the scenario; **(2) Impact duration:** the average number of steps per episode that the Red agent successfully impacts the operational server; **(3) Progress:** the average number of steps per episode that the Red agent took to penetrate *Enterprise subnet* and *Operational subnet*; and **(4) Action frequency:** the average proportion of command usage at each step in an episode.

3.1.3 Experiment

Action frequency The use of defensive commands by the Human defender shows a difference when confronting the agent $Beeline_{Red}$ and $Meander_{Red}$ in contrast to the agent $IBL_{Red}^{Trained}$.

Human participants are more passive and take more *Analyse, Monitor* actions than *Remove, Restore* actions. However, as shown in Fig. 3.4, human participants present consistent preference for the choice of action throughout the course of 7 episodes.

Size of Option Space Fig. 3.5 presents the number of options available to the human during the 25 steps of the episodes. Human participants can alleviate their cognitive load by narrowing the option space when paired with $Beeline_{Red}$. The size of option space remains approximately the same in the two stochastic conditions, i.e., $Meander_{Red}$ and $IBL_{Red}^{Trained}$.

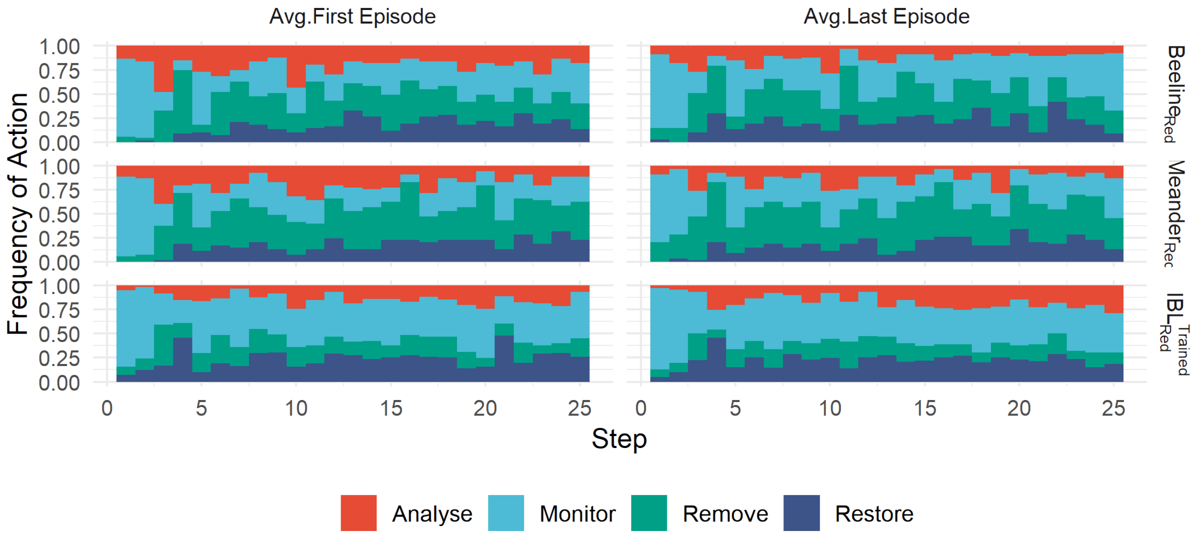


Figure 3.4: Evolution of average action frequency of Human Defender in the first (left) and the last episode (right)

Efficient and Inefficient Defenders In addition to comparing the average performance of red agents paired against all human defenders as is done in Figure **??**, we investigated the performance against efficient vs inefficient defenders. This split in performance was defined by over/under the mean attacker reward for each type of red agent. Comparing performance in this way allowed us to determine if there was a clear difference in how red agents fared against human participants who were better or worse able to learn attacker strategies. Additionally, to compare performance once human defenders had enough experience to learn the attacker strategy adequately, we limited the statistical analysis to later (¿4) trial episodes.

Fig.3.6 shows that the $IBL_{Red}^{Trained}$ had significantly higher reward against proficient defenders on later trials (Mean: 33.19 ± 33.31 (SD); Tukey's HSD p=0.040). Meanwhile, $Beeline_{Red}$ had a higher reward against the inefficient human defenders on later trials (Mean: 92.18 ± 53.36 (SD); Tukey's HSD p=0.041). These results demonstrate that the $IBL_{Red}^{Trained}$ strategy remained challenging for the higher-performing human defenders throughout the experiment. Additionally, the difficult but deterministic nature of the $Beeline_{Red}$ strategy was more difficult for lower-performing defenders.

One explanation for the worse defender performance in proficient defenders against $IBL_{Red}^{Trained}$ and inefficient defenders against $Beeline_{Red}$ is the strategies used by those groups. The four actions taken in the task can be described as either passive (monitor and analyze) or active (remove and restore) [55]. Since $Beeline_{Red}$ quickly attacks the operational server, passive human defenders could perform worse against $Beeline_{Red}$. Similarly, it could be more difficult for active human defenders because $IBL_{Red}^{Trained}$ is stochastic and adaptive to defender behavior.

To test this explanation, we compared the proportion of active type actions performed by proficient defenders paired against $IBL_{Red}^{Trained}$, $Beeline_{Red}$, and $Meander_{Red}$. This comparison demonstrated a higher rate of active actions in proficient defenders paired against $IBL_{Red}^{Trained}$

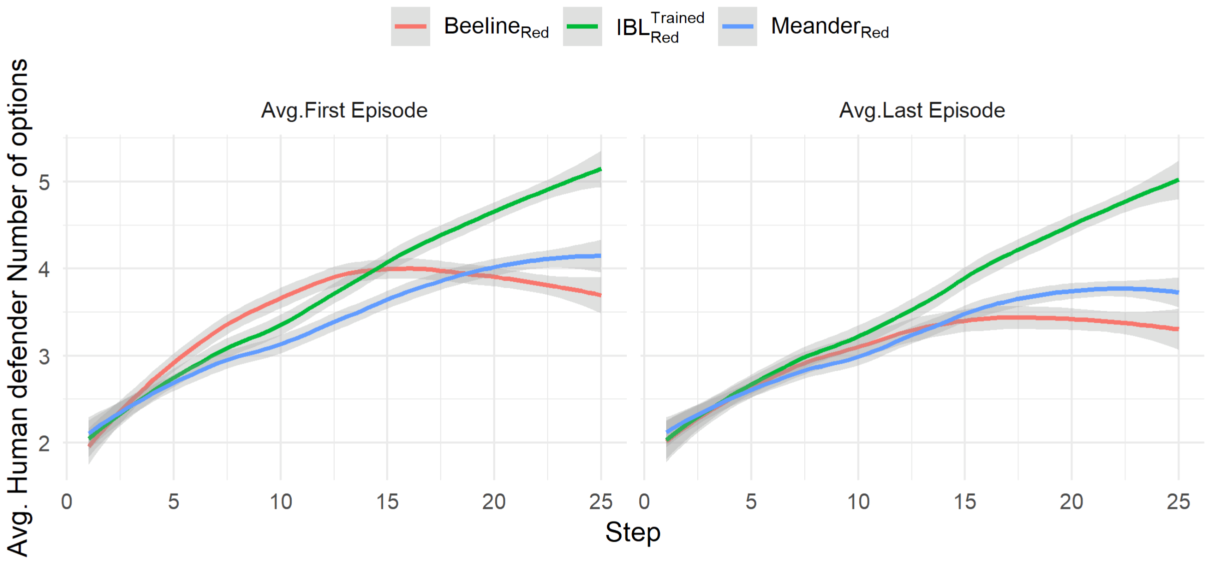


Figure 3.5: Average size of the Human defender's option space in the first left) and the last episode (right)

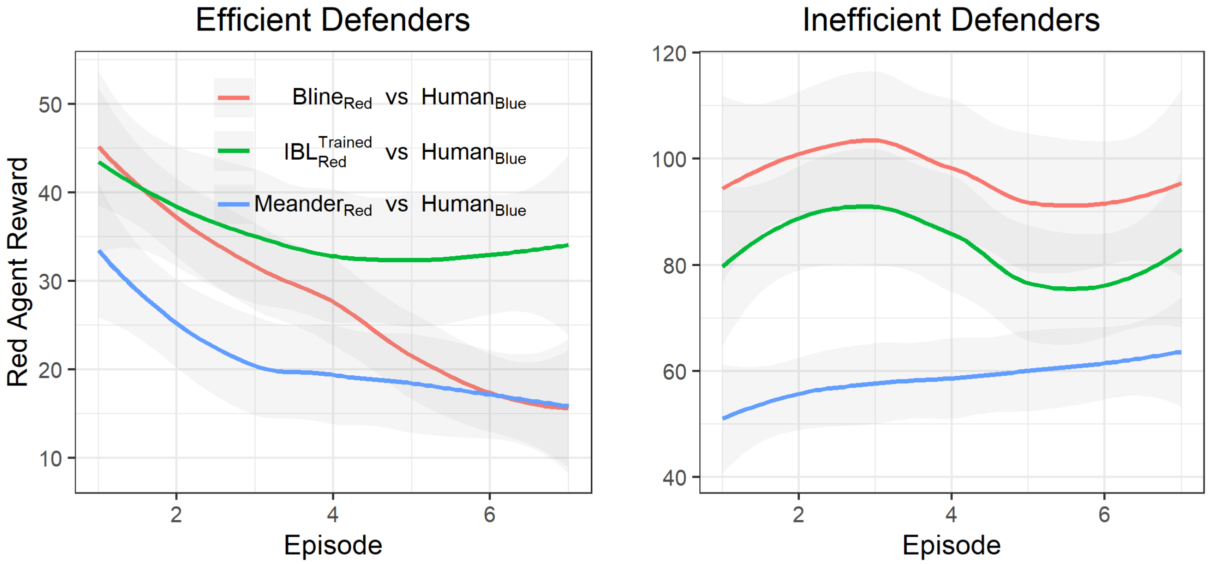


Figure 3.6: Average red agent reward by episode, split between efficient human defenders and inefficient human defenders.

than both $Beeline_{Red}$ (p=0.007) and $Meander_{Red}$ (p¡0.001), but no significant difference between $Beeline_{Red}$ and $Meander_{Red}$ (p=0.457). This demonstrates that achieving proficient performance against $IBL_{Red}^{Trained}$ required more active strategy actions.

In conclusion, we provide important steps towards establishing emulated adversaries that can effectively train cyber defenders and support the development of autonomous cyber defenders. We demonstrate that it is possible to use cognitive agents to produce adversaries that are adaptive to defenders' actions. These models can ultimately be more effective in learning cyber defense strategies than static and deterministic adversaries.

3.2 Learning to Play an Adaptive Cyber Deception Game

3.2.1 Adaptive Cyber Deception Game [16]

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 (within degree equals zero) and the possible initial status of the attacker in the beginning. Each node is assigned a value v_e that resembles the value of digital assets. Each edge is assigned a probability p_e to indicate the probability of successfully 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 the defender can 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 no more than one action 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.

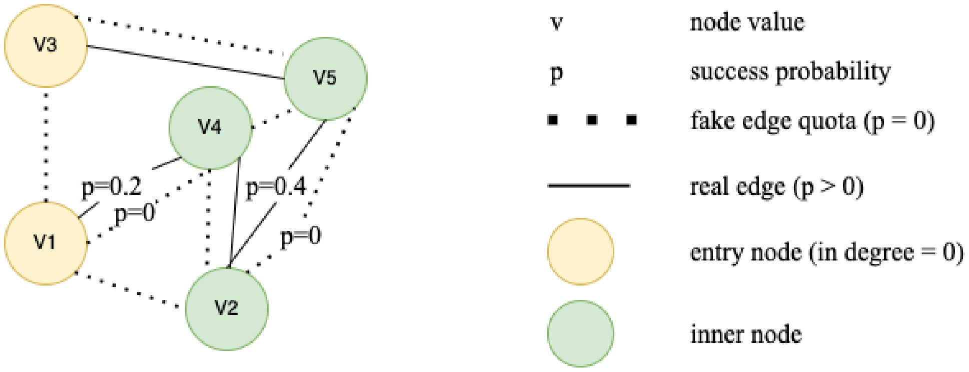


Figure 3.7: Attack Graph

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. Nature monitors the added fake edge. 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^t at 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 by 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 no precise defense is available. If the attacker's location is unknown, the defender will 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 the previous round. The defender is also allowed to opt out of edge hiding, adding, or monitoring. All the 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 must choose different real edges to hide or monitor if he uses 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, which is 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 observes 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 $O_d = (loc|h, V, V^{'}, E_{visible}, \textbf{budget})$: 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 an immediate reward for each successful establishment of a foothold on a node. If he finally gets caught, he will get a 10-point penalty but get to keep the values collected. The assumption is that the attacker can get the benefit out of the reached nodes and cause irreversible loss to the defender.

$$R_a(S_t) = \begin{cases} v_{n_t}, & \text{if t } \leq \text{ episode length } \& \ n_t \neq n_{t-1} \ \& \text{ safe} \\ 0, & \text{if t } \leq \text{ episode length } \& \ n_t = n_{t-1} \ \& \text{ safe} \\ -10, & \text{if caught} \end{cases} \tag{3.1}$$

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

$$R_d(S_t) = \begin{cases} 0, & \text{if t ¡ episode length} \\ \sum_{n \in N_{exploited}} v_n, & \text{if t = episode length}\& \text{ didn't catch} \\ 10 - \sum_{n \in N_{exploited}} v_n, & \text{if t = episode length } \& \text{ caught} \end{cases} \tag{3.2}$$

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 meantime. 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.

3.2.2 Learning to Play

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 and optimizing the policy directly. It has shown great success in various single-agent and multi-agent problems such as Atari games [71], real-time strategy games [10] and robotic control [64]. 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 invalid actions that 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, the probabilities of which are zero for invalid actions.

The attacker's set of valid actions is conditioned on its location on the attack graph and the visibility of real and fake edges. An attacker can only move along visible outgoing edges from its foothold. The defender's available actions depend on the occupied status of nodes, the remaining budget, and the knowledge about the 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. Only defenses on nodes at the next layer of the attacker's foothold are considered valid when the attacker's location is known.

The game state is partially observable for both the defender and the 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 $|E_{real}| + N(N-1)$. The defender's observation vector is the concatenation of the one-hot encoding of the 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 the remaining defense budget. The encoding of the attacker's location is a zero vector when it is unknown. The 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 [30] network for the attacker to observe conflicting node values and different visible edges in the history.

PPO was initially designed as a single-agent RL algorithm. To solve our game, we follow other works that use PPO for multi-agent settings, e.g., [10] 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 the one-step update for the defender's and attacker's network parameters using PPO separately.

We compare the performance of PPO-based policies against heuristic policies. We use the RLlib [42] implementation of the PPO policy network and LSTM network. The heuristic attacker always moves 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 that is non-differentiable from others. If the budget is not exhausted after masking, the defender will 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 only one real edge exists 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 $[1, U+1]$ range. The upper bound $U = \min_{v^* \in N} \sum_{i \in N} |v_i - v^*|$ is the minimum cost it takes to make the value of all nodes the same. The attacker randomly assigned to one of the entry nodes at the beginning.

We run 100,000 episodes to train 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

$$\begin{cases} Exp_a(\pi_a, \pi_d) = U_a(\pi_a, \pi_d) - U_a(\pi_a, BR_d(\pi_a)) \\ Exp_d(\pi_a, \pi_d) = U_d(\pi_a, \pi_d) - U_d(BR_a(\pi_d), \pi_d). \end{cases} \tag{3.3}$$

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 a defender's policy π_d. Specifically, in our zero-sum game,

$$Exp(\pi_a, \pi_d) = -\left(U_a(\pi_a, BR_d(\pi_a)) + U_d(BR_a(\pi_d), \pi_d)\right). \tag{3.4}$$

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 3.8 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.

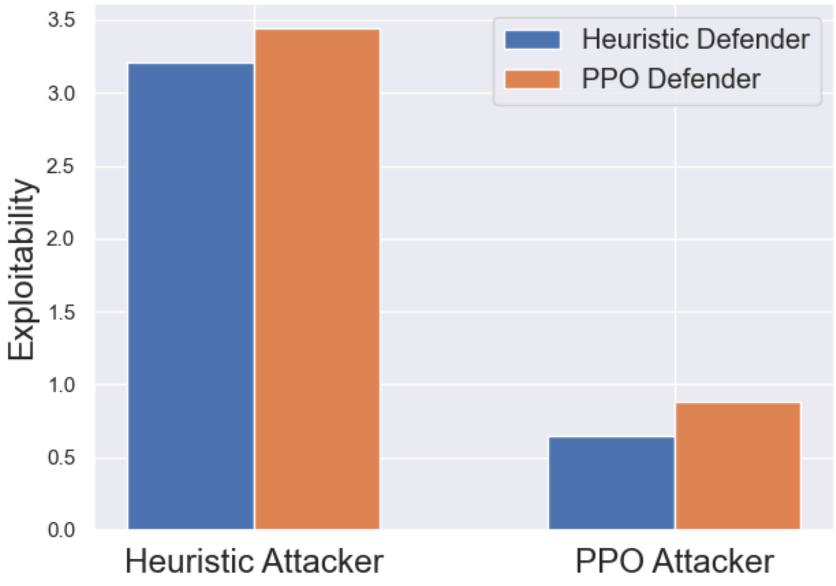


Figure 3.8: Exploitability of the heuristic defender and the PPO defender evaluated against the heuristic attacker and the PPO attacker

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

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 3.9 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.

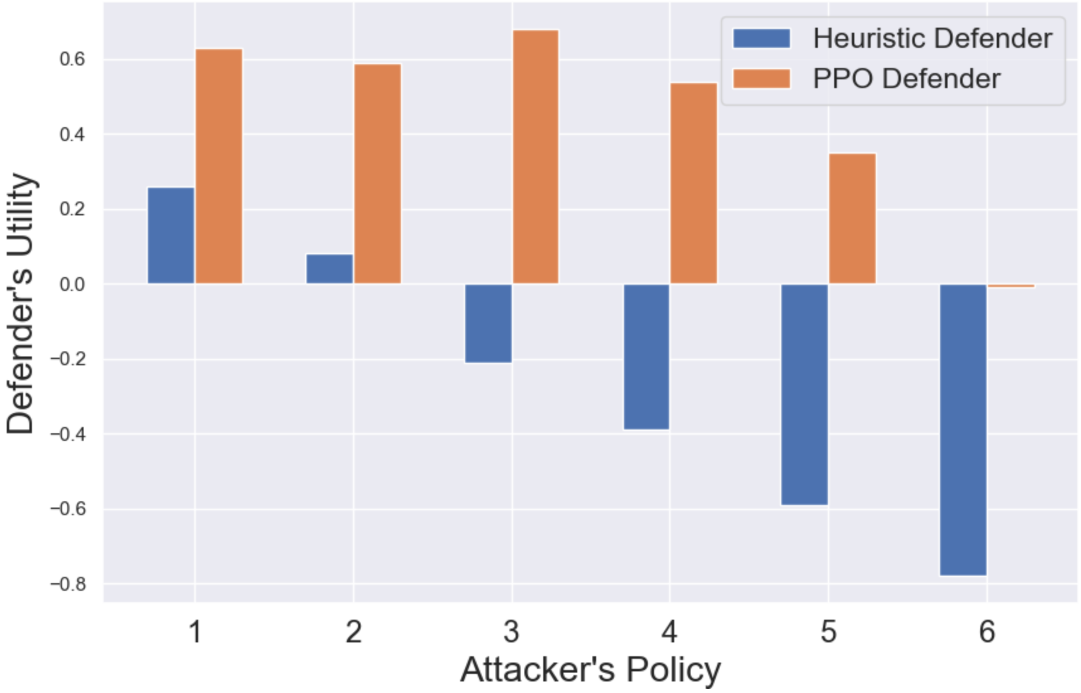


Figure 3.9: Defenders' utilities against multiple attackers

3.3 Exploring Human-AI Defense Teams with Cognitive Agents (In-Progress)

3.3.1 Team Defense Game

TDG is an extension of the interactive defense game. In TDG, human participants play the role of cyber analysts. They are tasked with protecting the computer network of a fictitious manufacturing company against external malicious activity. To do so, they are paired with an artificial teammate, an autonomous cyber defense agent, able to make decisions and partially act independently to collaborate with the human defender in a team. Human and autonomous agents must collaborate efficiently to monitor the network, detect suspicious activity, and take appropriate actions to protect it.

Agent Level of Autonomy In military settings, analysts in Cyber Protection Teams (CPT) are provided with a set of pre-approved actions that they can execute without consulting their hierarchy [**boyarchuk2021organizational**]. However, if they consider that the situation requires actions that are not pre-approved, they need to submit their intention to their chain of command for validation before acting. In TDG, we consider the autonomous agent to be a cyber analyst with a restricted set of pre-approved actions compared to humans. As with its human counterparts in real-world settings, the autonomous agent can select an action that is not pre-approved but must submit for approval or modification to its (human) teammate. This proposed collaboration structure creates an interdependence that necessitates an exchange of information (here, the intention of action) between the autonomous agent and the human, similar to what could happen in CPTs.

Interdependent team activities and outcomes In each step of the game, the human participant is first asked to choose an action and a target to perform it on (i.e., what computer or server to protect immediately) before being presented with the intention of the agent teammate.

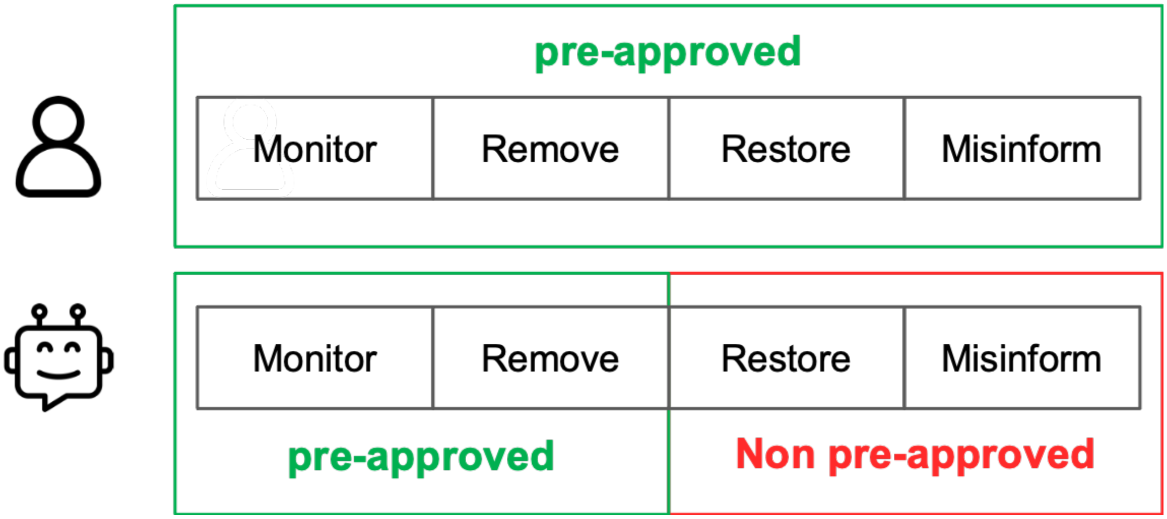


Figure 3.10: Differences in autonomy of action between human and agent teammates. The autonomous teammate has to ask the human for approval of the *Restore* and *Misinform* actions.

If the agent's intention incorporates one of the ”pre-approved” actions (i.e., Monitor or Remove - see Fig 3.10), the human is simply informed of the intention. However, two cases require special attention from the human participant: Overlaps and Supervision. If there is an overlap in the intentions of both the human and agent, humans must resolve the overlap. To do so, they can modify either one of the intentions (human's or agent's) or the target of the intended action. If there is no overlap of the intentions, then the agent actions that require pre-approval are investigated. If the agent chooses to perform a *Restore* or a *Misinform*, the human has the ability to modify this intention, changing the selected action or changing the target. The human can also validate the agent's intentions, agreeing to execute the intended action.

The human decides when to proceed to the next step of the game and to start the execution of the selected actions. This interaction flow is summarized in Fig. 3.11.

The defense team will then receive feedback resulting from their collective actions at this step. Feedback is the step loss calculated as follows:

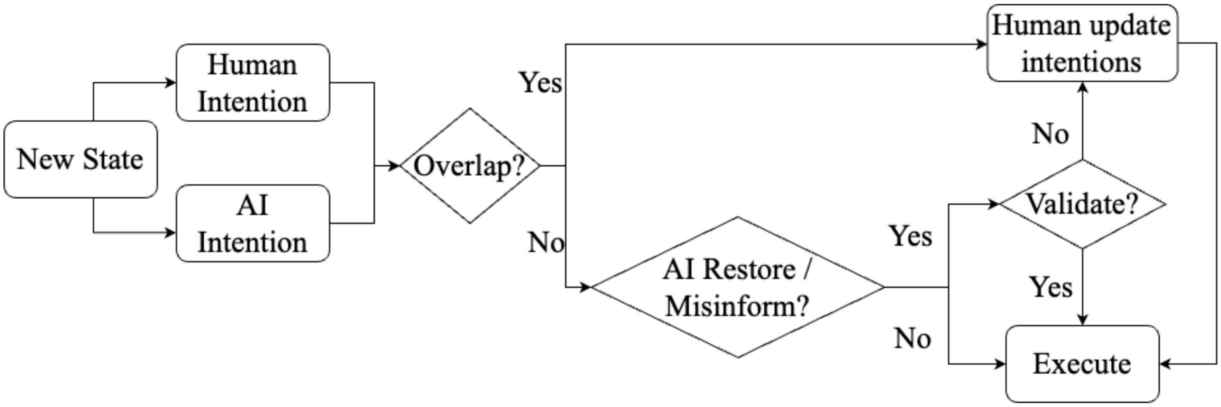


Figure 3.11: Control flow

$$\begin{aligned}\text{Step Loss} =& \text{Network Loss}+\\ & \text{Human Action Cost}+\\ & \text{AI Action Cost}\end{aligned} \tag{3.5}$$

3.3.2 Experiment

Autonomous Defenders Three types of autonomous partners were created: a **random agent**, a heuristic agent, and an instance-based agent. **Random agent** chooses actions **randomly** from the **40 possible options** $(1(\text{Monitor}) + 3\,(\text{Remove} \mid \text{Restore} \mid \text{Misinform}) * 13\,(\text{Hosts}) = 40)$ without regard to the network status. **Heuristic agent** has skills to correctly use commands and the knowledge of the network layout and the significance of the hosts. At each step, based on the observed state of the network and the consequences of the attacker's previous actions, a heuristic defense agent **randomly** chooses from the **4 correct options** conditional on the latest network status: 1) *Monitor*, 2) *Remove* [Exploited Host], 3) *Restore* [PrivEsced Host], 4) *Misinform* [Host on attack path]. When there is more than one Exploited or PrivEsced host, the related options are constructed with the most important host in such a state.

Instance-Based (IBL) Cognitive Agent determines the actions to take according to an Instance-Based Learning algorithm. This agent makes decisions **under the guidance of cognitive principles** and intends to make defense decisions similar to humans [23]. We used a modification of the agent proposed and tested in [55] as a partner to the human.

Team Performance Fig. 3.12 presents the team average loss and the team average recovery time for all the HATs in the three conditions. As shown in Table 3, the HAT loss is largest when humans are paired with Random agents, comparatively lower with Heuristic agents, and minimal when humans are paired with IBL agents. Likewise, the HAT recovery time follows a similar pattern, with a greater time taken to recover a breached host when humans are paired with Random agents, a shorter duration with Heuristic agents, and the shortest duration when paired with IBL agents. These observations strongly suggest that cognitive partners in HATs are humans' most effective team collaborators.

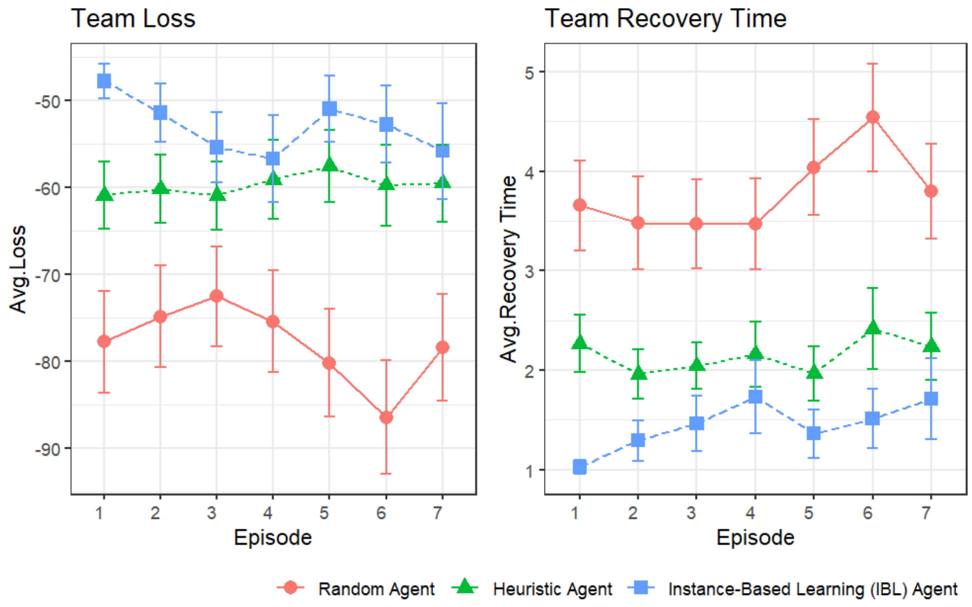


Figure 3.12: Evolution of team performance across the 7 episodes of the experiment composed of Team Loss per episode (left) and Team Recovery Time (right).

Human perception of cooperativeness and autonomy trustworthiness Despite the significant behavioral differences among the three conditions, autonomy characteristics were not found to have a significant effect on the subjective ratings. A one-way ANOVA was conducted to evaluate of human's perception of the cooperativeness and trustworthiness of the three types of agents. Results indicated no significant difference in either cooperativeness, F(2, 108) = 0.218, p = 0.805, or trustworthiness F(2, 108) = 0.545, p = 0.582.

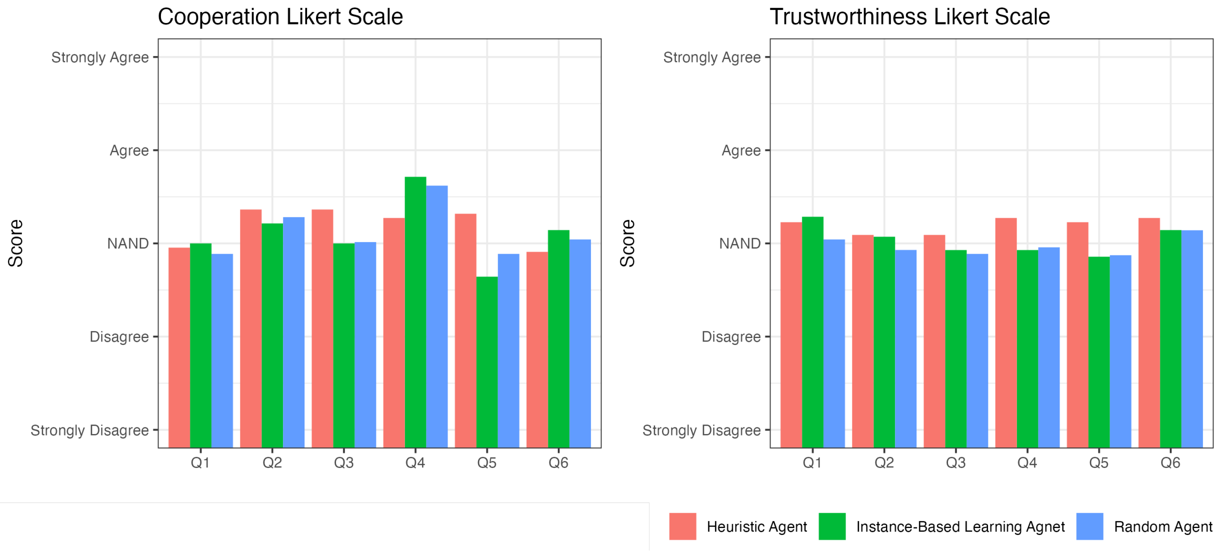


Figure 3.13: Left: Cooperativeness Likert Scale, Right: Trustworthiness Likert Scale

3.4 Fostering Cross-Organizational Information Sharing (In-Progress)

3.4.1 Multi-Defender-Game (MDG)

We have developed a Multi-Defender-Game (MDG) for data collection through human experiments. The MDG is designed for group experiments. In the MDG, a group of defenders (human participants) play an information-sharing game in a cyber-security scenario. The participants are assigned in groups of three players, who will be identified as Defender 1, Defender 2, and Defender 3, each defending their own network. Initially, each defender receives 1000 points as an endowment, which can be used to invest in security to defend their network. Each defender's network is independent; some defenders may be attacked when the others are not, and each may have a different chance of being attacked. Then, defenders start the game and play 50 trials of decision-making on sharing/not sharing information with other defenders in the network. The goal of each defender is to maximize their points in the game.

In each trial, t, the defender's network may or may not get attacked, determined by his *Probability of Breach* Pb^t. If the defender network gets attacked, it costs them -30 points (*attack status* $C_a^t = 1$). They need to choose to share or not to share information with other defenders in the network about the attack/not attack. They will receive feedback after the other two group members make their decisions.

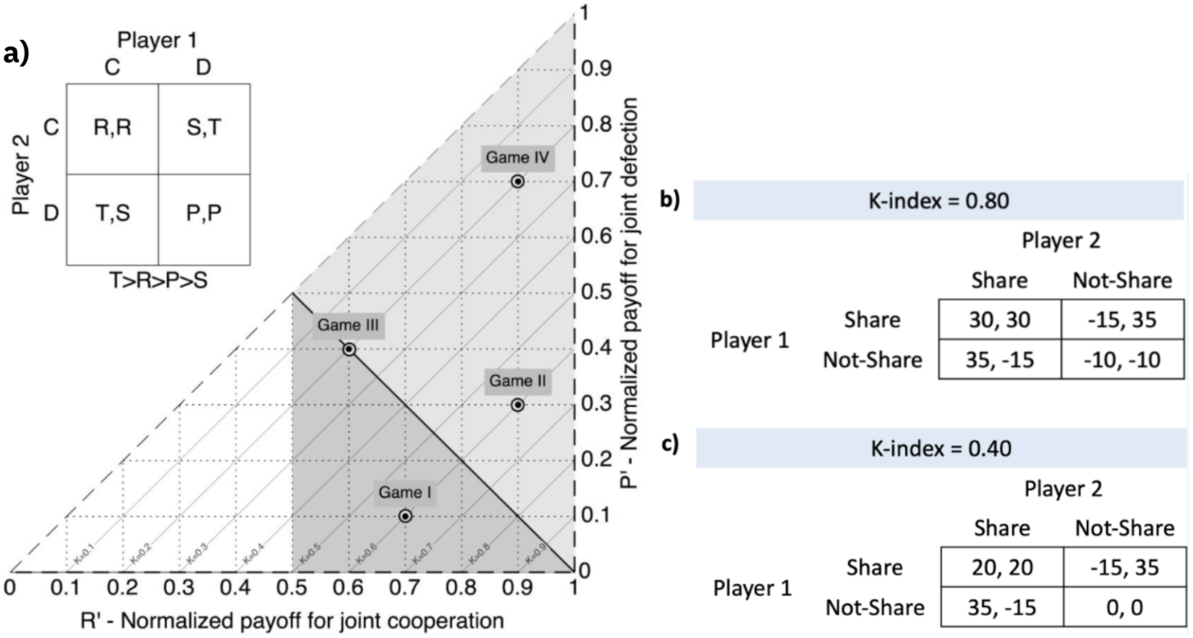
If defenders choose to share information with others, the cost of information sharing (-15 points) is deducted from the available points. The defender (receiver) is rewarded (35 points) for receiving information from each other defender. The sharing interaction between two defenders collectively forms a prisoner's dilemma. For example, the payoff in the share-share cell is $20 = 35 - 15$ for both the column and row players. *Sharing points* Z_i^t of defender i at trial t is the sum of receiving reward and sharing cost with the other two defenders in their group. The accumulated reward of player i at trial t of defender i is Eq.3.6. We assume the information shared is valuable and helps the receiver to strengthen their security. Thus, information sharing also affects the future probability of breach by Eq.3.7.

$$R_i^t = R_i^{t-1} + Z_i^t + (-30) \cdot C_a^t \tag{3.6}$$

$$Pb^{t+1} = Pb^t - \frac{0.95 \cdot Z_i^t}{2000} \tag{3.7}$$

3.4.2 Experiment on Incentive Structure

Moisan et al. [47] studied the interaction of two players using the prisoner's dilemma and assumed two factors that may influence the interaction: 1) the incentive structure and 2) the intrinsic social preferences of each player. Inspired by the theoretical work in [57], Moisan et al. [47] presented a normalized space for PD games. In this experiment, we relied on this theoretical work to design two experimental conditions based on Rapoport's K-index to study the cooperation of defenders in cyber-security scenarios (see Figure 3.14(a)).



b) K-index = 0.80

		Player 2	
		Share	Not-Share
Player 1	Share	30, 30	-15, 35
	Not-Share	35, -15	-10, -10

c) K-index = 0.40

		Player 2	
		Share	Not-Share
Player 1	Share	20, 20	-15, 35
	Not-Share	35, -15	0, 0

Figure 3.14: a) Rapoport K-index; Payoff Matrices b) Condition 1 with k-index = 0.80; c) Condition 2 with k-index = 0.40

3.4.3 Experiment on Information Level

As suggested by the Hierarchy of Social Information (HSI) in gonzalez2011scaling, increased cooperation can be promoted by additional information regarding the other players' actions and outcomes. Thus, knowledge about others' actions and outcomes might make the associations of reciprocity more clear and direct. The similarity of another's predicament to one's own can help strengthen a sense of reciprocity and thus lead to greater cooperation. The HSI proposed an increased level of social information from having no information about others to an increased level of descriptive social information, where increased information about the complete interaction structure may result in a more effective promotion of cooperation. [25] argued that ongoing visibility of the payoff matrix can assist in clarifying the trade-off between short-term and long-term rewards. The cognitive modeling work in [26] also suggests that humans tend to consider the outcome of their opponent, dynamically weighted by their interaction experience.

In this experiment, we designed two experimental conditions defined based on the information given to the participants regarding the sharing information of the other defenders in their group. The information levels were: *Own* and *Others*, where the *Own* condition provided no information on the actions and outcomes of the other defenders in the group, while the *Others* condition provided information about the information shared by the other players in the group. Participants received this information in a table 3.2 where the sharing decisions of each defender in the group, including the protagonist defender, were displayed in a separate column. The table also included their breach status when this information was shared by the other defenders in the group.

Table 3.2: An example output table provided as feedback in the *Others* condition of the [17] experiment

Defender 3 Decision (Me)	Defender 1 Decision	Defender 2 Decision
Information not shared with Defender 1, Information shared with Defender 2	Defender 1 shared information with me, He was attacked	Defender 2 didn't share any information
	My Payoff with Defender 1 : 35 Defender 1's Payoff with me : -15	My Payoff with Defender 2 : 0 Defender 2's Payoff with me : 0

3.4.4 Cooperative Behavior Modeling

We propose an Instance-Based Learning (IBL) cognitive model to make predictions about human sharing behavior in the MDG at different levels of information. The model, Multi-Defender IBL - Prisoner's dilemma (MDIBL-PD), is based on a model of individual learning and decisions from experience in repeated two-player prisoner's dilemma [26], and expands that concept to a multiplayer situation beyond a dyad. Like all IBL models, the MDIBL-PD model relies on the IBL Theory (i.e., IBLT) [24], a well-known cognitive theory of experiential decision-making. The key idea of this theory is that decisions are made by recognizing similar past experiences, their integration into the generation of expected utility of decision alternatives, and the selection of the alternative with the maximal expected utility. An IBL model can accurately represent the content of human memory, recognition, learning, and recall of experiences in decision-making.

The IBL model of the individual defender is primarily concerned with the learning processes determined by the various levels of information available to the model. We denote the within-group defender index as $x \in \{1, 2, 3\}$ and their sharing decisions as $D_x \in \{C(\text{Share}), D(\text{Not-Share})\}$.

The new MDIBL-PD model was developed for the information conditions described above and others. Each IBL agent in the MDIBL-PD model makes decisions using the same procedure defined in the previous section. The human participants in the condition *Others* receive information on the outcome and the breach status of other players (Table 3.2). To capture this interdependence, we modified the blending equation (Eq.**??**) to account for the outcome of the other player, as suggested in [26].

Actions a: In the MDG, the choice options are defined by the actions that each defender can take. The defender D_x can choose not to share information, to share information with one or both of the other defenders, denoted as None, $D_{(x+1) \mod 3}$, $D_{(x+2) \mod 3}$, *Both*.

State S_i^t: The situation state of the defender consists of four attributes: the breach status $A_x \in \{1(\text{attacked}), 0(\text{safe})\}, probabilityofbreach(\text{Pb}_i^t)$, and the expectation of receiving information from each player ($E^t_{D_x}$). Thus, the situation state s of participant i (Defender x) at trial t is $s_i^t = (A_i^t, Pb_i^t, E^t_{D_{(x+1) \mod 3}}, E^t_{D_{(x+2) \mod 3}})$.

Breach status A_it and probability of breach Pb_it have direct and indirect effects on the outcome of a trial; thus, they are included as context information whose pure appearance might affect humans' information-sharing tendency. As suggested by [72], beliefs and behavior correlate within rounds in repeated prisoners' dilemma games, and beliefs in one round vary with behavior in the previous round. Thus, we include $E^t_{D_x}$ to capture the association between the expectation of receiving information from peers and deciding whether to share information with them. It is approximated with the accumulated proportion of receiving information from D_x

(Eq.3.8). Here, we assume that participants can keep track of the interaction experience with their peers. This assumption can be relaxed by manipulating the window of proportion calculation. After receiving the actual sharing decisions at the trial t, the $E^t_{D_x}$ slots will be updated to $T^t_{D_x}$ to store the real interaction experience in memory. When the expectation $E^t_{D_x}$ is closer to 1, memory instances of receiving information from peer x ($T^{t'}_{D_x} = 1, t' \in [0, t)$) have greater similarities to the current situation, resulting in higher activation values (**??**), and higher likelihood to be recovered (**??**). Similarly, when the expectation $E^t_{D_x}$ is closer to 0, memories of defected by peer x ($T^{t'}_{D_x} = 0, t' \in [0, t)$) are more likely to be retrieved. The similarity of these numeric attributes is calculated linearly and normalized to $[0, 1]$.

$$E^t_{D_x} = \frac{\sum_{i=0}^{t-1} T^i_{D_x}}{t-1} \tag{3.8}$$

Utility U^t_x: Depending on the experimental condition, the players in the MDG received only information on their own actions (Own) or about the sharing decisions of other defenders and the effect on their outcome (Others). Therefore, the utility of the defender x in the trial t is the points gained or lost exclusively at that trial, constituted with the benefit of receiving information (35 points), the cost of sharing information (-15 points) and the cost of being attacked (Eq.**??**). The cost-benefit of information sharing forms the dyadic prisoner's dilemma, as shown in Table **??**. The cost of the breach is included as part of the utility since the status of the breach has an effect on the sharing decisions of human defenders.

$$w^t_1 = \frac{1 - Surprise^t_1}{2} \tag{3.9}$$

$$w^t_2 = \frac{1 - Surprise^t_2}{2} \tag{3.10}$$

$$U^t_x = \Delta^t_x = Z^t_x + (-30) \cdot A^t_x \tag{3.11}$$

$$U^t_x = \Delta^t_x + w^t_1 \cdot \Delta^t_{(x+1) \mod 3} + w^t_2 \cdot \Delta^t_{(x+2) \mod 3} \tag{3.12}$$

To simulate how humans account for the outcome of others, the utility for the blended value calculations is set as the weighted sum of the point update of the defender D_x and his peers (Eq.**??**). Inspired by the notion of *Social value orientation (SVO)* [8], w represents the degree to which a player is willing to consider the outcome of the other player for each option when making a decision that maximizes the gains in each trial.

Research in [26] finds that the dynamic w dependent on individual experiences can best explain human cooperation behavior. Under this hypothesis, a player will account for the outcome of the opponent as a function of a normalized gap between expected and actual outcomes (surprise). The value of w^t_i (with respect to the opponent's outcome in the trial t) will be reduced by surprise. We assume that the players evaluate the benefit of sharing information with each other independently with different weights, updated according to separate *surprises* and *gaps*.

The normalization of *surprises* limited the value of $Surprise^t_i$ within the range of $[0, 1]$, the value of w^t_i within $[0, 0.5]$, and the sum of weights on the benefit of *others* within $[0, 1]$. This formulation assumes that the way a player accounts for the opponent's outcomes will vary between *extreme selfish* when $w^t_1 = w^t_2 = 0$ and extreme fairness when $w^t_1 = w^t_2 = 0.5$.

$$Gap_i^t = Abs(V_j^{t-1} - (X_{ij} + O_{ij}) \tag{3.13}$$

$$Mean(Gap_i^t) = Mean(Gap_i^{t-1})(1 - \frac{1}{50}) + Gap_i^t(\frac{1}{50}) \tag{3.14}$$

$$Surprise_i^t = \frac{Gap_i^t}{[Mean(Gap_i^t) + Gap_i^t]} \tag{3.15}$$

Pre-Population: From human data, we observed that more than 70% of the human participants chose to share information with both peers at the beginning. [5] show that some fraction of the population actually has altruistic motives. This ingrained tendency to share between human subjects can be the consequence of the experience of cooperation in recent years, or it could be an experimental effect of human participants who expected to cooperate in a *Multi-Defender Collaboration Game*. To capture this preference, inspired by the conclusion in [35] that there are two stable types of individuals that can be described as cooperative and competitive, we prepopulate the IBL agents with instances that represent these initial tendencies. 70% of IBL agents are prepopulated with *Share* instances with positive rewards (0, 20, 40 for zero, one, two *sharing - receiving* with peers), while 30% of IBL agents are prepopulated with *Not-share* instances with negative rewards (0, -15, -30 for zero, one, two *sharing - not receiving* with peers). Cooperatively biased agents and defectively biased agents are randomly formed groups of three. Each group contains a random number (0 to 3) of cooperatively biased agents. The assumption is that the decrease in the proportion of information sharing is caused by the pairing of *cooperative* participants with *defective* participants.

Simulation Procedure: The MDIBL-PD model with default parameters was run for 100 simulated groups of players in each of the two information conditions. Each group plays the game for 50 trials.

Dependent Measures: We calculate the overall proportion of sharing in *Own* and *Others* conditions, the proportion of sharing with *Both*, *One*, or *None* of the other defenders, and the sequential dependencies that emerged from the interaction between IBL agents in a group [44]. Sequential dependencies measures include *Mistrust*, the decision a player makes to defect at time t after both players mutually defected at time $t-1$; *Forgiveness (Not Share - Share)*, the decision to continue cooperating at time t, although mutual cooperation was not achieved due to the defection of the other at time $t-1$; *Abuse (Share - Not Share)*, the decision to continue defecting at time t after a profitable defection at $t-1$; and *Trust*, the decision to continue cooperating at time t, after successful mutual cooperation at time $t-1$. To assess the precision of the predictions of the model with respect to human data, we calculated the mean squared deviation (MSD) using the average of the dependent measure (e.g., the average proportion of cooperation per trial) and using the Pearson correlation coefficient (r) to assess the similarity of time trends between the model and human data. The results demonstrate that the model performed similarly to the actions taken by humans. First, with more information on Others, individuals share information more often in the MDG. Second, humans tend to decrease the proportion of sharing with both players and increase the proportion of their no-sharing behavior over time. This happens

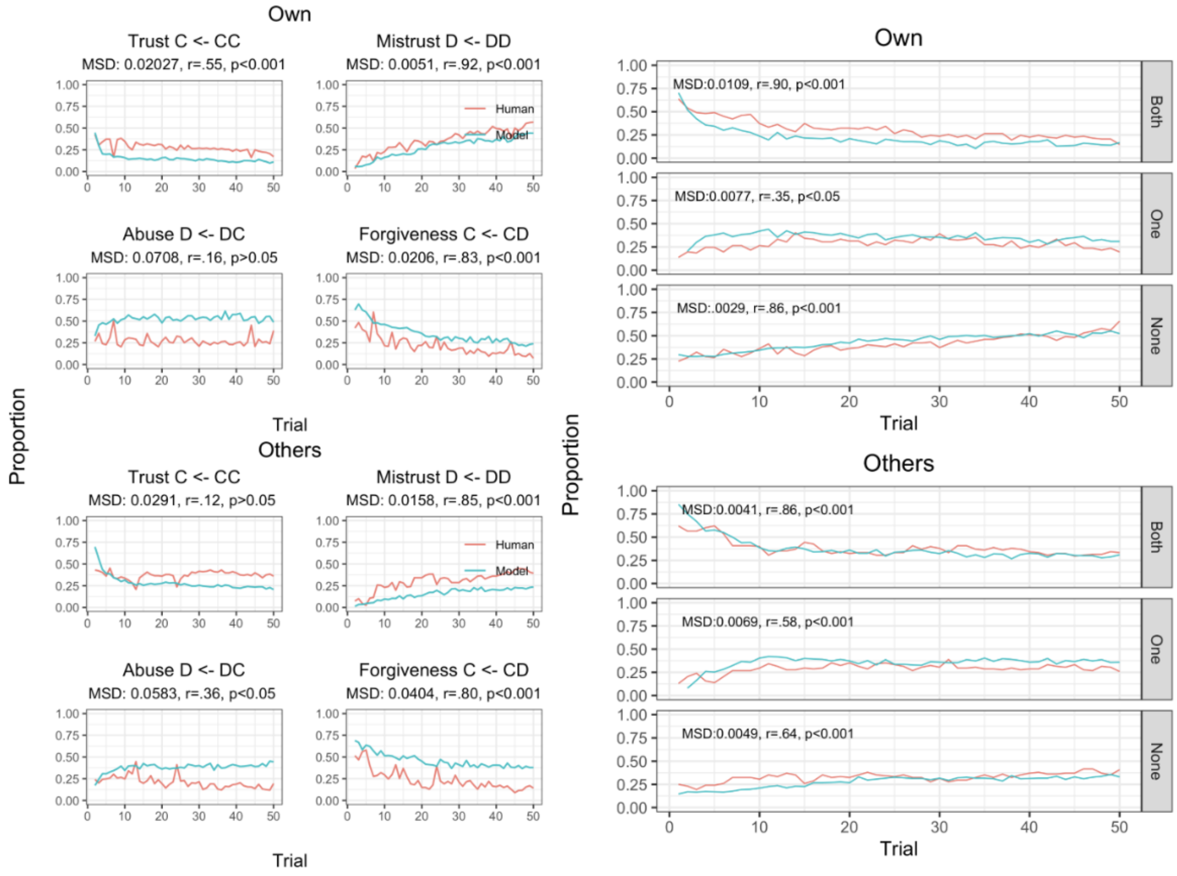


Figure 3.15: Sharing proportions with Both, One, or None of the other players for the *Own* condition (top panel) and the *Others* condition (bottom panel)

particularly in the Own condition. There are also some differences between the model's predictions and human data. For example, in the Own condition, the model initially tends to share more with one of the other players. The model also shows a higher proportion of ”abuse” of the other players, defined as the proportion of defections (not sharing) the model makes after the other player has cooperated (shared). It seems that the model is more ”selfish” than humans are regardless of the level of information, as clearly, the level of abuse in the model is higher than that of human participants.

Sequential dependencies also indicate that humans have difficulty sharing information with other players, increasing their mistrust of them over time. This pattern is particularly strong in the Own condition, and the model replicates such trends.

Chapter 4

Proposed Work

4.1 Modeling Information Sharing in Social Networks

The comparative analysis in section 3.4.4 demonstrated that the model performed similarly to the actions taken by humans at the average level. In agreement with human behavior, the model tends to decrease the proportion of sharing with both players and increase the proportion of their no-sharing behavior over time. However, we also observed discrepancies between the model prediction and human behavior. In general, models are more exploitative. Humans tend to share with every group member at the beginning, while models start with sharing with one of them. Humans are also more reciprocal. After receiving information from others, humans show more cooperative behavior than models. According to the computer tournament of prisoner's dilemma strategies organized by Axelrod et al., [6, 7], being ”nice” at the beginning and making sure not to ”defect” before the opponent are the key to maintaining cooperation between two parties. I conjecture that the model is more abusive due to (1) the lack of mental modeling about others and (2) not paying enough attention to the future. In the proposed work, I will improve the model in these aspects.

The assumptions in section 3.4.4 also make the model unscalable. The model assumes independent weight for each fellow group member. It also relies on perfect memory of past interactions with each member to form expectations about future collaboration. Both assumptions will be cognitively implausible in groups with a size larger than 6. In the real world, organizations are connected with each other as a collaboration network rather than isolated small groups. It is thus important to understand the human cooperative behavior in large networks. I hypothesize that in large networks, humans might (1) treat all neighbors in the network the same; (2) cluster neighbors to a limited number of categories and treat each of them differently; (3) terminate interaction with exploitative/unpredictable neighbors and only maintain relationships with ”friends”; In the proposed work, I will explore these hypotheses.

4.2 (Stretch Goal) Enhance MARL for cyber deception

In section 3.2, we proposed a two-play Markov game to capture the sequential moves in defender-attacker interaction. It is inevitable that there will still be the following limitations in our work: In terms of attack-defense game modeling, only one attacker is playing against one defender protecting an attack graph. In the real world, one network is attacked by multiple attackers with diverse motivations and different levels of attack capabilities. A cyber protection team typically consists of professionals with diverse roles, each bringing a unique set of skills to ensure comprehensive security measures. In the proposed work, I will extend the game model to a multi-attacker, multi-defender game with heterogeneous defense agents.

Regarding applying RL methods in the security field, section 3.2.2 showed promising preliminary results in simulation. However, the scale of the cyber scenario for training and evaluation in section 3.2 is not large. Each network contains only 4 nodes and 5 edges. Defenders only have four types of actions with a limited budget. The size of defense strategy space increases exponentially with the network size and the available actions for the defender. To achieve scalable, dynamic defense, I propose to employ the concept of hierarchical multi-agent RL [69] to decouple the defense problem and separate the choice of defense action from the choice of specific action parameters. The training of RL requires a large number of interactions with the environment to learn a policy that has comparable performance to heuristic strategies designed by human experts. To accelerate learning, I will incorporate expert knowledge with behavior priors [66].

Chapter 5

Timeline

To conclude this thesis proposal, I propose a timeline from 2024 June to 2025 May/August for completing the in-progress and proposed work introduced in Chapters 4 and 5.

- **June - July:** Complete the writing of the HAT submission, Experimental Evaluation of Autonomous Agents in Semi-Supervisory Team Defense
- **August - September**: Complete the writing of the MDG submission part I, Investigate Information Sharing Among Defenders as A Multi-Player Prisoner's Dilemma
- **October - December:** Complete the modeling and writing of the MDG submission part II, Modeling Information Sharing in Social Networks
- Plan A:
    - **January - May:** Dissertation writing & defense
- Plan B:
    - **January - May:** (Stretch Goal) Enhance MARL for cyber deception
    - **June- August:** Dissertation writing & defense

Bibliography

[1] International Data Corporation (IDC). *Worldwide Enterprise WLAN Forecast, 2024–2028*. `https://www.marketresearch.com/IDC-v2477/Worldwide-Enterprise-WLAN-Forecast-36971004/`. Accessed: 2024-05-25. 2024.

[2] Robert K Abercrombie, Bob G Schlicher, and Frederick T Sheldon. “Security analysis of selected AMI failure scenarios using agent based game theoretic simulation”. In: *2014 47th Hawaii International Conference on System Sciences*. IEEE. 2014, pp. 2015–2024.

[3] Nicholas Anderson, Robert Mitchell, and Ray Chen. “Parameterizing moving target defenses”. In: *2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE. 2016, pp. 1–6.

[4] Ross Anderson. *Security engineering: a guide to building dependable distributed systems*. John Wiley & Sons, 2020.

[5] James Andreoni and John H Miller. “Rational cooperation in the finitely repeated prisoner’s dilemma: Experimental evidence”. In: *The economic journal* 103.418 (1993), pp. 570–585.

[6] Robert Axelrod. “Effective choice in the prisoner’s dilemma”. In: *Journal of conflict resolution* 24.1 (1980), pp. 3–25.

[7] Robert Axelrod. “More effective choice in the prisoner’s dilemma”. In: *Journal of conflict resolution* 24.3 (1980), pp. 379–403.

[8] Daniel Balliet, Craig Parks, and Jeff Joireman. “Social value orientation and cooperation in social dilemmas: A meta-analysis”. In: *Group Processes & Intergroup Relations* 12.4 (2009), pp. 533–547.

[9] David Balson et al. *Cyber information sharing: Building collective security, https://www3.weforum.org/do* Oct. 2020.

[10] Christopher Berner et al. “Dota 2 with large scale deep reinforcement learning”. In: *arXiv preprint arXiv:1912.06680* (2019).

[11] Leslie Blaha et al. “Opportunities and Challenges for Human-Machine Teaming in Cybersecurity Operations [Panel Discussion]”. In: (2019).

[12] Identity Theft Resource Center. *2023 Annual Data Breach Report*. `https://www.idtheftcenter.org/wp-content/uploads/2024/01/ITRC_2023-Annual-Data-Breach-Report.pdf`. Accessed: 2024-05-25. 2024.

[13] Frederick B Cohen et al. “A note on distributed coordinated attacks”. In: *Computers & Security* 15.2 (1996), pp. 103–121.

[14] Michael J De Lucia, Allison Newcomb, and Alexander Kott. "Features and operation of an autonomous agent for cyber defense". In: *arXiv preprint arXiv:1905.05253* (2019).

[15] Yinuo Du et al. "A cyber-war between bots: human-like attackers are more challenging for defenders than deterministic attackers". In: *Proceedings of the 56th hawaii international conference on system sciences*. 2023.

[16] Yinuo Du et al. "Learning to play an adaptive cyber deception game". In: *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems. Auckland, New Zealand*. Vol. 6. 2022.

[17] Yinuo Du et al. "Multi-Defender Collaborations in a Cyber-security Scenario". In: *Human Factors* ().

[18] Yinuo Du et al. "Towards autonomous cyber defense: predictions from a cognitive model". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 66. 1. SAGE Publications Sage CA: Los Angeles, CA. 2022, pp. 1121–1125.

[19] Richard Elderman et al. "Adversarial Reinforcement Learning in a Cyber Security Simulation." In: *ICAART (2)*. 2017, pp. 559–566.

[20] Fei Fang, Peter Stone, and Milind Tambe. "When Security Games Go Green: Designing Defender Strategies to Prevent Poaching and Illegal Fishing." In: *IJCAI*. Vol. 15. 2015, pp. 2589–2595.

[21] Kimberly J Ferguson-Walter et al. "Cyber expert feedback: Experiences, expectations, and opinions about cyber deception". In: *Computers & Security* 130 (2023), p. 103268.

[22] Saira Ghafur et al. "A retrospective impact analysis of the WannaCry cyberattack on the NHS". In: *NPJ digital medicine* 2.1 (2019), p. 98.

[23] Cleotilde Gonzalez. "Building Human-Like Artificial Agents: A General Cognitive Algorithm for Emulating Human Decision-Making in Dynamic Environments". In: *Perspectives on Psychological Science* (2023), p. 17456916231196766.

[24] Cleotilde Gonzalez, Javier F Lerch, and Christian Lebiere. "Instance-based learning in dynamic decision making". In: *Cognitive Science* 27.4 (2003), pp. 591–635.

[25] Cleotilde Gonzalez and Jolie M Martin. "Scaling up instance-based learning theory to account for social interactions". In: *Negotiation and Conflict Management Research* 4.2 (2011), pp. 110–128.

[26] Cleotilde Gonzalez et al. "A cognitive model of dynamic cooperation with varied interdependency information". In: *Cognitive science* 39.3 (2015), pp. 457–495.

[27] Samuel N Hamilton and Wendy L Hamilton. "Adversary modeling and simulation in cyber warfare". In: *IFIP International Information Security Conference*. Springer. 2008, pp. 461–475.

[28] Allyson I Hauptman et al. "Adapt and overcome: Perceptions of adaptive autonomous agents for human-AI teaming". In: *Computers in Human Behavior* 138 (2023), p. 107451.

[29] Allyson I Hauptman et al. "Understanding the influence of AI autonomy on AI explainability levels in human-AI teams using a mixed methods approach". In: *Cognition, Technology & Work* (2024), pp. 1–21.

[30] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[31] Brian E. Humphreys. *Critical Infrastructure Policy Information Sharing and Disclosure Requirements After the Colonial Pipeline Attack*. PDF. 2021. URL: `https://crsreports.congress.gov/product/pdf/IN/IN11683`.

[32] Sushil Jajodia et al. *Adversarial and uncertain reasoning for adaptive cyber defense: control-and game-theoretic approaches to cyber security*. Vol. 11830. Springer Nature, 2019.

[33] Pontus Johnson, Robert Lagerström, and Mathias Ekstedt. “A meta language for threat modeling and attack simulations”. In: *Proceedings of the 13th international conference on availability, reliability and security*. 2018, pp. 1–8.

[34] Hamdi Kavak et al. “Simulation for cybersecurity: state of the art and future directions”. In: *Journal of Cybersecurity* 7.1 (2021), tyab005.

[35] Harold H Kelley and Anthony J Stahelski. “Social interaction basis of cooperators’ and competitors’ beliefs about others.” In: *Journal of personality and social psychology* 16.1 (1970), p. 66.

[36] Elmar Kiesling et al. “Simulation-based optimization of information security controls: An adversary-centric approach”. In: *2013 Winter Simulations Conference (WSC)*. IEEE. 2013, pp. 2054–2065.

[37] Glen Klien et al. “Ten challenges for making automation a” team player” in joint human-agent activity”. In: *IEEE Intelligent Systems* 19.6 (2004), pp. 91–95.

[38] Igor Kotenko. “Multi-agent modelling and simulation of cyber-attacks and cyber-defense for homeland security”. In: *2007 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*. IEEE. 2007, pp. 614–619.

[39] Pat Langley. “The computational gauntlet of human-like learning”. In: *Proceedings of the aaai conference on artificial intelligence*. Vol. 36. 11. 2022, pp. 12268–12273.

[40] Cheng Lei et al. “Moving target defense techniques: A survey”. In: *Security and Communication Networks* 2018 (2018).

[41] Chao Li et al. “Dynamic offloading for multiuser muti-CAP MEC networks: A deep reinforcement learning approach”. In: *IEEE Transactions on Vehicular Technology* 70.3 (2021), pp. 2922–2927.

[42] Eric Liang et al. “RLlib: Abstractions for Distributed Reinforcement Learning”. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, Oct. 2018, pp. 3053–3062. URL: `https://proceedings.mlr.press/v80/liang18b.html`.

[43] Richard Lippmann et al. “Validating and restoring defense in depth using attack graphs”. In: *MILCOM 2006-2006 IEEE Military Communications Conference*. IEEE. 2006, pp. 1–10.

[44] Jolie M Martin et al. “A description–experience gap in social interactions: Information about interdependence and its effects on cooperation”. In: *Journal of Behavioral Decision Making* 27.4 (2014), pp. 349–362.

[45] Stephanie Milani et al. “Harnessing the power of deception in attack graph-based security games”. In: *Decision and Game Theory for Security: 11th International Conference,*

GameSec 2020, College Park, MD, USA, October 28–30, 2020, Proceedings 11. Springer. 2020, pp. 147–167.

[46] Sarandis Mitropoulos, Dimitrios Patsos, and Christos Douligeris. “On Incident Handling and Response: A state-of-the-art approach”. In: *Computers & Security* 25.5 (2006), pp. 351–370.

[47] Frédéric Moisan et al. “Not all Prisoner’s Dilemma games are equal: Incentives, social preferences, and cooperation.” In: *Decision* 5.4 (2018), p. 306.

[48] Julio Navarro, Aline Deruyver, and Pierre Parrend. “A systematic survey on multi-step attack detection”. In: *Computers & Security* 76 (2018), pp. 214–249.

[49] *New Research Finds the SolarWinds Cyber Attack Cost Affected Companies in Key Sectors 11% of Total Annual Revenue on Average*. June 2021. URL: `https://www.businesswire.com/news/home/20210628005429/en/%5C%C2%5C%A0New-Research-Finds-the-SolarWinds-Cyber-Attack-Cost-Affected-Companies-in-Key-Sectors-11-of-Total-Annual-Revenue-on-Average`.

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

[51] Megan Nyre-Yu, Robert S Gutzwiller, and Barrett S Caldwell. “Observing cyber security incident response: qualitative themes from field research”. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 63. 1. SAGE Publications Sage CA: Los Angeles, CA. 2019, pp. 437–441.

[52] Xinming Ou, Sudhakar Govindavajhala, Andrew W Appel, et al. “MulVAL: A logic-based network security analyzer.” In: *USENIX security symposium*. Vol. 8. Baltimore, MD. 2005, pp. 113–128.

[53] Martina Panfili et al. “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. 2018, pp. 460–465.

[54] Donn B Parker. *Fighting computer crime: A new framework for protecting information*. John Wiley & Sons, Inc., 1998.

[55] Baptiste Prebot, Yinuo Du, and Cleotilde Gonzalez. “Learning about simulated adversaries from human defenders using interactive cyber-defense games”. In: *Journal of Cybersecurity* 9.1 (2023).

[56] Ratih Hikmah Puspita et al. “Reinforcement learning based 5G enabled cognitive radio networks”. In: *2019 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE. 2019, pp. 555–558.

[57] Anatol Rapoport, Albert M Chammah, and Carol J Orwant. *Prisoner’s dilemma: A study in conflict and cooperation*. Vol. 165. University of Michigan press, 1965.

[58] Shabana Razak, Mian Zhou, and Sheau-Dong Lang. “Network intrusion simulation using OPNET”. In: *Proceedings of 2002 OPNETWORKS Conference*. Citeseer. 2002.

[59] Erin Rubenstein. *Cybersecurity defense: Biden Administration Executive Order a great start towards a more robust national framework*. July 2021. URL: `https://content.lighthouseglobal.com/blog/cybersecurity-defense-biden-administration-executive-order`.

[60] Elaine M Sedenberg and James X Dempsey. “Cybersecurity information sharing governance structures: An ecosystem of diversity, trust, and tradeoffs”. In: *arXiv preprint arXiv:1805.12266* (2018).

[61] Maxwell Standen et al. “CAGE Challenge 1”. In: *IJCAI-21 1st International Workshop on Adaptive Cyber Defense.* arXiv, 2021.

[62] Maxwell Standen et al. “CybORG: A gym for the development of autonomous cyber agents”. In: (2021). arXiv: `2108.09118v1`.

[63] Branka Stojanović, Katharina Hofer-Schmitz, and Ulrike Kleb. “APT datasets and attack modeling for automated detection methods: A review”. In: *Computers & Security* 92 (2020), p. 101734.

[64] Jie Tan et al. “Sim-to-real: Learning agile locomotion for quadruped robots”. In: *arXiv preprint arXiv:1804.10332* (2018).

[65] Omkar Thakoor et al. “Cyber camouflage games for strategic deception”. In: *Decision and Game Theory for Security: 10th International Conference, GameSec 2019, Stockholm, Sweden, October 30–November 1, 2019, Proceedings 10*. Springer. 2019, pp. 525–541.

[66] Dhruva Tirumala et al. “Behavior priors for efficient reinforcement learning”. In: *Journal of Machine Learning Research* 23.221 (2022), pp. 1–68.

[67] N Eric Weiss. *Legislation to facilitate cybersecurity information sharing: Economic analysis*. Congressional Research Service Washington, DC, 2015.

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

[69] Zhiwei Xu et al. “HAVEN: hierarchical cooperative multi-agent reinforcement learning with dual coordination mechanism”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 10. 2023, pp. 11735–11743.

[70] Tarun Yadav and Arvind Mallari Rao. “Technical aspects of cyber kill chain”. In: *Security in Computing and Communications: Third International Symposium, SSCC 2015, Kochi, India, August 10-13, 2015. Proceedings 3*. Springer. 2015, pp. 438–452.

[71] Chao Yu et al. “The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games”. In: *arXiv preprint arXiv:2103.01955* (2021).

[72] Dandan Zhang et al. “The dynamics of belief updating in human cooperation: findings from inter-brain ERP hyperscanning”. In: *NeuroImage* 198 (2019), pp. 1–12.

[73] Jing Zhang, Jun Zhuang, and Victor Richmond R Jose. “The role of risk preferences in a multi-target defender-attacker resource allocation game”. In: *Reliability Engineering & System Safety* 169 (2018), pp. 95–104.

[74] Rui Zhang et al. “” An ideal human” expectations of AI teammates in human-AI teaming”. In: *Proceedings of the ACM on Human-Computer Interaction* 4.CSCW3 (2021), pp. 1–25.

[75] Wanying Zhao and Gregory White. “A collaborative information sharing framework for community cyber security”. In: *2012 IEEE Conference on Technologies for Homeland Security (HST)*. IEEE. 2012, pp. 457–462.