Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

CYBER DECEPTION METRICS FOR INTERCONNECTED COMPLEX SYSTEMS

Md ali reza Al amin
Sachin Shetty

Computational Modeling and Simulation Engineering
Old Dominion University
5115 Terminal Blvd
Norfolk, VA 23529, USA

Charles Kamhoua

US Army Research Laboratory

2800 Powder Mill Rd
Adelphi, MD 20783, USA

ABSTRACT

Cyber attackers' evolving skills cause it challenging to secure the network. Thus it is paramount to characterize adversarial strategies and estimate the attacker's capability. Furthermore, estimating the adversarial capability can aid the cyber defender when deciding to place deceptive elements in the network. In this paper, we address the problem of characterizing adversarial strategies and develop a suite of metrics that quantify the opportunity and capability of the adversary. Using these metrics, the cyber defender can estimate the attacker's capability. In our simulation, we incorporated the developed metrics to estimate adversary capabilities based on the attacker's aggression, knowledge, and stealthiness level. To minimize the adversarial impact, we consider placing decoy nodes as deceptive elements in the network and measure the effectiveness of having decoy nodes. Our experimental evaluation suggests that placing decoy nodes in the network can effectively increase the attacker's resource usage and decrease the win percentage.

1 INTRODUCTION

Cyber deception has also emerged as a defense approach to secure our cyber infrastructure from opponent forces. The cyber deception technique allows deploying a network of decoy assets in a complex networked system environment, aiming to exhaust the opponent's resources and time, gather information about their strategies, tactics, capabilities, and intent, and redirect the opponent towards less desirable states. The cyber deception approach will mitigate the information asymmetry that exists between an opponent and friendly systems by converting the friendly's disadvantageous position to a position of strength. However, in order to deploy cyber-deception to realize a resilient cyber infrastructure, several research challenges need to be addressed: a) Design of network decoys, b) Placement of network decoys, c) Maximizing information uncertainty for an opponent, d) Anticipating opponent attack strategy and nodes targeted, e) Quantifying the performance of deception mechanism to assess the level of asymmetry between an opponent and friendly systems.

Practical deployment of cyber deception relies on friendly forces' ability to optimally place decoy nodes along the networked paths. We have developed a cyber deception approach focused on predicting the most likely sequence of attack paths and deploying decoy nodes along the predicted path (Al Amin et al. 2021). Our proposed approach combines reactive (graph analysis) and proactive (cyber deception technology) defense to thwart the adversaries' lateral movement. The proposed approach is realized through two phases. The first phase predicts the most likely attack path based on Intrusion Detection System (IDS) alerts and network trace. The second phase determines the optimal deployment of decoy nodes along the predicted path. We employ transition probabilities in a Hidden Markov Model to predict the path. In the second phase, we utilize the predicted attack path to deploy decoy nodes. However, it is likely that the attacker will not follow that predicted path to move laterally. To address this challenge, we employ

978-1-6654-7661-4/22/$31.00 ©2022 IEEE

a Partially Observable Monte-Carlo Planning (POMCP) framework. POMCP helps the defender assess several defense actions to block the opponent when it deviates from the predicted path.

In this paper we address the following research challenge, a set of metrics that will provide insights into the level of complexity a sequence of events will impose on an opponent. A set of mixed true and false information increases an aspect of the complexity of the environment in a way that makes it more difficult for an opponent to make decisions or shape conditions in one's favor. If the attacker believes that all the information, he is receiving is accurate, the probability that the attack campaign will fail is very high. On the other hand, if the attacker knows that the information is mixed with true and false information, attackers need to spend more time differentiating that information. To define the metrics that will provide insights into the level of complexity a sequence of events will impose on an opponent, we consider two dimensions of evolution in terms of the opponent's progression: timeliness and effectiveness. Timeliness metrics measures the time it takes to generate new strategies while effectiveness reflects of these generations and impact measure the causeeffect relationship whenever the attacker or defender takes an action. Timeliness suite of metrics measures how quickly the attacker or defender evolves its strategies with or without considering the resulting effectiveness. On the other hand, effectiveness suite of metrics measures the effectiveness of generations over the course of the evolution. In summary, we provide the following contributions:

- Developed a suite metrics that quantify the opportunity and capability of the adversary based on based on aggression, knowledge, and stealthiness.
- To quantify more effective evolutionary strategy to aid in selecting effective deception action by using an autonomous cyber attacker agents.

2 Related Work

The importance of security metrics to capture the attacker-defender evolution strategy and related challenges in developing those metrics have been recognized by the security communities (Cyberspace 2011).

Some literature has been proposed dynamic security metrics (Pendleton et al. 2016) where it mentioned metrics for measuring the strength of the preventive defense. The authors in (Kührer et al. 2017) proposed metrics for measuring the reactive defense, and metrics to measure the overall defense are proposed in (Levin 2003). A few kinds of literature (Zhan et al. 2015) have proposed some time-related metrics to forecast cyber threats and incidents. er to characterizing adversarial strategies in the cyber deception domain.

The authors in (Sugrim et al. 2018) proposed two metrics, Reconnaissance Surface measure (RSM) and Attacker's Belief Error (ABE), to quantify the effectiveness of network deception. Furthermore, the authors model an attacker's evolving knowledge during their interaction with the target system as a belief system. These two metrics are from the network perspective to disrupt the reconnaissance effort. At the same time, we measure the effectiveness of the deception strategy from the system perspective to disrupt the adversarial lateral propagation. To the best of our knowledge, we are the first to study the dynamic security metrics in the cyber deception domain to measure the deception strategy evolution and incorporating it with the POMCP framework to push the adversary towards the decoy network.

3 Metrics to Capture Attacker's Capability

Cyber attackers and defenders are continuously evolving their strategies to maximize the effectiveness of cyber-attacks or defense. Sometimes cyber attackers are more agile than defenders because cyber defenders take reactive responses against new cyber attacks. The proposing framework aims to define metrics to capture the insights of the complexity a sequence of actions will impose on an opponent's decision calculus. The imposition of complexity is to take actions that increase an aspect of the complexity of the environment in a way that makes it more difficult for an opponent to make decisions or shapes conditions in one's favor. For example, in a game of chess, a player receives a notification of a "win' or "lose" after a long sequence of moves. Strategic moves are complex ones that present an opponent with a multitude of dilemmas, which obviate any simple or singular response.

The framework is designed to consider the evolution of both attack and defense strategies. The defenders need to improve the strategy over time. The metrics will try to understand and improve the defenders' evolution. A single metric can not adequately measure the effectiveness of attack and defense generation over time; we consider a set of metrics. Then these metrics can be aggregated using a weighted average.

3.1 Overview of the metrics

Here, we consider two sets of metrics in terms of the attacker-defender evolution: timeliness and effectiveness. Timeliness measures the time to evolve new strategy generation from attackers' and defenders' perspectives. Effectiveness measures how effective the strategy is over each other (i.e., attacker and defender). Thus, we use timeliness as a reference of effectiveness and vice versa.

1. **Timeliness-Oriented Metrics:** This suite of metrics measures how an attacker or defender emerges its strategies with or without taking into account the resulting effectiveness. Timeliness-oriented metrics contains 4 metrics for both the attacker and defender as follows:
   - *Generation-Time (GT)* measures the time between two consecutive generations of strategies that are observed by the measuring party (i.e., an attacker or defender).
   - *Effective-Generation-Time (EGT)* measures the time it takes for a party to evolve a generation which indeed increases the effectiveness against the opponent.
2. **Effectiveness-Oriented Metrics:** This suite of metrics measures the effectiveness of generations over the course of the evolution. This suite contains 2 metrics, which are equally applicable to both an attacker and defender, leading to 4 metrics in total. The 2 metrics are as follows:
   - *Evolutionary-Effectiveness (EE)* measures the overall effectiveness of generations with respect to the opponent's generation. This is a random variable over $t \in [0,T]$.
   - Aggregated-Generation (AG) measures the gain in the effectiveness of all generations $t \in [0,T]$.

4 Attack Defense Strategy Evolution

We now describe the graph-theoretic representation of the Bayesian attack graph and metrics mathematical model. Here, we restricted the attention to directed acyclic graphs based on the assumption *monotonicity* (Ammann et al. 2002) on the attacker's behavior. The monotonicity assumption states that the success of the previous exploit will not interfere with the success of the future exploit. The nodes in a Bayesian attack graph represent *attributes* whereas edges represent *exploits*. Attacker capabilities, vulnerabilities of a service or system, interpreted as attributes and exploits, allow the attacker to obtain further capabilities using their current capabilities.

DEFINITION 5.1 (Poolsappasit et al. 2011) *A Bayesian attack graph, G, is defined as the tuple $\mathscr{G}$ = ($\mathscr{N}$,$\mathscr{T}$,$\mathscr{E}$,$\mathscr{P}$)* where

- $\mathscr{N}$ = {1,...,N} is the set of nodes.
- $\mathscr{T}$ is the set of node types. The node type can one of two types, $\mathscr{T}_\rangle \in \{\wedge(AND), \vee(OR)\}$.
- $\mathscr{E}$ is the set of directed edges.
- $\mathscr{P}$ is the set of exploit probabilities associated with edges.

Here, each of the nodes $i \in \mathscr{N}$ (attributes) can be either enabled or disabled, which means the attacker possesses a certain capability or not. So, at time t, the network state denoted by $X_t = (X_t^1, ..., X_t^N)$ where X_t^i is the state of attribute i at time t. We have AND and OR configurations for different attack scenarios for a complex networked system (i.e., power grid) from the Bayesian attack graph. The metrics we defined earlier help us to estimate the attackers' opportunity and capability. It also creates a path to assess the effectiveness of different strategies an attacker or defender might take. Our very first suite of metrics, Strategy Evolution - Timeliness, captures the opportunity for the adversary to make a successful attack and the defender to make a successful defense. The second suite of metrics, Strategy Evolution - Effectiveness, captures the capability of the defender and attacker.

4.1 Strategy Evolution - Timeliness

4.1.1 Generation-Time (GT)

Strategical evolution of attack-defense in terms of timeliness measures the time it takes for the attacker or defender to evolve its strategy. The first metric is Generation-Time (GT), a random variable because the strategy generation is often a stochastic process. We assume that the defender will not evolve its strategy before the attacker's movement. So that at time $t = 0$ there will be no defender's impact in the initial attack.

Defenders' Generation-Time: Suppose the defense is evolved at $t_0 = 0, t_1, ..., t_n \leq T$, namely $t_0, t_1, ..., t_n \subseteq [0, T]$. The defender's GT, namely random variable $GT(D)$, is sampled by $GT(D,i)$ (Mireles et al. 2019)

$$G_T(D,i) = t_{i+1} - t_i \ for \ i = 0,1,...,n-1. \tag{1}$$

This implies that $D_{t_i+\Delta t} = D_{t_i}$ for any $\Delta t < t_{i+1} - t_i$ and $D_{t_i+GT(D,i)} = D_{t_{i+1}} \neq D_{t_i}$ because the defense is not evolved until time t_{i+1}. At time t_{i+1}, the defender's strategy evolve and will have an impact on the attacker's success probability. So, the defender's success probability in terms of blocking the attack,

$$P(X^i_{t+1}|X_t) = \begin{cases} \prod\limits_{y \in p_x} \alpha_{yx} \times (1 \setminus G_T(D,i), \ if \ X^y_t = 1 \forall j \in p_x \\ 0, \quad otherwise \end{cases} \tag{2}$$

where $P(X^i_{t+1}|X_t)$ represents the probability of defender's success probability from the current security state X_t to next security state X_{t+1}. α_{yx} represents the number of available exploits to the attacker and p_x defines as the vulnerability where $x = 1,2,3...n$.

Attackers' Generation-Time: Suppose the attack evolves at $t^{'}_0 = 0, t^{'}_1, ..., t^{'}_k \leq T$, namely $t^{'}_0, t^{'}_1, ..., t^{'}_k \subseteq [0, T]$. Here, the notation $t^{'}$ is used to further highlight the perspective of the attacker's. So, the random variable $GT(A)$ is sampled by $GT(A, j)$, (Mireles et al. 2019)

$$G_T(A, j) = t^{'}_{j+1} - t^{'}_i \ for \ j = 0,1,...,k-1. \tag{3}$$

This means that $A_{t^{'}_j+\Delta t} = A_{t^{'}_j}$ for any $\Delta t < t^{'}_{j+1} - t^{'}_j$ and $A_{t^{'}_j+GT(A,j)} = A_{t^{'}_{i+1}} \neq A^{'}_{t_j}$. The attacker's attack success probability defined as follows,

$$P(Y^i_{t+1}|Y_t) = \begin{cases} \prod\limits_{y \in p_x} \beta_{yx} \times (1 \setminus G_T(A, j), \ \ if \ X^y_t = 1 \forall j \in p_x \\ 0, \quad otherwise \end{cases} \tag{4}$$

The above terms are defined from attacker's perspective as terms are defined in Eq.(2).

4.1.2 Effective-Generation-Time (EGT)

Effective Generation Time (EGT) measures the time to make an effective strategy generation as a whole attack-defense generation. The earlier metric GT, only measure the generation time without considering the effectiveness of the action from both attackers and defenders perspective. This is why we may not get an relationship with respect to the opponent's action. In that case EGT helps to devise the relationship between the attackers and defenders action.

EGT Defenders': Suppose defense generations evolve at $t_0 = 0, t_1, ..., t_n \leq T$, namely $t_0, t_1, ..., t_n \subseteq [0,T]$. The defender's EGT is a *random variable*, denoted by $EGT(D)$, because the evolution of defense generations are stochastic in nature. The random variable $EGT(D)$ is sampled by $EGT(D,i)$ for $i = 0, ..., n-1$ such that $D_{t_i+EGT(d,i)}$ is the nearest future generation that leads to a higher than $D_{t_i}(A_{t_i}, M)$ defense effectiveness. Formally (Mireles et al. 2019),

$$EGT(D,i) = t_{i^*} - t_i \tag{5}$$

when there exists some $t_{i^*} \in t_{i+1}, ..., t_n$ such that

$$D_{t_i+\Delta t}(A_{t_i}, M) \leq D_{t_i}(A_{t_i}, M) \tag{6}$$

and

$$D_{t_i+EGT(d,i)}(A_{t_i}, M) = D_{t_i^*}(A_{t_i}, M) > D_{t_i}(A_{t_i}, M) \tag{7}$$

EGT Attackers': Suppose the attack generations evolve at $t_0^{'} = 0, t_1^{'}, ..., t_k^{'} \leq T$, namely $t_0^{'}, t_1^{'}, ..., t_k^{'} \subseteq [0,T]$. The attacker's EGT is a *random variable*, denoted by $EGT(A)$, because the evolution of attack generations are stochastic in nature. The random variable $EGT(A)$ is sampled by $EGT(A,j)$ for $j = 0, ..., n-1$ such that $D_{t_j^{'}+EGT(d,j)}$ is the nearest future generation that leads to a smaller than $D_{t_j^{'}}(A_{t_j^{'}}, M)$ defense effectiveness. Formally (Mireles et al. 2019),

$$EGT(A,j) = t_{j^*}^{'} - t_i^{'} \tag{8}$$

when there exists some $t_{j^*}^{'} \in t_{j+1}^{'}, ..., t_k^{'}$ such that

$$D_{t_j^{'}+\Delta t}(A_{t_i^{'}}, M) \geq D_{t_i^{'}}(A_{t_j^{'}}, M) \tag{9}$$

and

$$D_{t_j^{'}+EGT(A,j)}(A_{t_j^{'}}, M) = D_{t_{j^*}^{'}}(A_{t_{j^*}^{'}}, M) < D_{t_j^{'}}(A_{t_j^{'}}, M) \tag{10}$$

4.2 Strategy Evolution - Effectiveness

4.2.1 Evolutionary Effectiveness

Now, we need to compare each generation with respect to a reference generation. Evolutionary Effectiveness metrics measures the difference between two generations. The measurement of generation is a random variable sampled by the opponent's generations. Suppose defense generations are evolved at time $t_0 = 0, t_1, ..., t_l$ and attack generation are evolved at time $t_0^{'} = 0, t_1^1, ..., t_k^{'}$. Here we assume that, defender's generation as the reference generation and attacker's generation with respect to defender's generation (Mireles et al. 2019),

$$EE(\mathscr{A}, i) = \frac{1}{T+1} \sum_{t^{'}=0}^{T} [D_{ti}(A_{t^{'}}, M)] \tag{11}$$

where defender's EE is defined by a random variable,

$$EE(\mathscr{D}, i) = \frac{1}{T+1} \sum_{t=0}^{T} [D_t(A_{t_j^{'}}, M)]. \tag{12}$$

Now, we can define the attacker's capability in terms of the defender's EE($\mathscr{D}$,j) random variable, available exploits a_e, pre-conditions of a node $\mathscr{P}_{re}$

$$\mathscr{C}_a = EE(\mathscr{D},i) \times \sum_{t=1}^{T} a_e \times P_{re} \tag{13}$$

where the pre-conditions of a node depends on the network configuration. For example, pre-conditions might include the presence of certain vulnerable programs, sufficient user privileges, or a particular form of connectivity to other node. From the available set of exploits, attacker will attempt exploits based on his capabilities.

4.2.2 Aggregated-Generation

We need to estimate the overall security gained by the defender when the defender is successful in blocking an attack. Aggregated-Generation (AG) metrics measure the security gained over a time horizon $[0,T]$. Here, the security gain is the advantage over the attacker's progression throughout the network. The gain is quantified by how many times the defender blocked the attacker. The gain is represented by $\mathscr{G}(i)$ where $i = 1,...,T$ (Mireles et al. 2019),

$$AG(\mathscr{D}) = \frac{1}{T}\sum_{i=1}^{T}\mathscr{G}(i). \tag{14}$$

As of now, we have characterized how to capture defender and attacker evolving strategies, we need to assess defense action to achieve higher security gain. To assess various defense action, we employ Partially Observable Monte-Carlo Planning (POMCP) framework that simulates future possible state trajectories from the current security state. The POMCP framework maps the current security state and attacker strategy to a defense action. To quantify the security state, we define the security state as the set of currently enabled security conditions. In this sense, the security state at any given time represents the current capabilities of the attacker. One of our paper's objective is to quantify the level of security of the system as attacker progress. To capture the security level, we define the security state as a current level of the network's attacker progression. In the following section we describes the framework.

5 Control Over Attacker's Decision Making Process

5.1 Decision Calculus

We begin our analysis by laying out an adversary's decision calculus as it applies to the attack context. This represents the base state of adversarial decision making and defender operations, both traditional and complexity focused, that attempt to shape the outcome of this decision flow to achieve a desired set of operational effects. In some cases, the outcome might be more robust deterrence as the attacker sees that the likelihood of arriving at a desired end state is diminished. In other cases, the outcome might be an inability of the attacker to act quickly as necessary information is denied and risk-reward trade-offs are more complex. We focused on decision making as the target of these attacks because, as we explored possible analytical frames, this provides the clearest articulation of ways that complexity exerts its influence in operational practice. In this paper, we assume that the defender can collectively formulate the attackers decision calculus. Rather, we focused on how the defender can control the attackers' decision making process if decision calculus is given. In the previous section we identified and formulated metrics to provide the insights onto the level of complexity a sequence of events will impose on an opponent. In the following section, we describe how the metrics will help in capturing and controlling the attacker decision making process.

5.2 Decision Making Process

Most of the cyber security algorithm assumes that the attackers make rational decision. However, human decisions making process are only boundedly rational and based on the similarity of the present contextual features to past experience. Attackers' Decision-Making is a complex process. To capture the attacker's decision making process, we categorize the attacker based on attacker's i)Knowledge, ii) Stealthiness, and iii) Aggression level. *Aggression* level is defined by the conditional attack probabilities and success. From the reconnaissance stage of attack phase, attacker collects critical information from the network and make the attack plan. Identifying the available set of exploits defined the level of the attacker's *knowledge*. *Stealthiness* is described by the false alarm and the probabilities of detection. The attacker types we consider is presented in Table-I. To capture the attacker types, we consider numerical values from 0 to 10 range for different levels of attacker's *knowledge*, *aggression*, and *stealthiness* presented in Table 1. The defender needs to make the decision based on the attacker types which sometimes turn the defender life tough. However, an autonomous agent can interact with the defender and make decision by automatically. So that we need a model-free technique where it can find the best course of action given its current state. Q-Learning is a reinforcement learning algorithm where it can come up with rules of its own, or it may operate outside the policy given it to follow.

Table 1: Attacker types.

Attacker Types	Knowledge	Aggression	Stealthiness
Type-I	High (9)	Moderate (7)	High (7)
Type-II	Moderate (7)	High (9)	High (9)

6 Experimental Evaluation

In this section, we demonstrated that using reinforcement based learning algorithm i.e., Q-Learning we can gain control over the attacker decision making process. We placed decoy nodes along with the real nodes in the network and measure the effectiveness of when the attacker is an autonomous agent.

6.1 Simulation Setup

We adopted the simulation environment from the toy capture the flag (ToyCTF) example provided in the CyberBattleSim code (Microsoft Defender Research Team. 2022). CyberBattleSim is an experimentation research platform to investigate the interaction of automated agents operating in a simulated abstract enterprise network environment. The network topology in this simulation platform is fixed, and a set of vulnerabilities are present, which can be used by the attacker to move laterally. We modified the ToyCTF code to include the metrics defined in section 4. On each run of the simulation, we first measure the defender's success probability and the attacker's attack probability which helps to limit the attacker's progression into the network. Then, the Q-learning part is run to measure the attacker's win percentage over the number of decoy networks.

6.1.1 Action Space

We consider three exploits which includes local, remote, and connect and control for the attacker as the attacker action space. *Local exploits* are executable on a node the attacker controls, *remote exploits* the attacker can exploits from the local node to remote node. *Connect and control* exploits can be executed with a matching credential object on a node which is visible to the attacker.

6.1.2 State Space and State Transitions

The scenario presented in Figure 1 involves a small network with machine running different OSes, software. We assume that a single attacker (agent) is present in the network and the goal of the attacker is to maximize the reward by discovering and exploits nodes in the network. Here, the environment from the attacker perspective is partially observable. The attacker can not see all the nodes and edges of the network graph in advance. Instead, the attacker take actions to laterally move and observe the environment. For

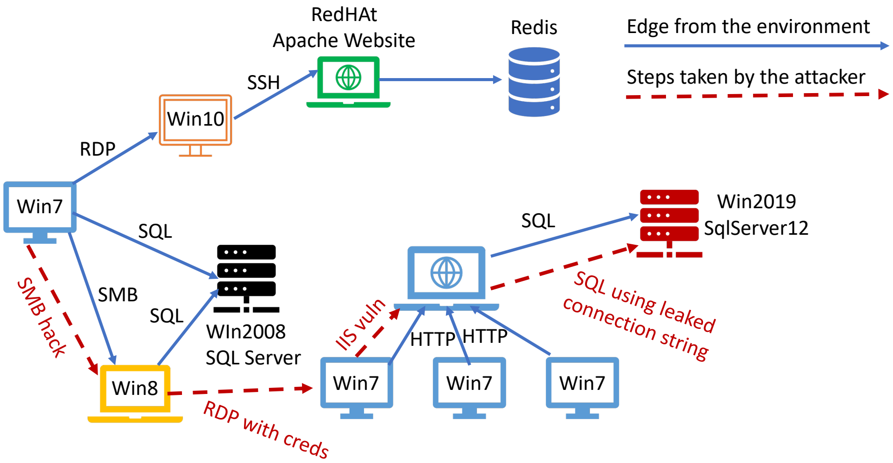


Figure 1: A toy example of network with machines running different OSes, software without deception.

the simulation, we consider three types actions the attacker can take to offer a mix of exploitation and exploration. The *local exploits* reveals credentials to exploit the local node and it can used later with a *connect and control exploit* to gain control of the specific node. *Local exploit* can also discover the path to neighboring nodes. *Remote exploits* are used form the local node to exploit the remote node. The reward represents the the intrinsic value of a node. The attacker breaches into the network as follows (represented by red arrow in Figure 2):

$$Win7 \rightarrow Win8 \rightarrow Win7 \rightarrow IIS \rightarrow SQLDB$$

6.1.3 Decoy Nodes

We consider decoy nodes (Al Amin et al. 2019) as deceptive elements. Decoy node is appeared as real node and the attacker can exploit and control the decoy node. Attacker can find the credentials for decoy node in the real node and leading to the decoy node. Any new connection established to the decoy node generates a defined penalty of -100 for the attacker. We do not increase the penalty when the attacker make the repeating actions. Instead, the repeating action incurred a penalty of -1.

6.1.4 Reward Function

To facilitate penalties with the deception the original code was modified as per following reward function:

- Decoy connection: -100
- Exploit worked: +50
- Exploit use: -1
- Control of node: +1000
- Win condition: +5000
- Repeated mistake: -1

6.1.5 Learning Algorithms

We consider the attacker agents provided by the CyberBattleSim which includes the following algorithms:

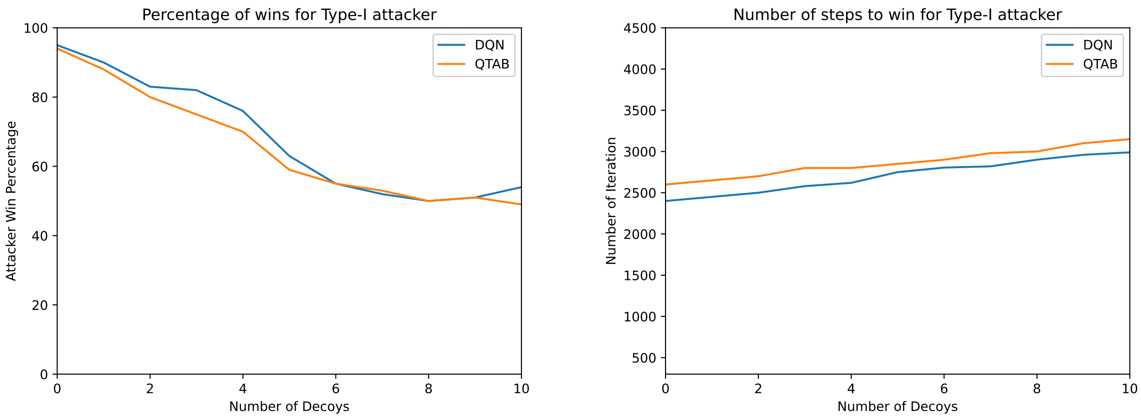
Deep Q-Learning: Deep Q Networks (DQN) are neural networks that utilize deep Q learning to provide models. DQL uses Neural Network as the Q-value function approximator. The agent interacts with the environment and applies rewards based on the current state and action. It is also could be defined as value based RL agent. The following parameter is used in the simulation $\varepsilon = 0.9$, epsilon-min= 0.1, epochs= 300, steps= 5000.

Tabular Q-Learning: Q-learning is a sample based version of Q-value iteration. This method attempts to directly find optimal Q-values, instead of computing Q-values of a given policy. The value for $\gamma = 0.025$ is selected for the simulation.

6.2 Results

In our first simulation, we consider attacker Type-I and the toy example network presented in Figure 1. We deployed the decoy nodes along the real nodes. The goal of the RL agent is to learn a policy so that the expected cumulative reward can be maximized. We consider two metrics to assess the deception framework. Firstly, we calculate the percentage of the attacker wins for defined attacker types in Table 1. Secondly, we measure the resources the attacker needs to spend because of the deception is present.

Attacker's win is the percentage of episode where the attacker meets the win criteria. The win condition for the attacker is defined when the attacker takes control of the real nodes. Figure 2 represents that for type-I attacker agents using different algorithms that can vary overall success on the cyber attack goal. When the number decoy nodes are increasing, the win percentage is also decreasing. Deep Q-network (DQN) outperforms the QTAB algorithm. It is evident from the Figure 3 that the attacker the attacker spending more resources (number of iteration increases) when the number of decoy nodes increases.



(a) Attacker's win percentage when the number of decoy increases.

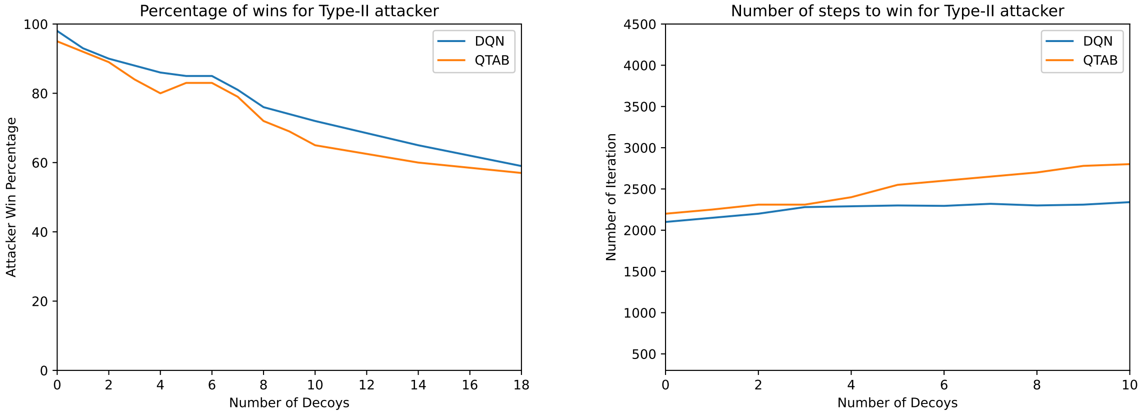
(b) The number of steps required for the attacker to win as a function of the number of decoys.

Figure 2: Attacker's performance analysis.

For type-II attacker, we find that attacker win percentage increases compare to type-I attacker demonstrated in Figure 4. In type-II, we increased the value of attacker's aggression and stealthiness value. Because of high aggression and stealthiness, the number of iteration to enter the win state for the attacker also decreased as shown in Figure 5. Our findings suggest that even though the attacker is highly skilled, adding decoy nodes can significantly improve the overall network security. Although defender agent is not employed in this paper but our results can provide insight for an autonomous cyber defender. Early placement of decoy nodes in the network can effectively block the attacker from lateral movement. Our next steps include addition of more deceptive elements with an autonomous deceptive cyber defender.

7 Conclusion

In this paper, we propose the framework for characterizing adversarial strategies in complex networked systems by developing a set of metrics and provides evaluation to measure the effectiveness of deception.



(a) Attacker's win percentage when the number of decoy increases.

(b) The number of steps required for the attacker to win as a function of the number of decoys

Figure 3: Attacker's performance analysis.

We formulated two sets of metrics to capture the defender and attacker strategy evolution in terms of time and effectiveness and attacker's progression based on the Bayesian attack graph. While defender's strategy evolution not employing in the simulation but or findings measure the effectiveness of employing decoy nodes in the network. We categorize two type of attacker based on attacker's aggression, knowledge, and stealthiness. Increasing decoy nodes in the network can effectively reduce the highly skilled attacker win percentage. Thus, in the future work we will consider adding more deceptive elements i.e., honeypots, honey-tokens. We will also incorporate autonomous cyber deceptive agent in the simulation.

Acknowledgment

This work is supported by DoD Center of Excellence in AI and Machine Learning (CoE-AIML) under Contract Number W911NF-20-2-0277 with the U.S. Army Research Laboratory. The U.S. Government is authorized to reproduce and distribute copies for Governmental purposes notwithstanding any copyright or other restrictive legends.The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force or the U.S. Government.

REFERENCES

Al Amin, M., S. Shetty, L. Njilla, D. Tosh, and C. Kamouha. 2019. "Attacker capability based dynamic deception model for large-scale networks". *EAI Endorsed Transactions on Security and Safety* 6(21):e2.

Al Amin, M. A. R., S. Shetty, L. Njilla, D. K. Tosh, and C. Kamhoua. 2021. "Hidden Markov Model and Cyber Deception for the Prevention of Adversarial Lateral Movement". *IEEE Access* 9:49662–49682.

Ammann, P., D. Wijesekera, and S. Kaushik. 2002. "Scalable, graph-based network vulnerability analysis". In *Proceedings of the 9th ACM Conference on Computer and Communications Security*. November 18th-22nd, Washington DC, USA, 217–224.

Cyberspace, T. 2011. "Strategic plan for the federal cybersecurity research and development program". *Executive Office of the President National Science and Technology Council*.

Kührer, M., C. Rossow, and T. Holz. 2017. "Paint it black: Evaluating the effectiveness of malware blacklists". In *Proceedings of the 17th International Symposium of Research in Attacks, Intrusions and Defenses (RAID)*. September 17th-19th, Gothenburg, Sweden, 1-21.

Levin, D. 2003. "Lessons learned in using live red teams in IA experiments". In *Proceedings DARPA Information Survivability Conference and Exposition*. April 22nd-24th,Washington DC, USA, 110–119.

Microsoft Defender Research Team. 2022. "https://github.com/microsoft/cyberbattlesim". accessed 12th January 2022.

Mireles, J. D., E. Ficke, J.-H. Cho, P. Hurley, and S. Xu. 2019. "Metrics towards measuring cyber agility". *IEEE Transactions on Information Forensics and Security* 14(12):3217–3232.

Pendleton, M., R. Garcia-Lebron, J.-H. Cho, and S. Xu. 2016. "A survey on systems security metrics". *ACM Computing Surveys (CSUR)* 49(4):1–35.

Poolsappasit, N., R. Dewri, and I. Ray. 2011. "Dynamic security risk management using bayesian attack graphs". *IEEE Transactions on Dependable and Secure Computing* 9(1):61–74.

Sugrim, S., S. Venkatesan, J. A. Youzwak, C.-Y. J. Chiang, R. Chadha, M. Albanese, and H. Cam. 2018. "Measuring the effectiveness of network deception". In *Proceedings of the 2018 IEEE International Conference on Intelligence and Security Informatics (ISI)*. November 9th-11th, Miami, Florida, 142–147.

Zhan, Z., M. Xu, and S. Xu. 2015. "Predicting cyber attack rates with extreme values". *IEEE Transactions on Information Forensics and Security* 10(8):1666–1677.

AUTHOR BIOGRAPHIES

MD ALI REZA AL AMIN received the M.S. degree in computer information and system engineering from Tennessee State University, Nashville, USA, in 2016. He is currently pursuing the Ph.D. degree in computational modeling and simulation engineering with Old Dominion University, Norfolk, USA. His research interests include cyber security, cyber deception technology, and AI in cyber security. His email address is malam002@odu.edu.

SACHIN SHETTY received the Ph.D. degree in modeling and simulation from Old Dominion University, in 2007. He is currently the Associate Director of the Virginia Modeling, Analysis, and Simulation Center, Old Dominion University. He holds a joint appointment as an Professor with the Department of Computational, Modeling, and Simulation Engineering. His research interests include the intersection of computer networking, network security, and machine learning. His email address is sshetty@odu.edu.

CHARLES KAMHOUA received the M.S. degree in telecommunication and networking and the Ph.D. degree in electrical engineering from Florida International University (FIU), in 2008 and 2011, respectively. He is currently a Researcher with the Network Security Branch, U.S. Army Research Laboratory (ARL), Adelphi, MD, where he is responsible for conducting and directing basic research in the area of game theory applied to cyber security. His email address is charles.a.kamhoua.civ@army.mil.