Game Theory with Learning for Cyber Security Monitoring

Keywhan Chung*, Charles A. Kamhoua†, Kevin A. Kwiat†, Zbigniew T. Kalbarczyk* and Ravishankar K. Iyer*
*Coordinated Science Laboratory, University of Illinois, Urbana, IL
kchung10, kalbarcz, rkiyer}@illinois.edu
†Cyber Assurance Branch, Information Directorate, Air Force Research Laboratory, Rome, NY
{charles.kamhoua.1, kevin.kwiat}@us.af.mil

***Abstract*—Recent attacks show that threats to cyber infrastructure are not only increasing in volume, but are getting more sophisticated. The attacks may comprise multiple actions that are hard to differentiate from benign activity, and therefore common detection techniques have to deal with high false positive rates. Because of the imperfect performance of automated detection techniques, responses to such attacks are highly dependent on human-driven decision-making processes. While game theory has been applied to many problems that require rational decision-making, we find limitation on applying such method on security games. In this work, we propose Q-Learning to react automatically to the adversarial behavior of a suspicious user to secure the system. This work compares variations of Q-Learning with a traditional stochastic game. Simulation results show the possibility of Naive Q-Learning, despite restricted information on opponents.**

I. Introduction

Computer systems are tempting targets for attackers. Successful invasion of data and control can threaten the availability of the computer system and compromise the integrity and confidentiality of information processed or stored in the system. To defend systems from exploits, sensors and monitors are deployed at different layers of the system, and system and user activities are logged and audited to filter out suspicious or malicious activities and/or trigger an in-depth investigation or response to a potential attack. While that type of monitoring and response has been an effective method for detecting hostile activities against systems, recent analysis shows a change in attack trends against cyber systems. For example, according to [1], attacks are not only increasing in number, but are also getting more sophisticated and intelligent, which is accelerating an increase in the number and variety of security measures applied to systems. However, naive deployment of more security monitors and policies does not always lead to better detection. While such increments bring better security coverage, they also increase the complexity of analysis and overhead in terms of performance degradation. In [2], an analysis of security incidents shows that a significant portion of alarms that trigger human investigation turn out to be false positives. In addition, it was shown that most of the incidents were detected only after actual damage was done. Those observations reveal a need for an automated intrusion response that can react to malicious actions threatening a computer system. In this paper, we propose a game-theoretic model to emulate the decision-making process in responding to cybersecurity incidents. Given an attack model and a reward model based on expert knowledge, our approach determines the optimal action that minimizes damage. We focus on intrusion response; detection of zero-day attacks and unknown attacks are outside the scope of this paper.

In order to better model learning in security games, we considered variations of Q-Learning where Q-Learning (QL) [3] is a model-free reinforcement learning technique, used to learn the optimal policy that maximizes the expected reward (e.g., the monetary value of the information exploited or an estimated loss caused by compromised system availability). We claim that our approach is more realistic than pure game theoretic approaches[4], as it uses an algorithm that releases the restrictions on the rationality of the players or the completeness of information. Q-Learning algorithms are discussed in more detail in section IV.

With respect to learning, earlier work has shown the effectiveness of applying machine learning for cyber security. However, we find that rationality has been neglected in the existing machine learning approaches. For the models trained on the dataset, the model becomes specific to the observations from history. Hence, such approaches take time to adapt to unforeseen patterns. Q-Learning on the other hand makes decisions based on a model that was derived from human intelligence. Moreover, assuming the possible incompleteness of the model, the approach reinforces the model by adapting to the patterns from the previous iterations.

In this paper, the performance of Q-Learning algorithms (MMQL, NQL) for detecting execution of a multistage attack is evaluated through comparisons of the cumulative earnings of the attacker after multiple iterations over the game. Simulating a security game of attackers and defenders with different abilities and knowledge, we show that Nave Q-Learning has a potential on minimizing the loss against non-fully rational attackers. The main contributions of this paper are:

- Motivates the approach through a study of real incidents. From an analysis of the Target data breach and earlier work on security incidents, we show the need for automation in incident response.
- Models the battle of an attacker and a defender as

[0] Approved for Public Release; Distribution Unlimited: 88ABW-2015-3067, 16 Jun 2015

[0] This material is based on research sponsored by the Air Force Research Laboratory and the Air Force Office of Scientific Research, under agreement number FA8750-11-2-0084. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

TABLE I: Summary of attacks in a large organization

ID	Time	Event
1	09:52	log in from known host using public key authentication
	09:59	changed 'authorized_key'
	10:05	logged in from new host / updates 'known_hosts'
	11:19	download malware from remote host
	11:20	attempt root escalation
2	17:34	log in through WinVNC exploit
	17:49	download malware from remote host
	17:52	connection to blacklisted command & control systems
3	07:38	access network with weak credentials
	07:39	download malware from remote host
	07:40	start bruteforce ssh scan
	09:15	attemp ssh scan on a DDoS black list

a security game. Using real incident data from the Organization X, we derive an attack model that reflects both the attackers and defenders perspective, and we use the model to formulate a security game. In terms of a security game, this model represents the worse case where the attacker can perform all attacks shown in the dataset.

- Presents an experimental result showing the possibility of applying Nave Q-Learning for effectively learning the opponents behavior and making a proper decision. Comparing the performance of different decision making algorithms, we present simulation results that show Naive Q-Learning performing better than algorithms with restricted assumptions, especially against irrational attackers, and show that Naive Q-Learning performs as well as Minmax Q-Learning, despite the relatively limited information.

II. Motivaton

Unlike earlier attacks, whose goal was to invade a target and leave as quickly as possible after causing damage, recent attacks show that attackers are willing to remain in the system undetected. Hence, the attack sequences are designed to consist of a set of actions that are hard to differentiate from legitimate ones. These type of attacks are often called Advanced Persistent Threats (APT). [5] For example, from an analysis of a recent data breach attack [6], it was shown that the attackers resided in the system undetected for more than a month. We notice two interesting aspects of that incident. One is the decision model that lay underneath the attack. The attack consisted of multiple states (phases), and a decision on the next action to be taken had to be made in each state. From that decision model, we recognized the possibility of applying game-theoretic approaches to counteract malicious intent. Also, we concentrate on the fact that the system was able to detect the intrusion but no proper response was made for easily assuming the alarms as false positives. Often, alarms lack on confidentiality as they rely on limited observation. Instead, an attack has to be understood as a sequence of events that calls for the detection/response model to encompass observations from varying dimensions. In addition, we note the insights from a study of security incidents at a large computing organization. [2] That study has shown how many alerts turn out to be false positives, how they affect the detection of real incidents, and how dependent the organization is on human expertise. According to the paper, the majority of incidents are not detected until the actual damage to the system has been initiated.

From those observations, we see a need for an automated decision-making process to respond to potential attacks. Such a process should provide a decision based not only on the current observation, but also on results from the past and the expected result of taking each action available at the decision-making state. In this paper, we discuss a method to automatically determine the response, given the observations on the system states from a set of monitors.

III. Attack Model

In modeling an attack, we are considering parties with a conflict of interests: the attacker and the defender. The defender, often a system administrator, manages the system. The main interest of the defender is to secure the cyber infrastructure from malicious activities. The attacker, on the other hand, is a malicious opponent who attempts to compromise the target system. We model the interaction between the attacker and the defender based on data on actual security incidents.

A. Attacker

The attacker is an opponent who accesses the system with the intention of threatening its security. Attacks can vary from a single action to a sequence of activities. In this paper, we limit our interest to attacks that consist of multiple activities that lead to an ultimate goal.

Attack State AS_x represents the state of the attack, i.e., the depth/degree of intrusion. Each attack state is assigned a numeric value(reward) which quantifies the damage to the target system. The bigger the impact, the more severe the damage to the system and/or the greater the unauthorized control over the system. Transition from one state to another depends on the result of the *action*.

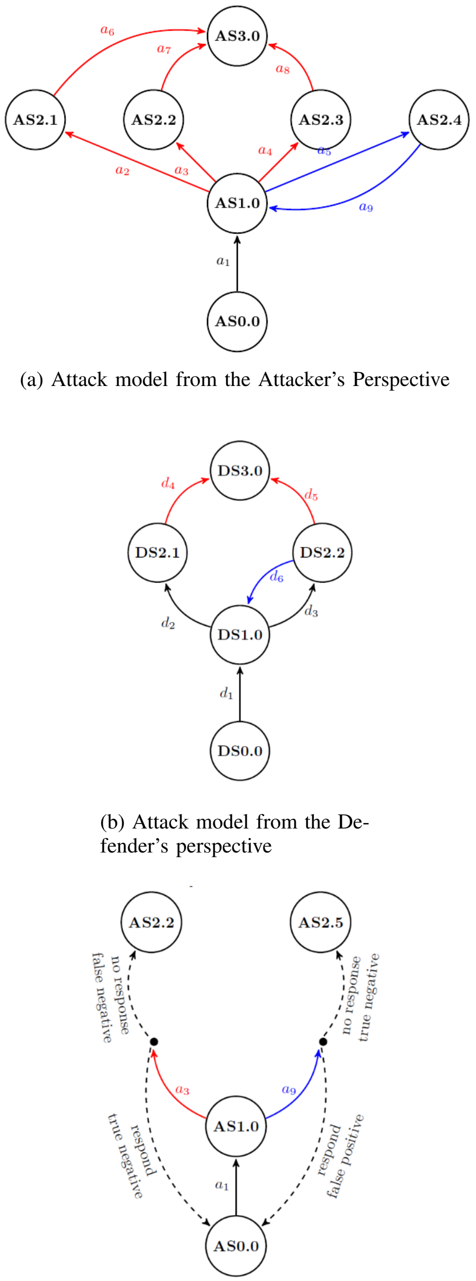
Activity A is a set of actions a_i available to the attacker. It can lead to malicious control over the system, or if the attacker decides to remain in the current state, the transition will result in a loop. The set of available activities in state AS_x is denoted by A_x. Therefore, A_x is a subset of A. The causal relation between activities and attack states can be represented as a state diagram.

Transition Matrix $P_a(s, s')$ is the probability that an action from state s will lead to a transition to the next state s'. In an attack model, a transition matrix represents the probability of a successful attack. Depending on the monitoring system configured on the defender's side, an attack can be either detected or missed. The transaction matrix models the uncertainty of the result of an action.

Immediate Reward $R_a(s, s')$ is the reward of the attacker as a result of a transition from state s to s' for performing action a. The reward is a quantitative representation of the earnings that the attacker can get from a successful attack.

B. Defender

The defender is a party that is in charge of making proper responses to secure the system from malicious attacks. The defender has a set of monitors to protect the system. The



(a) Attack model from the Attacker's Perspective

(b) Attack model from the Defender's perspective

(c) Snapshot of the Security Game

Fig. 1: State Diagram Representations of the Attack Phase (a, b) and a Snapshot of the Security Game (c).

main objective of this player is to make prooper responses in a preemptive manner based on a limited view of the system status, relying on monitors.

Attack State DS_x represents the state of the attack from the defender's perspective. The observations that defenders use rely on the monitoring systems, and lack the granularity needed to reveal the details of users' actions.

Defender Action D is a set of actions(d) available to the defender in a given state. For security incident detection and response, a monitor detects changes in system status. However, such detections do not directly map to the attacker's definite actions. The monitor may miss an action (false negative) or misidentify a benign action as malicious (false positive). Hence, the defender needs to take an appropriate action while relying on imperfect information. Assuming that there are proper responses for each action, we abstract the defender action to either "Reponse" or "No Response," where "No Response" is useful for monitored events that are hard to differentiate from benign ones, and/or events that do not cause immediate harm to the system.

C. Attacker-Defender interaction

While each attacker has a logic flow for making decisions, his or her decisions are not independent, but are related to the opponents decision process. Hence, we model the interaction between the two players. In Figure 1c, we show a subset of the security game. Once an attacker has taken an action, the defender chooses his or her action based on the information from the monitoring system. An attackers action results in a transition to the intended state only if the defender does not make a proper response. Once the defender has responded to the observed action, the attacker is forced to transit to the default state. Assuming a zero-sum game, a successful attack will result in an immediate reward, and the defender will have a symmetric loss. As a result of the execution of the attack, the attack state will change accordingly. Otherwise, if the defender detects the attack and makes a proper response, the attack state will be reset to the default for the identified attacker. In that case, a reward will be assigned to the defender, with an equivalent loss to the attacker.

D. Study of Organization X Data

In [2], it was shown that 62% of the incidents in the study were detected only after attacks have already damaged the system, for the monitoring relies heavily on alerts from the IDS; no significant evidence is available until the actual attack has started. Analyzing the incidents at Organization X, we find a common pattern in the attack phase. For example, as shown in the sample incidents summarized in Table I, suppose a malware file has to be downloaded for an attack to progress. Unless the attacker has downloaded well-known malware whose signature will match the malware detection database, the monitoring system will report the event as a general file download. Since downloading of a file is a general action often performed by benign users, it would be difficult to use a file download as the basis for determining that an attack is occurring. However, once the malware has been executed, the system has already been exploited, with visible damage. Hence, it would be ideal if a useful decision on the response could be made at the download phase.

Figure 1 represents the attack model from the attackers and defenders perspectives; the meanings of the labels are explained in the table below the figure. Figure 1a models an attack in four phases. $AS_{0.0}$ is a base state. It is the default state, in which the attacker has not taken any malicious action. In this state, an attacker cannot be differentiated from a benign user. Using stolen credentials or already exploited backdoors, attackers gain access to the system, which leads to $AS_{1.0}$. In $AS_{1.0}$, the attacker has 4 options: to download

TABLE II: Comparison on Different Approaches

	MG [4]	QL [3]	MMQL [7]	NQL [8]
Number of Agents	multiple	single	multiple	multiple
Required Opponent Info	full	n/a	limited	no
Learning	no	yes	yes	yes
Adapt to Opponent	no	n/a	partial	partial

malware (well-known, rare, or new) or download benign software. Downloading benign software would not give immediate benefit to the attacker, but we model it to represent a case in which the attacker is trying to obfuscate the user profile. A successful download leads to $AS_{2.X}$. Once the attacker has reached $AS_{2.1}$, $AS_{2.2}$, or $AS_{2.3}$, he or she can execute the malware and exploit the system. Our goal is to be able to make a proper response before the exploit happens. Figure 1b depicts the defenders view of the same attack model depicted in Figure 1a. Because of the limitations of the monitoring system, the defender has a limited view of the files downloaded in $AS_{2.X}$. If the malware has a known signature predefined in the monitoring system, the file can be recognized as malware (d2). Otherwise, a malware download will be seen as a benign file download (d3).

IV. Game Model

In this section, the interaction of an attacker and a defender is discussed in terms of games. We refer to [7], [9] and [4] for the definitions and equations for formulating the game. The game consists of two rational players with conflict, and their goal is to maximize their reward by deriving the optimal policy for each state. The comparison of different methods is summarized in table II.

A. Stochastic Game

First we define the terminologies used for solving the game as a Stochastic game[4].
Set of actions A contains all possible actions, a that are available to the player. We use o for the opponent's action.
Reward $R(s, a, o)$ defines the immediate reward based on the attack state s and player's actions, a and the opponent's action o in the tth iteration.

A stochastic game that consists of multiple stages is often called a Markov game. In a Markov game, the concepts of quality of state and value of state are introduced to represent the expected reward of the player's decision.
Value of state $V(s)$ is the expected reward when the player, starting from state s, follows the optimal policy. It is equivalent to the maximum reward that the player can expect, assuming that the opponent's action o will be the action that minimizes the expected reward. The player maximizes the value of state by deriving the optimal policy, i.e., the probability distribution among the actions available to the player in a given state.

$$V(s) = \max_{\pi} \min_{o' \in O_s} \sum_{a' \in A_s} \pi(s, a') Q(s, a', o') \tag{1}$$

Quality of state $Q(s, a, o)$ is the expected reward each player can gain by taking actions a and o from state s and then following the optimal policy from then on. The quality of state is a sum of the immediate reward from this iteration (R^{t-1}) and the reward expected as a result of transitioning to state s' ($V^t(s')$), which was derived from the previous t iterations. Note that the value of state is weighted by a discount factor (γ).

$$Q^{t+1}(s, a, o) = R^{t+1}(s, a, o) + \gamma V^t(s') \tag{2}$$

Discount factor γ is assigned by the user's intention on balancing between future and current rewards. A myoptic player, who only considers current reward, is modeled by a value of 0 while 1 is assigned for a player who strives for a long-term high reward.
Optimal policy π is the set representing the probability distribution of actions ($\pi(s, .)$) available at each state(s). It is chosen to maximize the value of state($V(s)$) which represents the expected reward of the player if the player follows the optimal policy. $\pi(s, a)$ indicates the likelihood of taking action a in state s where π is the overall distribution that maximizes the value of state ($V(s)$).

$$\pi(s, .) = \arg \max_{\pi'(s,.)} \min_{o \in O_s} \sum_{a' \in A_s} \pi(s, a') Q(s, a', o') \tag{3}$$

B. Minimax Q-Learning

To solve stage-based games, a Markov game assumes full rationality and complete information about the opponent. However, an empirical study involving a guessing game has shown that the assumption on complete information and full rationality is not realistic in all cases [10].

In a security game, the assumption of complete information and rationality is even more unrealistic. In security games, players generally make decisions with limited information, and compendate for their lack of information with learning[11]. To account for that characteristic of security games, we apply Minimax Q-Learning as a decision making algorithm. Instead for a need of complete information on the attack model, the Minmax Q-Learning algorithm allocates partial weight on its earlier results to combine knowledge of history, the actual earnings on the current iteration, and the future expected reward.
Quality of state for Minimax Q-Learning is defined as follows to embed the learning aspect into the algorithm.

$$Q^{t+1}(s, a, o) = \alpha Q^t(s, a, o) + (1 - \alpha) R^{t+1}(s, a, o) + \gamma V^t(s') \tag{4}$$

Learning rate α leverages the ability of the player by assigning a real value between 0 and 1. A learning rate of zero represents full learning ability for the player while a rate of one models the case where the player only considers only the most recent information. In full learning, the player would not consider the immediate reward $R(s, a, o)$ and the expected future award $V(ns)$ but keep the quality of state constant. To account for the absence of prior results to learn from at the initial stage of the game, an α of 1.0 is assigned; α then decays as Q(s, a, o) accumulates information on the performance of previous iterations.[7]
Exploration rate exp is a distinct parameter for Q-Learning which determines the degree of variation from the optimal policy. Unlike the Markov game, in which the optimal solution is known from the initial iteration, Q-Learning has to learn the optimal policy by trial and error. The exploration rate

determines the relative rate of the action not following the optimal policy to learn the results of different actions. An exp value closer to 0 results to a Makov game while a value closer to 1 means that the player will take random actions.

C. Naive Q-Learning

In a security game, information about the opponent is not always available. The attacker often has information about the target system from public resources. However, the amount of information is limited. Similarly, the defender is playing a game against an unspecified opponent. In order to model this situation, Naive Q-Learning from [12] is applied. Naive Q-Learning optimizes the strategy without information about the opponent, such as the the opponent's action o. It utilizes limited information of the immediate reward and its own information to derive the optimal policy.
Quality of state is updated accordingly to reflect the limited information. Note that the opponent's action is no longer considered for differentiating the Quality of state.

$$Q^{t+1}(s,a) = \alpha Q^{t}(s,a) + (1-\alpha)R^{t+1}(s,a) + \gamma V^{t}(s') \quad (5)$$

Value of state is the maximum expected reward when following the optimal policy. Note that because of the lack of information about the opponent, o is no longer considered.

$$V(s) = \max_{\pi} \sum_{a' \in A_s} \pi(s,a')Q(s,a') \quad (6)$$

Optimal policy is the optimal policy that maximizes the value of state ($V(s)$). Note that the quality of state (Q) is only defined for s and a but not o.

$$\pi(s,.) = \arg \max_{\pi'(s,.)} \sum_{a' \in A_s} \pi(s,a')Q(s,a') \quad (7)$$

V. Experiment

To evaluate and analyze the game, a simulation was performed. The simulation started with initialization of the value V and quality Q of all states, optimal policy and the learning rate(α). For Q-Learning, initially there is no information that the algorithm can learn from. Therefore, α is set to 1.0 indicating that initially, the decision relies on the rationality(game model) only. Markov game would evaluate the game model and determines the optimal policy before determining what action to take. Once the model has converged to a convergence coefficient(ϵ), the optimal policy is fixed. Q-Learning, on the other hand, needs to take actions before updating the quality and value of state. The next action is decided based on the *exp* value. A random action is chosen with a probability of *exp*; otherwise the action follows the optimal policy. Once the opponent's action has been determined in a similar manner, the quality of state is updated. Then through the use of linear programming, the optimal policy (π) and the value of state (*V(s)*) can be derived. Recall that the optimal policy is the probability distribution of available actions at a given state that maximizes the value of state V.

Using the simulator, we compared the performances of the algorithms by assuming a random, Makov, Minimax Q-Learning and Naive Q-Learning attacker and pairing with a Markov, Minimax Q-Learning and Naive Q-Learning defenders. We did not consider the unrealistic situation in which a defender is a random player. For the attacker, on the other hand, a random player is a considerable assumption for modeling unprofessional attackers such as script kiddies.

To compare the performance of different algorithms, we evaluated the accumulated immediate reward of the attacker. In addition, we studied how the parameters affect the performance of the Q-Learning based players.

VI. Result

A. Comparison between algorithms

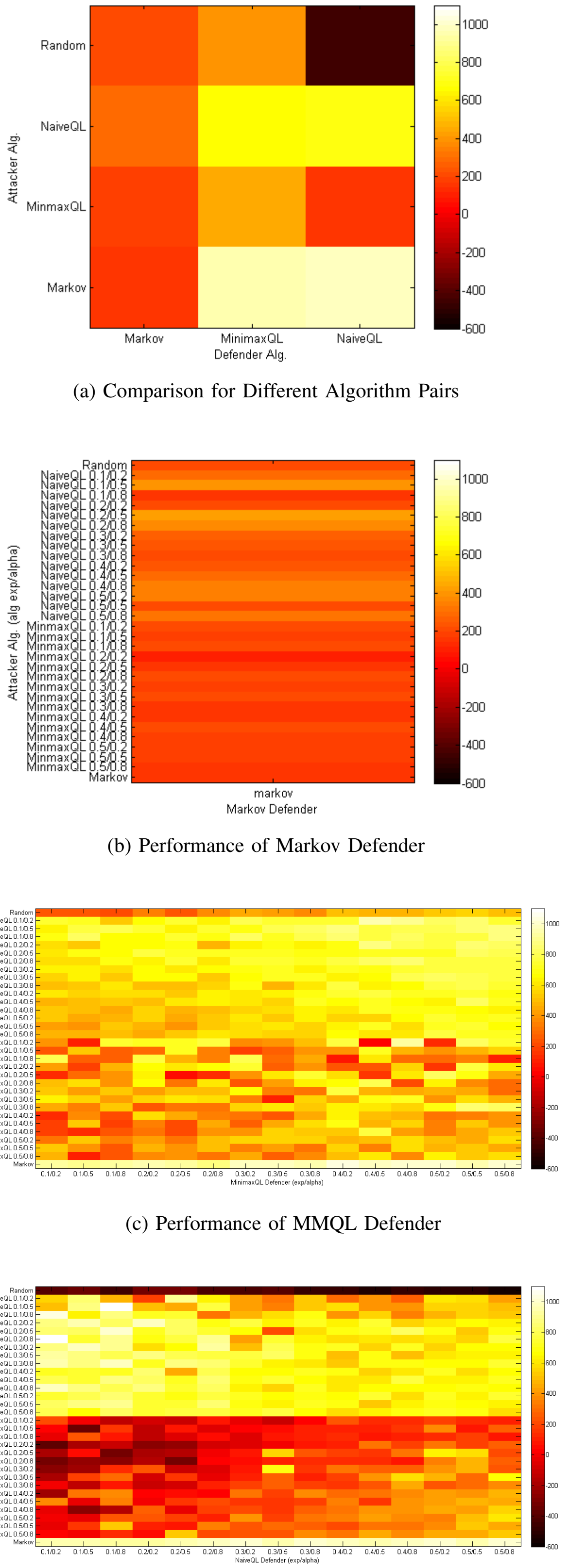
First, we compare how each algorithms with varying parameters perform against different opponents. Figure 2a provides an overview of the comparison. We represent the accumulated reward of the attacker using a heat map in which a lighter (i.e., closer to white) color indicates higher reward. Looking at the first column, we confirm that the defender, on average, shows the best performance when the decision-making is based on Markov games. In addition, we find low variance within the column. From that observation, we see that when the defender has full information about the attacker and hence is playing a Markov game, the attackers choice of algorithm does not make a significant difference. That insight becomes obvious when we consider the assumption upon which the algorithm relies. However, the same insight does not always apply to all algorithm pairs. An interesting observation is that Naive Q-Learning performs better than Minmax Q-Learning, despite the limited information. Comparing the second and third columns, we find that attacker performance when played against a Naive Q-Learning defender (third column) is represented by a darker color (lower accumulatedreward). In addition, as shown in the upper-right corner of Figure 2a, we find that NQL performs better than the Markov game when played against a random attacker. Because the Markov game and Minmax Q-Learning algorithms have more information about the opponent, they are able to formulate a more accurate model. However, as these models assume the rationality of the opponent, they cannot adapt well to the behavior of a random (irrational) attacker. Naive Q-Learning lacks the information about the opponent that would be needed to formulate a complete game model, and hence makes decisions based on its own decision model; that gives it the flexibility to adapt to the attackers empirical behavior. By observing the changes of the optimal policy, we confirm that Naive Q-Learning defenders converge to an optimal policy that has higher probability for counteraction against attacks with higher immediate reward.

Figures 2b through 2d show, in detail, how the attackers performance changes for different the combinations of algorithms, exp, and . In Figure 2b, we confirm the insignificant impact of the parameters (exp and) on the performance. While different exploration rates and learning rates are applied, the rows have low variance for the defenders algorithm driving the game. From Figures 2c and 2d, we can see how the parameters affect the performance of the decision-maker. When the attacker is playing the strongest algorithm (MG), the parameters have no impact. That claim can be confirmed by observing the consistent color across columns in the last row of both figures. While the different columns stand for different parameter pairs, no difference in attacker performance was found. As defined in Section IV, Q-Learning introduces a new parameter called the exploration rate (exp). This rate defines the ratio of actions that are randomly chosen (rather than being chosen by following the optimal policy). From looking at Figures 2c and 2d, we find that there is no single trend for exp, but rather it depends on the algorithms for both players. When the defender is playing MMQL against an NQL attacker, as shown in the upper half of Figure 2c, a higher exp rate of the attacker leads to a higher accumulated reward, while a higher exp rate of the defender lowers the accumulated reward of the attacker. On the other hand, the lower half of Figure 2d shows that when the defender is deploying NQL against an MMQL attacker, the defender reduces the accumulated reward of the attacker with a smaller exploration rate, while the attacker gains more with a higher exploration rate. For both cases, we find that the defender only needs a minimal exploration rate to assure discovery of all possible actions. We find that deviating from the defenders optimal policy does not benefit the defender. Studying the impact of the learning rate, namely the relative weight between the game model and the learning model, we find no clear pattern in the accumulated reward. Instead, we find a potential relationship to the time to convergence, which we discuss in the following subsection.

B. *Time to Convergence for Different Learning Rates*

In Figure 3, we can see how the learning rate affects the time to convergence. When we assume that the attacker is playing the game under a consistent strategy, the time to convergence indicates the time it takes for the defender to derive the strategy that minimizes the loss. From Figure 2b to figure 2d, we saw that the learning rate of the players did not have a significant impact on the accumulated reward of the attacker. To confirm how the learning rate affects the decision-making process, we compare the values of the state at the initial state of the attack model. That value of state ($V(s_0)$) represents the expected reward that the attacker can earn when the attacker follows the optimal policy from the starting state and thereafter. Once $V(s_0)$ becomes constant, we claim that the optimal policy of the defender becomes constant. Note that this analysis does not assign meanings to the value of state for its nature of representing the expected reward, not the actual reward that has or could be earned.

From the formulation of Minmax Q-Learning algorithms, a zero-sum game is expected to converge [7]. In this experiment, we checked whether the learning rate affects the time to convergence. Recall that a low learning rate indicates intensive learning, as more weight is assigned to the previous quality of state than to the sum of the immediate reward and the future expected reward. From Figure 3, we observe two things. One is that the expected reward of the attacker is larger if the defenders learning rate is larger than or equal to that of the attacker. That insight can be confirmed in the figure through comparison of 0.8 0.5, 0.2 0.5, and 0.5 0.5, and comparison of 0.8 0.8 and 0.2 0.8. Recall that 0.8 0.5 is interpreted as Attacker with learning rate 0.8 against a defender with learning rate 0.5. Based on that observation, we can verify that if both parties apply Minmax Q-Learning for decision-making, then it is more likely for the defender to be able to protect the system against a Minmax Q-Learning attacker when the defender weights learning more than rationality. Another observation from the



(a) Comparison for Different Algorithm Pairs

(b) Performance of Markov Defender

(c) Performance of MMQL Defender

(d) Performance of NQL Defender

Fig. 2: Simulation Results: Comparing the Attackers Accumulated Rewards from Playing a Security Game with Different Decision-making Algorithm Pairs.

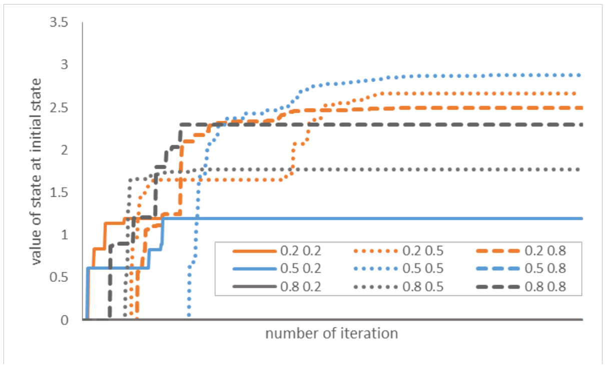


Fig. 3: Comparison of Value of state at initial state. The label indicates the learning rate pair (learning rate of the attacker, learning rate of the defender).

TABLE III: Summary of Related work

	information	repeated game	attack type
Lye 2002[13]	complete	stochastic	battle
Liu 2006[14]	incomplete	no (static Bayesian)	single
Alpcan 2006[8]	complete/ incomplete	stochastic	single
Zonouz2009[15]	complete	yes	single
Becker 2011[16]	complete	no	single
Markov	complete	stochastic	battle
MMQL	incomplete	stochastic + learning	battle
NQL	incomplete	stochastic + learning	battle

figure is that the game converges faster for the attackers with lower learning rates. The learning rate of the opponent (the defender, in this case) also affects the time to convergence of the player. While there is a slight deviation between 0.5 and 0.8, a lower learning rate for the defender (opponent) also accelerates the time to convergence of the attacker. A similar analysis for Naive Q-Learning players is not applicable, because the Naive Q-Learning algorithm is not guaranteed to converge.

VII. Related Work

Modeling attacker intent or attack flow in a graphical model has been a well studied problem in security. For the applicability and variation of game theoretic models, numerous approaches exist for modeling security as a game. Bier[17] provides a good study of a defender securing a set of potential targets with limited resource. The author solves problem in a resource constrained environment to answer policy questions to better secure the physical target against threat of terror. Though this paper solves a physical security problem, it provides a good framework for cyber security games. Liu et. al present a game theoretic model to infer attacker intent, objectives, and strategies(AIOS)[18]. Considering the incomplete knowledge of each players on the opponents, the authors choose a Bayesian Game model. The model also introduces the state of the attack. The attack state is normally predicted from observable events.

A number of previous works apply traditional game theoretic approach for cyber security problems. In [19] and [20], the authors apply a Markov decision process(MDP) to secure information sharing in online social networks. In addition, [21] and [22] apply a Markov game framework for optimal data management in online social network. Nguyen et. al, apply fictitious play[23] for solving a security game with incomplete information[9]. Similar to [12], information on the opponent's payoff matrix is not available, hence the agent derives the belief of its prediction on the opponents strategy by monitoring the result of its own move. However, such approach has limitation on applying to a Markov game consisting of numerous states. Alpcan and Basar presents a comprehensive study on modeling security games under different level of information about the opponent[12]. In their model, the interaction of two players are modeled as a stochastic(Markov) game. Each player has an option of attack/no attack or respond/not respond. They present a simulation model under three different conditions: perfect information about the system (Q learning), partial(action set, transaction history) information about the opponent(Minimax Q-Learning) and no information about the opponent(naive Q learning).

VIII. Limitations and Future Work

Unlike many approaches using machine learning [24], [25], our approach is not intended to detect new attacks. Instead, our game theoretic approach, like other decision making applications, suggests a likelihood of taking a certain action to maximize the benefit of the security administrator.

Also, while our approach is based on the attack and reward model, because of the lack of agreement on security metrics, there is no true measure for quantifying the rewards of a successful attack or attack detection. Therefore, the current configuration relies on expert knowledge to enumerate the potential damage or overhead for taking a certain action.

One last limitation is that the attack models performance depends on expert knowledge. Because the decision-making process is based on the attack model, the granularity and completeness of the attack model affect the performance. That limitation is intrinsic to pure game-theoretic models, and we claim that our approach can compensate for that shortcoming of the attack model by adding the ability to learn from past iterations. However, lack of coverage of undefined actions or states still remains as a limitation.

While the present work was focused on automated intrusion detection, we plan to test and evaluate our work by embedding it in a security analytics testbed [26]. With real incident data fed into the testbed and factor graph [27] detecting malicious intentions; our approach will then determine the right response to take. By combining our game theory based response model with the detection framework, we expect to verify the timeliness of intrusion detection and response and determine the accuracy and impact of misdetection.

IX. Conclusion

In this paper, we presented our approach for modeling the decision-making process of cyber security monitoring using a game-theoretic approach. To reflect the realistic conditions of decision-making in a security game, we considered variations of Q-Learning algorithms. Minmax and Naive Q-Learning, compared to traditional Markov games, are more realistic when applied to security games, because they relax the requirement for full information about the opponent. We compensated for the lack of information by enabling learning of the optimal policy, which has the advantage that it resembles situations in which attackers probe system vulnerabilities (through techniques like scanning) and defenders train and renew security policies and devices based on earlier data. We noted that the rich literature in online learning theory lacks efforts to reason about pattern to capture the rationality of attackers in security games.

From the experiments based on simulation, it was shown that Naive Q-Learning performs well against irrational (non-Markov) attackers, i.e., random decision-makers or attackers based on probing and learning. When played against a Markov attacker, the Naive Q-Learning approach was able to perform at least as well as a Minmax Q-Learning defender. In the real space of security games, players, especially the defenders, have limited ability to obtain information about their opponents. Any parties with access to the system are potential attackers, and their ability and knowledge not only vary but are hidden. Hence, a Markovian attacker, which represents the worst case for the defender, is unrealistic. The simulation results show that despite the limited information on which decisions are based, our approach is promising compared to the traditional Markov game approach and Minmax Q-Learning.

While the paper presents the possibility of Naive Q-Learning as a decision-making logic in security games, some limitations remain. For the lack of agreement in metrics that represent impacts, there is no clear definition of the reward model. In this paper, the reward was abstracted as the relative severity under an assumption of a zero-sum game, indicating that a players reward is his or her opponents loss. In addition, for the dependency of the parameters to the opponent, it is necessary to do further study on how to tune the parameters of a Naive Q-Learning algorithm against an attacker from real data, and to test the approach embedded in a framework with real-time logs and specific detection logic.

References

[1] S. J. Templeton and K. Levitt, "A requires/provides model for computer attacks," in *Proceedings of the 2000 workshop on New security paradigms*. ACM, 2001, pp. 31–38.

[2] A. Sharma, Z. Kalbarczyk, J. Barlow, and R. Iyer, "Analysis of security data from a large computing organization," in *Dependable Systems & Networks (DSN), 2011 IEEE/IFIP 41st International Conference on*. IEEE, 2011, pp. 506–517.

[3] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[4] L. S. Shapley, "Stochastic games," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 39, no. 10, p. 1095, 1953.

[5] S. Bodmer, M. Kilger, G. Carpenter, and J. Jones, *Reverse Deception: Organized Cyber Threat Counter-Exploitation*. McGraw Hill Professional, 2012.

[6] U.S. Senate, Committee on Commerce, Science and Transportaion, "A 'kill chain' analysis of the 2013 target data breach," Tech. Rep., March 2014.

[7] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning." in *ICML*, vol. 94, 1994, pp. 157–163.

[8] T. Alpcan and T. Basar, "An intrusion detection game with limited observations," in *12th Int. Symp. on Dynamic Ga author=Alpcan, Tansu and Basar, Tamer,mes and Applications, Sophia Antipolis, France*, 2006.

[9] K. C. Nguyen, T. Alpcan, and T. Basar, "Security games with incomplete information," in *Communications, 2009. ICC'09. IEEE International Conference on*. IEEE, 2009, pp. 1–6.

[10] R. Nagel, "Unraveling in guessing games: An experimental study," *The American Economic Review*, pp. 1313–1326, 1995.

[11] J. Bethencourt, J. Franklin, and M. Vernon, "Mapping internet sensors with probe response attacks." in *USENIX Security*, 2005.

[12] T. Alpcan and T. Basar, "A game theoretic approach to decision and analysis in network intrusion detection," in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 3. IEEE, 2003, pp. 2595–2600.

[13] K.-W. Lye and J. M. Wing, "Game strategies in network security," in *Foundations of Computer Security*, 2002, p. 13.

[14] Y. Liu, C. Comaniciu, and H. Man, "A bayesian game approach for intrusion detection in wireless ad hoc networks," in *Proceeding from the 2006 workshop on Game theory for communications and networks*. ACM, 2006, p. 4.

[15] S. A. Zonouz, H. Khurana, W. H. Sanders, and T. M. Yardley, "Rre: A game-theoretic intrusion response and recovery engine," in *Dependable Systems & Networks, 2009. DSN'09. IEEE/IFIP International Conference on*. IEEE, 2009, pp. 439–448.

[16] S. Becker, J. Seibert, D. Zage, C. Nita-Rotaru, and R. State, "Applying game theory to analyze attacks and defenses in virtual coordinate systems," in *Dependable Systems & Networks (DSN), 2011 IEEE/IFIP 41st International Conference on*. IEEE, 2011, pp. 133–144.

[17] V. Bier, S. Oliveros, and L. Samuelson, "Choosing what to protect: Strategic defensive allocation against an unknown attacker," *Journal of Public Economic Theory*, vol. 9, no. 4, pp. 563–587, 2007.

[18] P. Liu, W. Zang, and M. Yu, "Incentive-based modeling and inference of attacker intent, objectives, and strategies," *ACM Transactions on Information and System Security (TISSEC)*, vol. 8, no. 1, pp. 78–118, 2005.

[19] J. White, J. S. Park, C. A. Kamhoua, and K. A. Kwiat, "Social network attack simulation with honeytokens," *Social Network Analysis and Mining*, vol. 4, no. 1, pp. 1–14, 2014.

[20] J. White and et. al., "Game theoretic attack analysis in online social network (osn) services," in *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. ACM, 2013, pp. 1012–1019.

[21] C. A. Kamhoua, K. A. Kwiat, and J. S. Park, "A game theoretic approach for modeling optimal data sharing on online social networks," in *CCE*, 2012, pp. 1–6.

[22] J. S. Park, K. A. Kwiat, C. A. Kamhoua, J. White, and S. Kim, "Trusted online social network (osn) services with optimal data management," *Computers & Security*, vol. 42, pp. 116–136, 2014.

[23] D. Fudenberg and D. Levine, "Learning in games," *European economic review*, vol. 42, no. 3, pp. 631–639, 1998.

[24] R. A. Maxion and T. N. Townsend, "Masquerade detection augmented with error analysis," *Reliability, IEEE Transactions on*, vol. 53, no. 1, pp. 124–147, 2004.

[25] F. Bergadano, D. Gunetti, and C. Picardi, "User authentication through keystroke dynamics," *ACM Transactions on Information and System Security (TISSEC)*, vol. 5, no. 4, pp. 367–397, 2002.

[26] P. Cao, E. C. Badger, Z. T. Kalbarczyk, R. K. Iyer, A. Withers, and A. J. Slagell, "Towards an unified security testbed and security analytics framework," in *Proceedings of the 2015 Symposium and Bootcamp on the Science of Security*. ACM, 2015, p. 24.

[27] P. Cao, E. Badger, Z. Kalbarczyk, R. Iyer, and A. Slagell, "Preemptive intrusion detection: Theoretical framework and real-world measurements," in *Proceedings of the 2015 Symposium and Bootcamp on the Science of Security*. ACM, 2015, p. 5.