A Constraint Partially Observable Semi-Markov Decision Process for the Attack-Defense Relationships in various Critical Infrastructures

Nadia Niknami and Jie Wu
Department of Computer and Information Sciences
Temple University, Philadelphia, PA, USA
{nadia.niknami, jiewu}@temple.edu

***Abstract*—Gaining a better understanding of the relationship between attackers and defenders in cybersecurity domains in order to protect computer systems is of great importance. From the defender's side, it is critical to choose the best reaction to maintain the system in a safe state, based on a given estimate of the attacker. One of the main challenges is that the defender may not be able to correctly detect a current attack due to incomplete and noisy information presented to them. Another important factor in the attack-defense interaction is the limited budget of both attackers and defenders. Therefore, both sides want to perform the best actions to maximize their gains. This paper focuses on an approach based on interactions between the attacker and defender by considering the problem of uncertainty and limitation of resources for the defender, given that the attacker's actions are given in all states of a Markov chain. The best actions by the defender can be characterized by a Markov Decision Process in a case of partially observability and importance of time in the expected reward, which is a Partially Observable Semi-Markov Decision model. Our simulation on a trace-based data set demonstrates that the proposed approach handles analyzing interactions of the attacker and defender with the limited budgets for both sides along with imperfect information for the defender.**

***Index Terms*—Attackers, best actions, defenders, imperfect information, Markov chain, Markov Decision Process (MDP), Partially Observable MDP, utility**

I. Introduction

Analyzing behaviors of the attacker and defender can shed light on improving defense methods to protect a system. In the real world, a major factor for a defender is that they do not have prior nor enough information about the current status and or security level of the system, and they have partial observations. In contrast, most of the time the attacker, has full observations about systems and defending actions. In addition, resources, such as money and time, are not infinite, and agents should take actions efficiently due to this limitation. Finite creates a limitation for taking more action for both the attacker and defender. Time is another important parameter that should be considered in analyzing attack-defense. From the attacker's view, the more time that a system stays in an unsafe state, the more damage for the system and the more benefits for the attacker. Conversely, from the defender's view, detecting and recovering the system to a safe state should be as quick as possible in order to prevent more damage. It should be noted that staying for a long time in a particular state of the system increases the chance of revealing the attacker by the defender, thereby the attacker must restart the attack, resulting in more cost for him. In other words, the defender tries to minimize cost and get the highest reward. It is critical for the defender to recover the system to a safe state as soon as possible. On the other side, the attacker tries to the minimizing cost and makes high damage in order to reach to a crash state of the system. The behaviors of attackers and defenders can be modeled/characterized with the help of Markov chain [1].

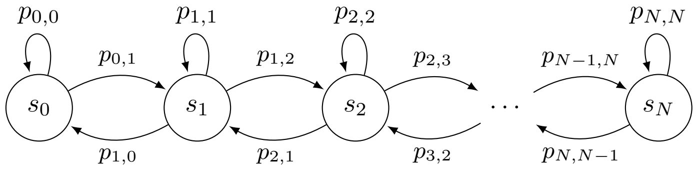


Fig. 1. Markov model for relationship between different state of the system.

The possible states of the given system can be shown with the nodes of a Markov chain. In Fig. 1, s represents the state of the system which can be from s_0 to s_N for a system with N different states. Any change in the state of system is the result of taking a particular action. Directed edges between nodes in a Markov chain represent possible transitions between the states of the system. The probability of a transition between state i and j can be shown with $p_{i,j}$. For example, in Fig. 1, when the system is in the current state s_2, taking a particular action, say a, will transit the system from s_2 to s_3 with the probability of $p_{2,3}$ and will transit the system from s_2 to s_1 with the probability of $p_{2,1}$. It is possible that the system remains in the same state i after a given action. This transition is shown with the self-transition of $p_{i,i}$. Certainly, both sides try to earn rewards as much as possible by reaching their target states. Therefore, estimating the cost that the attacker or defender would have to pay to succeed and the time they would stay in a particular state is important to analyze the attacker's and defender's behavior.

In this paper, we use a simple the attacker and defender interaction example to illustrate different types of Markov models. Given that the attacker acts according to a particular probability distribution, our focus is on how the defender determines their best strategy to counter under various conditions, including limited time and budget along with partial and incomplete view. The quality of a strategy is determined by utility which is based weighted summation of percentages at different states, from a safe state with the highest weight, a compromised state with a reduced weight, to a totally damaged state with zero weight. We start with a Markov Decision Processing (MDP) using a regular Markov chain to represent the transition between states based on the given probability distribution and total budget of the attacker and defender. Then we use a semi-Markov chain to include the time it takes after taking action to transit to the next state. We consider the fact that the defender stays in each state for a different amounts of time and this leads to varying amounts of rewards for him or her. Finally, due to the defender's lack of perfect information, a Partially Observable Markov chain can model the actions of the defender. Therefore, a state-based model that captures the attacker's and the defender's behaviors can be cast as a Partially Observable Semi-Markov Decision Process.

The primary contribution of our work is to model the relationship of the defender with the environment with uncertainty about the underlying states, the past and present status of the environment. The defender has to predict the best action by considering the belief state, rate of detection and available budget. We will analyze this model on the Markov model associated with real data on intrusion detection service. All models are simulated using a trace-based data set on the attacker model. We compare different models in terms of utility under different settings including the budget for attacker and defender, detection rates, partial view, and initial information states.

The remainder of the paper is as follows: Section II describes related works on Partially Observable MDP and Semi-Markov Decision Process. Section III provides background related to Markov chain models and differences between different kinds of MC. Section IV describes different scenarios for attackers and defenders and explains the model that we consider with numeric examples. Section V contains some results on real data in order to evaluate our model and solution. And finally, we conclude the paper in Section VI with some suggestions for the future.

II. Related Works

This section presents an explanation about related works of different Markov models and how they are used in modeling behaviours of attackers and defenders. Authors in [2] and [3] applied Markov chain to model security threats based on graph theory. They used different security metrics and vulnerability scores to compute the probability distributions. Paper [4] presented different methods of intrusion detection and used a hidden Markov chain to detect anomalies. Approaches of [5], [6] are related to Partially Observable Semi-Markov modeling for machine maintenance field. They consider frameworks under both state transitions and observation uncertainties. A different idea based on the Hidden Semi-Markov model that considers state duration and state interval for the sequential data events is proposed in [7]. In their model, there is only some predictions for sequential states rather than decisions about actions.

Many works use Markov chains to analyze behaviors of the attacker and defender [8],[9],[10],[11],[12]. For instance, [10] presents a MDP to model moving target defense policies. The authors considered different Markov models for different defenders' actions and used complete Markov Decision model which is a combination of MDPs associated with actions and varying transition probabilities and costs. Authors in [13] proposed a model for moving target defense and considered the defender's problem as a semi-Markov Decision Process in which the attacker and defender are as follower and leader, respectively. Migration cost, loss of the system, and amount of needed time are used to find optimal defense strategy, and it has a time-average cost objective. In [14], a stackelberg, zero-sum, semi-markov model has been suggested to capture relationships between advanced persistent threats and dynamic information flow tracking while considering false negative and required time for the defender, DIFT. Their model considers the trade off between quickest detection and efficient resource use. All of these papers did not consider the fact that the defender does not have complete information about the underlying system. Also, they consider infinite budgets for the attacker and defender. Authors in [15] consider a defender simultaneously making decisions about the state that the system should be migrated to it and when this migration should happened. In their approach, the target is the minimize the losses which are caused by compromises of systems and the cost of migration. They assume that the defender has prior information about the distribution of the attacker type, and there is no updating for the knowledge of the defender in their approach. Authors in [16] propose an agent-centric approach to address equilibrium in partially observable MDP from a cyber defender's perspective against a fixed attacker. They design a Monte Carlo based sampling for dynamic policy implementation under uncertainty. However, they did not investigate the importance of time and constraint budget for the defender. The approach of the paper in [17] is a state-based model that consideres real time preserving availability for realistic-sized cyber networks. The defender in this model can operate under uncertainty about the attacker's actions and noisy security information. The defender in the given model uses imperfect information to find the optimal strategy for the future.

Our approach considers the fact that in the real world, a defender does not have complete information about the real state of the system and the actions of the attacker. Most of the attackers' actions and the amount of damage they make are partially observable to the defender, and he or she needs to update their knowledge after any new changes in the system if they want to make right decisions about defending. Additionally, this paper models the relationship between the attacker and defender by taking into account the limited budget for taking actions. Most of the approaches related to modeling attack-defense interaction with Markov chain consider an infinite budget for the attacker and defender and try to minimize the cost and use infinite resources, while in reality, there is a finite budget with a limited amount of money and time. Another important factor in our approach is that we consider the time of staying in each state as weight in the reward or cost function for agents. Therefore, our model is a constraint one-sided Partially Observable semi-Markov Decision Process which considers constraint budget for the defender.

III. Markov Chain

In this section, we will give a brief review of different Markov chains. A Markov chain is a random process with a discrete sequence of states in which the future state is only relevant to the present state and the past one is irrelevant. A triple (S, P, Q) can be used to represent the simple Markov chain in which S denotes all possible states in the system, and P denotes the state transition probability matrices and Q denotes the initial probability of states. Transition probabilities matrix presents as:

$$P = \begin{bmatrix} p_{00} & p_{01} & p_{02} & \cdots & p_{0N} \\ p_{10} & p_{11} & p_{12} & \cdots & p_{1N} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ p_{N0} & p_{N1} & p_{N2} & \cdots & p_{NN} \end{bmatrix}, \tag{1}$$

where p_{ij} stands for transition probability from state i to j. Initial probability shows the probability of start from each state. If there is no particular information about the initial status of the system, Q will be equal for all states. In such a MC, computing steady states helps to analyze long-term behaviors of the agent. Consider P as matrix of transition probabilities, steady states $\pi = [\pi_0, \pi_1, ..., \pi_N]$ is:

$$\pi \cdot P = \pi$$

when

$$\sum_{i=0}^{N} \pi_i = 1, \tag{2}$$

where N is total number of possible states.

A. Markov Decision Process

When an agent who can make decisions about taking action based on the reward in each state is with a simple Markov chain, such a decision process is called Markov Decision Process (MDP) [18]. In other words, MC describes the states of the system according to the actions of agents, and MDP can be considered as a stochastic game with a single player. MDP uses the reward to guide planing and choosing the next state based on the associated reward. A MDP is a five-tuple (S, A, P, R, γ) which stands for the set of possible states, the set of all the possible actions, the state transitions matrix, the benefit or reward of state transition by performing the action, and a discount factor, respectively. Discount factor, γ helps to consider more weight for the current obtained rewards in comparison with the future ones. By considering γ equal to 1, there is no difference between the reward that the agent obtains now and in the future.

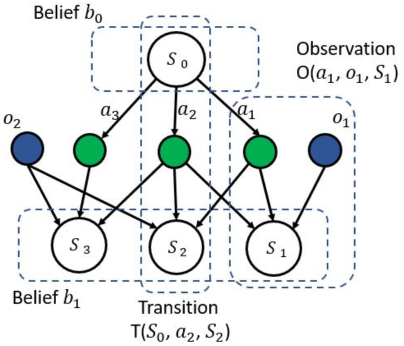


Fig. 2. Updating belief state at any time slot based on the current belief states, Transition probability and Observations probability for particular action

B. Hidden Markov Model

In some systems there is uncertainty about the current state, and information about the true state of the system can be estimated through relevant and possible observations. A Markov chain with hidden states is called Hidden Markov Model (HMM), which can be represented by a five-tuple (S, O, A, B, P_I). S stands for the set of hidden states, O is the set of observation states, A is the state transition probabilities matrix, B shows the observation symbol probability matrix, and the initial state distribution of this Markov model is presented by P_I. The agent in HMM can learn something about its environment and actual state by sensing and observing, and then the sequence of states in the system can be predicted.

C. Partially Observable Markov Processor

The combination of MDP and HMM is called Partially Observable Markov Processor (POMDP) which is represented in the model by using tuples (S, A, T, R, O, γ). New variables in this model in comparison with previous ones are T and O which represent the relation between states and actions and the observation probability in resulting state $s \in S$ because of taking action $a \in A$ respectively. In POMDPs, the steady states is not decidable, and the agent cannot ensure about the current state of the system. The agent uses their beliefs $b(s)$ on the probability distribution over states [19]. The belief state is computed based on the history of actions and observations seen by the agent.

In addition to the incomplete observations in a given system, utility function might be unknown [20]. This means that the cost and reward should be estimated in order to make decisions about taking actions, and there is no exact information or value for them. The choice of the agent as action is based on the belief state and the best value. For all $s \in S$, the summation of belief states is unity, $\sum_{s \in S} b_0(s) = 1$; that is, b_0 is probability distribution over initial states [21]. A belief state summarizes the knowledge of the agent at a given time and shows the likelihood of being in each state. Formally, belief state for S_t based on the past experience, initial belief states, actions $a_t \leftarrow A$, and observations, o_t, is:

$$b(S_t) = Pr(S_t = s | s_0, a_1, o_1, a_2, o_2, ..., a_{t-1}, o_{t-1}). \tag{3}$$

Action always results in a transition to a new belief state, depending on the observation that is received, Fig. 2. When the agent chooses an action, state likelihoods are updated based on new observations and Bayes' rule. Since the current state is s with probability $T(s, a, s')$ which is probability of transition from state s to state s' due to the particular action a and receiving observation o in s' with the probability of $O(a, s', o)$, the new belief state is:

$$b'(s') = \frac{\sum_{s \in S} b(s) \cdot T(s, a, s') \cdot O(a, s', o)}{p_z}, \tag{4}$$

where

$$p_z = O(a, s', o) \sum_{s \in S} b(s) \sum_{s' \in S} T(s, a, s'). \tag{5}$$

In order to have a valid probability distribution over states, belief states must be normalized by dividing by the total probability of observing, p_z.

Thanks to the updating belief states, POMDP will be converted into a fully observable MDP in which agents can make decisions to take proper action. In this version of Markov chain, the expected reward obtained for taking action $a \in A$ can be computed based on belief state and the reward function $R(s, a, s')$ which shows obtaining reward of taking action a in state s and ending up in s':

$$r_a = \sum_{s \in S} \sum_{s' \in S} b(s) \cdot R(s, a, s'). \tag{6}$$

Solving a POMDP in order to find the optimal policy can be done by the frameworks of Value iteration (VI) and Policy Iteration (PI) [22] which are based on iteration over value space and policy space. The iterative calculation is called dynamic programming and the value function, $V(b)$, according to to the belief state b is:

$$\begin{aligned} V(s) = & \max_a \{ b(s) \cdot R(s, a) \\ & + \gamma \sum_{s' \in S} T(s, a, s') \cdot O(a, s', o) \cdot V(s') \}, \end{aligned} \tag{7}$$

where the first term, $R(s, a)$, is immediate reward, and the second term, $\sum_{s' \in S} T(s, a, s') \cdot V(s')$ is the expected reward of future with parameter γ that is a discount factor. After enough iterations, the value function converges to a stationary policy, $V_{t+1}(s) \approx V_t(s)$, and does not change much any more. This means that with a small number of ϵ if $|V_{t+1}(s) - V_t(s)| \leq \epsilon$, no need to find better value. Although POMDP is very close to a real scenario, there is some limitation in this model. First of all, it is assumed that the agent knows the complete POMDP model, such as transition probabilities, observation probabilities, and rewards, which is not realistic for many problems. Secondly, it is required time to compute belief state if the resource and information for updating belief states are not available [23].

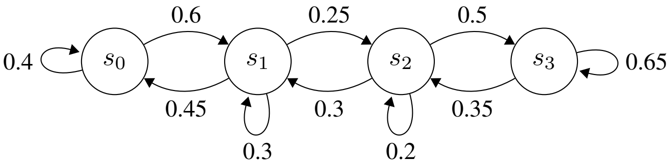


Fig. 3. Simple Markov chain with transition probabilities

D. Semi-markov Decision Process

In the types of Markov chains mentioned above, every transition is based on time units; while actions need some time to execute and states can be defined for any given time. To shed light on this statement, consider an agent takes action a in state s_i and remains in this state for time τ_n, then transits to the next state s_j and receives the reward r. In other words, taking action a in s_i and transitioning to the next state takes an amount of time which is presented by τ_i. Therefore, the transition from one state to another one depends not only on the current state and action, but also on the time the action has been taken. This kind of MDP is called Semi-Markov Decision Process which is considered time for transitioning to the following state. Formally:

$$p_{ij} = P\{X(\tau_n + 1) = j | X(\tau_n) = i\}. \tag{8}$$

It should be noted that Semi-Markov Decision Process does not consider the uncertainty about the current state of the system. Therefore, combination of Semi-Markov with POMDP would be useful in modeling scenarios in the real world.

IV. Modelling relationship

In this section, we discuss modeling the defender's and the attacker's relationships. Most of the attacks have a sequential set of attack steps and can be decomposed to some sub-stages. When all sub-stage pass successfully, the attack succeeds. As an application, consider installing a malware on a host for DDoS attack [24][25]. Generally a sophisticated DDoS attack can be decomposed with five steps as a multi-step attack scenario. These phases are:

1) Sending echo-request for live hosts.
2) Finding live IP's to look for the sadmind daemon on hosts.
3) Breakins by the sadmind vulnerability on hosts.
4) Installing trojan mstream DDoS software on hosts.
5) Launching the DDoS.

For simplicity, we can consider four phases: "Intrusion Attempt", "Compromised System", "Planting Virus", and "Denial of Service" as the states of the system under a sequential DDoS attack. Fig. 3 shows a Markov model for such an attack where each state shows one of the steps of DDoS. First state, s_0, shows "Intrusion Attempt" when there is not any attack over the system, and the system is in a safe status. If the attacker sends broadcasting ping to the system and finds live hosts, the system will transit to the "Compromised System" which state s_1 in Fig. 3. There is a 60% chance that this transition will happen . On the other hand, there is a 40% chance that the system will stay in s_0 and the attacker will not be successful in finding live hosts. Executing a daemon in background by the attacker is the next step after finding hosts.

With a 25% probability the attacker can perform this step and system transits to "Critical State" or state s_2. Ultimately, by installing malware or trojans with the help of daemon and launching DDoS attack, total damage is done, and system will transit to state s_3 which is status of "Totally Damaged" for given system. It is worth to mentioning that in each state, the defender can take actions in order to defend and recover the system to a safer state.

A. Simple Markov Model

The simplest scenario for Markov model in Fig. 3 is when there is an infinite budget for the attacker and defender. The attacker tries to make more damage and transits system to a more unsafe state, while the defender tries to recover the system to a safer state. Note that there are different probabilities for transitions and self-transition (an option for agents to wait in a given state which shows staying in state without any actions). For example, suppose that system is in state s_1 and the attacker does attack in order to transit system to the next state. According to the Markov model in Fig. 3, there is a chance 25% the attacker is successful and transits system to s_2, and there is a 45% chance, defending action recovers system to a safer state s_0. There is a 30% chance, the attacker cannot further damage the system, and the system remains in s_1. After a long-round, the steady states, π, for the Markov Model of Fig. 3 can be computed:

$$\pi \cdot \begin{bmatrix} 0.4 & 0.6 & 0 & 0 \\ 0.45 & 0.3 & 0.25 & 0 \\ 0 & 0.3 & 0.2 & 0.5 \\ 0 & 0 & 0.35 & 0.65 \end{bmatrix} = \pi$$

and

$$\pi = \begin{bmatrix} 0.20 & 0.26 & 0.22 & 0.32 \end{bmatrix}$$

This means that after a long period of time, system will be in state s_0 with a 20% chance, in state s_1 with the 26% probability, and so on. Staying in each state has a reward for agent and formally long-term total reward, R, can be computed with steady states with:

$$R = \sum_{i=0}^{N} \pi_i \cdot r(s_i), \tag{9}$$

where $r(s_i)$ is the reward function that represents obtained reward of being in state s_i. Suppose that the reward function for the Markov model of Fig. 3 is $r(s_i) = 10 * i$, then the expected total reward after a long time will be 334.6.

B. Markov Chain with Budget Constraint

Taking action has cost, and certainly, there is not an infinite budget for agents. In case of limited budget, the relationships between the attacker and defender will not continue forever and this process will be finished the case that they run out of their budget. For example, consider the attacker and defender have only \$120 to spend for their actions. Fig. 4 shows a Markov model with different costs for different transitions. In this model, although two-hop transitions costs more for agents, they convert system to a safer state in the defender's view or a state with more damage in the attacker's view. If after some relationships between the attacker and defender, defender runs out their budget, he or she cannot do anything in order to defend the system and the attacker will continue to make more damage.

Algorithm 1 POMDP

```
Input: Initial Belief $b$, Observation Function $O$, State Transition $T$
Output: Expected reward and Updated belief state.
  $\pi \leftarrow$ Value Iteration ()
  a $\leftarrow \pi$(s)
  for each $s' \in S$ do
      $b(s') \leftarrow O(o, a, s') \cdot \sum_s T(s, a, s') \cdot b(s)$
```

It is possible that the attacker and defender do not have same budget. Analysing the behaviors of the attacker and defender in such a different scenario that defender has more budget in comparison with the attacker or the attacker has more budget would be interesting. Consider a scenario for Fig. 4 in which the attacker has \$150 and the defender has \$100 as budget and the attacker starts from s_0. The attacker does $s_0 \xrightarrow{\text{attack}} s_1 \xrightarrow{\text{attack}} s_2 \xrightarrow{\text{attack}} s_3$, thereby spends \$90 for his or her actions. In this state, the defender detects this attack and recovers system to s_1 which is safer by spending \$75 and then recover to s_0, ($s_3 \xrightarrow{\text{defend}} s_1 \xrightarrow{\text{defend}} s_0$), thereby running out his or her total budget (\$75 + \$25). In such a situation, the defender cannot continue defending because he or she does not have any budget to spend for making actions, so the attacker, with the help of the rest of his or her budget, \$60, can continue attacking and making more damages without any interference from the defender. Therefore, the attacker transits system from s_0 to state s_2 which has more security damage for system and stays there. The amount of budget for the attacker and defender and the sequence of relationships have a vital impact on the final results of the scenario.

C. Partially Observable Markov Decision Model

In the real world, the defender does not have complete information about exact state of the attacker or the impact of actions on system. Therefore, the defender has to make decisions to prevent attackers with incomplete information. On the other side, most of the time attackers have a full observation, and their information about states of systems and type of defenders is complete. Such a scenario can be modeled by a MDP with partially observable states, or POMDP which is more close to reality compared to MDP. In POMDP, there is an initial belief states about the probability of being in states at the first time slot. After any action, the defender has some new observations that helps him or her to update information about the current state of system, but not exactly. Actually, based on these new observations, belief states can be updated to show the probability of being in states after taking action a. As mentioned before, the defender does not have full observation, so updating belief states based on observations helps him or

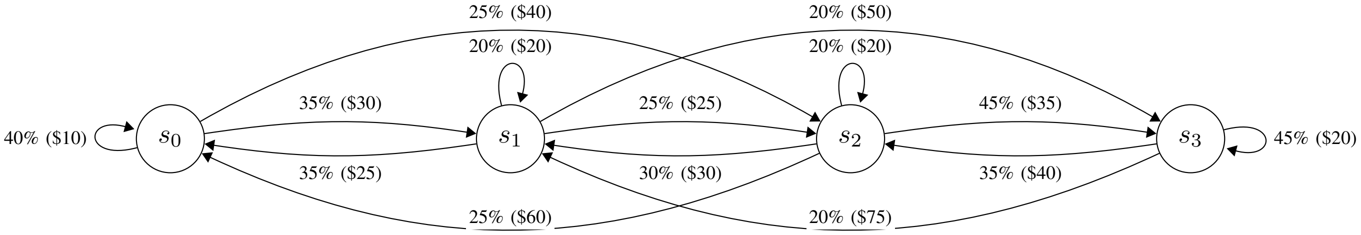


Fig. 4. Attack-defense Markov model with considering cost for actions.

her to estimate the current state of system and to make the right decision about required actions. Certainly, with more knowledge, there would be a better estimation about states for the defender.

Suppose that based on the prior knowledge of the defender, the initial belief states in the defense-attack model of Fig. 4 is $[50\%, 30\%, 20\%, 0\%]$. This means that the state of system at the start, the first time slot, is state s_0 with the chance of 50%, state s_1 with the chance of 30%, state s_2 with 20%, and there is no chance to be in state s_3. Also, the defender has knowledge about the chance of detecting the attack in different status of the system. For example, 90% for detection rate means that agent after taking action can detect the following state with 90% probability. According to the equation 4, new belief state for each state is summation of all possible transitions from other states to this state due to taking particular action $a \leftarrow A$. Consider three actions $A = \{NoAction, Defend, Reset\}$ for the defender in Fig. 4. Suppose the action is $a = Defend$, the observation probability O in resulted probable states is:

$$O(s, a = Defend) = \begin{bmatrix} 0.9 & 0.1 \\ 0.85 & 0.15 \\ 0.6 & 0.4 \\ 0.4 & 0.6 \end{bmatrix}$$

Therefore if the defender takes action $Defend$ and has initial belief $b(s) = [50\%, 30\%, 20\%, 0\%]$, the new belief for state s_0 according to the equation 4 will be $0.9[(0.3 \times 0.35) + (0.20 \times 0.25)] = 0.14$. As mentioned before, new belief states should be computed for all states, then normalization for obtained values is needed to bring probabilities to the same range. Ultimately, the normalized new belief state after the first action, $Defend$, for Fig. 4 will be $[69\%, 25\%, 8\%, 0\%]$. As mentioned before, with the help of value iteration, the optimal action based on the maximum reward is obtained (Algorithm 2 shows associated pseudo code). Value iteration function in any MDP helps agents to select optimal decision based on the immediate reward and an estimation about the expected reward which will be obtained in the future .

Algorithm 2 Value Iteration

```
Initialize $V_0(s) \leftarrow 0$, $\Delta \leftarrow 0, i \leftarrow 0, \epsilon \leftarrow$ Small number
repeat
    for $\forall s \in S$ do
        $V_{i+1}(s) \leftarrow \max_{a \in A}\{P(s,a,s') \cdot b(s) \cdot [R(s,a,s') + \gamma \sum_{s' \in S} V_i(s')]\}$
        $\Delta \leftarrow \max(\Delta, |V_{i+1}(s) - V_i(s)|)$
    $i \leftarrow i + 1$
until $\Delta < \epsilon$
Output: $\pi(s) \leftarrow \arg\max_{a \in A}(V(s))$
```

D. Partially Observable Semi-Markov Decision Model

In a real scenario, in addition to considering partial view for the defender, it is of great importance to find how long the agent stays in a given state. In other words, time of being in each state is how much it takes for transitioning from one state to another after taking particular action. In such a case when the defender has partial view and the time it takes to take action is important, Partially Observable Semi-Markov Decision Process or POSMDP can be used for modeling and analysing relationships between the attacker and defender, Fig. 5. Needs to find the probability of staying in a particular state i for period time of m after taking action a. Taking action a results in the state j [6]. Conditional time distribution $F_{ij}(m, a_i)$ shows average time to stay in each state i and the following state j for every action a. Formally, F is defined as:

$$F_{ij}(m, a_{t_i}) = p(m | X_{t_i} = i, X_{t_j} = j, a_{t_i}), \tag{10}$$

where $m = |t_i - t_j|$ and is the length of the time between when system is in state i at t_i and in state j at t_j. By choosing action a in state i at time t_i after m amount of time, the system will be in state j with the probability of:

$$Q_{ij}(m, a_{t_i}) = P_{ij}(a_i) \cdot F_{ij}(m, a_{t_i}), \tag{11}$$

where P_{ij} is the transition probability of transition from s_i to s_j with taking action a. In POSMDP, in order to find the optimal decision to get the most reward, we need to consider $Q_{ij}(m, a_{t_i})$ in updating belief regarding the probability of being in a particular condition at the specific time.

The quality of a strategy in a partially observable semi-Markov model is determined by utility which is based on the weighted summation of percentages at different states, from a safe state with the highest weight, a compromised state with a reduced weight, to a totally damaged state with zero weight. The objective function is to maximize sum of immediate reward and future expected reward associated with

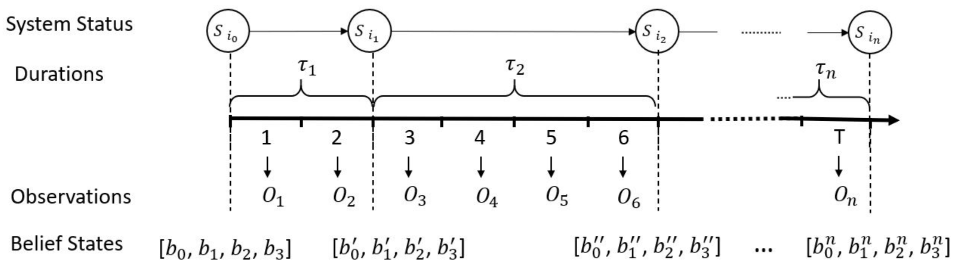


Fig. 5. Updating belief state at the end of each period. τ_i shows the amount of time that system stays in a particular state.

given decision. The utility function for state s and action a if the system stays in s for time with length of m is:

$$Utility(s,a) = R(s,a) + F(t,a) \cdot r(s,t) - C(s,a), \quad (12)$$

where $R(s,a)$ is the obtained immediate reward in state s due to taking action a, $r(i,a,t)$ is the reward per unit time in transition period, and $C(s,a)$ is the cost associated with taking action a in state s. Certainly staying longer time in states which are safer will have more rewards for the defender, and time can be used as a weight in finding total reward of the defender. As an example, suppose that immediate reward, $R(s,a)$, for taking action $Defend$ in state s_1 in Markov model of Fig. 4 is 50 and reward per time, $r(s,a,t)$, is $20t$. If defender takes action $Defend$ in s_1 and can keep system in this state for 10 minutes, he or she will find $50+20\times10 = 250$ as reward. In order to find the best action in each state, the defender needs to have some estimations about the future of taking action. Value function for POSMDP is a modified version of VI which includes the immediate reward of the current time, the expected transition time and previous time period:

$$\begin{aligned} V(s) = \max_a \{ & b(s) \cdot (R(s,a) + F(t,a) \cdot r(s,a,t) - C(s,a)) \\ & + \gamma \cdot \sum_{s' \in S} Q(m,a) \cdot O(s,a,s') \cdot V(s') \}. \end{aligned} \quad (13)$$

In a nutshell, with a semi partial observable MDP, both of the time of being in each state and partial view of the defender has an effect on the value of utility. Partial view plays an important role in updating belief of the defender, and the impact of time in each state as a weight in the reward function is considerable.

V. Simulations Results

In this section, we present the results of analysing optimal policy and impact of different parameters on utility. We try to show what is the impact of different amounts of total budgets for the attacker and defender, varying initial belief states, different detection rates (partially observation), cost of $Reset$ action and the probability of defense and attack on total utility for the defender.

A. Dataset and Settings

For simulation, we used Markov model of data that was drawn from MIT 2000 DARPA [26]. This dataset is related to an off-line intrusion detection dataset, LLDOS 1.0, and includes different scenarios of a distributed denial-of-service attack based on the stealthy level of attack and slightly modified as demand. The target of the attacker in these scenarios is to install components and then carry out a DDOS attack. Each step of this attack can be considered as one of the states of the system. Any action that transits the system to a safer state can be considered as a defend or reset action. Thus, this implementation is a mimic of a real application. P. Holgado et al. [25] designed a HMM with the help of clustering tag of alerts which detect intrusive activity as the observations for each step of attacking in order to train the Markov chain with both unsupervised and supervised methods. He found the transition probability matrix, the observation probability matrix, and the probability of starting intrusion from each state. Clustering is done based on the tags of denial-of-service, buffer overflow, cross-site scripting, remote attackers, remote users, and information for alerts. This clustering helps to predict the transition probability of attacks over states which show the status of the system. According to the HMM, the predicted transition probability for S =\{Intrusion Attempt, Compromised system, Planting Virus, Denial of Service\} are:

$$P = \begin{bmatrix} 0.48 & 0.52 & 0 & 0 \\ 0.33 & 0.33 & 0.34 & 0 \\ 0.24 & 0.24 & 0.25 & 0.27 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and initial probability distribution over states is:

$$\pi = \begin{bmatrix} 0.348 & 0.238 & 0.414 & 0 \end{bmatrix}$$

Transition between states in this model is based on the probability of defense and probability of progressing attack in each state. Authors in [25] considered a naive defender that only has two simple actions $Silent$ and $Alert$. In this paper, we consider a defender with more reaction related to the status of the system whose actions are $A = \{NoAction, Defend, Reset\}$. Another trace we consider to analyze the results in Markov model is moving target defense [10] which has different transitions between states. It is important to mention again that the following state of each action is based on some probabilities, transition probability. For example, if system is in the state of Intrusion Attempt and takes action $NoAction$, with the chance of P_a, probability of finding live hosts, system will transit to state of Compromised System and with the chance of $1-P_a$, system will stay in state of Intrusion Attempt, as a self transition. There is a similar situation for other actions, $Defend$ and $Reset$.

We assume that there is a cost for each transition between states that comes from the cost of taking action for the agent. This cost depends on the type of action. For example, there is a cost imposed on the defender when he takes defense or does a reset action. The final expected reward of the defend action is the baseline reward subtracted by the cost associated with the defend action. We consider different costs and rewards associated with different actions in states of system, and the cost of $Defend$ and $Reset$ actions should be added to the cost of attacking. Fig. 4 shows the cost of each transition between states for a particular action. The unit of the cost and budget is dollars. The cost of reset is set to be larger than the cost of defend. That is because for the reset action, the defender needs to pay a higher cost for the system come back to the first state. The reset action is taken only when there are no other defensive solutions.

There is an initial total budget for the defender and attacker. To take any action, an agent should spend the cost of a given action from his budget. The initial total budget of the defender and attacker can be the same or different. In simulation, there is an analysis of scenarios when the attacker has a higher total budget than the defender and vice versa. In addition, we assume there is a reward R that is received due to defense action and a general reward due to making decisions against attackers. The values of future discount factor $\gamma = 0.9$ and $\epsilon = 0.001$ are fixed. Generally, we considered initial belief state $[25\%, 25\%, 25\%, 25\%]$ and observation probability 85% rather than scenarios which investigate impacts of changing initial belief states and observation probability. We analyze two scenarios for the POMDP model. In the first model, the agent selects the action according to his belief about the current state of system and tries to select the best action. In the second model, the agent selects an action randomly when he has a partial view.

The time of being in each state plays an important role in the value of the reward. Therefore, time of being in each state can be considered as a weight in computing total utility. To this end, the more time the system is in a safer state, the more rewards the defender will have. On the other hand, being more unsafe in compromised states have more cost and then less reward for him or her. In order to find the optimal policy at any particular state, the maximum of value function for three actions is computed repeatedly according to the associated transition probability and cost of different actions. Finally, at the convergence, proper action in each state will be selected as the optimal policy. It should be noted that after taking any action, the defender has to update his belief, and in the next value iteration he will use this new belief in order to find the best policy. Therefore, after any action, because of new knowledge about the current state of system, the defender will have new optimal policy. Analysing the impact of costs, initial belief states, detection rates of the defender and the amount of total budget in finding optimal policy and utility will be illustrated in the following parts.

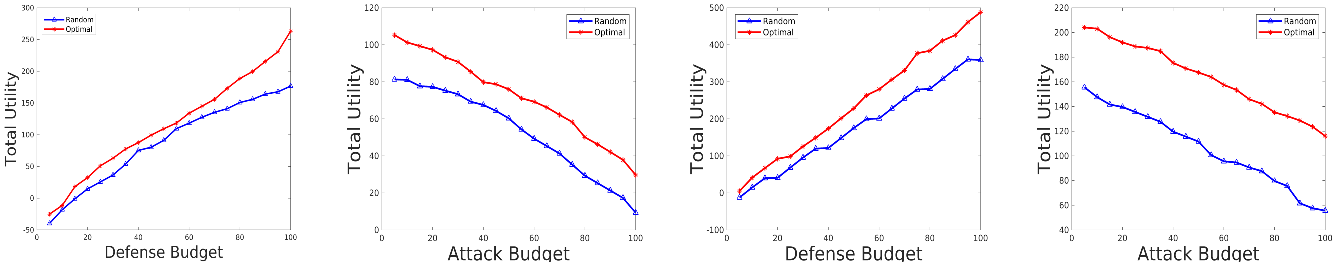
B. Impact of Attack and Defense Budgets on Utility

In this part, we investigate the impact of increasing the attacker's budget on total utility. If either the attacker or defender uses all of his or her budget, then the opponent agent can continue their actions and obtain more rewards due to no interference from the. Fig. 6 shows impacts of increasing total budget of one side when opponent side's budget is fixed on utility. Figs. 6(a) and (c) illustrate that increasing the defender's total budget increases the utility. This is because of the attacker does not have any budget to progress his attack to target state, while the defender has enough budget to recover the system to the safest state without any concern about damage or cost from the attacker. On the other hand, increasing the attacker's total budget has more cost for the defender as Figs. 6(b) and (d) represent. This is because after running out of their budget, the defender cannot recover the system to a safe state, while the attacker transits system to the states which have more damage for the defender. By doing so, the more the attacker's budget, the less utility for the defender.

In addition, in Fig. 6, we can see the difference between two model: a POMDP model in which the agent selects the best action according to the current state of system and a model that the agent selects action randomly when he or she has a partial view. Although the trend of utility due to changing the budget of the defender and attacker is approximately the same for these two models, selecting the best action based on the estimated value of each action and new belief has more utility in all situations. In addition, Figs. 7 and 8 show error bar diagram for the impacts of changing total budget on total utility for two models.

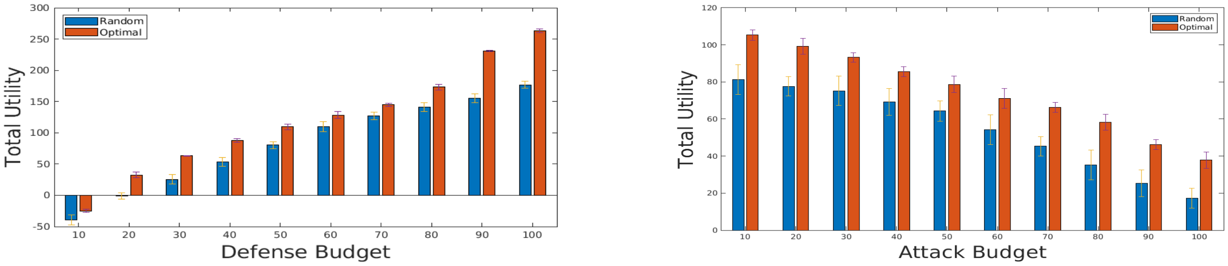
C. Impacts of Detection Rate

As mentioned before, in a POMDP, the agent does not have complete information about the current status of the system and needs to update his belief to figure out with which probability can the system be in each state. After any action, the agent will have some new observation about the underlying environment, and this observation will be used in updating their belief. In this part we want to consider impacts of different observations on total utility. For a defender, observation can be the power of detecting the status of the system, so we can call it detection rate. Fig. 9 shows the impact of different detection rates in state Compromised System on utility. As depicted in figures, increasing detection rate or partially view of the defender increases total utility value. This is because with larger detection rates, defenders find current critical states of systems sooner. By doing so, there is less damage for the system and less cost for the defender. In other words, as a result of a better detection rate, the agent will have better and more precise belief states.



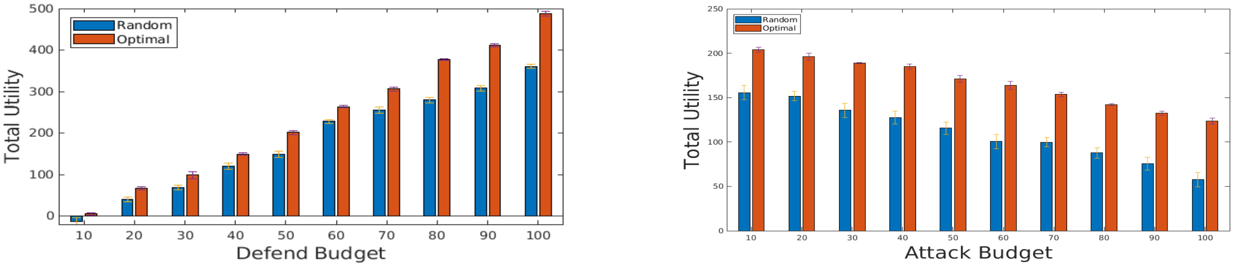
(a) (b) (c) (d)

Fig. 6. Impact of changing total budget of one side while keeping budget of other side on total utility. (a) and (b) show result when agent selects an action based on his belief of current state. (c) and (d) show results when agent selects an action randomly.



(a) (b)

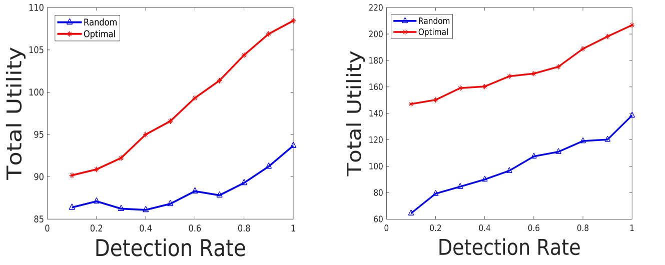
Fig. 7. Impact of changing budget on total utility when agent selects an action based on his belief of current state



(a) (b)

Fig. 8. Impact of changing budget on total utility when agent selects an action randomly.

There is a similar trend in the scenario with random action, but selecting an action totally randomly has less utility compared to selecting the best action. In the random scenario, the agent selects the next action randomly without considering probable effects of this action in the future. Conversely, in the best action scenario the agent uses the value function to select the next action according to the current belief and also estimation of obtained value in following. Detection rate or observation probability has a direct effect on estimation value of future. Therefore, it plays an important role in value function and then in selecting the best action.



(a) Agent selects an action based on the current state.

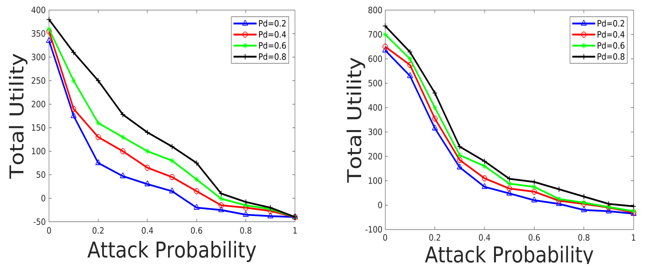
(b) Agent selects an action randomly.

Fig. 9. Impacts of changing detection rate on total utility in the Compromised state of system.

D. Impact of Probability of Attack and Defense on Utility

In this section we investigate the impact of attack and defense probabilities on utility. Probability of attack presents frequent of attacking and larger probability shows more aggressive attack and therefore more cost. In the Markov model for IDS, probability of attack is the probability of transit from the first state, intrusion attempt, to the next state which is the state of compromised system. Larger chance for this transition shows larger frequency of attack. Defense probability represents the probability that the defender selects $Defense$ action rather than $NoAction$ or $Reset$. A higher defense probability shows a more effective response of the defender and notices the success rate of defending against the attacker.

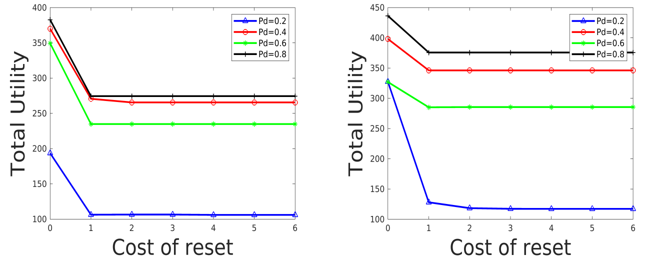
Fig. 10 shows the impact of increasing attack and defense probabilities on utility. According to the diagram, increasing probability of attack decreases total utility because higher probability attack, P_a, leads to transit system to the states with more damage and cost for the defender. On the other hand, the



(a) Agent selects an action based on the current state.

(b) Agent selects an action randomly.

Fig. 10. Impact of changing attack and defense probabilities



(a) Agent selects an action based on the current state.

(b) Agent selects an action randomly.

Fig. 11. Impact of changing detection rate on total utility in state Compromised System.

higher defense probability, P_d, the better utility due to help system to transit to the safer states. As a result, analysing these probabilities and their impacts on total utility would be helpful in analysing the attacker's and defender's strategies.

E. Impact of Reset Cost on Policy and Utility

This part illustrates the relationship between cost of $Reset$ action on optimal policy and total utility. Fig. 11 shows the effect of reset cost for different defense probabilities on total utility for two models.It is noticeable that in some points, in spite of increasing cost of reset, there is no change in the value of utility. This because when the cost of $Reset$ action is more than the cost of taking $Defend$ or $NoAction$ actions agent will not select this action as the best one and other two actions based on the current state and other parameters will be selected. Therefore, there is a threshold point for cost where after this point, increasing cost of $Reset$ will not have any impact on the total utility.

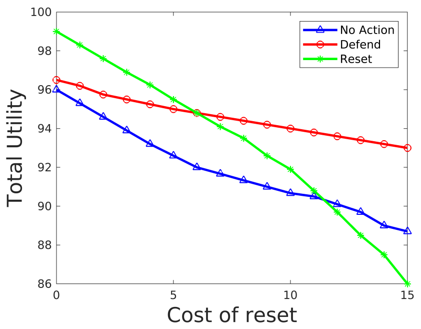


Fig. 12. Impact of reset cost on utility of different actions in particular state S_3.

TABLE I
UTILITY AND NUMBER OF ITERATION FOR VARYING INITIAL BELIEF STATES

Belief State	Utility	Iterations
[1,0,0,0]	73.9021	32
[0.25,0.25,0.25,0.25]	73.8914	29
[0.5,0.5,0,0]	73.931	32
[0.2,0,0.8,0]	73.9011	29
[0,1,0,0]	73.9142	30
[0,0.5,0.5,0]	73.8942	29
[0.1, 0.2, 0.4, 0.3]	73.9021	29
[0, 0, 1, 0]	73.9101	30
[0, 0, 0, 1]	73.9573	31
[0, 0, 0.5, 0.5]	73.9226	30
[0, 0, 0.5, 0.5]	73.9131	30

Let us consider the impact of cost of $Reset$ action in more detail. The observation in Fig. 12 shows the impact of changing cost of $Reset$ action on utility and selecting the best action in at a particular state of Denial of Service. It implies that before point 1, the defender prefers to select reset action because this action has less cost and therefore more rewards for him. In the following, with increasing cost of $Reset$ action, the total utility decreases for this given action and the best action is $Defend$ due to more utility in comparison with $Reset$ and $NoAction$. After point 2 in Fig. 12, there is less utility associated with taking action $Reset$ than utility which is obtained with action $NoAction$. However, $Defend$ has the most utility and is selected as the best action. It should be noted that in some states, such as Denial of Service state in DDoS, there is no defend action for agent and he has to select between $NoAction$ and $Reset$. In such a case, although generally the cost of $Reset$ is higher than $NoAction$, the agent selects $Reset$ action because staying in this particular state has more damage and cost than cost of $Reset$. But with high cost of $Reset$ action, the agent will select $NoAction$ instead of $Reset$.

F. Impact of Belief States on Policy

In this part, we aim to investigate the impact of initial belief states about an environment on utility and select optimal policy in scenarios with different uncertainty about the current state of system. Initial knowledge about history of the underlying system helps a defender with partially observation to predict the current state of system and select the best action associated with the current status. Certainly, how much the agent knows about the current position of system has an effect on his or her decision making. Different assumptions have different cascading effects and lead to different following a sequence of actions. In our models, because of the fact that there is close reward functions for different states, changing initial belief does not have considerable impact on total utility.

Table I shows total utility and number of iterations to converge for different initial belief states. If there is considerable reward for a particular state in comparison with other states, initial information states or belief states will have noticeable

TABLE II
OPTIMAL DECISION FOR STATE DENIAL OF SERVICE WHEN
$b(s) = [20\%, 80\%, 0\%, 0\%]$

Time	Optimal Decision	Belief State
1	No Action	[0.2, 0.8, 0, 0]
2	Defend	[0.1,0.7,0.2,0.1]
3	Defend	[0.1,0.6,0.2,0.2]
4	Defend	[0,0.4,0.2,0.3]
5	Defend	[0,0.3,0.2,0.5]
6	No Action	[1,0,0,0]
7	Defend	[0.7,0.2,0.1,0]
8	Defend	[0.4,0.4,0.1,0]
9	Reset	[0.1,0.3,0.2,0.4]
10	No Action	[1,0,0,0]

TABLE III
OPTIMAL DECISION FOR STATE DENIAL OF SERVICE WHEN
$b(s) = [25\%, 25\%, 25\%, 25\%]$

Time	Optimal Decision	Belief State
1	Defend	[0.25, 0.25, 0.25, 0.25]
2	Reset	[0.1,0.2,0.2,0.5]
3	No Action	[1,0,0,0]
4	No Action	[0.7,0.2,0.1,0]
5	Defend	[0.4,0.4,0.1,0.1]
6	Defend	[0.2,0.4,0.2,0.2]
7	Reset	[0.1,0.3,0.2,0.4]
8	No Action	[1,0,0,0]
9	Defend	[0.7,0.2,0.1,0]
10	No Action	[0.4,0.4,0.1,0.1]

impact on total utility. As mentioned before, agent uses beliefs to have approximate view from the current status of the system and then makes decision about the best action. The initial belief states with low probability for being in a state with more rewards guides agent towards states with less reward. As a result the total utility will be less than the scenario with an initial belief which assigns more probability for being in a given state. Hence, the impact of initial utility on total utility and speed of convergence in value iteration is based on the reward function associated with the given model.

In addition, Table II shows details of updating belief states and optimal action in each step for initial belief states $b_0(s) = [20\%, 80\%, 0\%, 0\%]$. The best action in any given time is selected based on the best action associated with the current state of system. More probability for safer state leads to select $NoAction$ because this action has less cost for the defender. Conversely, if the probability of being in states which have more cost and damage for the defender are larger, then the defender will select $Defend$ or $Reset$ based on the current situation. $NoAction$ in such states has more cost for the defender.

In a case that the defender has high uncertainty about the status of underlying system and does not have enough prior information to distinguish between states in order to find what is the real current of system, initial belief state will be $b_0(s) = [25\%, 25\%, 25\%, 25\%]$. Table III shows optimal decision and details of updating belief for a defender with high uncertainty. A defender with high certainty has initial belief $b_0(s) = [100\%, 0\%, 0\%, 0\%]$. He or she knows that system in the first state at first without any doubt. Therefore, such a defender can select the best action according to the maximum value function, Table IV shows the best action in each time based on the new belief states with a high certainty belief.

TABLE IV
OPTIMAL DECISION FOR STATE DENIAL OF SERVICE WHEN
$b(s) = [100\%, 0\%, 0\%, 0\%]$

Time	Optimal Decision	Belief State
1	No Action	[1, 0 ,0, 0]
2	No Action	[0.7,0.2,0.1,0]
3	Defend	[0.4,0.4,0.1,0.1]
4	Defend	[0.2,0.4,0.2,0.2]
5	Reset	[0.1,0.3,0.2,0.4
6	No Action	[1,0,0,0,]
7	Defend	[0.7,0.2,0.1,0]
8	No Action	[0.4,0.4,0.1,0.1]
9	Defend	[0.2,0.4,0.2,0.2]
10	Reset	[0.1,0.3,0.2,0.4]

Finally, from above simulation results we can conclude that total utility of the defender and the action selected as the best one for defending in each state are under effect of different parameters. It is also observed that a partially observable semi-Markov decision process with budget constraint is a proper model to analyze the behaviors of the defender in order to improve defending strategies. Because it considers partial view and importance of time for the defender, limitation of budget for both the attacker and defender exist in the real scenarios in security domain.

VI. CONCLUSION

In cybersecurity, one of the major obstacles to achieve effective defense is the fact that an attacker knows more about the defender than the defender knows about the attacker. In this paper, we proposed to use finite-state, finite-action, and stationary Partially Observable Semi-Markov Decision Process to model the relationship between a defender and an attacker. We investigated different scenarios for behaviors of the attacker and defender and analysed optimal decision and utility in respect of cost, total budget, initial belief and rate of observation. Also, we considered the impact of staying time associated weight based on the security level of each state. When there is a lack of information on one side, say the attacker side, the attacker does not have enough information to select the best action against the action of the defender. He needs to learn more and try to make his belief of the defender more accurate. Certainly, it will take more time in comparison with the complete information. This approach can be used when both sides have partial information. The only difference is the duration of learning time. In this case, both the attacker and defender need to update their beliefs in multiple rounds, based on what they receive from the actions of other side in previous rounds.

In the future, we would like to work on different behaviors for the attacker and defender. For example, a “Patient” defender pays more attention to resource efficient and waits until detecting a critical situation to consume his or her budget, while an “Impatient” defender considers short term reward and penalty, rather than resource efficiency and tries to detect attacks quickly and make reactions, in spite of finishing resources. Also, a "Smart" attacker prefers to make small damage in order to stay undetected and can obtain more rewards because of transition to more unsafe states. Also it is possible that there are different type of attacks some of which may spend more budget compared with others. Analysing different scenarios with varying behaviors for the attacker and defender would be interesting and useful. Another issue as a future work is taking into account the reward function as the feedback for agent in order to reduce the uncertainty about the state of the system. In general POMDP, agent only uses prior belief states and new observations after taking action to have a better view from underlying system. The amount of the reward that the agent receives as a result of taking particular action can help him have more certainty about state of system.

ACKNOWLEDGMENTS

This research was supported in part by NSF grants CNS 1824440, CNS 1828363, CNS 1757533, CNS 1629746, CNS 1651947, and CNS 1564128.

REFERENCES

[1] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, *Handbook of markov chain monte carlo*. CRC press, 2011.

[2] N. T. Le and D. B. Hoang, "Security threat probability computation using markov chain and common vulnerability scoring system," in *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*. IEEE, 2018, pp. 1–6.

[3] S. Abraham and S. Nair, "Cyber security analytics: a stochastic model for security quantification using absorbing markov chains," *Journal of Communications*, vol. 9, no. 12, pp. 899–907, 2014.

[4] H. Sukhwani, V. Sharma, and S. Sharma, "A survey of anomaly detection techniques and hidden markov model," *International Journal of Computer Applications*, vol. 93, no. 18, 2014.

[5] M. Zhang and M. Revie, "Continuous-observation partially observable semi-markov decision processes for machine maintenance," *IEEE Transactions on Reliability*, vol. 66, no. 1, pp. 202–218, 2016.

[6] R. Srinivasan and A. K. Parlikad, "Semi-markov decision process with partial information for maintenance decisions," *IEEE Transactions on Reliability*, vol. 63, no. 4, pp. 891–898, 2014.

[7] H. Narimatsu and H. Kasai, "Duration and interval hidden markov model for sequential data analysis," in *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2015, pp. 1–8.

[8] O. P. Kreidl, "Analysis of a markov decision process model for intrusion tolerance," in *2010 International Conference on Dependable Systems and Networks Workshops (DSN-W)*. IEEE, 2010, pp. 156–161.

[9] O. Hayatle, H. Otrok, and A. Youssef, "A markov decision process model for high interaction honeypots," *Information Security Journal: A Global Perspective*, vol. 22, no. 4, pp. 159–170, 2013.

[10] J. Zheng and A. S. Namin, "Markov decision process to enforce moving target defence policies," *arXiv preprint arXiv:1905.09222*, 2019.

[11] Q. Liu, L. Xing, C. Zhou, and Y. Wang, "Probabilistic security risk assessment of systems subject to sequential attacks," in *2018 12th International Conference on Reliability, Maintainability, and Safety (ICRMS)*. IEEE, 2018, pp. 161–166.

[12] K. Horák and B. Bošanskỳ, "Solving partially observable stochastic games with public observations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 2029–2036.

[13] H. Li and Z. Zheng, "Optimal timing of moving target defense: A stackelberg game model," in *MILCOM 2019-2019 IEEE Military Communications Conference (MILCOM)*. IEEE, 2019, pp. 1–6.

[14] D. Sahabandu, J. Allen, S. Moothedath, L. Bushnell, W. Lee, and R. Poovendran, "Quickest detection of advanced persistent threats: A semi-markov game approach," in *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2020, pp. 9–19.

[15] H. Li, W. Shen, and Z. Zheng, "Spatial-temporal moving target defense: A markov stackelberg game model," *arXiv preprint arXiv:2002.10390*, 2020.

[16] R. Tipireddy, S. Chatterjee, P. Paulson, M. Oster, and M. Halappanavar, "Agent-centric approach for cybersecurity decision-support with partial observability," in *2017 IEEE International Symposium on Technologies for Homeland Security (HST)*. IEEE, 2017, pp. 1–6.

[17] E. Miehling, M. Rasouli, and D. Teneketzis, "A pomdp approach to the dynamic defense of large-scale cyber networks," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 10, pp. 2490–2505, 2018.

[18] K. Hazeghi, "Markov decision processes: Discrete stochastic dynamic programming," *Journal of the American Statistical Association*, vol. 90, no. 429, pp. 392–394, 1995.

[19] A. R. Cassandra, L. P. Kaelbling, and M. L. Littman, "Acting optimally in partially observable stochastic domains," in *Aaai*, vol. 94, 1994, pp. 1023–1028.

[20] Z. Hu, M. Zhu, and P. Liu, "Online algorithms for adaptive cyber defense on bayesian attack graphs," in *Proceedings of the 2017 Workshop on moving target defense*, 2017, pp. 99–109.

[21] M. L. Littman, "A tutorial on partially observable markov decision processes," *Journal of Mathematical Psychology*, vol. 53, no. 3, pp. 119–125, 2009.

[22] K. P. Murphy, "A survey of pomdp solution techniques," *environment*, vol. 2, p. X3, 2000.

[23] S. Jajodia, G. Cybenko, P. Liu, C. Wang, and M. Wellman, *Adversarial and Uncertain Reasoning for Adaptive Cyber Defense: Control-and Game-theoretic Approaches to Cyber Security*. Springer Nature, 2019, vol. 11830.

[24] Z. Zhang, F. Nait-Abdesselam, and P.-H. Ho, "Boosting markov reward models for probabilistic security evaluation by characterizing behaviors of attacker and defender," in *2008 Third International Conference on Availability, Reliability and Security*. IEEE, 2008, pp. 352–359.

[25] P. Holgado, V. A. Villagrá, and L. Vázquez, "Real-time multistep attack prediction based on hidden markov models," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 1, pp. 134–147, 2020.

[26] "Darpa - intrusion detection evaluation dataset," https://www.ll.mit.edu/ideval/data/2000data.html, 2000.

Nadia Niknami received her B.S. degree in Computer Science from University of Isfahan, Iran, in 2011, and MSc degrees From Tarbiat Modares University, Tehran, Iran in 2015. She is currently pursuing the Ph.D. degree in the Department of Computer and Information Sciences, Temple University, Philadelphia. Her current research focuses on security, privacy-preserving, attack-defense scenarios, Markov chain, and game theory.

Jie Wu is the Director of the Center for Networked Computing and Laura H. Carnell professor at Temple University. He also serves as the Director of International Affairs at College of Science and Technology. He served as Chair of Department of Computer and Information Sciences from the summer of 2009 to the summer of 2016 and Associate Vice Provost for International Affairs from the fall of 2015 to the summer of 2017. Prior to joining Temple University, he was a program director at the National Science Foundation and was a distinguished professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. Dr. Wu regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Mobile Computing, IEEE Transactions on Service Computing, Journal of Parallel and Distributed Computing, and Journal of Computer Science and Technology. Dr. Wu was general cochair for IEEE MASS 2006, IEEE IPDPS 2008, IEEE ICDCS 2013, ACM MobiHoc 2014, ICPP 2016, and IEEE CNS 2016, as well as program chair/cochair for IEEE INFOCOM 2011, CCF CNCC 2013, and ICCCN 2020. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a Fellow of the AAAS and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.