POSTER: A Markov Decision Process to Determine Optimal Policies in Moving Target

Jianjun Zheng
Computer Science Department
Texas Tech University
jianjun.zheng@ttu.edu

Akbar Siami Namin
Computer Science Department
Texas Tech University
akbar.namin@ttu.edu

ABSTRACT

Moving Target Defense (MTD) has been introduced as a new game changer strategy in cybersecurity to strengthen defenders and conversely weaken adversaries. The successful implementation of an MTD system can be influenced by several factors including the effectiveness of the employed technique, the deployment strategy, the cost of the MTD implementation, and the impact from the enforced security policies. Several efforts have been spent on introducing various forms of MTD techniques. However, insufficient research work has been conducted on cost and policy analysis and more importantly the selection of these policies in an MTD-based setting.

This poster paper proposes a Markov Decision Process (MDP) modeling-based approach to analyze security policies and further select optimal policies for moving target defense implementation and deployment. The adapted value iteration method would solve the Bellman Optimality Equation for optimal policy selection for each state of the system. The results of some simulations indicate that such modeling can be used to analyze the impact of costs of possible actions towards the optimal policies.

CCS CONCEPTS

• **Networks** → **Network security**; • **Security and privacy** → *Formal security models*; • **Moving Target Defense**;

KEYWORDS

Moving Target Defense; Markov Decision Process; Optimal Policy

ACM Reference Format:
Jianjun Zheng and Akbar Siami Namin. 2018. POSTER: A Markov Decision Process to Determine Optimal Policies in Moving Target. In *2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18), October 15–19, 2018, Toronto, ON, Canada.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3243734.3278489

1 INTRODUCTION

Moving Target Defense has been an ongoing active research since its official introduction at the National Cyber Leap Year Summit in 2009 [1, 6–8]. The essence of Moving Target Defense is security through diversification which dynamically and randomly changes the configurations and properties of a target system (e.g., a host or a network). This creates a complex and unpredictable moving target for attackers and thus makes it computationally expensive for them to exploit exposed and known vulnerabilities. While increasing the attack cost and reducing attackers' financial incentive, effective Moving Target Defense implementation can also come with cost and thus impose financial burden on defenders and accordingly on the network infrastructure. Therefore, it is important to take into account cost in the factors that can affect the effectiveness of MTD. Examples of possible factors include: the type of attack [4], network environment and MTD deployment [7], and MTD strategy change frequency [8].

Security policies are imposing additional restricting factor of the implementation of MTD in practice. These security policies are usually defined on the network, on which the prospective MTD system would be deployed. The existing security policies regulate actions that are permitted or prohibited under certain circumstances (i.e., access controls) and might also cause conflicts with possible actions required by the MTD.

Many MTD-based techniques have been introduced to address many real-world security challenges and these research efforts focus on introducing practical MTD techniques and conduct simulation to evaluate the effectiveness of their techniques. A major problem with these approaches is that they are technique-specific and the evaluation mechanism of a certain MTD-based technique can hardly be applied to another technique. Therefore, an appropriate mathematical model is needed to evaluate MTD techniques from a higher and more abstract level for a better evaluation of MTD.

To meet this challenge, this poster paper proposes to use Markov Decision Process (MDP) to model the state transitions of a system based on the interaction between a defender and an attacker. A Markov model is a stochastic model used to describe the state transition of a system. Combined with game theory, a Markov game model can be built to describe the interaction between defenders and attackers and then analyze the outcome of the system when it is in a certain state. The Markov chain game model is helpful for providing information for a defender to choose the best strategy for the next move. However, the network defenders in some situations face time constraints when making decision with respect to the outcome obtained from a Markov model. Therefore, A model is needed that can make decision with the goal of implementing best security policies (i.e., actions) in certain situations.

Our model incorporates the costs of players' actions and the existing security policies in a system and further uses Bellman Optimality Equations to find the optimal defense strategies or policies under different scenarios. The results are used to analyze the impact of the policy change by the cost of chosen strategy.

CCS '18, October 15–19, 2018, Toronto, ON, Canada

ACM ISBN 978-1-4503-5693-0/18/10.
https://doi.org/10.1145/3243734.3278489

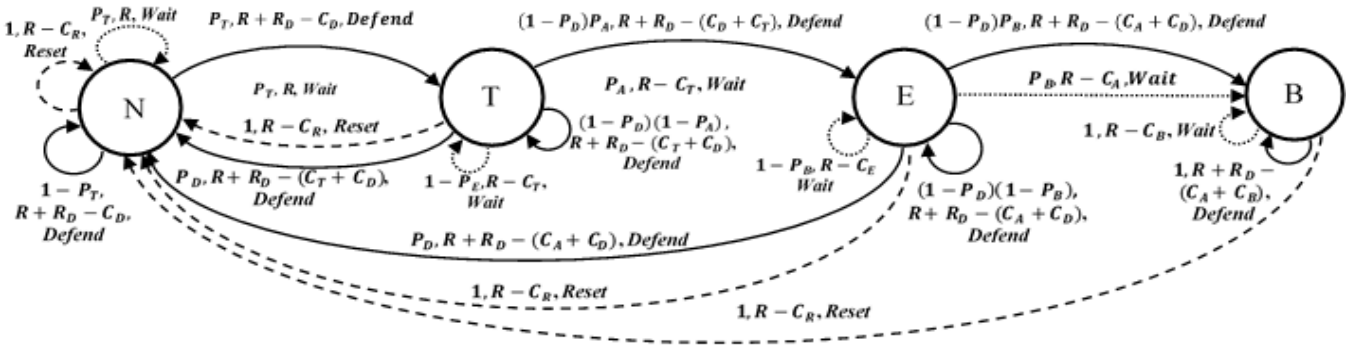

Figure 1: MDP model with state transition probabilities and costs.

The remainder of this paper is organized as follows: Section 2 describes Markov Decision Process game model and Bellman Optimality Equation. Section 3 presents the model simulation setup. Section 4 presents our future work and concludes the poster.

2 A MARKOV DECISION-BASED MODEL

In the proposed Markov decision-based model, the interaction between a defender and an attacker is abstracted out as a discrete, finite-state, and finite-action Markov Decision Process (MDP) as a 4-tuple (S, A, P, R), where:

- S is a finite set of states.
- A is a finite set of control actions.
- P is the probability of a state transitioning to a new state upon performing an action.
- R is the expected immediate rewards received after state transition, due to the control action performed.

Figure 1 depicts the proposed MDP-based model. In this model, the security defense mechanism is abstracted out into four states (S) and three control actions (A):

$$S \in \begin{cases} N & \textit{System Running Normally} \\ T & \textit{System Being Targeted} \\ E & \textit{System Being Exploited} \\ B & \textit{System Breached} \end{cases} \tag{1}$$

$$A \in \{\textit{Wait}, \textit{Defend}, \textit{Reset}\} \tag{2}$$

The ultimate goal is to find an optimal policy for the defender, by which the defender needs to know what best action needs to be taken in each state with the goal of maximizing the rewards.

2.1 Key Concepts of MDP

In a typical MDP, the most critical property that must be satisfied is known as *Markov property*. This property states that the effects of an action taken in any state depend only on that state and not on the prior history or knowledge.

A *policy* π in MDP is a mapping function from states to actions: $\pi : S \rightarrow A$. In other words, a policy dictates each process (i.e., agent) to take certain actions in each state.

The *value function*, denoted by $V_\pi(s)$, represents the expected value of rewards received starting from state s and following policy π. It is also called state value function or utility function:

$$V_\pi(s) = \sum_{s' \in S} P(s, \pi, s')[R(s, \pi, s') + \gamma V_\pi(s')] \tag{3}$$

where:

- $P(s, \pi, s')$ is the transition probability starting from state s and ending at state s' after following policy π.
- $R(s, \pi, s')$ is the expected rewards received after state transition from s to s' after following policy π.
- γ is the discount factor.

The *discount factor* in MDP, denoted by $\gamma \in (0, 1)$, indicates what portion of the future rewards will be lost in comparison to the present rewards. Smaller γ means the rewards received in the future will be worth much less than the present rewards due to the discount, so the reward should be collected sooner than later.

An *optimal policy* π^* is a control action $a \in A$ that generates the maximum state value function and is expressed by *Bellman Optimality Equation* [2]:

$$V^*_{i+1}(s) = \max_{a \in A} \sum_{s' \in S} P(s, a, s')[R(s, a, s') + \gamma V^*_i(s')] \tag{4}$$

The optimal policy can be obtained by solving the MDP problem or the Bellman Optimality Equation.

2.2 Solving MDP

Before discussing how to solve an MDP problem, we need to introduce some important theorems regarding MDP [2, 5]:

Theorem 2.1. *For any finite Markov Decision Process (MDP), there exists an optimal policy that is always better than or equal to all other policies,* $\pi^* \geq \pi, \forall \pi$.

Theorem 2.2. *All optimal policies in any finite Markov Decision Process achieve the optimal value function,* $V_{\pi^*}(s) = V^*(s)$.

The formal proof of Theorem 2.1 and Theorem 2.2 can be found in [5] and [2], respectively. Based on these two theorems, the optimal policy is obtained by solving the Bellman Optimality Equation such that [2]:

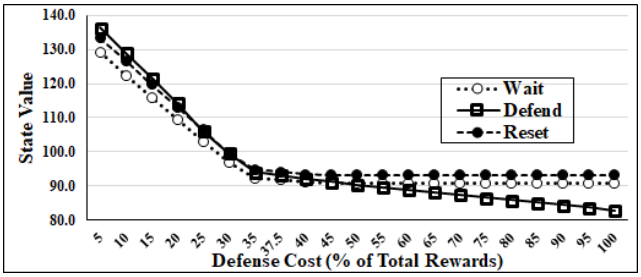

Figure 2: Impact of defense cost on value function at state $S = E$.

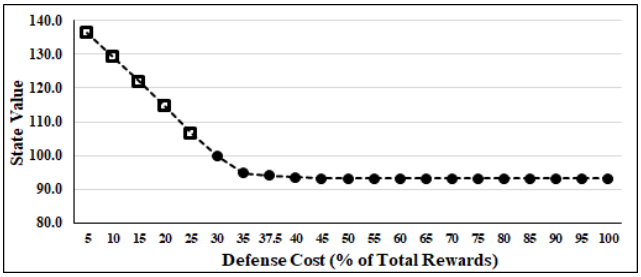

Figure 3: The optimal policy changes as the defense cost increases.

$$\pi^*(s) \leftarrow arg\max_{a \in A} \sum_{s' \in S} P(s,a,s')[R(s,a,s') + \gamma V_i^*(s')] \quad (5)$$

Value Iteration is a method developed by Bellman [2] to solve MDP. In the proposed model, the value iteration method is chosen because due to its simplicity.

2.3 The Impact of Cost on Optimal Policy

In MDP, the optimal policy can be controlled by manipulating the rewards. In our model, we introduce the concept of cost factor and define the expected reward as the result of the baseline reward R subtracting the cost incurred by an action during a state transition. The action can be initiated by the attacker or the defender. After incorporating the cost factor into the computation, the Bellman equation will be:

$$V_{i+1}^*(s) = \max_{a \in A} \sum_{s' \in S} P(s,a,s')[(R - C(s,a,s')) + \gamma V_i^*(s')] \quad (6)$$

where $C(s,a,s')$ is the cost incurred after state transition from s to s' due to action a. This equation will enable us to analyze the cost impact on the optimal policy. The action $a \in \{\text{Wait, Defend, Reset}\}$ that yields the maximum value will be chosen as the optimal policy.

3 SIMULATION EXPERIMENTS

We implemented the value iteration method and calculated the value function for each policy (wait, defend, reset) at each state $S \in \{N, T, E, B\}$ with different defense cost through simulation. As an example, the value function at the state $S = E$ for each control action versus the defense cost is plotted in Figure 2, where x-axis and y-axis show the defense cost and the value function at each state, respectively. As Figure 2 shows, when the defense cost is below a certain value (called the *turning point*), the "Defense' action is the optimal policy and the best decision to make. On the other hand, when the defense cost is above the turning point, the "Reset" action turns out to be the optimal policy because it generates higher rewards than the other two actions.

This optimal policy shift can be better demonstrated when the optimal state value is plotted against each level of the defense cost for state $S = E$. Figure 3 shows such this plot, where x-axis and y-axis show the defense cost and the value function at the state, respectively. The first 5 data points indicate the "Defend" action and the rest data points indicate the "Reset" action.

4 CONCLUSION AND FUTURE WORK

This poster paper introduced the idea of modeling MTD and the problem of making optimal decisions through MDP. The model defined four states, in which optimal policies can be made with respect to the actions. The simulation results show that the optimal policy changes with in accordance with associated costs for each action.

As future work, it is important to apply the MDP-based model to other network dynamics and investigate how various cost factors impact the decision about the optimal policy. We also plan to apply the MDP-based model to some existing MTD techniques to demonstrate how to select the optimal policy and provide additional insights on the feasibility of the MTD techniques. We would also like to compare our work to the evidence theory [3] which is also applicable to this problem. Finally we plan to address some challenges in the introduced model such as the estimation of the initial probability values in a real-world dataset.

ACKNOWLEDGEMENT

This project is funded in part by a grant (Awards No: 1516636 and 1564293) from National Science Foundation.

REFERENCES

[1] 2009. National Cyber Leap Year Summit 2009. Retrieved June 1, 2018 from https://www.nitrd.gov/nitrdgroups/index.php?title=National_Cyber_Leap_Year_Summit_2009

[2] Richard E. Bellman. 2010. *Dynamic Programming* (reprint ed.). Princeton University Press.

[3] M. Chatterjee and A. S. Namin. 2018. Detecting Web Spams Using Evidence Theory. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, Vol. 02. 695–700. https://doi.org/10.1109/COMPSAC.2018.10321

[4] David Evans, Anh Nguyen-Tuong, and John Knight. 2011. *Effectiveness of Moving Target Defenses.* Springer New York, New York, NY, 29–48. https://doi.org/10.1007/978-1-4614-0977-9_2

[5] Ronald A. Howard. 1960. *Dynamic Programming and Markov Decision Processes.* Technology Press of MIT.

[6] Jafar Haadi Jafarian, Ehab Al-Shaer, and Qi Duan. 2012. Openflow Random Host Mutation: Transparent Moving Target Defense Using Software Defined Networking. In *Proceedings of the First Workshop on Hot Topics in Software Defined Networks (HotSDN '12).* ACM, New York, NY, USA, 127–132. https://doi.org/10.1145/2342441.2342467

[7] Richard Skowyra, Kevin Bauer, Veer Dedhia, and Hamed Okhravi. 2016. Have No PHEAR: Networks Without Identifiers. In *Proceedings of the 2016 ACM Workshop on Moving Target Defense (MTD '16).* ACM, New York, NY, USA, 3–14. https://doi.org/10.1145/2995272.2995276

[8] Jianjun Zheng and Akbar Siami-Namin. 2016. The Impact of Address Changes and Host Diversity on the Effectiveness of Moving Target Defense Strategy. In *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, Vol. 2. 553–558. https://doi.org/10.1109/COMPSAC.2016.233