9-2015

An adaptive Markov strategy for effective network intrusion detection

Jianye HAO

Yinxing XUE

Mahinthan CHANDRAMOHAN

Yang LIU

Jun SUN
Singapore Management University, junsun@smu.edu.sg

An Adaptive Markov Strategy for Effective Network Intrusion Detection

Conference Paper · November 2015

DOI: 10.1109/ICTAI.2015.154

CITATIONS
2

READS
6

5 authors, including:

Chandramohan Mahinthan
Nanyang Technological University
23 PUBLICATIONS **447** CITATIONS

SEE PROFILE

Yang Liu
Nanyang Technological University
279 PUBLICATIONS **2,672** CITATIONS

SEE PROFILE

Jun Sun
Singapore University of Technology and Design
231 PUBLICATIONS **2,247** CITATIONS

SEE PROFILE

An Adaptive Markov Strategy for Effective Network Intrusion Detection

Jianye Hao*, Yinxing Xue†, Mahinthan Chandramohan†, Yang Liu† and Jun Sun*
* *Singapore University of Technology and Design*
Email: {jianye_hao,sunjun}@sutd.edu.sg
† *Nanyang Technologocial University*
Email: yinxingxue@gmail.com, {commahintha001, yangliu}@ntu.edu.sg

***Abstract*—Network monitoring is an important way to ensure the security of hosts from being attacked by malicious attackers. One challenging problem for network operators is how to distribute the limited monitoring resources (e.g., intrusion detectors) among the network to detect attacks effectively, especially when the attacking strategies can be changing dynamically and unpredictable. To this end, we adopt Markov game to model the interactions between the network operator and the attacker and propose an adaptive Markov strategy (AMS) to determine how the detectors should be placed on the network against possible attacks to minimize the network's accumulated cost over time. The AMS is guaranteed to converge to the best response strategy when the attacker's strategy is fixed (*rationality*), converge to a fixed strategy under self-play (*convergence*) and obtain a payoff no less than that under the precomputed Nash equilibrium strategy of the Markov game (*safety*). The experimental results show that the AMS can achieve better protection for the network compared with both previous approaches based on the prediction of attack paths and Nash equilibrium strategy.**

I. Introduction

An important way of securing a network to leverage network monitoring to protect hosts from being attacked from malicious attackers [7]. Among various ways of network monitoring, one important technique is to place malware detectors in the network to proactively detect possible malicious packages and take necessary actions accordingly.

To determine how the detectors should be placed, the typical approach in network security is to predict all possible attacking paths in terms of attack tree [9] or attack graph [17], [13], and then place detectors to cover all possible predicted paths. However, the coverage of all possible attack paths might require too much monitoring, which would incur significant deployment cost and significantly decrease network performance due to the overhead of monitoring. Thus, it is vital to identify manageably small number of strategic points in the network to install malware detectors. Besides, two major characteristics can be usually observed in practical attacks. First, the way that attackers choose their attacking targets (e.g., Distributed denial-of-service (DDoS) attack) is usually not random, but a deliberate and calculated process [15]. Instead of *naively* selecting a target node and *carelessly* launching the attack, the attackers usually are well-prepared beforehand by studying the potential target nodes, investigating the possible vulnerabilities and evaluating the risk of being detected for different attacking options. Second, the attacks can be "multi-stage" [3], i.e., the attacker may penetrate node A first, and then use it as a platform to penetrate another node B, which is in turn used to penetrate node C.

The *targeted* and *multi-stage* nature of the attacker's behaviors makes it necessary for the network operator (or defender) to determine its detector placement strategy in a strategic way instead of taking into consideration all possible attacking paths. One natural solution is to adopt Markov game-theoretic framework to model and analyze the strategic interaction between them. Both defender and attacker can be considered as individually rational entities interested in minimizing (or maximizing) the long-term damage to the network. A number of work [16], [20], [8] investigated the detector placement problem by modeling the interaction between the attacker and defender as a Markov game and proposed using the Nash equilibrium (NE) solution as the deploying strategy. The rationale for adopting a NE solution lies in the stable property of NE, e.g., neither the attacker nor the defender can do better by choosing a different strategy under a NE.

However, always choosing a NE strategy to deploy the detectors is not necessarily optimal for the defender, since it depends on the assumption that the attacker would always choose the same NE strategy to attack the network, which might not hold for the following reasons: (1) computing a NE relies on a perfect knowledge of the network; however, in reality, the attacker might not have the capability to collect enough information to construct an accurate model of the network or compute the NE attacking strategy beforehand; (2) since attacking is performed by intelligent human attackers, it is likely that they would choose their attacking strategy based on their intuition or past experience, which might deviate from the NE strategy; (3) even if the attacker follows a NE strategy, it is unclear whether they would coordinate on the same pair of NE strategy if multiple equilibria coexist. Miscoordination might incur significant unnecessary cost for the defender.

To handle these challenges, in this paper, we model the interactions between the network operator and the attacker as a Markov game and propose an adaptive Markov strategy

(AMS) to adaptively determine how to place the detectors on the network to minimize the network's cost based on the estimation of the attacker's behaviors. The AMS is guaranteed to converge to the best response strategy when the attacker's strategy is fixed (*rationality*) and always obtain a payoff no less than that under the Nash equilibrium strategy of the Markov game (*safety*) no matter how the attacker may behave. We also empirically evaluate the performance of AMS and the results demonstrate that it leads to better protection for the network compared with both approaches based on the prediction of attack paths and Nash equilibrium strategy.

The rest of the paper is organized as follows. In Section II, we give the problem description and the Markov game modeling of the problem. Section III presents the AMS strategy and analyzes its properties. In Section IV, we present the experimental results of the AMS strategy and compare it with previous approaches. Lastly Section V concludes the paper.

II. Malware Detector Placement Problem: Markov Game Modeling

In this work, we investigate the malware detector placement problem, where an attacker, from compromised nodes (e.g., hosts and routers), launches a strategic and multi-stage attack to take control of target nodes of interest (e.g., database servers); the defender (e.g., network operator), on the other hand, places malware detectors in strategic points (i.e., routers in the path of attack) to prevent the attack measures taken by the adversaries.

Formally, given a network consisting of a set G of nodes, we assume that the set $G_T \subset G$ of nodes are potential target nodes and the set $G_C \subset G$ ($G_C \cap G_T = \emptyset$) of nodes are compromised nodes controlled by the attacker and served as the starting penetration points. The attacker may start the attack by sending malicious packages from any already compromised node to any target node. For any malicious package traveling through any node k, it can be detected with probability p if a detector is placed on this node (*detection success rate*). If malicious packages successfully bypass the detectors and reach a target node, we assume that the target node is compromised with probability q (*attack success rate*). Given the attacker and defender's actions, the actual travel paths of malicious packages through the network vary depending on the current routing configuration. This indirectly influences the outcome of the attack, i.e., whether the target nodes can be successfully compromised. The routing configuration of the network is usually changed in a dynamic way based on the outcome of previous stage. The attacker's goal is to maximize the overall damage to the network by compromising as many target nodes as possible, and the defender suffers equivalently from the gain of the attacker.

We model the interactions between the *attacker* and the *defender* as a zero-sum Markov game as follows,

- S: a finite set of system states. We abstract the routing tables used in the network determining package delivering paths as the system states. Formally we have $S = \{RT_1, ..., RT_k\}$, where RT_i $(1 \leq i \leq k)$ denotes one routing table (one network routing configuration).
- N: a finite number of players. In our setting, there are two players (defender and attacker), i.e., $N = \{d, a\}$.
- A_a and A_d: the set of actions for each player. An action of the attacker corresponds to its attacking plan, i.e., the pairs of its attacking starting node and the targeting node. For example, an action of the attacker can be denoted as $\langle n_i, t_j \rangle, n_i \in G_C, t_j \in G_T$. An action of the defender corresponds to the specification of the set of nodes that the detectors are placed on. As we mentioned previously, in real world, due to various constraints such as deployment cost and performance overhead, it is not feasible to apply network monitoring measure on all possible nodes. Thus the defender's action set only consists of all the feasible actions subject to its resource constraint.
- Pr: transition probability function. Given the current state s and the joint action (d, a), $Pr(d, a, s, s')$ models the probability that the network state (routing configuration) transits from s to s' when the defender and the attacker perform actions d and a, respectively.
- R_a and R_d: payoff function of the players. Given $s \in S$, $a \in A_a$, and $d \in A_d$, $R_d(s, d, a)$ and $R_a(s, d, a)$ return the immediate expected payoff of the defender and attacker respectively when the joint action (d, a) is performed under state s. Intuitively, the attacker's payoff is determined by the cost of launching attack plus the benefit from successfully compromising certain nodes. The defender's payoff $R_d(s, d, a)$ is simply the negation of $R_a(s, d, a)$, since we assume that the defender benefits equally from the attacker's cost and vice versa. Let us use a_t to denote the set of target nodes in the attacker's action a, and formally we have

$$R_a(s, d, a) = \sum_{t \in a_t} [f(s, d, a, p, q, t) \times c_1(t) + f'(s, d, a, p, q, t) \times c_2(t)], \quad (1)$$

  where p and q are *detection success rate* and *attack success rate* respectively, and $f(s, d, a, p, q, t)$ is the overall probability of compromising target node t, which can be obtained through simulation; $f'(s, d, a, p, q, t) = 1 - f(s, d, a, p, q, t)$; and $c_1(t)$ and $c_2(t)$ are the damage to the system if node t is compromised and the attack's loss if an attack to node t fails, respectively.

We define a player's strategy ϕ as a function that given some state s, returns a probability distribution over the set of actions that the player may perform in state s. The long-term goal of the defender (or attacker) is to maximize its

overall payoff along the repeated interactions between them. We adopt the γ-discounted criterion and define the overall payoff $V()$ of a player as the sum of the expected discounted payoff of each round over an infinite number of interactions. Formally for each starting state s, its corresponding overall payoff is defined as follows,

$$V_i(s) = \sum_{t=0}^{\infty} \gamma^t E[R_i(s_t, d_t, a_t)|s_0 = s], \text{where } i \in \{d, a\} \tag{2}$$

III. Adaptive Markov Strategy

In the previous section, we have modeled the interactions between an attacker and a network defender (operator) as a Markov game. The question for defender is how to place the *limited* number of detectors in the network in a strategic way to maximize its own long-term payoff. A commonly adopted approach is employing a *Nash equilibrium* (NE) strategy: that is, both the attacker and the defender play a strategy that would maximize their individual payoffs given none of them changes its strategy [16], [20], [8].

However, simply adopting NE strategy might not be the optimal solution due to a number of reasons mentioned previously. An effective detector placement strategy should be *adaptive*, i.e., it should be able to learn the attacker's strategy and dynamically compute the *best response* strategy to the attacking strategy in terms of where the detectors should be placed. However, assuming that an attacker may change its strategy arbitrarily is neither useful nor practical. Besides, putting too much restriction on the attacker's behavior might make the defending strategy not very useful in practice since the attacker is outside of our control.

In this paper, we define the following three criteria which are desirable for an effective detector placement strategy to satisfy in the context of Markov game [1], [14].

Rationality - A rational detector placement strategy must eventually learn to play the best response strategy if the attacker eventually converges to a fixed strategy. Intuitively, satisfying this property guarantees that the overall network damage can be minimized as long as it is possible to achieve.

Convergence - The detector placement strategy must always converge to a fixed strategy under self-play. This property considers the case when the attacker might be as intelligent as the defender and employ the same adaptive strategy. We can see that under self-play, if both rationality and convergence properties are satisfied, the defender and attacker will eventually converge to a NE. This means that the maximum cost to the network can be bounded to the cost under a NE, even when the attacker is as intelligent as the defender.

Safety - The average overall payoff $V(s)$ of the defender for each state s in the limit should have certain minimum guarantees, no matter how the attacker may behave. We require that it should be no less than the corresponding payoff under the precomputed NE of the Markov game.

A number of learning strategies have been proposed to satisfy some of the above properties in the multiagent learning literature, however, all of them suffer from either of the following two problems: 1) long learning periods are required before converging to the best response strategy, thus resulting in significant losses during learning period and failing to make timely response [2], [14]; 2) some strategies are designed for repeated game setting only and also do not satisfy all the above properties [4], [6]. Thus we cannot directly apply the existing learning strategies into the malware detector placement problem. In this paper, we propose an adaptive Markov strategy (AMS) for Markov games which satisfies all the above three properties. The AMS strategy can be considered as an extension of the AWESOME strategy [4] from repeated games setting to Markov game setting.

A. Overview of AMS and Action Space Reduction

The high-level idea of the AMS algorithm is explained as follows. It begins by assigning an NE strategy as the defending strategy, and observes the behavior of the attacker for some fixed number of rounds (called a period). If the estimated strategy of the attacker is consistent with its NE strategy, then AMS keeps the original NE as the defending strategy. Otherwise, it computes a new best response strategy to play against its current estimation of the attacker's strategy. After playing the new strategy for another period of rounds, AMS checks whether the attacker's strategy remains the same as the one from the previous period; if not, this implies that the previous estimation of the attacker's strategy was incorrect, and so AMS restarts the whole process again by retreating to the original equilibrium strategy.

The action spaces of the players have significant influence on the learning efficiency since the increase of action space would necessarily increase the computational cost of calculating the best-response strategy. Thus it would be desirable to reduce unnecessary (or unrealistic) actions from the original action space beforehand by taking into consideration the problem domain's characteristics to reduce the computational cost of the AMS strategy.

Defender - The defender's action space can be reduced by taking into consideration the network topology characteristics. For example, if a node is the only one connecting one subnetwork consisting of one or more target nodes with the rest of the network, we can always put a detector on that node to detect any possible attack towards those target nodes within that subnetwork. Thus any action consisting of placing some detectors on some nodes within that subnetwork can be removed from the action space. From the game-theoretic perspective, those actions correspond to (weakly) dominated actions and can be safely removed without affecting the analytical results [18].

Attacker - In practical attacks (e.g., DDoS attack), to successfully compromise the node, it usually requires coordinated attacks from multiple nodes towards the same target node [10]. Similar to the defender's case, those actions consisting of attacking target nodes from a single node controlled by the attacker are (weakly) dominated and thus can be safely removed without affecting the analytical results. Thus any action violating the above constraint can be removed from the attacker's action space.

B. AMS: Adaptive Markov Strategy

Before introducing the AMS algorithm in details, we need to explain a few terms first. First, to determine whether the attacker is employing the precomputed NE or any other stationary strategy, we define the *distance* between two stationary strategies to compare whether they are the same or not.

Definition 1. *The* distance $Distance(\phi_1, \phi_2)$ *between two stationary strategy* ϕ_1 *and* ϕ_2 *is:*

$$Distance(\phi_1, \phi_2) = \max |\phi_1(s,a) - \phi_2(s,a)|, \forall a \in A_s, s \in S \tag{3}$$

where A_s *is the action space at state* s *and* S *is the state space, and* $\phi_1(s,a)$ *and* $\phi_2(s,a)$ *is the probability that action* a *is played at state* s *for strategy* ϕ_1 *and* ϕ_2 *respectively.*

Second, given two strategies ϕ_1 and ϕ_2, we define the value $V(s, \phi_1, \phi_2)$ of playing strategy ϕ_1 against strategy ϕ_2 under state s, which is defined as the sum of the discounted expected payoff obtained over infinite number of interactions.

Definition 2. *The value* $V(s, \phi_1, \phi_2)$ *of playing strategy* ϕ_1 *against strategy* ϕ_2 *under state* s *is defined as follows,*

$$\begin{aligned} V(s, \phi_1, \phi_2) =& R(s, \phi_1(s), \phi_2(s)) + \\ & \delta \sum_{s \in S} Pr(\phi_1(s), \phi_2(s), s, s') V(s', \phi_1, \phi_2) \end{aligned} \tag{4}$$

where δ is the discounting factor reflecting the relative importance of future payoffs and $Pr(\phi_1(s), \phi_2(s), s, s')$ is the probability that the system state transits from s to s' given that the players choose actions $\phi_1(s)$ and $\phi_2(s)$ respectively. We can construct one equation for V-value of each state $s \in S$ following Definition 2, and thus the value of each state can be calculated by solving a system of $|S|$ linear equations using techniques such as iterative methods [5].

The AMS algorithm (Algorithm 1) takes place over consecutive *periods* (where each period is some number of rounds). Initially, the AMS begins by playing the precomputed NE strategy for the initial period N^0 (Line 5) (described in details later) and estimates the strategy of the attacker based on the actions taken in this period (Line 7 to 9). If the *distance* between the estimated strategy h_a^{curr} and the NE strategy π_a^* of the attacker is larger than the given threshold (line 13 - 14), the attacker is considered playing a non-NE strategy, and the first while-loop is terminated by setting APPE to False. After that, AMS computes the best response strategy ϕ_d' (described in details later) against the current estimated strategy h_a^{curr} of the attacker based on the last period's interaction (Line 16). Then AMS first checks whether the the attacker's stately is unchanged in previous two consecutive rounds by comparing the estimated strategy h_a^{curr} and h_a^{prev} of the attacker (Line 17). If changed, AMS continues adopting the precomputed NE strategy (Line 18); otherwise, AMS checks whether for every state $s \in S$, the difference between the V-value of ϕ_d' against the h_a^{curr} (see Definition 2) and that of ϕ_d is larger than the given threshold $2|A|^{|S|}\epsilon_s^{t+1}\mu$ (where $|A|^{|S|}$ represents the total number of pure strategies of the Markov game and μ is the maximum payoff difference between the AMS player's best and worse outcomes among all states). If true, the current NE defending strategy ϕ_d is replaced by a more optimal strategy ϕ_d' (Line 19-20).

At the end of the each following period, AMS checks whether the attacker's stately is indeed unchanged by comparing the estimated strategy h_a^{curr} and h_a^{prev} of the attacker in the current and preceding periods (Line 25 - 27). If the distance between these two is larger than the given threshold ϵ_s^t, it indicates that the opponent is not playing according to the estimated strategy h_a^{prev}, and the AMS will restart by breaking from the second while loop (APS = False). Otherwise, the AMS recomputes a best response strategy ϕ_d' based on the last period's interaction, and employs ϕ_d' as its strategy if it is more optimal than ϕ_d (Line 19-20). This overall process repeats as indicated by the outer *Repeat* loop. Note that we also check the empirical strategy of the AMS strategy (Line 12-14 and 25-27) to ensure the synchronization when both players adopt AMS strategy.

The remaining question is how the parameters of the AMS algorithm should be adjusted, which is described as follows.

Definition 3. *A schedule of adjusting the parameters* $\{\epsilon_e^t, \epsilon_s^t, N^t\}$ *is valid if*

- $\epsilon_e^t, \epsilon_s^t$ *and* ϵ_c^t *are decreased monotonically and converge to zero eventually.*
- *the value of* N^t *is increased monotonically at the end of each period* t *to infinity.*
- $\Pi_{t \in \{1,2,\ldots\}}(1 - A_S \frac{1}{N^t(\epsilon_s^{t+1})^2}) > 0$*, where* A_S *is the total number of actions of the defender summed over all states.*

Compute Nash Equilibrium strategy of Markov Game

Since the Markov game modeling the interaction between attacker and defender is a zero-sum Markov game (the sum of the attacker and defender's payoffs is always 0), the maxmin/minmax strategy of the Markov game for each

Algorithm 1: Description of AMS

Compute a NE strategy $(\pi_i^*, \forall i \in \{d, a\})$, initialize $t = 0$;
repeat
  Initialize h_i^{prev}, h_i^{curr} to 0, $\forall i \in \{d, a\}$;
  $s = s_0$, $APS = True$, $APPE = True$;
  Set defender strategy ϕ_d to be the NE strategy $(\phi_d = \pi_d^*)$;
  while *APPE = True* **do**
    for r *: 0 to* N^t **do**
      Play($\phi_d(s)$);
      Update(h_i^{curr}), $\forall i \in \{d, a\}$;
    $h_a^{prev} = h_a^{curr}$;
    $t := t + 1$;
    for *each player* $i \in \{d, a\}$ **do**
      if *Distance*(h_i^{curr}, π_i^*) $> \epsilon_e^t$ **then**
        APPE = False;
  while *APS = True* **do**
    ϕ_d' := BestResponseStrategy(h_a^{curr});
    if *Distance*(h_a^{curr}, h_a^{prev}) $> \epsilon_c^t$ **then**
      $\phi_d = \pi_d^*$;
    else if $V(s, \phi_d', h_a^{curr}) > V(s, \phi_d, h_a^{curr}) + 2|A|^{|S|}\epsilon_s^{t+1}\mu$, $\forall s \in S$ **then**
      $\phi_d = \phi_d'$;
    for r *: 0 to* N^t **do**
      Play($\phi_d(s)$);
      Update(h_i^{curr}), $\forall i \in \{d, a\}$;
    $t := t + 1$;
    for *each player* $i \in \{d, a\}$ **do**
      if *Distance*(h_i^{curr}, h_i^{prev}) $> \epsilon_s^t$ **then**
        APS = False;
    $h_i^{prev} = h_i^{curr}$, $\forall i \in \{d, a\}$;
until;

player is equivalent with its corresponding Nash equilibrium strategy [18]. Thus we only need to compute the minmax/maxmin strategy profile of the Markov game instead. We first define the Q-value $Q_d(s, d, a)$ of the defender as its expected long-term value starting at state s by choosing action d (the attacker chooses action a) and both players choose the minmax strategy thereafter. Formally we can have,

$$Q_d(s, d, a) = R_d(s, d, a) + \delta \sum_{s' \in S} Pr(d, a, s, s') V_d(s') \quad (5)$$

where $Pr(s, d, a, s')$ is the transition probability from state s to state s' under the joint action (d, a), δ is the discounting factor, and $V_d(s')$ is the long-term expected payoff of the defender if both players always choose their corresponding maxmin strategy. The value of $V_d(s)$ can be defined based on the $Q_d(s, a, d)$ as follows,

$$V_d(s) = \max_{\phi_d(s) \in \Pi(A_d)} \min_{a \in A_a} \sum_{d \in A_d} Q_d(s, d, a) \phi_d(s, d) \quad (6)$$

where $\Pi(A_d)$ is the set of all the probability distributions (mixed strategies) over the action set A_d of the defender.

The value of V_d and Q_d can be solved based on the generalization of the value iteration technique [19], which is omitted due to space limitation. The defender's maxmin strategy is already obtained when we calculate $V_d(s)$ for each state. We can define Q-value and V-value for the attacker in a similar way as Equation 5 and 6, and then compute its corresponding Nash equilibrium strategy.

Calculate the Best-Resonse Strategy in Markov Game

Given the estimated strategy of the attacker, the AMS strategy needs to compute its best-response strategy in the Markov game. The way of calculating the best-response strategy for the defender is similar to that of calculating its maxmin strategy in Markov game, except that it is based on the assumption that the attacker is employing the estimated strategy instead of minimizing the defender's payoff.

We first define the Q-value $Q_d(s, d, a)$ of the defender as its expected long-term value starting at state s by choosing action d (the attacker chooses action a), and the attacker and defender choose its estimated strategy ϕ_a and the best-response strategy against the estimated strategy of the attacker thereafter. The Q-value function is the same as Equation 5, except that $V_d'(s')$ is the long-term expected payoff of the defender if the attacker and defender choose its estimated strategy ϕ_a and the best-response strategy against the attacker respectively.

Next, the value of $V_d'(s)$ for any state s can be defined based on $Q_d(s, a, d)$ as follows,

$$V_d'(s) = \max_{\phi_d(s) \in \Pi(A_d)} \sum_{d \in A_d} \Big(\sum_{a \in A_a} Q_d(s, d, a) \phi_a(s, a) \Big) \phi_d(s, d) \quad (7)$$

where $\Pi(A_d)$ is the set of all the probability distributions (mixed strategies) over the action set A_d of the defender.

Based on the generalization of the value iteration technique, we can obtain the best-response strategy for the defender against the estimated strategy of the attacker by repeatedly updating the V-values and Q-values in Equation (5) and (7) respectively until convergence.

C. Properties of the AMS

It can be theoretically proved that the AMS satisfies all these properties (i.e., rationality, convergence and safety), which are formalized as the following three theorems:

Theorem 1. *(*Rationality*) Given a valid schedule of adjusting the parameters, if the attacker employs (or converges to) a fixed attacking strategy, the defender adopting AMS eventually converges to a best response to the attacker's strategy with probability one.*

Proof: (Sketch) This theorem can be proved by dividing it into two parts. First, we prove that with non-zero probability, the AMS will never restart based on the triangle inequality and Chebyshev's inequality theorem. Second, we prove that the probability that the AMS never restarts and does not converge to a best response strategy against

the attacker is 0 by continuity and Chebyshev's inequality theorem. By proving both parts, we can conclude that the AMS will converge to a best response against the attacker with probability 1. ■

Theorem 2. *(*Convergence*) Given a valid schedule, if both the defender and attacker employ the AMS, they eventually converge to a fixed strategy with probability one.*

Proof: (Sketch) We prove that the players converge to a (precomputed) NE strategy (i.e., a fixed strategy) with probability one. Similar to the proof of Theorem 1 and using the same technique, we prove this theorem by dividing it into two parts. First, we prove that with a positive probability, the AMSs for both players will never restart and are always within the first while-loop. Second, we prove that the probability that the AMS strategy never restarts but does not converge to equilibrium strategy is zero. We omit the details of the proof due to space constraint. ■

Theorem 3. *(*Safety*) Given a valid schedule, if the defender employs the AMS, its average overall payoff for each state in the limit is guaranteed to be no less than that obtained under the precomputed NE of the Markov game.*

Proof: (Sketch) If the attacker never converges to a fixed strategy, the defender adopting AMS would restart the outer repeat-loop for infinite number of times, and also stationary check (Line 17) would be satisfied eventually infinitely ($N^t \to \infty$). Thus AMS would choose the precomputed NE strategy infinitely. Based on the definition of a NE and the property of zero-sum game, its average overall payoff for each state in the limit will be no less than the corresponding payoff under the precomputed NE of the Markov game. ■

IV. Experimental Evaluation

In this section, we evaluate the AMS strategy using a variety of testbed networks and compare with both traditional approach based on the prediction of attack path [12] and the approach using NE strategy [16], [20]. The traditional approach based on attack path prediction determines the deployment strategy to cover all possible attacking paths as much as possible (equivalent as solving a graph coloring problem (GCP)) for each state based on the prediction. The NE-based approach adopts the NE strategy of the Markov game model as the defending strategy. The Markov game model is experimentally obtained following the definitions in Section II[1], and the NE strategy is computed following Equation 5 and 6. We denote these two approaches as GCP and NE respectively in the following descriptions.

The attacker's strategy is unavailable to the defender at the beginning of each round of attack. We assume that the attacker may employ any feasible Markov strategy (including any NE strategy). The attacker's strategy is generated randomly at the beginning of each run of the simulation

[1]No abstraction and action reduction technique are adopted since the testbed is relatively simple and thus unnecessary.

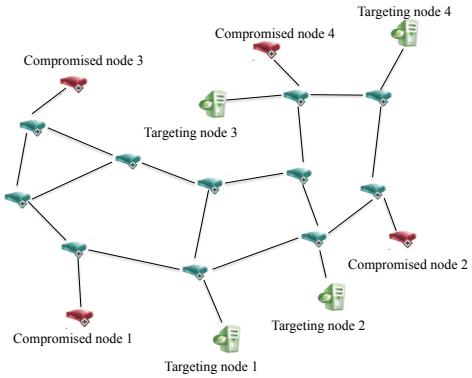


Figure 1: Network with 4 attacker controlled and targeting nodes

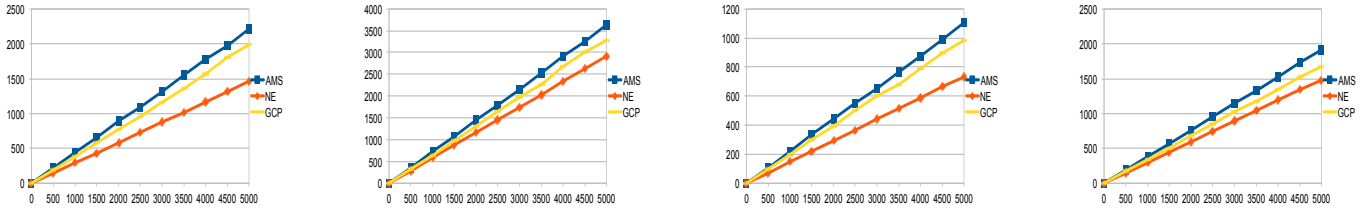
to model the diversity and unpredictability of the attacker's behaviors [11]. One representative testbed network is shown in Figure 1 inspired from the Abilene network[2], in which there are four target nodes (in green) and four compromised nodes (in red). Note that each compromised node may consist of a number of bots whose activities are synchronized, which can be abstracted as one node. In this testbed network, we assume that the defender can only deploy at most two detectors simultaneously in the network to not significantly degrade the network performance. We compare the performance of the above three strategies against the same attacker in two different scenarios in Section IV-A and IV-B respectively. For each pair of comparison, the results are obtained averaging over 20 networks with the same set of nodes as Figure 1 but randomly generated topologies.

A. Case 1: Single pair of attack

In this case, we assume that the attacker only launches attacks from one controlled node to any single targeting node each round. Without loss of generality, we assume that the damage to the network in case of successfully compromising any target node t is the same, and set $c_1(t) = 1$ and $c_2(t) = -1$. Both detection success rate p and attack success rate q are set to 1. For the defender, we consider two different cases: 1) it can only deploy one detector; 2) it can deploy two detectors. Figures 2a and 2b show the dynamics of the average overall payoff of the defender for the above two cases when it employs the AMS strategy, GCP and NE, respectively.

From Figures 2a and 2b, we can observe that the defender using AMS can always obtain the highest overall payoff with GCP defender ranking the second and the NE defender ranking last. For NE strategy, it works best only when the attacker actually employs the same NE strategy as it expects, while suffers for most cases when the attacker adopts any non-NE strategy. For GCP defender, it chooses to deploy detectors to cover as many attacking paths as possible based on its predication, thus its performance is better than NE strategy. However, it does not take into consideration

[2]http://abilene.internet2.edu



(a) Case 1: 1 detector deployment (p = 1, q = 1)

(b) Case 1: 2 detectors deployment (p =1, q = 1)

(c) Case 1: 1 detector deployment (p = 0.5, q = 1)

(d) Case 1: 2 detectors deployment (p = 0.5, q = 1)

Figure 2: Averaged overall payoff of the defender over rounds for different defending strategies

the *multi-stage* feature of the attacking strategy, thus its defending strategy may not be optimal in the long run. Also its strategy is fixed and not adaptive to the changes of the attack's strategy. Another observation is that better protection (higher overall payoff) can be achieved when the number of detectors available is increased from one to two for all three defending strategies.

Next we consider a more interesting case when detection becomes more difficult by setting detection success rate p to 0.5, with other parameters unchanged. In this case, the optimal selection of which nodes to place the limited detectors become more important. The performance comparison results are shown in Figure 2c and 2d respectively. Overall similar trends to previous cases ($p = 1$) can be observed: $AMS > GCP > NE$. However, compared with previous case ($p = 1$), two main differences are observed here. First, the expected overall payoff over rounds of the defender are decreased significantly. For the one detector case, given the same round, the defender's payoff is decreased by approximately 50%. Second, the payoff differences between the AMS defender and other two strategies are significantly increased: the AMS defender can obtain higher payoff than GCP and NE by approximately 20% and 40% respectively. These observations are reasonable since reducing detection success rate essentially is equivalent with reducing the number of effective detectors, and thus it is expected that the defender would suffer more from the attacks given the number of detectors unchanged. Besides, due to the decrease of effective detectors, the selection of strategic points becomes more critical to prevent target nodes from being compromised. Therefore, the AMS strategy's advantage becomes more obvious in this kind of situations.

B. Case 2: Double pairs of attacks

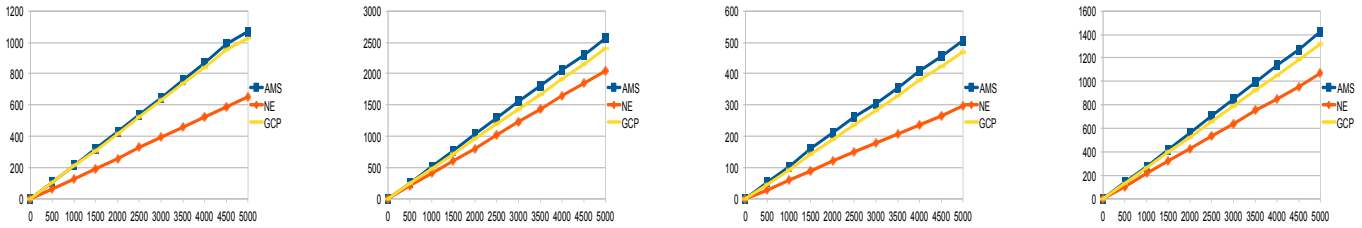
In this case, we assume that the attacker may launch attacks from two different compromised nodes simultaneously to any target node(s). For the players' payoff functions, we set $c_1(t) = 1$ and $c_2(t) = -1$. We first consider the case when both the detection success rate p and the attack success rate q are set to 1. Two different cases are considered for the defender where it can deploy either one detector or two detectors, and the results compared with previous approaches (GCP and NE) are shown in Figure 3a and 3b respectively. From both figures, we can observe that the AMS is able to achieve significantly higher overall payoff for the defender than both CGP and NE approaches. Again the GCP approach achieves second-best performance and the NE approach comes last. This observation is similar to what we found in Case 1 and can be explained in a similar way.

Next we further decrease the values of the detection success rate p to 0.75 while keeping the attack success rate unchanged. As explained in Section IV-A, the decrease of the detection success rate is in essence equivalent with reducing the number of effective detectors and thus makes the detector placement problem more difficult. Again two different cases are considered here in terms of whether the defender can deploy either one or two detectors. Figures 3c and 3d show the dynamics of the defender's expected overall payoff when it employs AMS, GCP, and NE respectively.

Similar observation as Section IV-A can be found here. First, given all other parameters unchanged, either decreasing the number of defenders or the detection success rate would reduce the expected overall payoff of the defender, and vice versa. Second, for the same number of detectors, reducing the detection success rate would increase the performance gap between the AMS and the other two approaches. For example, for the 1-detector deployment case with $p = 1$ and $q = 1$, by using the AMS, the defender's expected overall payoff by round 5000 is approximately 2% and 39% higher than that of the defender using GCP and NE respectively. In contrast, for the case of $p = 0.75$ while all other parameters unchanged, the defender's payoff using the AMS becomes 7% and 41% higher than that of the defender using GCP and NE.

V. Conclusion

In this paper, we tackle the intrusion detector placement problem by modeling the strategic interaction between the defender and the attacker as a Markov game, and then proposed the AMS strategy to dynamically determine the optimal detector deployment plan. Three desirable properties can be theoretically guaranteed: rationality, convergence and safety. Apart from its nice properties, the AMS strategy is also empirically shown to be more effective than the



(a) Case 2: 1 detector deployment (p=1, q = 1)

(b) Case 2: 2 detectors deployment (p=1, q = 1)

(c) Case 2: 1 detector deployment (p=0.75, q = 1)

(d) Case 2: 2 detectors deployment (p=0.75, q = 1)

Figure 3: Averaged overall payoff of the defender over rounds for different defending strategies

traditional attacking path prediction based approach and game-theoretic approach using NE strategy. Besides, from the practical deployment perspective, the AMS strategy only needs to recompute its strategy at certain time interval which is increased gradually, thus enables it to provide real-time deployment plans. As future work, more extensive evaluations on practical network testbeds will be performed to further evaluate the performance of the AMS strategy.

References

[1] Michael Bowling and Manuela Veloso. Convergence of gradient dynamics with a variable learning rate. In *Proceedings of ICML(2001)*, pages 27–34, 2001.

[2] Michael Bowling and Manuela Veloso. Rational and convergent learning in stochastic games. In *Proceedings of IJCAI(2001)*, volume 17, pages 1021–1026, 2001.

[3] David D Clark and Susan Landau. The problem isn't attribution: it's multi-stage attacks. In *Proceedings of the Re-architecting the Internet Workshop*, page 11. ACM, 2010.

[4] Vincent Conitzer and Tuomas Sandholm. Awesome: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *Machine Learning*, 67(1-2):23–43, 2007.

[5] Stanley C Eisenstat, Howard C Elman, and Martin H Schultz. Variational iterative methods for nonsymmetric systems of linear equations. *SIAM Journal on Numerical Analysis*, 20(2):345–357, 1983.

[6] Mohamed Elidrisi, Nicholas Johnson, and Maria Gini. Fast learning against adaptive adversarial opponents. In *Proceedings of AAMAS12*, 2012.

[7] Sugih Jamin, Cheng Jin, Yixin Jin, Danny Raz, Yuval Shavitt, and Lixia Zhang. On the placement of internet instrumentation. In *Proceedings of INFOCOM(2000)*, volume 1, pages 295–304. IEEE, 2000.

[8] Mohammad Hossein Manshaei, Quanyan Zhu, Tansu Alpcan, Tamer Bacşar, and Jean-Pierre Hubaux. Game theory meets network security and privacy. *ACM Computing Surveys (CSUR)*, 45(3):25, 2013.

[9] Sjouke Mauw and Martijn Oostdijk. Foundations of attack trees. In *Information Security and Cryptology-ICISC 2005*, pages 186–198. Springer, 2006.

[10] Jelena Mirkovic and Peter Reiher. A taxonomy of ddos attack and ddos defense mechanisms. *ACM SIGCOMM Computer Communication Review*, 34(2):39–53, 2004.

[11] Anupama Mishra, BB Gupta, and Ramesh Chandra Joshi. A comparative study of distributed denial of service attacks, intrusion tolerance and mitigation techniques. In *Proceedings of European Conference on Intelligence and Security Informatics Conference*, pages 286–289. IEEE, 2011.

[12] Steven Noel and Sushil Jajodia. Advanced vulnerability analysis and intrusion detection through predictive attack graphs. *Critical Issues in C4I, Armed Forces Communications and Electronics Association (AFCEA) Solutions Series. International Journal of Command and Control*, 2009.

[13] Xinming Ou, Wayne F Boyer, and Miles A McQueen. A scalable approach to attack graph generation. In *Proceedings of CCS(2006)*, pages 336–345. ACM, 2006.

[14] Rob Powers, Yoav Shoham, and Thuc Vu. A general criterion and an algorithmic framework for learning in multi-agent systems. *Machine Learning*, 67(1-2):45–76, 2007.

[15] Radware. White paper: Pre-attack planning causes successful dos/ddos attacks research brief. Technical report, Radware, Ltd, 2013.

[16] Stephan Schmidt, Tansu Alpcan, Şahin Albayrak, Tamer Başar, and Achim Mueller. A malware detector placement game for intrusion detection. In *Critical Information Infrastructures Security*, pages 311–326. Springer, 2008.

[17] Oleg Sheyner, Joshua Haines, Somesh Jha, Richard Lippmann, and Jeannette M Wing. Automated generation and analysis of attack graphs. In *Proceedings of IEEE Symposium on Security and privacy*, pages 273–284. IEEE, 2002.

[18] Y. Shoham and K. L. Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2009.

[19] Olivier Sigaud and Olivier Buffet. *Markov decision processes in artificial intelligence*. John Wiley & Sons, 2013.

[20] Quanyan Zhu and Tamer Basar. Dynamic policy-based ids configuration. In *Proceedings of the 48th IEEE Conference on Decision and Control, held jointly with the 2009 28th Chinese Control Conference.*, pages 8600–8605. IEEE, 2009.