SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2014; **7**:473–491
Published online 28 February 2013 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.730

RESEARCH ARTICLE

A fictitious play-based response strategy for multistage intrusion defense systems

Yi Luo[1,*†], Ferenc Szidarovszky[2], Youssif Al-Nashif[3] and Salim Hariri[3]

[1] Department of Systems and Industrial Engineering, University of Arizona, USA
[2] Department of Applied Mathematics, University of Pécs, Hungary
[3] Department of Electrical and Computer Engineering, University of Arizona, USA

ABSTRACT

The recent developments of advanced intrusion detection systems in the cyber security field provide opportunities to proactively protect the computer network systems and minimize the impacts of attackers on network operations. This paper is intended to assist the network defender find its best actions to defend against multistage attacks. The possible sequences of interactions between the attackers and the network defender are modeled as a two-player non-zero-sum non-cooperative dynamic multistage game with incomplete information. The players are assumed to be rational. They take turns in making decisions by considering previous and possible future interactions with the opponent and use Bayesian analysis after each interaction to update their knowledge about the opponents. We propose a Dynamic game tree-based Fictitious Play (DFP) approach to describe the repeated interactive decisions of the players. Each player finds its best moves at its decision nodes of the game tree by using multi-objective analysis. All possibilities are considered with their uncertain future interactions, which are based on learning of the opponent's decision process (including risk attitude and objectives). Instead of searching the entire game tree, appropriate future time horizons are dynamically determined for both players. In the DFP approach, the defender keeps tracking the opponent's actions, predicts the probabilities of future possible attacks, and then chooses its best moves. Thus, a new defense algorithm, called Response by DFP (RDFP), is developed. Numerical experiments show that this approach significantly reduces the damage caused by multistage attacks and it is also more efficient than other related algorithms.

KEYWORDS
fictitious play; decision making under uncertainty; intrusion defense

***Correspondence**
Yi Luo, University of Michigan Health System, USA.
E-mail: yiyiLuo@med.umich.edu
[†]Present address: University of Michigan Health System, USA

1. INTRODUCTION

With the wide deployment of web services and technologies in many areas covering all aspects of our life, the security of networks become critically important, and reducing or eliminating the negative effects of an intrusion is a fundamental issue of network security. There is a clear conflict between the intruders and the defenders. Because classic game theory is the usual methodology to examine decision making problems with multiple decision makers and conflicting interests, intrusion defense became one of the most important application fields of game theory. There are many examples of the applications of game theory in engineering, military, economics, and finance to mention only a few [1], but it has not been applied to the field of network security until the recent decade [2]. There are some limitations in applying it in this area, because the attackers' strategies are uncertain, intrusion steps are not instantaneous and the rules of the game might change dynamically and so on. However, game theory concepts are still the best mathematical tools to model the complex defense systems providing intuitions and best protection strategies.

We focus on multistage attacks that can last days, weeks, and even months. For example, the 'Conficker' computer worm, one of the newest botnets attacks, had infiltrated as many as 15 million machines around the world. One way it spreads is by infecting the USB thumb drives that carry data from one machine to the next, and it can exploit vulnerabilities in windows servers for several months [3]. In general, the attackers can start by stealing secrets and then use their access to destroy the same systems they have been exploiting, corrupt the backup files,

and bring the whole system down. Insider attacks are very difficult to detect and they are the most destructive and most difficult ones for the intrusion defense system to respond to, because the inner attackers, who have full knowledge and access to internal computer systems and data, use their internal privileges and intelligence to break the defense of the system. Existing techniques to respond against such attacks such as intrusion detection systems and firewall hardware/software systems are manual intensive. This makes them too slow to respond efficiently to complex and multi-stage network attacks. Furthermore, countermeasures such as signatures for intrusion detection systems cannot detect new types of attacks. Therefore, Autonomic Network Defense System (ANDS) as shown in Figure 1 is developed by our team at the University of Arizona to detect known or unknown network attacks based on anomaly behavior analysis of networks and applications and proactively prevent or minimize their impact on network operations and services in real time. The main modules to implement the proposed ANDS include online monitoring and filtering, multi-level behavior analysis, adaptive learning, risk analysis and protective action [4–6].

Our ANDS includes intrusion response subsystem (IRSS) and intrusion detection subsystem (IDSS), and both of them share the adaptive learning module as illustrated in Figure 1. The former is designed to find protective actions to defend against multistage attacks, and its performance depends on that of the latter. Although our IDSS is based on anomaly-based techniques that are known to be ineffective because of the large false alarms they produce, multi-level and fine-grain behavior analysis of each layer of the communications protocols is employed to overcome this limitation and produce high detection rates with low false alarms [4–6].

The objective of this paper is to introduce a fictitious play approach leading to a new and efficient response algorithm, which is employed by our IRSS to defend against attackers with minimal total losses to the system. As we know, the attackers' malicious goals are different in reality. After breaking the defense of network systems, the attacker might be able to obtain financial gain and confidential data, alter important information, destroy network control functions, make the entire network dysfunction, and sometimes just want to show off. The defender chooses appropriate responses to reduce the impacts of the attacks to the systems. In the meantime, both of them want to reduce the risk of their activities. On the whole, we assume that the attacker and the defender are rational and have enough resources to maximize their payoffs based on their own risk taking attitudes. The attacker's believed that potential impact and the defender's potential reward can be calculated on the basis of *risk and impact analysis*, and they are not necessarily the same. For example, if the defender places a 'honeypot' as an easy target, then it will be taken with the belief of a light impact without any actual damage. It also might supply valuable information to the defender about the attacker [7]. Because multistage attacks require repeated interactions between them, the consequence of the attacker's or the defender's decision also depends on the current and future decisions of the opponent. This situation can be modeled appropriately as a dynamic game. So the interaction between them can be considered as a two-player non-cooperative nonzero-sum dynamic multistage game with incomplete information, where the attacker is the leader and the defender is the follower. Their decision nodes, the possible attacks, and responses during their interactions can be represented by a *dynamic game tree*.

The attacker or the defender at each period chooses its best move at its decision node against the last selected strategy of the opponent. The repeated interactions of the players are similar to those of a two-player '*fictitious play*', because at different steps, different strategies and payoffs are used. However, it is not a classical fictitious play (CFP) algorithm [8] when the players want to find Nash equilibrium. In our case, a player at its decision node tries to assess the 'best' moves of the opponent and to maximize its payoffs sequentially and to choose its own best moves by considering previous and future possible interactions with the opponent. Therefore, our DFP approach is developed on the basis of the risk and impact analysis, the dynamic game tree, and the CFP. The main differences between our

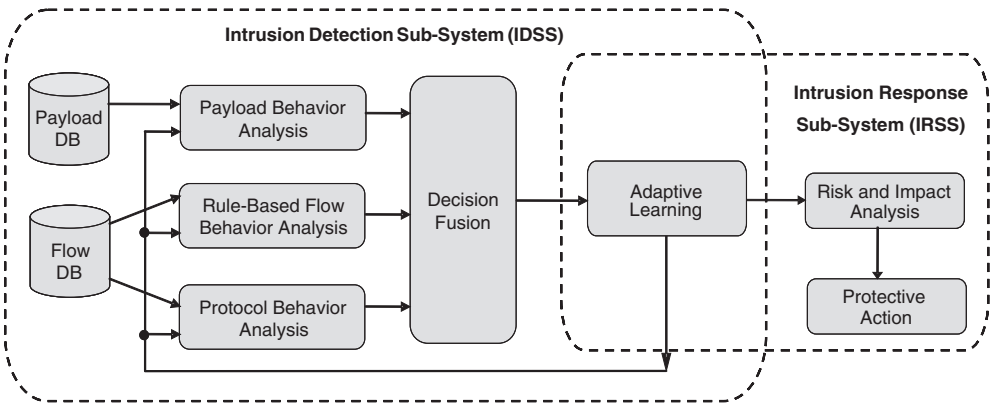

Figure 1. Autonomic network defense system.

Security Comm. Networks 2014; **7**:473–491 © 2013 John Wiley & Sons, Ltd.
DOI: 10.1002/sec

method and other classical approaches are discussed in Section 3. One of the important applications of the DFP is to develop a fast, efficient response algorithm RDFP for multistage intrusion defense systems.

In our ANDS, after the IDSS accurately identifies anomalous behavior in progress, the IRSS needs to evaluate changes in the network operations, update the knowledge about the attacker, analyze the two players' payoffs, and then choose the best action to defend against the detected attacks. Our RDFP algorithm is therefore designed to work online. The current advances in high performance computing, storage, and networks provided us with unprecedented computing capabilities, and they will make intrusion detection, adaptive learning, risk and impact analysis implementable in real time. For example, recently, IBM Watson Robot beat the world champion Jeopardy game. However, because of limited knowledge of the attacker's characteristic, objectives, and strategies in the dynamic game, the process of estimating the attacker's payoff to determine the best responses in the RDFP algorithm is very complex. For the sake of clear description and without loss of generality, we assume that the risk taking attitude of the intruder is risk neutral when introducing the details of our response algorithm in Section 4.

The rest of the paper is organized as follows. Section 2 proposes the basic ideas of the DFP approach and the RDFP algorithm. Section 3 contains a discussion on related works. The main steps and details of the RDFP algorithm are introduced in Section 4. General analysis and simulation results are presented in Section 5 to compare our approach to other known algorithms. Conclusions and future research directions are outlined in Section 6.

2. THE DYNAMIC GAME TREE-BASED FICTITIOUS PLAY APPROACH

2.1. Dynamic game tree

In the non-cooperative game, the interactions between the attacker and the defender can be represented by a dynamic Game tree (G) as shown in Figure 2. Each interaction between the players is equivalent to one time horizon or two stages. Let d be the index of the stages from the beginning of the game ($d = 1, 2\ldots$), and H_d or D_d a decision node of the attacker or the defender, respectively. The bold lines show an example of the first two possible interactions between the two players. The root of the tree is the initial decision node of the attacker. Because the attacker can run a script and launch one attack or several adversarial actions before the defender gets a chance to react, each arc originating at this root represents one of the possible combinations of these attacks (including a single attack). The nodes at the end point of each arc represent the decision nodes of the defender. Each arc starting from them at the second stage of the game tree denotes a possible combination of responses (including a single response) to defend against the previous attack(s). Then, the intruder makes the next move at its decision nodes. The possible attack scenarios are represented by the next layer of the graph and so on. In this way, dynamic game tree G is constructed on the basis of the analysis of all possible interactions between the two players. The tree structure continues to change according to current attack and defense activities along the game, and it terminates when the attacker gives up attacking or reaches its goals.

Generally, the value of a service is evaluated on the basis of the losses to the users due to the attacks when the service is completely not available. Then, the effort of the aforementioned intrusion or response combination can be measured by the damage or the reduced loss of the affected services. 'Intensity level' was employed by McGill *et al*. [9] to describe the impact of a threat on the attacker's target and to calculate the total loss resulting from the attack. Similarly, the intensity level is used in our paper to quantify the impact of each possible attack or defense combination on the network services. Assuming the relationship between them is linear, the latter can be calculated from the former and the value of the attacker's potential objective as it was demonstrated in our previous work [10]. Furthermore, the number of arcs originating from decision nodes of game tree G is determined by that of intensity levels. As we know, the increment of the number of intensity levels (arcs) can increase the accuracy of the players' best moves. However, the almost unlimited choices of the attacker's and the defender's activities lead to a very large game tree and make the computation of the best moves time consuming. Then, the accuracy of the best decision and the responsive time are balanced in

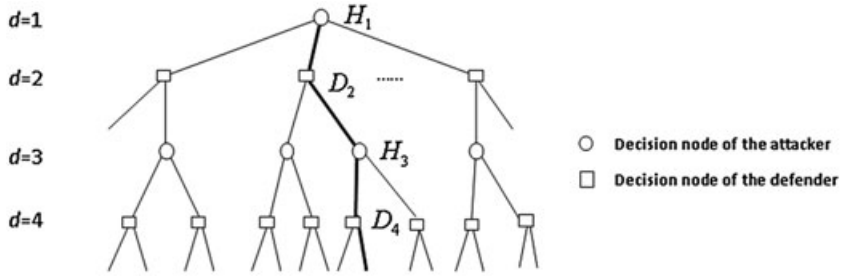

Figure 2. A dynamic game tree (G).

our DFP approach by classifying the intensive levels of the attack and the defense into appropriate categories and assigning them to the arcs originating from their decision nodes in G.

2.2. Risk and impact analysis

Any actions of the attacker damage the network, and any responses to reduce the impact of the attack will have a cost and may cause additional losses to the system. In an extreme case, the defender might block all traffics to a server that provides an important service if he/she detects an attack on the server. Although this leads to a big loss to the system, the attacker, on the other hand, has been blocked completely from his ability to launch attacks against the server. The sum of the damage/loss and the defender's cost is the outcome of the attack in the point of view of the attacker. This outcome is uncertain because of limited knowledge about the network, the objectives, and the possible future moves of the counterpart. Because of limited information about the types and objectives of the attacker, the outcome of the defender's response is uncertain as well.

The nature of the distribution of a random variable mathematically can be described by the central moments, where the first, second, and third central moments of a random variable are the mean, variance, and skewness, respectively [11]. Clearly, the characterization of a random variable becomes more accurate if higher moments are employed. However, the complexity of the computation process increases as well. In this paper, we assume that this uncertainty is modeled by considering the outcome of the attacker as a random variable $\widetilde{x}$. As it is well-known from the economic literature, the certainty equivalent of the random variable can be expressed by a linear combination of its expectation and variance

$$\hat{x} = E(\widetilde{x}) - \lambda \cdot Var(\widetilde{x}), \tag{1}$$

when we assume that larger value of $\widetilde{x}$ is better. Here, an important factor $\lambda \geq 0$ represents the risk attitude of the attacker. In general, $\lambda = 0$ is selected when the attacker does not consider risk at all (risk seeking) and $\lambda = \infty$ is selected when the attacker considers only risk (risk aversion). The value of λ assigned to the attacker is a fixed number between 0 and ∞. The certainty equivalent of the defender's outcome can be obtained similarly to Equation (1) by maximizing the expected reward of responses or minimizing the losses of the system while minimizing its uncertainty with its own risk attitude.

The attackers with different risk types may make different choices when facing the same options in the multistage game. It is better to consider the attackers' risk attitudes in modeling their behavioral decision making process. The factors to affect the attacker's decision making may include, but not limited to, the professional level of computer skills, intrusion experience, being inside or outside of the network system, and so on. All these elements can be described by the attacker's risk attitude in the DFP approach. For example, with the same computer technology level and the same previous intrusion experience, inside attackers are more risk averters. They may not accept risks and choose an action with as many as possible previous interactions because they do not want to be caught and to face all consequences. However, outside attackers pay less attention to their possible failures, because they can launch the attack again and again, so they are risk seekers.

In the DFP approach, the computation of expectation and variance of the impact at the attacker' decision nodes of game tree G is based on the total impact of attacks along the entire sequence of interactions of the two players and the probability of the occurrence of the impact. The latter can be obtained on the basis of the probabilities of the possible responses. The former includes the actual impacts, the costs of the responses within an appropriate time horizon, and all potential future impacts and costs afterwards. The computation of the actual impacts and costs has been studied in our earlier paper [10]. For the computation of the potential future impacts and costs, we consider different phases of the dynamic game based on the life cycle of the multistage attacks. On the other hand, the network defender needs to respond against each attack once it is detected and to minimize the expected loss of the system and the associated risks. The payoff function of the defender at its decision nodes can be formulated in a similar way as that of the attacker. Although the players may have limited information on the risk taking attitude of the opponent at the beginning of the game, they can predict the opponent's characteristic and use Bayesian analysis to update their estimations after each interaction between them. Then, the players can improve the knowledge of the opponent's activities in their future possible interactions and make the best current decision.

2.3. The response by dynamic game tree-based fictitious play algorithm

In our DFP approach, because both players consider all possible future interactions with the opponent at all of its decision nodes, any game tree rooted at any decision node of any player can be also considered a game tree of the player considering all possible interactions starting at this decision node. Therefore, the dynamic game tree can be decomposed into the game trees of the attacker and those of the defender starting at their decision nodes, respectively. The RDFP algorithm is developed on the basis of the DFP approach from the viewpoint of the defender to choose its best response at each of its decision nodes after each detected attack. The response payoffs are determined by the current and possible future interactions with the attacker. In order to find its best responses, the defender needs to estimate the opponent's payoffs from the game tree of the attacker. Because the dynamic game tree continuously grows unless the attacker gives up attacking or

reaches its goal, the sizes of the two players' game trees remain uncertain. In fact, the more interactions a player considers with its opponent, the more accurate payoff values can be obtained.

In our response algorithm, we assume that the time horizon considered by the attacker in the game depends on its risk taking attitude. In general, in order to make current decision, the risk seeker considers only few interactions with its opponent, but the risk averter would like to think about the whole process of achieving its goal. However, the defender has limited knowledge about the attacker's risk attitude at the beginning of the game. By assuming that the initial risk attitude of the attacker is risk neutral, the defender predicts the probability distribution of future possible attacks based on historical data and then implements its response. After detecting new intrusion activity, the defender employs Bayesian learning to update its knowledge of the attacker's risk attitude based on the discrepancy of the prediction and the observation. The size of the attacker's game trees can be adjusted dynamically as the defender updates its knowledge about the attacker along the game. On the other hand, the intrusion defense system in reality needs to respond to the most current attack as soon as possible and cannot wait for the decision until the defender finishes searching the opponent's and its own entire game trees. Therefore, appropriate time horizons are developed to find the trade-off between accuracy and the speed of solution, and they reduce the size of the defender's game trees that he/she needs to consider in finding best responses. One of our approaches to determine the time horizons is introduced in Section 4.4.

These reduced game trees are still uncertain because the players do not have complete information about the opponents. Even though the attacker's strategy sets and possible future moves can be obtained from the analysis of attack graphs and historical records, the defender does not know the probabilities of the possible future attacks. These probabilities can only be predicted from the attacker's reduced game tree. The probability distribution of the defender's future responses is also unknown to the attacker; however, we assume that the attacker is able to assess these probabilities from its observations during previous attacks. Because the defender also knows the probability distributions of its previous actions from its own records, the attacker's reduced game tree becomes a decision tree.

Therefore, the payoffs of the attacker can be obtained on the basis of this decision tree and the updated knowledge of the attacker. The defender is able to assess the probability distribution of all possible attacks at each of the attacker's decision nodes. Consequently, the reduced game tree of the defender becomes a decision tree as well, and the payoff values computed from this decision tree determine his/her best response. The details of the tree systems are introduced in Section 4. The convergence of the learning process can be examined on the basis of stability theory [12]. In the case of convergence, after several interactions of the players, their payoffs can closely approximate the optimal solution of the game.

3. RELATED WORKS

Network intrusion response mechanisms have been intensively studied in recent years due to their importance in cyber security. Some scholars design automated response mechanism to deploy the appropriate responses [13–16]. Considering the conflicting interests of the decision makers, game theoretical approach is used by many researchers to find strategies for both the defender and the attacker. Lye and Wing [17] view the interactions between an attacker and the defender as a two-player stochastic (Markov) game and construct the game model based on the analysis of the intrusion graph. The cost of any response is the amount of recovery effort (time) required by the defender. The reward of the attacker is defined in terms of the amount of effort the defender has to spend in order to bring the network back to normal state. The Nash equilibrium strategies are determined for both players by using nonlinear programming for infinite-horizon stochastic games and applying dynamic programming for finite-horizon games. These methods are also presented in [18,19]. The main disadvantage of the stochastic (Markov) model is that the full state space may become extremely large, and therefore, solutions for stochastic models are hard to compute. Markov game is also employed by other researchers, for example, by Shen *et al.* [20].

Several researchers notice that in real intrusion defense systems, the players cannot observe system states explicitly, they need to estimate the system states from observations related to current running services, data, and so on and then a partially observable Markov decision process (POMDP) can be used to model the attacker's actions. The research works presented in [21–25] belong to this category. The application of POMDP is a more realistic approach to capture the probabilistic nature of the state transition relations. This methodology has to consider a very large system and consequently it makes the computation of best responses more complicated and more expensive. Liu *et al.* [26] introduce a general incentive-based method to model attacker intent, objectives and strategies (AIOS) and a game theoretic approach to implement AIOS. They also mention that in cases of high correlation among the attacker's actions, the application of dynamic, multistage games is more appropriate in comparison with the combination of probabilistic 'attack states' and stochastic game models, because they are cheaper, more accurate, and have smaller search space, and there is no need to know the complete state space of the attack.

The fictitious play approach is employed by some scholars to describe the interactions between the attacker and the defender. Alpcan and Basar [27] model the attack–defense game by using stochastic fictitious play (STFP). They consider the expected payoffs of the players and use an entropy term to randomize the players' strategies in their utility functions. It can be interpreted as a way of concealing their true strategies, and a coefficient is assigned to represent the willingness of the players to randomize their own actions. If these coefficients are zero, then the security game reduces

to the CFP. If these coefficients are larger than zero, then the game leads to the STFP. They also discuss the Bayesian security game. Buttyan and HuBaux [28] describe a distributed sample fictitious play (DSFP) algorithm in which each player tries to improve its payoff in each step. If the players' responses are close to the Nash equilibrium strategies, then the payoffs remain close to the Nash equilibrium payoffs as well.

As the continuation of our previous work [10], the DFP approach combines the advantages of the dynamic multistage games and the CFP. A game tree is constructed to show the repeated interactions between the attacker and the defender. Appropriate time horizons are developed to reduce the size of the attacker's and the defender's game trees. The potential final impact and potential final loss are also considered in the payoffs. The milestones in the attack's life cycle are introduced as different phases. While each attack and responsive action transform the state of the system in the stochastic (Markov) game, the system cannot reach the next phase until the attack and the responsive action meet the requirements of the next phase. Using phases can largely reduce computation complexity compared with the use of all possible attack states, because the number of phases is much less than that of the possible attack states, and the potential impact or loss needs to be updated only after reaching the next phase.

Our approach is different from the STFP and the DSFP, because the players in a multistage game consider their possible future interactions, they are rational, strategic, and can predict the probabilities of the opponents' future activities based on an updating procedure. The DFP is similar to the extensive forms of Bayesian games; however, Nash equilibrium strategies are not determined at the decision nodes of the players due to the following reasons. First, it is time consuming to find the Nash equilibrium by considering all possible future interactions between the players, whereas the intrusion defense system needs fast responses. Second, the players still cannot find unique decision if there are more than one pure or mixed Nash equilibria. Furthermore, in addition to considering the expected rewards, profits, or gains as the player's objectives as it is always the case in the literature, our approach also considers the player's willingness to take risk, which is a more realistic approach to model the player's decision making process. Therefore, a new and efficient response algorithm RDFP is developed on the basis of the DFP approach. Suppose an intruder with a neutral risk attitude launches multistage attacks, the best move to defend against the attacker is calculated on the basis of the RDFP algorithm as will be shown in the next section.

4. DEFENDING AGAINST A RISK NEUTRAL ATTACKER

4.1. The tree system of the attacker

In our RDFP algorithm, tree G can be decomposed into the game trees $GA(H_d)$ of the attacker with roots H_d and the game trees $GD(D_d)$ of the defender with roots D_d. The processes of obtaining $GA(H_d)$ and $GD(D_d)$ are indicated by line 1 of procedures FIND_$RDA(H_d)$ and FIND_$RGD(D_d)$ as shown in Figures A1 and A2, respectively. The attacker chooses its moves at its decision nodes H_d to maximize its payoffs by considering all future interactions with the defender on its game tree. The payoff of the attacker cannot be obtained directly from the $GA(H_d)$ because he/she has limited information about the possible responses of the defender.

In a $GA(H_d)$, let AA be the set of the possible attack combinations at root H_d, a be the index of these scenarios ($a \in AA$), and A be the total number of elements of set AA. Then $GA(H_d)$ can be further decomposed into A sub-game trees $SGA(H_d,a)$ with common root H_d starting with different attacks a. The depth of $SGA(H_d,a)$ is related to the number of future interactions of the two players considered by the attacker. This is however unknown to the defender. On the basis of our assumptions that the future time horizon that the player considers depends on its risk attitude and the risk type of the attacker is risk neutral, the risk type λ of the attacker and the number of future interactions τ that the attacker considers on his/her game tree with the defender in making decision are known. Tree $SGA(H_d,a)$ becomes the reduced sub-game tree $RSGA(H_d,a,\tau)$ of attack a with depth τ. The reduced game tree $RGA(H_d)$ of the attacker consists of all $RSGA(H_d,a,\tau)$ sub-game trees with root H_d. The process of obtaining $RGA(H_d)$ is indicated by line 2 of procedure 'FIND_$RDA(H_d)$' as shown in Figure A1.

Consider a subtree $RSGA(H_d,a,\tau)$. Let m be the index of the interactions during time horizon τ ($m = 1, 2,\ldots,\tau$). Let K_a be the total number of paths originating at attack a to the endpoints of $RSGA(H_d,a,\tau)$($a = 1, 2,\ldots,A$), and k_a be the index of these paths (k_a=1, 2,. . .,K_a). In addition, let $p_{k_a}^{d+2m-1}$ be the probability that the defender chooses its response on path k_a at stage $d+2m-1$. It is assumed that the attacker is able to assess it based on previous attacks and its knowledge of the defender's risk attitude. Because the defender is also able to obtain this value from its own records, probability $p_{k_a}^{d+2m-1}$ can be estimated as shown in line 6 of procedure 'FIND_$RDA(H_d)$' (Figure A1). Thus, tree $RSGA(H_d,a,\tau)$ becomes the reduced sub-decision tree $RSDA(H_d,a,\tau)$ starting at attack a. All sub-graphs $RSDA(H_d,a,\tau)$ from root H_d constitute the attacker's reduced decision tree $RDA(H_d)$. The process of finding $RDA(H_d)$ is shown in Figure A1. The notations of the above described trees and sub-trees are listed in Table I, and their relationships are illustrated in Figure 3.

4.2. The impact of attacks in tree $RDA(H_d)$

The cost function to defend against a multistage attack includes three parts: actual damage AI, defensive cost RC, and potential damage and cost PI. As we mentioned before, although defensive action can reduce the damage caused by the attacks, the defensive action itself, such as

Table I. Notations of the dynamic tree systems.

Symbols	Definitions
G	The dynamic game tree
d	The index of the stages of tree G
H_d	The decision node of the attacker at stage d in tree G
D_d	The decision node of the defender at stage d in tree G
$GA(H_d)$	The game tree of the attacker at root H_d in tree G
$GD(D_d)$	The game tree of the defender at root D_d in tree G
$SGA(H_d,a)$	The sub-game tree of $GA(H_d)$ starting with attack a
$SGD(D_d,r)$	The sub-game tree of $GD(D_d)$ starting with response r
$RSGA(H_d,a,\tau)$	The reduced sub-game tree of $SGA(H_d,a)$ with depth τ
$RSGD(D_d,r,\pi)$	The reduced sub-game tree of $SGD(D_d,r)$ with depth π
$RSDA(H_d,a,\tau)$	The reduced sub-decision tree of $SGA(H_d,a)$ with depth τ
$RSDD(D_d,r,\pi)$	The reduced sub-decision tree of $SGD(D_d,r)$ with depth π
$RGA(H_d)$	The union of reduced sub-game trees $RSGA(H_d,a,\tau)$ with root H_d
$RGD(D_d)$	The union of reduced sub-game trees $RSGD(D_d,r,\pi)$ with root D_d
$RDA(H_d)$	The union of reduced sub-game trees $RSDA(H_d,a,\tau)$ with root H_d
$RDD(D_d)$	The union of reduced sub-game trees $RSDD(D_d,r,\pi)$ with root D_d

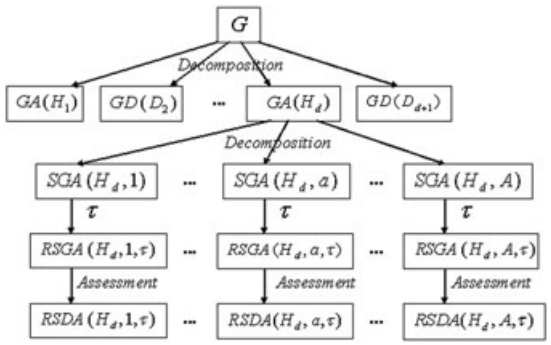

Figure 3. The tree system of the attacker.

blocking traffic on the router, will lead to additional losses to the system as well. Then, the actual damage includes the reduced impact of the attack and the additional losses due to defensive activity. In the meantime, the defender needs to use the resources of the system to defend against the attack which results in additional defensive cost. Moreover, the potential damage and cost is determined by the value of potential targets and the current phase in the life cycle of multistage attack. Considering τ interactions between the attacker and the defender, the total impact y_{k_a} of the attack at the endpoint of path k_a is the sum of all impacts $AI_{k_a}^{d+2m-1}$, costs $RC_{k_a}^{d+2m-1}$ of the responses along the path and the potential final impact/cost $PI_{k_a}^{d+2\tau}$ of the attack beyond the time horizon. That is,

$$y_{k_a} = \sum_{m=1}^{\tau}\left(AI_{k_a}^{d+2m-1} + RC_{k_a}^{d+2m-1}\right) + PI_{k_a}^{d+2\tau} \quad (2)$$
$$(a = 1, 2 \ldots A, k_a = 1, 2, \ldots, K_a).$$

The definition and computation of the values of the actual impacts and costs can be found in our earlier paper [10]. In order to estimate the potential final impact of the attack, the phases of the dynamic game are developed as follows. Multistage attacks include reconnaissance, penetration, attack, and exploit in their life cycles [29]. In general, each phase indicates a progress of the multistage attack to reach its objective. Higher phase indicates that the attacker is closer to its objective. Each phase consists of a sequence of interactions between the attacker and the defender. That is, the attacker–defender game cannot reach the next phase until the actions of the two players meet its requirements. All phases of the game can be analyzed and classified on the basis of the interactions of the players. Multistage attacks involving different vulnerabilities of the network system could have different numbers of phases, and any step of interaction of the players can belong to different phases. Let O be the number of possible objectives of the attacker in the network system, o be the index of these objectives ($o = 1, 2, \ldots, O$). Furthermore, let F_o be the total number of phases of the dynamic game in reaching objective o, and f_o the index of these phases ($f_o = 1, 2, \ldots, F_o$). Assume that C_o is the value of objective o. Define the binary variable b_{f_o} as $b_{f_o} = 1$ if the attacker–defender game to objective o reaches phase f_o, and $b_{f_o} = 0$ otherwise. On the basis of the interactions of the players in each phase, the value of b_{f_o} is updated after the game enters the next phase. The updating process is represented by line 9 in the formal description of the RDFP algorithm (Figure 4). The potential final impact PI of the attack clearly depends on the progress in reaching the objectives of the attacker. In general, it can be calculated from equation

$$PI = \sum_{o=1}^{O}\sum_{f_o=1}^{F_o}\frac{f_o}{F_o}b_{f_o}C_o. \quad (3)$$

The potential losses PL of the system can be computed in a similar way. The impact of the attack on any objective may include different impact categories such as economic,

INPUT:
Tree G, λ, d=1.
Step-1:
1: if detect_an_attack then
2: Set d=d+1, and then go to Step-2;
3: else
4: STOP.
5: end if
Step-2:
6: FIND_$RGD(D_d)$;
7: FIND_$RDD(D_d)$;
8: FIND_r^* at node D_d;
9: UPDATE_b_{f_0} at node D_d;
10: Set d=d+1, and then go to Step-1.

Figure 4. The response by dynamic game tree-based fictitious play algorithm.

reputation, reliability, safety, and social effect. Then, it is very important to consider all of these impact categories to find the correct values of the objective functions of the players [30]. For example, if the network system cannot provide any service at all when the attacker realizes its objective, then the value of the objective function can be obtained on the basis of the impact of losing the services. Therefore, in computing C_o, we have to consider the values of these impacts and their corresponding weights in all impact categories. Let J_o be the total number of impact categories of objective o, and j_o be the index of these categories ($j_o = 1, 2, \ldots, J_o$). For objective o, pair-wise comparisons are conducted to indicate the relative important factors of the categories. This is a well-known procedure from multi-criteria decision making [31]. The weight W_{j_o} of each impact category can be obtained by using, for example, the analytic hierarchy process [32]. Because the measures of the impacts in different categories are usually given in different units, they have to be transformed into unitless values by using appropriate utility functions. The values of some impact categories such as reputation and safety need to be evaluated from historical data and by the assessment of a group of experts. If G_{j_o} is the transformed measure of impact in category j_o when the attacker realizes objective o, then the evaluation of objective o can be obtained as the weighted average

$$C_o = \sum_{j_o=1}^{J_o} W_{j_o} G_{j_o}. \qquad (o = 1, 2 \ldots O) \qquad (4)$$

The evaluation of the impact of the attacks is a standard process that is applied to evaluate risk and impact against any enterprise assets (physical, logical, and data resources) that can be targeted by malicious threats. Our cost functions will utilize these techniques to quantify the damage that can occur to the company assets or services.

4.3. The probabilities of future attacks at the root of tree *RDA*(H_d)

As previously mentioned, the probability distribution of possible defensive actions in the attacker's decision tree $RDA(H_d)$ can be estimated by the defender's own records. The probability that the interaction of the players will follow a path k_a in the tree is given as

$$Q_{k_a} = \frac{\prod_{m=1}^{\tau} p_{k_a}^{d+2m-1}}{Q_a} \qquad (a = 1, 2 \ldots A, k_a = 1, 2 \ldots K_a) \qquad (5)$$

with $Q_a = \sum_{k=1}^{K} \prod_{m=1}^{\tau} p_{k_a}^{d+2m-1}$ being a normalizing constant. Let $E_{k_a}^{H_d}$ be the expected value of the impact of a particular path k_a starting at decision node H_d of $RDA(H_d)$. It can be obtained as

$$E_{k_a}^{H_d} = y_{k_a} Q_{k_a}, \qquad (a = 1, 2 \ldots A, k_a = 1, 2 \ldots K_a) \qquad (6)$$

where y_{k_a} can be obtained similar to Equation (2). Let KK_a be the set of all paths starting at attack a in tree $RDA(H_d)$. If

$$\max_{k_{a'} \in KK_{a'}} \left\{ E_{k_{a'}}^{H_d} \right\} \leq \min_{\substack{k_a \in KK_a \\ a \neq a'}} \left\{ E_{k_a}^{H_d} \right\}, \qquad \left(a', a \in AA \right) \qquad (7)$$

then attack a' will not be chosen by the attacker because this attack is dominated by others, and it is eliminated from set AA. This step is indicated in lines 12 and 13 of procedure 'FIND_$RDD(D_d)$' (Figure A3). Let AA' be the subset of AA where attack a' is not included. The expected value of the total impact of all future paths starting at attack a can be obtained as

$$E_a^{H_d} = \sum_{k=1}^{K} E_{k_a}^{H_d} = \sum_{k=1}^{K} y_{k_a} Q_{k_a}, \qquad \left(a \in AA' \right) \qquad (8)$$

The variance of the impact is similarly

$$V_a^{H_d} = \sum_{k=1}^{K} y_{k_a}^2 Q_{k_a} - \left(E_a^{H_d} \right)^2. \qquad \left(a \in AA' \right) \qquad (9)$$

The certainty equivalent of this random impact is obtained from Equation (1) as

$$U_a^{H_d} = E_a^{H_d} - \lambda \cdot V_a^{H_d}. \qquad \left(a \in AA' \right) \qquad (10)$$

After calculating $U_a^{H_d}$ for all possible attacks at the root H_d of $RDA(H_d)$, the probability of the occurrence of attack a at this decision node can be obtained from relation

$$pp_a^{H_d} = \frac{U_a^{H_d}}{\sum_{a=1}^{A} U_a^{H_d}}. \qquad \left(a \in AA'\right) \quad (11)$$

4.4. The best response at the defender's decision node

As we mentioned before, the defender wants to optimize its payoffs by selecting its best moves to minimize the total costs and losses to the system and the risk of its responses. Suppose an attack is detected at node H_{d-1} of tree G, and the defender with risk attitude μ needs to choose its best move at decision node D_d. Let R be the total number of the possible response combinations at root D_d, RR be the set of these strategies, and r be their index ($r = 1, 2,\ldots,R$). Tree $GD(D_d)$ can be decomposed into a set of sub-game trees $SGD(D_d,r)$ at root D_d based on the different responses r. In order to protect the important targets in the network, the defender is assumed to be a risk averter in defending against the attacker. Starting from response r, the defender predicts the potential objectives of the attacker, uses the breath-first search (BFS) to find the 'shortest path' with length π where the attacker is able to achieve its objectives with the least number of interactions in $SGD(D_d,r)$ resulting in the reduced sub-game tree $RSGD(D_d,r,\pi)$ of response r. The application of the BFS algorithm is shown in line 3 of procedure 'FIND_$RGD(D_d)$' in Figure A2. As we know, both space and time complexities in breath-first search are exponential in the number of arcs (stages or time horizons) of the path to reach the attacker's goals in the game tree. Because the defender needs to find the probability distribution of possible future attacks at each of the attacker's decision nodes in tree $RSGD(D_d,r,\pi)$, the assumption of risk neutrality for the attacker can reduce the space and time complexities of the RDFP algorithm while maintaining a good prediction of the attacker's future possible activities. The union of the sub-game trees $RSGD(D_d,r,\pi)$ with all responses r constitutes the reduced game tree $RGD(D_d)$ of the defender. The process of finding $RGD(D_d)$ is indicated by line 6 in the formal description of the RDFP algorithm (Figure 4).

In a $RSGD(D_d,r,\pi)$, let n denote the indices of the interactions ($n = 1, 2,\ldots,\pi$), let K_r be the total number of paths originating at response r to the endpoints of $RSGD(D_d,r,\pi)$, and let k_r denote the indices of these paths ($k_r = 1, 2,\ldots,K_r$). In addition, let H_{d+2n-1} be the decision node of the attacker at stage $d+2n-1$ of $RSGD(D_d,r,\pi)$. The probability $p_{k_r}^{H_{d+2n-1}}$ that the attacker selects its attack at a decision node H_{d+2n-1} on path k_r can be predicted by the defender from Section 4.3 based on previous interactions and its knowledge of the attacker's risk attitude. After obtaining the probability values $p_{k_r}^{H_{d+2n-1}}$ at every decision node of the attacker on the corresponding tree $RDA(H_{d+2n-1})$, the reduced sub-decision tree $RSDD(D_d,r,\pi)$ of response r is found. The union of all sub-decision trees $RSDD(D_d,r,\pi)$ at root D_d constitutes a reduced decision tree $RDD(D_d)$ of the defender. The process of finding $RDD(D_d)$ is indicated by line 7 of the RQFP algorithm (Figure 4). If the defender obtains its reduced decision tree, then it can compute the payoffs to all possible responses and is able to select its best response at the root. The notations of these trees are listed in Table I, and their relationships are similar to those of the attacker shown earlier in Figure 3. Then, the defender is able to compute the probability Q_{k_r} that the players will follow path k_r and the total loss y_{k_r} of the network at the endpoint of this path using equations similar to (5) and (2), respectively. The expected loss $E_{k_r}^{D_d}$ of the network along path k_r can be calculated similarly to Equation (6). Let KK_r be the set of all paths starting from response r in $RDD(D_d)$, and let $k_r \in KK_r$ be a path. If

$$\min_{k_{r'} \in KK_{r'}} \left\{E_{k_{r'}}^{D_d}\right\} \geq \max_{\substack{k_r \in KK_r \\ r \neq r'}} \left\{E_{k_r}^{D_d}\right\}, \qquad \left(r', r \in RR\right) \quad (12)$$

then response r' is not considered as a candidate of the best response because this response is dominated by others. The aforementioned process is described by lines 8 and 9 in Figure A4. Let RR' be the subset of responses RR where response r' is eliminated. For each response r, the defender computes expectation $E_r^{D_d}$ and variance $V_r^{D_d}$ of the sum of losses, response costs, and potential final losses similarly to (8) and (9), respectively, on the basis of the tree $RDD(D_d)$. The certainty equivalent in this case can be given as

$$U_r^{D_d} = E_r^{D_d} + \mu \cdot V_r^{D_d}, \qquad \left(r \in RR'\right) \quad (13)$$

because the defender wants to minimize the total losses and costs and the risk of its responses.

The responsive action with the lowest certainty equivalent is selected as the next move of the defender at root D_d:

$$r^* = \arg\min_{r \in RR'} U_r^{D_d}. \quad (14)$$

5. COMPARISONS OF THE RESPONSE BY DYNAMIC GAME TREE-BASED FICTITIOUS PLAY AND RELATED ALGORITHMS

5.1. General analysis

By employing the RDFP algorithm to defend against the attack to a given computer network, the defender needs to analyze possible multistage attacks in the attacker–defender game in advance and to keep tracking the characteristics of the attacker after each of their interactions. It is assumed that the defender had been professionally trained to protect the network system and accumulated many years of intrusion defense experience, and therefore he/she can evaluate the values of the services based on the demands of the users, can identify the vulnerabilities of the computer

network and is able to analyze the life cycle of multistage attacks based on the interactions of the attacker and the defender in historical records. Then, the defender is also able to quantify the intensity levels of possible attack and defense activities, predict the attacker's possible future attacks and strategy sets, develop dynamic game tree G to describe their interactions, and find the cost function for each possible attack–defense scenario. Clearly, the aforementioned analysis can be conducted by the defender offline before the beginning of the attack.

Once a known multistage attack is launched, the defender's best responses are determined by its knowledge of the opponent's risk attitude. As mentioned previously, Bayesian learning can be used by the defender to update this knowledge based on the discrepancy of the detected and the predicted attacks. If the intruder changes its behavior significantly; for example, launching a new multistage attack to break or bypass the defense system, then the attacker's activities may not appear in the predicted attack strategy sets. Our intrusion detection system, which is based on anomaly behavior can detect such type of attacks and consequently help the defender predict the intruder's malicious goals. In reality, the defender would not allow their network to be attacked in order to learn the unknown attacks, and then placing 'honeypots' is a good approach to learn about the opponent's computer skills, intrusion experience, and location in the network after several iterations of the game. Thus, our RDFP algorithm can help the defender find its best responses to defend against the unknown attacks. Afterwards, the attacker's activities are recorded in the defender's database for future attack assessment.

It is assumed that there are multiple vulnerabilities in the computer network system, and the attacker tries to launch multistage attack with objectives unknown by the defender. The formal description of the RDFP algorithm is described in Figure 4. For simplicity of describing our approach, the risk attitude of the attacker is assumed to be risk neutral. The figure illustrates the offline version of the RDFP algorithm, where the special procedures shown in lines 6, 7, and 8 are described in the Appendix.

After presenting the details of the general RDFP algorithm, two special method variants are described. One is a Greedy (GA) algorithm in which the defender completely blocks the traffic of the corresponding services on routers, firewall, or disconnects the machines by using managed switches regardless of the type and the intensity level of the attack. The other algorithm is myopic, called the single-interaction optimization (SO) algorithm in which at each interaction, the defender tries to minimize the total costs and losses from the most current attack without considering future actions.

Compared with the GA and SO algorithms in different computer network systems, the RDFP algorithm has its advantages and disadvantages. First of all, the defender employs the GA and SO algorithms to protect the network system mainly based on of its observations, and their implementation costs are very low compared with that of intrusion detection system that can detect and monitor the known and unknown multistage attacks to the computer network. Secondly, because responsive actions are subdivided into different intensive levels to capture future possible interactions with the attacker and they can be allocated to multiple stages in the dynamic game, the defender has more responsive actions and sequences to select within the structure of the RDFP algorithm. On the other hand, the defender can only choose limited responsive actions and time horizons in the GA and the SO algorithms. Thirdly, in order to defend against an identical attack, the total cost and damage of the computer network by employing the RDFP algorithm is not larger than that of using the GA algorithm or the SO algorithm because the strategy set of the latter is a subset of the former. Fourthly, because the two players are assumed to be rational and try to maximize their potential impacts in a non-cooperative game, the best response based on the RDFP algorithm is obtained from minimizing the damage/loss of the system due to the attack in the worst case scenario. If there are no valuable targets in the network system, the RDFP algorithm has no advantage over the SO algorithm due to a little damage reduction and a more expensive detection cost. The simulation study of how to defend against a multistage attack with different algorithms is conducted in the next subsection.

5.2. Simulation study

All multistage attacker–defender games consist of tedious and complicated processes in real computer network systems. In this section, numerical experiments are reported by simulating how the defender uses the RDFP, GA, and SO algorithms to defend against multistage attacks. In order to explain and illustrate the fundamentals of the new algorithm, the scenarios are created simply from the distributed denial of service (DoS). It is a typical botnets attack that involves multiple sites and may last for a long time. The basic structure of the network system is shown in Figure 5, which includes an inner attacker in Subnet 1, chief executive officer (CEO) and secretary in Subnet 2, marketing and accounting managers in Subnet 3 and two clients in external network. It is assumed that the information in the CEO computer ($o=1$), the important data in the accounting system ($o=2$), the HTTP server ($o=3$), Database 2 ($o=4$), the FTP1 server ($o=5$) are the possible objectives of the attacker in the network system. As previously stated, the value of a service is defined by the loss to the users due to the attacks when the service becomes unavailable. Being the important parts of the network systems, information and data are usually the targets of a multistage attack, and consequently, the impact of losing them is huge on system users. In order to simulate these real situations and compare the performances of three responsive approaches, we created the values of the system services for different types of users as shown in Table II.

In our simulation study, it is assumed that the attacker with a neutral risk attitude employs different strategies to launch multistage attacks to achieve its goal and to minimize the uncertainty in the consequences of its actions.

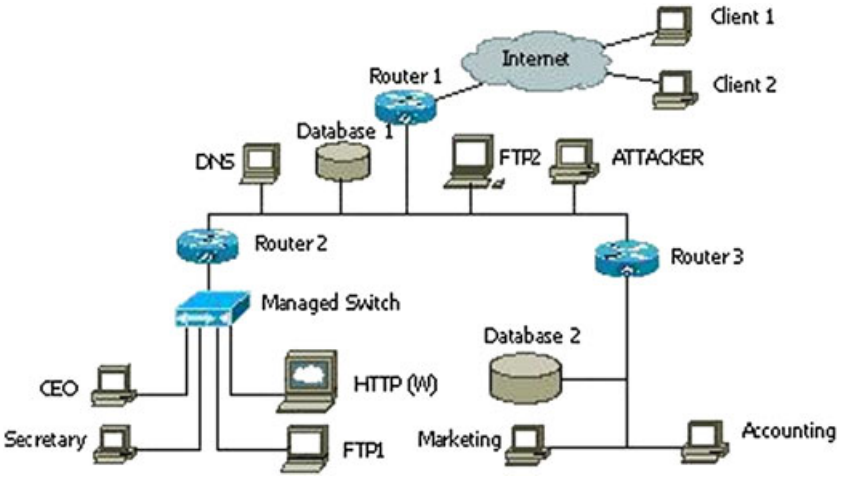

Figure 5. Network structure.

Table II. The values of the services in the computer network.

Subnet 2			Subnet 3			External network		
Users	Services	Values	Users	Services	Values	Users	Services	Values
CEO	HTTP (Subnet 2)	40	Marketing	HTTP (Subnet 2)	30	Client 1	HTTP (Subnet 2)	24
	HTTP (Outside)	15		HTTP (Outside)	30		HTTP (Outside)	40
	Database 2	1000		Database 2	2000			
	Important Info	100,000	Accounting	HTTP (Subnet 2)	20	Client 2	HTTP (Subnet 2)	30
	FTP1	50		HTTP (Outside)	10		HTTP (Outside)	40
Secretary	HTTP (Subnet 2)	20		Important Data	100,000		Database 2	2000
	HTTP (Outside)	40		Database 2	2000			

In the meantime, the defender does not know the objectives of the attacker when the game starts. The life cycle of the multistage attacks is defined from the beginning of the attack to the stage where the attacker reaches his/her objectives or stops attacking due to the responsive actions. The possible interactions between the two players on game tree G are developed on the basis of the attack graphs and the historical data. The probability distribution of the defender's future responses in the reduced game tree of the attacker is assumed to be uniform.

Both the intruder and the defender are assumed to employ one action or move at each stage for the sake of simplicity. Each action is measured by its impact on the target services with certain intensity levels, which could be 0 or 1 or a certain number between 0 and 1. The first and the second situations can be interpreted as the players select to do nothing and choose to take an action, respectively. The numbers in the third case are roughly classified into simple categories in our simulation study. For instance, weak or strong attack level is used to describe that the intruder attacks a server with 10%–30% or 70%–90% intensity levels. Similarly, low or high blocking level is employed to indicate that the defender blocks 10%–30% or 70%–90% of traffic to the server. Table III shows only their representative activities based on the network structure. Two main scenarios are introduced in the following subsections.

5.2.1. Scenario one: multistage attacks with one objective.

In this scenario, the attacker's main objective is to access the CEO computer to obtain information ($o = 1$) with value C_1. That is, the attacker has one objective ($O = 1$). On the basis of Table II, the CEO computer needs services provided by the HTTP server, Database 2, and the FTP1 server. The attacker can launch multistage attacks to access the information from the CEO computer in many different ways. The phases f_1 of the attacker–defender game can be generally described on the basis of the dynamic tree G as follows.

After any attack is detected either to Database 2 or to the HTTP server or to the FTP1 server at a node H_d, the attacker–defender game to the information source reaches phase one (that is, $f_1 = 1$). If Database 2 is attacked with strong intensity level ('A_DB2_S') and the defender chooses to block its traffic on Router 3 with high block-level ('B_DB2_H_R3') at node D_{d+1}, then the game reaches phase two (that is, $f_1 = 2$). Assume that the intruder attacks the HTTP server (or the FTP1 server) with strong

Table III. The possible actions of the attacker and the defender in the computer network.

Attacks				Responses			
Indices	Contents	Indices	Contents	Indices	Contents	Indices	Contents
*A*1	A_DB2_S	*A*11	Act_HTTP_CEO	*B*1	B_DB2_H_R3	*B*11	B_FTP1_L_R2
*A*2	A_DB2_W	*A*12	Act_FTP1_CEO	*B*2	B_DB2_L_R3	*B*12	Disc_HTTP
*A*3	A_HTTP_S	*A*13	Act_HTTP_Accounting	*B*3	B_HTTP_H_R3	*B*13	Disc_FTP1
*A*4	A_HTTP_W	*A*14	Inject_Code_CEO	*B*4	B_HTTP_L_R3	*B*14	Disc_CEO
*A*5	A_FTP1_S	*A*15	Inject_Code_Accounting	*B*5	B_All_R3	*B*15	Do_Nothing
*A*6	A_FTP1_W	*A*16	Get_Info_CEO	*B*6	B_DB2_H_R2	*B*16	B_DB2_All_R3
*A*7	DoS_HTTP	*A*17	Get_Data_Accounting	*B*7	B_DB2_L_R2	*B*17	B_DB2_All_R2
*A*8	DoS_FTP1	*A*18	Get_Info & Act	*B*8	B_HTTP_H_R2	*B*18	B_HTTP_All_R2
*A*9	DoS_DB2	*A*19	Do_Nothing	*B*9	B_HTTP_L_R2	*B*19	B_HTTP_All_R3
*A*10	Act_DB2_CEO	*A*20	A_DB2_M	*B*10	B_FTP1_H_R2	*B*20	B_FTP1_All_R2

intensity level ('A_HTTP_S' (or 'A_FTP1_S')) at node H_d. Then, the defender selects to block the HTTP traffic (or the FTP1 traffic) on Router 2 with low level ('B_HTTP_L_R2' (or 'B_FTP1_L_R2')) at node D_{d+1}. Then, the intrusion is detected to continue with denying the HTTP service (or the FTP1 service) ('DoS_HTTP' (or 'DoS_FTP1')) at node H_{d+2}. In this case, the defender chooses to disconnect the HTTP server (or the FTP1 server) from the network system using the managed switch ('Disc_HTTP' (or 'Disc_FTP1')) at node D_{d+3}. At this stage, the attacker–defender game to the information source also arrives at phase two (that is, $f_1 = 2$). If the attack is detected to spoof Database 2 and the HTTP server (or the FTP1 server) and start sniffing the traffics of these services from the CEO ('Act_DB2_CEO' and 'Act_HTTP_CEO' (or 'Act_FTP1_CEO')) at nodes H_{d+2} and H_{d+4}, and the defender chooses other responsive actions than blocking the traffic to Database 2 and HTTP (or FTP1) on Router 2 with high level ('B_DB2_H_R2' and 'B_HTTP_H_R2' (or 'B_FTP1_H_R2')) at nodes D_{d+3} and D_{d+5}, respectively, then the game arrives at phase three (that is, $f_1 = 3$). If the intruder is found to reply to requests from the CEO, and begin to inject codes into its machine ('Inject_Code_CEO') at nodes H_{d+4} and H_{d+6}, and the defender does not choose to disconnect the machine of the CEO using the managed switch ('Disc_CEO') at nodes D_{d+5} and D_{d+7}, respectively, then the game reaches the final phase (which is, $f_1 = F_1 = 4$). At this stage, the attacker can compromise the machine and is able to access information from the CEO computer ('Get_Info_CEO') at nodes H_{d+6} and H_{d+8}. When the game is in phase i ($i = 1$, 2, 3, 4) to the information in the CEO computer ($o = 1$), then the value of the binary variable b_{f_1} is equal to 1.

On the basis of the RDFP algorithm, the defender can respond to the multistage attack as follows. At the beginning of the game, the defender assigns λ as the risk neutral type of the attacker based on our assumption, analyzes the phases of the attacker–defender game on tree G. Suppose an attack to Database 2 with strong intensity level ('A_DB2_S') is detected at node H_{d-1}, then $f_1 = 1$. The defender now needs to select its best move at node D_d. Figure 6 shows a reduced game tree $RGD(D_d)$ to this attack with possible responses 'B_DB2_H_R3' and 'B_DB2_L_R3'. Starting at each possible response, the defender finds the shortest path toward

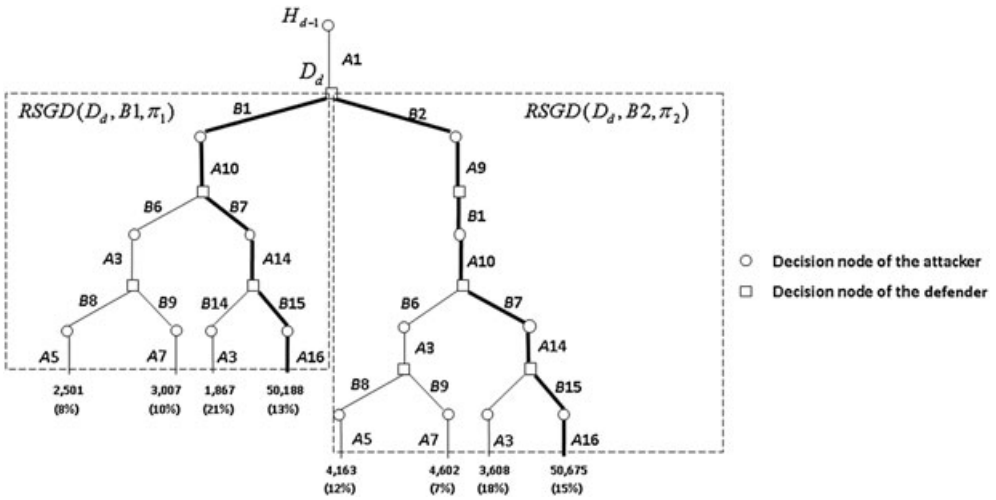

Figure 6. The reduced game tree $RGD(D_d)$ of the defender for Scenario 1.

a node when the attacker reaches its goal. The two corresponding 'shortest paths' are indicated with bold lines. Let π_1 and π_2 denote the lengths of these shortest paths, respectively. The defender then constructs trees $RSGD(D_d,B_1,\pi_1)$ and $RSGD(D_d,B_2,\pi_2)$ for its responses. They are shown inside the rectangles of Figure 6.

Let H_{d+2n-1} be an arbitrary decision node of the attacker in $RGD(D_d)$. Then, tree $RGA(H_{d+2n-1})$ of the attacker with depth τ can be found for each node H_{d+2n-1}. On the basis of the assessed probability distribution of the possible future responses, the game tree $RGA(H_{d+2n-1})$ becomes a decision tree $RDA(H_{d+2n-1})$. The values of $E_a^{H_{d+2n-1}}$ and $V_a^{H_{d+2n-1}}$ in $RDA(H_{d+2n-1})$ can be calculated from relations (8) and (9). Equations (10) and (11) are used to compute $U_a^{H_{d+2n-1}}$ and $pp_a^{H_{d+2n-1}}$ at each node H_{d+2n-1} of $RGD(D_d)$, and so the game tree $RGD(D_d)$ becomes the decision tree $RDD(D_d)$. Notice that Figure 6 shows only one attack with the highest probability under each decision node of the attacker due to the limited space. The first number at the end of each path in $RDD(D_d)$ indicates the value of y_{k_r}. It can be computed from relation (2), which is used by the attacker to calculate the total impact of the attack. The second number shows the value of Q_{k_r} where relation (5) is applied. The values of $E_{k_r}^{D_d}$ are computed by using Equation (6). Then, relation (12) is used to see that no response is dominated by any other, so none of them can be eliminated. The expectation $E_r^{D_d}$ and variance $V_r^{D_d}$ of response 'B_DB2_H_R3' or 'B_DB2_L_R3' can be computed on the basis of (8) and (9), respectively, and then (13) is used to find the payoff value $U_r^{D_d}$ for each possible response. Finally, on the basis of (14), the best response of the defender is 'B_DB2_H_R3', and then $f_1 = 2$.

Assume that the next detected attack at H_{d+1} is 'Act_DB2_CEO'. A game tree $RGD(D_{d+2})$ with two possible responses is partially shown in Figure 7. All parts of the game tree that do not lead to the 'shortest paths' are ignored. As before, thicker lines indicate the 'shortest paths' along which the attacker is able to reach its objective starting at the given responses of the defender. The values of y_{k_r} and Q_{k_r} can be obtained after tree $RGD(D_{d+2})$ becomes the decision tree $RDD(D_{d+2})$. Using the computed values of $E_{k_r}^{D_{d+2}}$, relation (12) shows that no responsive action at D_{d+2} can be deleted. By comparing the payoff values $U_r^{D_{d+2}}$ of all candidate responses at node D_{d+2}, we can see that action 'B_DB2_L_ R2' is the best, so it is chosen, and then $f_1 = 3$. If the next detected attack at H_{d+1} is 'Inject_Code_CEO', the best response can be found as 'Disc_CEO' following a similar approach.

In attacker–defender game, the attacker launches multistage attacks based on his/her own possible strategies and the defender's possible responsive actions. For the same attack, different responsive algorithms may give different responses even in cases when the attacker uses the same strategy, because the interactions between the two players may be different on the basis of the three different response approaches. Different interactions also cause different total losses to the system. The left most, the middle, and the right most paths in Figure 8 show the interactions between the players when the intruder launches a multistage attack with one objective and the defender employs the SO, the RDFP, and the GA algorithm to defend against the attack, respectively. The arcs under the attacker's decision nodes represent the detected attacks, and the arcs under the defender's decision nodes are the responsive actions based on different response schemes. The numbers under the last nodes of each path show the total losses of the system during the entire life cycle of the multistage attack. While the responses with the GA algorithm result in 3980 units of total

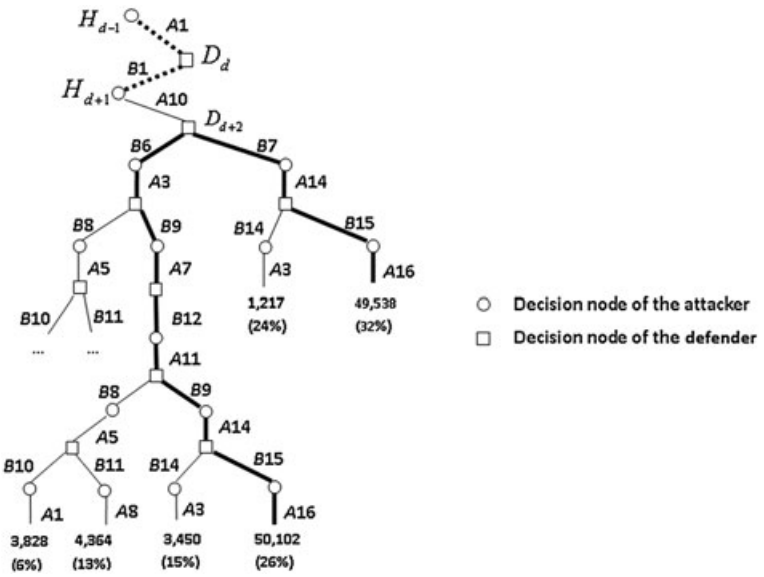

Figure 7. The reduced game tree $RGD(D_{d+2})$ of the defender for Scenario 1.

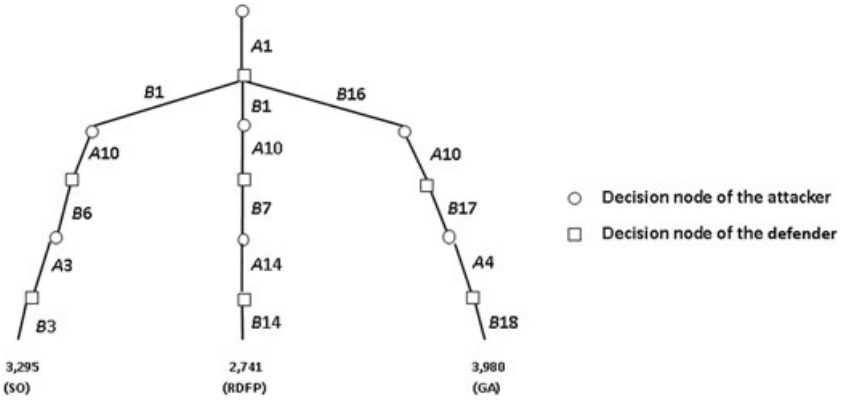

Figure 8. Defending against a multistage attack with one objective.

Table IV. Performances of the greedy algorithm, the single-interaction optimization and the response by dynamic game tree-based fictitious play algorithms in Scenario 1.

The total losses of the computer network		Attacker							
		One objective							
Instances		1	2	3	4	5	6	7	8
Defender	GA Algorithm	3980	4062	2612	4470	3285	2850	5072	4238
	SO Algorithm	3295	3425	2440	4280	2960	2537	4635	4010
	RDFP Algorithm	2741	2910	2250	3340	2470	1760	4120	3780

GA, greedy algorithm; SO, single-interaction optimization; RDFP, response by dynamic game tree-based fictitious play.

loss to the computer network, the total loss of the system is only 2741 units by employing the RDFP algorithm.

Although the total loss of the system calculated from the SO algorithm is less than that of the GA algorithm, it is still higher than our approach. We also repeated this experiment with other multistage attacks containing the different strategies. The total losses with different defensive approaches were calculated similarly. The results are shown in Table IV.

5.2.2. Scenario two: multistage attacks with two objectives.

In this scenario, we assume that the attacker has two goals (that is, $O=2$): obtaining the information from the CEO ($o=1$) and accessing the data from the accounting system ($o=2$). The accounting system also needs services provided by Database 2 and the HTTP server. The phases f_1 of the multistage game with respect to gaining access to the CEO information have been already introduced in the previous section. The phases f_2 of the game to access the data from the accounting system can be described as follows. Starting with phase one (that is, $f_2=1$), the intruder attacks the HTTP server by flooding it with strong intensity level ('A_HTTP_S'). If the defender blocks the HTTP traffic to the HTTP server with high block level on Router 2 ('B_HTTP_H_R2'), then the attacker–defender game to the data in the Accounting reaches phase two (that is, $f_2=2$). Then, the attacker spoofs the HTTP server and starts sniffing the HTTP traffic to it from the accounting system ('Act_HTTP_Accounting'). If the defender chooses other responsive actions than blocking the HTTP traffic on Router 3 with high level ('B_HTTP_H_R3'), then the game arrives at phase three (that is, $f_2=3$). The attacker then replies to the requests from the accounting system and begins to inject codes into it ('Inject_Code_Accounting'). If the defender does not block all traffic on Router 3 ('B_All_R3'), then the attack arrives at the final phase (that is, $f_2=F_2=4$). The attacker compromises the machine of the Accounting, obtains the data ('Get_Data_Accounting') reaching its goal.

The defender has to choose his/her best responses by considering both objectives of the attacker in this scenario. Figure 9 partially illustrates a game tree $RGD(D_d)$ for possible responses to the detected attack ('A_HTTP_S') at node H_{d-1} with $f_1=1$ and $f_2=1$. The notation of each action is given in Table III. 'Get_info & Act' denotes the attacker's selection 'Get Info' and then its choice 'Act_HTTP_Accounting' as shown in Figure 9. The bold lines starting at the possible responses indicate the 'shortest paths' that the attacker is able to obtain the information from the CEO and the data from the accounting system. The lengths of the 'shortest paths' determine the depths of the game trees $RSGD(D_d,r,\pi)$. The payoffs $U_r^{D_d}$ of possible responses $r(r\in RR)$ can be calculated in a similar way as it was shown in the case of a single objective, and then the candidate response with the smallest payoff value

DOI: 10.1002/sec

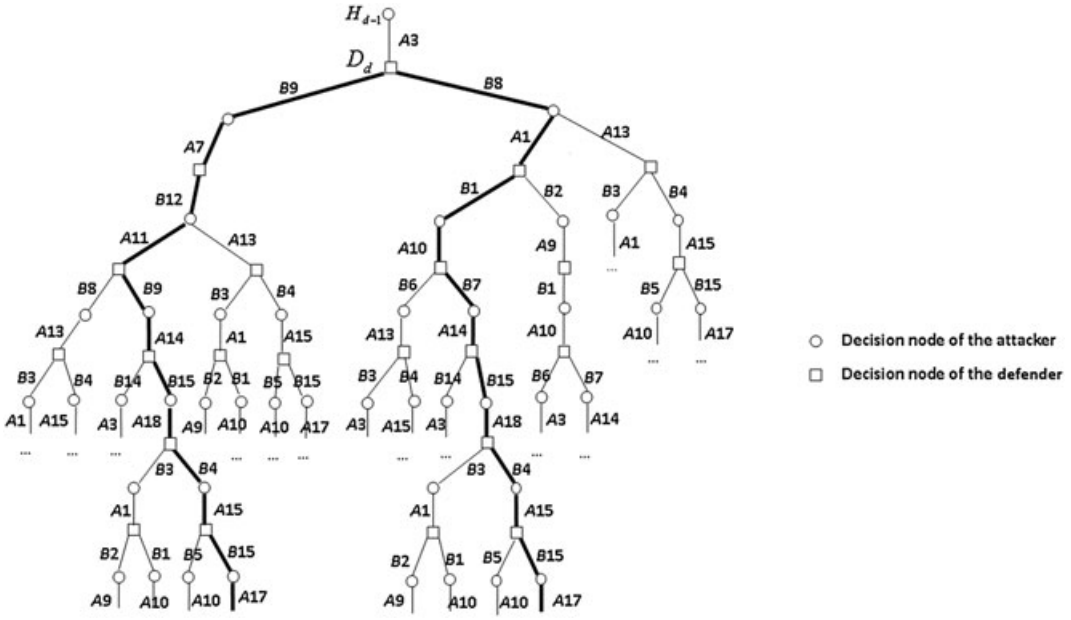

Figure 9. The reduced game tree $RGD(D_d)$ of the defender for Scenario 2.

is chosen as the best action r^* of the defender at root D_d. Because of limited space, Figure 9 shows only the structure of the reduced game tree $RGD(D_d)$ without showing the values of y_{k_r} and Q_{k_r} for the different paths.

Figure 10 shows the interactions of the two players based on a strategy of the attacker with two objectives. The left most, middle, and right most paths shown in Figure 10 illustrate the interactions between the players based on the SO, the RDFP, and the GA algorithms, respectively. The total losses of the system during the life cycle are given at the terminating nodes of these paths. Clearly, the RDFP algorithm is the best again in comparing the results of the three methods. It is also clear that the relative reduction in loss is larger with two objectives than in the case of only a single objective. More instances with two objectives were simulated on the basis of different attack strategies. The total losses of the system in these instances are listed in Table V. Tables IV and V show that the total losses of the system during a life cycle with the RDFP algorithm is on the average 36% smaller than the

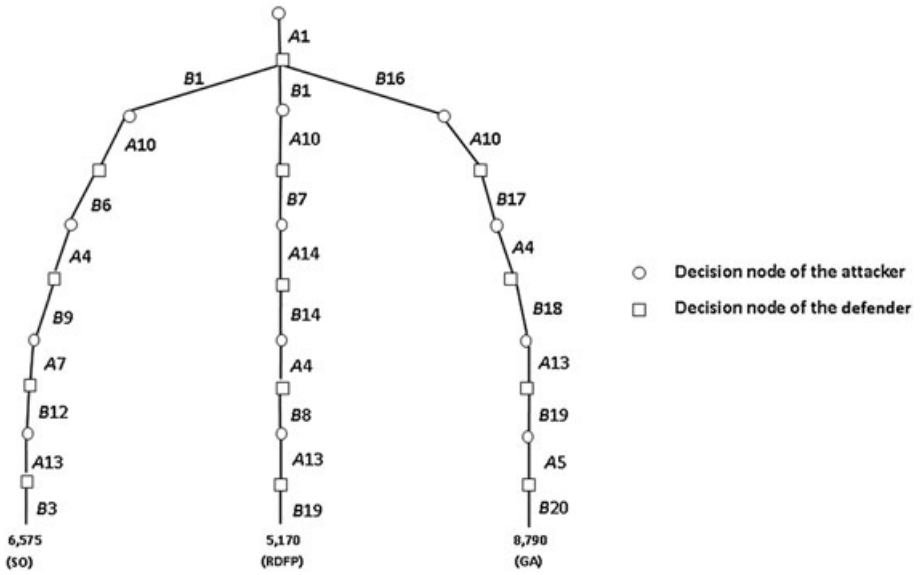

Figure 10. Defending against a multistage attack with two objectives.

Table V. Performances of the greedy algorithm, the single-interaction optimization and the response by dynamic game tree-based fictitious play algorithms in Scenario 2.

The total losses of the computer network		Attacker							
		Two objectives							
instances		1	2	3	4	5	6	7	8
Defender	GA Algorithm	8790	8016	7973	6891	8758	9450	10,507	8253
	SO Algorithm	6575	7680	7288	6384	7984	8625	9630	7390
	RDFP Algorithm	5170	5338	6005	4740	5736	7860	8743	6542

GA, greedy algorithm; SO, single-interaction optimization; RDFP, response by dynamic game tree-based fictitious play.

GA algorithm and 22% smaller than the SO algorithm. Paired *T*-test shows that both reduced losses are statistically significant with 99.5% confidence level.

In reality, the objectives of the attacker are unknown to the defender at the beginning of the game. As we mentioned before, the defender can identify the vulnerabilities of the computer network as the possible targets of the attacker. We assume again that the CEO computer and the accounting system are the possible targets of the attacker. Both of them need other services to function correctly as in the previous cases. On the basis of repeated observations, the defender can continuously update the attacker's objectives.

Suppose an intrusion is detected to attack Database 2 with strong intensity level ('A_DB2_S') at node H_{d-1}. Because the detected attack belongs to phase one ($f_1 = 1$) of attacking the CEO computer and it is not in any phase to gain access to the accounting system, game trees $RSGD(D_d,r,\pi)$ are similar to those shown in Figure 6. In order to find his/her best response at node D_d, the defender selects the depth of the attacker's game tree at each of the attacker's decision nodes in $RGD(D_d)$ and calculates the payoffs and the probability distributions of the possible attacks at these decision nodes. Therefore, the decision tree $RDD(D_d)$ is obtained, and $U_r^{D_d}$ can be computed on the basis of it. The best response of the defender at node D_d is the same as obtained in Scenario 1, and then $f_1 = 2$.

Suppose the next detected attack is 'A_HTTP_S'. The detected attack belongs to phase one of attacking both the CEO computer and the accounting system, so the CEO information ($o = 1$) and the accounting data ($o = 2$) could be the objectives of the attacker. Because the attacker–defender game to the CEO information has already reached phase two, we have $f_1 = 2, f_2 = 1$. Then, the method shown for Scenario 2 can be applied to this situation. Following these basic scenarios, our intrusion defense system based on the RDFP algorithm can efficiently defend against more complicated multistage attacks in reality.

6. CONCLUSION

This paper introduced a multistage intrusion defense system, where the sequence of interactions between the attacker and the defender was modeled as a two-player non-cooperative non-zero-sum dynamic game with incomplete information. The two players conducted the DFP along the game tree of the dynamic game. The attacker was assumed to be rational, and the defender could update his/her knowledge on the decision process of the attacker after each detected attack and could use it to predict future attacks more accurately. Multistage attacks have a life cycle consisting of reconnaissance, penetration, attack, and exploit. With the development of intrusion detection technologies, the defender becomes more familiar with the properties of multistage attacks. Our RDFP algorithm can help the defender find the best defending strategies quickly. The performance of the RDFP algorithm is better than that of the other algorithms as it was demonstrated in our simulation study.

In order to clearly introduce the framework of the RDFP algorithm, we assume that the attacker is risk neutral for mathematical simplicity. The algorithm is described so far in an offline mode, because the learning of the risk attitude of the opponent is not considered. A fast-converging approach based on Bayesian learning had been developed and is being evaluated to update the player's knowledge of the opponent after each interaction so that the defender will be able to make the best online decisions dynamically. In our future research, we will consider other attack scenarios in more complex network systems, such as when the intruder can launch multi-attack before the defender gets a chance to react. Moreover, the current RDFP algorithm considers only the game between two players, the defender and an attacker or a group of attackers that can collaborate in launching a multistage attack. In reality, different attackers or different groups of attackers may cooperate to attack the system resources. Although the overall system can be described as a non-cooperative, non-zero-sum game, the interaction between the attackers is a cooperative game. This multiplayer scenario is more complex and will be considered in our future research work.

REFERENCES

1. Forgo F, Szep J, Szidarovszky F. *Introduction to the Theory of Games*. Kluwer Academic Publishers: Dordrecht, The Netherlands, 1999.
2. Alazzawe A, Nawaz A, Bayraktar MM. Game theory and intrusion detection systems. http://www.qatar. cmu. edu/iliano/courses/06S-GMU-ISA767/project/papers/alazzawe-mehmet-nawaz.pdf

3. Stewart B. *Skating on Stilts, Why We Aren't Stopping Tomorrow's Terrorism*. Hoover Institution Press: Stanford, CA, 2010.
4. Chen H, Al-Nashif Y, Qu G, Hariri S. Self-configuration of network security. *Proceedings of the 11th IEEE International Enterprise Distributed Object Computing Conference*, *EDOC'07*, 2007; 97–108.
5. Al-Nashif Y, Kumar A, Hariri S, Luo Y, Szidarovszky F, Qu G. Multi-level intrusion detection system (ML-IDS). *Proceedings of the 2008 International Conference on Autonomic Computing, ICAC'08*, 2008; 131–140.
6. Al-Nashif Y. Multi-level anomaly based autonomic intrusion detection system. *Doctoral dissertation*, retrieved from ProQuest Dissertations and Theses, 2008.
7. Jin H, Vel O, Zhang K, Liu N. Knowledge discovery from honeypot data for monitoring malicious attacks. *Lecture Notes in Artificial Intelligence* 2008; **5360**:470–481.
8. Fudenberg D, Levine DK. *The Theory of Learning in Games*. MIT Press: Cambridge, MA, 1998.
9. McGill WL, Ayyub BM, Kaminskiy M. Risk analysis for critical asset protection. *Risk Analysis* 2007; **27**(5):1265–1281.
10. Luo Y, Szidarovszky F, Al-Nashif Y, Hariri S. A game theory based risk and impact analysis method for intrusion defense systems. *Proceedings of the Seventh ACS/IEEE International Conference on Computer Systems and Applications, AICCSA'09*, 2009; 975–982.
11. Samuelson PA. The fundamental approximation theorem of portfolio analysis in terms of means, variances and higher moments. *The Review of Economic Studies* 1970; **37**(4):537–542.
12. Szidarovszky F, Bahill AT. *Linear Systems Theory* (2nd edn). CRC Press: Boca Raton, FL, 1999.
13. Foo B, Wu YS, Mao YC, Bagchi S, Spafford E. ADEPTS: adaptive intrusion response using attack graphs in an e-commerce environment. *Proceedings of the 2005 International Conference on Dependable Systems and Networks, DSN'05*, 2005; 508–517.
14. Toth T, Kruegel C. Evaluating the impact of automated intrusion response mechanisms. *Proceedings of the 18th Annual Computer Security Applications Conference, ACSAC'02*, 2002; 301–310.
15. Stakhanova N, Basu S, Wong J. A taxonomy of intrusion response systems. *International Journal of Information and Computer Security* 2007; **1**(1/2):169–184.
16. Foo B, Glause MW, Howard GM, Wu YS, Bagchi S, Spafford EH. Intrusion response systems: a survey. In *Information Assurance: Dependability and Security in Networked Systems*, Spafford EH (ed.). Morgan Kaufmann Publishers: Burlington, MA, 2008; 377–412.
17. Lye K, Wing J. Game strategies in network security. *International Journal of Information Security* 2005; **4**(1-2):71–86.
18. Filar F, Vrieze K. *Competitive Markov Decision Processes*. Springer-Verlag: New York, 1996.
19. Fudenberg D, Tirole J. *Game Theory*. MIT Press: Cambridge, MA, 1991.
20. Shen D, Chen G, Blasch E, Tadda G. Adaptive Markov game theoretic data fusion approach for cyber network defense. *Proceedings of the 2007 IEEE Military Communications Conference, MILCOM'07*, 2007; 1–7.
21. Carin L, Cybenko G, Hughes J. Cybersecurity strategies: the QuERIES methodology. *Computer* 2008; **41**(8):20–26.
22. Luo Y, Szidarovszky F, Al-Nashif Y, Hariri S. Game tree based partially observable stochastic game model for intrusion defense systems (IDS). *Proceedings of the 2009 IIE Annual Conference and Expo, IERC'09*, 2009; 880–885.
23. Zhang Z, Ho P. Janus: a dual-purpose analytical model for understanding, characterizing and countermining multistage collusive attacks in enterprise networks. *Journal of Network and Computer Applications* 2009; **32**(3):710–720.
24. Zonouz SA, Khurana H, Sanders WH, Yardley TM. RRE: a game-theoretic intrusion response and recovery engine. *Proceedings of the 39th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN'09*, 2009; 439–448.
25. Ourston D, Matzner S, Stump W, Hopkins B. Coordinated Internet attacks: responding to attack complexity. *Journal of Computer Security* 2004; **12**(2):165–190.
26. Liu P, Zang W, Yu M. Incentive-based modeling and inference of attack intent, objectives, and strategies. *ACM Transactions on Information and System Security* 2005; **8**(1):78–118.
27. Alpcan T, Basar T. *Network Security: A Decision and Game Theoretic Approach*. Cambridge University Press: Cambridge, U.K, 2011.
28. Buttyan L, Hubaux JP. *Security and Cooperation in Wireless Networks*. Cambridge University Press: Cambridge, U.K, 2008.
29. Ourston D, Matzner S, Stump W, Hopkins B. Applications of hidden Markov models to detecting multistage network attacks. *Proceedings of the 36th Hawaii International Conference on System Sciences, HICSS'03*, 2003.
30. Richardson BT, Chavez L. National SCADA test bed consequence modeling tool. *SANDIA report*, http://energy.sandia.gov/wp/wp-content/gallery/uploads/Consequence_Modeling_Tool_Report_Final.pdf
31. Szidarovszky F, Gershon M, Duckstein L. *Techniques for Multiobjective Decision Making in Systems Management*. Elsevier: Amsterdam, The Netherlands, 1986.
32. Forman EH, Gass SI. The analytic hierarchy process – an exposition. *Operations Research* 2001; **49**(4):469–486.

APPENDIX A

```
procedure FIND_RDA(H_d)
1:    FIND_GA(H_d) based on G;
2:    FIND_RGA(H_d) based on τ;
3:    for all a ∈ AA at the root of RSGA(H_d, a, τ) do
4:       for all k_a ∈ KK_a at the root of RSGA(H_d, a, τ) do
5:          for all m ∈ M
6:             ASSESS_ p_{k_a}^{d+2m-1};
7:          end for
8:       end for
9:          RDA(H_d) = RDA(H_d) ∪ RSDA(H_d, a, τ);
10:   end for
end procedure
```

Figure A1. Procedure 'FIND_$RDA(H_d)$'.

```
procedure FIND_RGD(D_d)
1:    FIND_GD(D_d) based on G;
2:    for all r ∈ RR at the root of SGD(D_d, r) do
3:       π = BFS (Shortest path in SGD(D_d, r));
4:       RGD(D_d) = RGD(D_d) ∪ RSGD(D_d, r, π);
5:    end for
end procedure
```

Figure A2. Procedure 'FIND_$RGD(D_d)$'.

DOI: 10.1002/sec

procedure FIND_$RDD(D_d)$

1: **for all** $r \in RR$ at the root of $RGD(D_d)$ **do**
2: **for all** $n \in N$ in $RSGD(D_d, r, \pi)$ **do**
3: **for all** H_{d+2n-1} in $RSGD(D_d, r, \pi)$
4: FIND_$RDA(H_{d+2n-1})$;
5: **for all** $a \in AA$ at the root of $RSDA(H_{d+2n-1}, a, \tau)$ **do**
6: **for all** $k_a \in KK_a$ at the root of $RSDA(H_{d+2n-1}, a, \tau)$
7: Compute y_{k_a} from equation (2);
8: Compute Q_{k_a} from equation (5);
9: Compute $E_{k_a}^{H_{d+2n-1}}$ from equation (6);
10: **end for**
11: **end for**
12: **if** $\max_{k_{d'} \in KK_{a'}} \{E_{k_{d'}}^{H_{d+2n-1}}\} \le \min_{\substack{k_a \in KK_a \\ a \neq a'}} \{E_{k_a}^{H_{d+2n-1}}\}$ **then**
13: Eliminate d' from AA at node H_{d+2n-1}, and then go to line 15;
14: **else**
15: Compute $U_a^{H_{d+2n-1}}$ from equation (10);
16: Compute $pp_a^{H_{d+2n-1}}$ from equation (11);
17: **end if**
18: **end for**
19: **end for**
20: $RDD(D_d) = RDD(D_d) \cup RSDD(D_d, r, \pi)$;
21: **end for**
end procedure

Figure A3. Procedure 'FIND_$RDD(D_d)$'.

procedure FIND_r^* at node D_d

1: **for all** $r \in RR$ at the root of $RDD(D_d)$ **do**
2: **for all** $k_r \in KK_r$ at the root of $RDD(D_d)$
3: Compute y_{k_r} similar to equation (2);
4: Compute Q_{k_r} similar to equation (5);
5: Compute $E_{k_r}^{D_d}$ similar to equation (6);
6: **end for**
7: **end for**
8: **if** $\min_{k_{r'} \in KK_{r'}} \{E_{k_{r'}}^{D_d}\} \ge \max_{\substack{k_r \in KK_r \\ r \neq r'}} \{E_{k_r}^{D_d}\}$ **then**
9: Eliminate r' from RR, and then go to line 11;
10: **else**
11: Compute $U_r^{D_d}$ from equation (13);
12: Find r^* from equation (14);
end procedure

Figure A4. Procedure 'FIND_r^* at node D_d'.