Intrusion intention recognition and response based on weighed plan knowledge graph

Zengyu Cai*, Qikun Zhang, Ran Zhang, Yong Gan

School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zheng Zhou 450002, China

Received 25 October 2014, www.cmnt.lv

Abstract

With the development of the network, security has become the focus problem of network. To be effective, current intrusion prevention systems must incorporate artificial intelligence methods, such as plan recognition and adversarial plan. Plan recognition is critical for predicting the future actions of attackers and the adversarial plan is critical for planning appropriate responses to attacks. In this paper, an attack intention and plan recognition method based on weighted planning knowledge graph is presented to predict the anomaly intentions of potential intruders to a computer system according to the observation data. And the adversarial planning method based on HTN planning to response the future actions of attackers is also presented. The experimental results show that the plan recognition method based on weighed planning knowledge graph has a good accuracy in predicting the intrusion intentions. The experimental results also show that the adversarial planning method can prevent computer system correctly and effectively.

Keywords: adversarial plan, plan recognition, plan knowledge graph, intrusion

1 Introduction

With the rapid development of the network application, the traditional passive safety mechanism has become difficult to meet the needs of the security situation. Intrusion Prevention System (IPS) as a basic network security technology already has attracted attentions of many researchers [1-4]. The IPS must be able to analyze the actions of an attacker, infer the attacker's goals, and make predictions about their future actions. But it is difficult to infer intentions and predict future actions of attackers in IPS. In the artificial intelligence literature this process of deducing an agent's goals from observed actions is called plan recognition or intention recognition. In order to improve the intelligence level of IPS, plan recognition has a number of successful applications in the intrusion detection system, and has played an important role in improve the performance of the system [3-6]. However most existing plan recognition method can't predict intentions of agents. In our previous work, we had described an approach to predict the future action based on Planning Knowledge Graph (PKG) [7]. In this paper we discuss the improvement of it and its application to the network security domain.

On the other hand, planning appropriate responses to attackers is another factor for IPS. The adversarial plan is important for planning appropriate responses according the future actions of attackers. Adversarial plan was first suggested by Geib and Goldman [3] as an addition to the traditional models of keyhole and intended recognition. In this paper we discuss the adversarial plan in network security domain also. Hierarchical Task Network (HTN) planning has been extensively studied in the AI planning. Many Adversarial planners used HTN plan [8, 9].The PKG and HTN intelligent planning are almost entirely similar in planning knowledge storage and inference mechanism, so which is very beneficial to knowledge sharing of intelligent planning and plan recognition.

This paper constructs an intrusion prevention method based on Weighed Planning Knowledge Graph (WPKG). It can recognize intention of attackers, through the observation of invasion action. And it also can predict the attack action and take response strategies effectively. The paper is organized as follows. In Section 2, we discuss the related work to our paper. Section 3 describes the intrusion intention recognition and response algorithm based on WPKG in network. Section 4 presents the experiments of our intrusion prevention method. The paper concludes with a brief summary of results.

2 Background and related work

2.1 INTRUSION DETECTION AND INTRUSION PREVENTION

Intrusion Detection System (IDS) mainly includes the following four steps: data collection, data pre-processing, behavioral analysis and response. Intrusion analysis is the core of IDS, the essence of which is to use the reasoning or pattern matching etc. intelligence technology to determine whether user's behavior is intrusion according to the abstract description of user behavior and existing security policies.

* ***Corresponding author's*** e-mail: mailczy@163.com

There is no a clear definition to IPS. We define it as any hardware or software systems can detect the attacks or security threats, block the attack protective system effectively. Technically speaking, the IPS combines firewall and IDS, which purpose is to provide safety protection of in-depth and effective for networks. The response of traditional IPS is mostly achieved by simple rule-based trigger. There are several IPS using adversarial plan and adversarial plan recognition [3-6]. However these papers have focused on the requirements for adversarial plan in IDS or recognition algorithm for network attack.

2.2 PLAN AND PLAN RECOGNITION

McDermott and James Hemdeler thought a plan is devising the sequence of actions for an agent [10]. We define it as a set of actions that can achieve the goals of a planning problem. The planning problem references a STRIPS-like domain (a set of operators), a set of objects, a set of propositions (literals) called the initial conditions and a set of problem goals which are propositions that are required to be true at the end of a plan. The planner can find a valid plan, that is a set of actions and specified time steps in which each is to be carried out. A valid plan must make all the problem goals true at the final time step. In the intrusion prevention, intelligent planning can give a reasonable response action on the basis of the targets have been identified of the intruders and the next action protected.

Plan recognition involves inferring the intention of an agent from a set of observed actions [7]. In intrusion detection, the plan recognition can determine whether there invasion or threaten and can predict the next possible action of the attacker.

2.3 ADVERSARIAL PLAN RECOGNITION AND ADVERSARIAL PLANNING

Adversarial plan recognition was first suggested by Geib and Goldman [3] as an addition to the traditional models of keyhole and intended recognition. It has been also independently proposed by Jensen et al. for predicting the opponent’s moves in robotic games [11]. In adversarial recognition, the observed agent is hostile to the observation of his actions and attempts to thwart the recognition. We define adversarial planning as it uses plan recognition to infer the goals of hostile agents, predicts their future actions, and blocks the hostile agents’ goals realizing. Although there has been significant recent work in adversarial plan recognition [4, 8, 9, 11, 12], little thought has been given to the question of how to oppose attacks in network.

2.4 PLAN KNOWLEDGE GRAPH (PKG) AND SUPPORTING DEGREE

PKG is an acyclic AND/OR graph G=(N, E), where N and E denote the set of nodes and edges respectively, in which nodes denote plans (events) and edges denote the supporting relation between nodes [13]. AND nodes present that they are component nodes of their parents. The children and their parents have relationships of the whole and the part which is presented by arc lines in the graph. OR nodes present that their parents and they have relationships of abstraction and specialization. All nodes are joined by edges which are used to connect the parent and its children.

Plan recognition algorithms based on PKG choose candidate plans by computing probability of every event in really world. Two kinds of data are needed in computing the probability. Firstly, it is the probability of every event in the real world. Secondly, it is the probability of one event induced by another event (also called supporting degree). The supporting degree means the probability of a plan (event) induced by another plan (event). There are only two relationships between events, abstraction and specialization or whole and part. Supporting degree under the two relationships is simply prescribed in [13]: the supporting degree of the appearance of specialization plan to the abstraction plan is 1; the sum supporting degree of the appearance of all part plans to the whole plan is 1. See details in [13].

2.5 HTN PLANNING

HTN planning is based on three types of object: Goals, Operators and Plan Schemas. Operators are actions which can be performed in the world. Goals are more abstract and express aims in the world. Schemas (also called Task Networks or Methods), specify the sub goals which must be achieved in order to satisfy the goal[12] . The objective of an HTN planner is to produce a sequence of actions that perform some activity or task. The description of a planning domain includes a set of operators similar to those of classical planning, and also a set of methods, each of which is a prescription for how to decompose a task into subtasks (smaller tasks) [14].

Informally, an HTN planning problem can be viewed as a generalization of the classical planning paradigm. An HTN domain contains, besides regular primitive actions, a set of tasks or high-level actions. Tasks can be successively refined or decomposed by the application of so-called methods. When this happens, the task is replaced by a new, intuitively more specific task network. In short, a task network is a set of tasks plus a set of restrictions (often ordering constraints) that its tasks should satisfy. The HTN planning problem consists of finding a primitive decomposition of a given (initial) task network [15].

3 Intrusion intention recognition and response algorithm based on weighed plan knowledge graph

The plan recognition method based on PKG was put forward by Jiang [13]. Compared with Kautz’s formalism used widely in plan recognition, this method is simpler and more direct. Jiang’s method changes the plan recognition problem into the graph searching one. This method not only prompts efficiency but also gives the same result as Kautz's. It can recognize the agent's plan according agent's acts. However, as Kautz's formalism, the Jiang's method can't predict future actions of agent. So, it can't work well in IDS. This paper adds edge's weight based on it, which will be expanded to a WPKG. It can more accurately recognize the attack planning and predict future attack action.

3.1 WEIGHED PLAN KNOWLEDGE GRAPH

The WPKG is similar to PKG, and it is also a directed acyclic AND/OR graph. There are three types of nodes, "OR", "AND" and "LEAF" nodes which are depicted by circles, rectangles and triangles respectively in Figure 1. These nodes have attributes including name and time-slice, which are depicted as (name, time-slice). AND node presents the whole-part relation that its children nodes are its component events. OR node presents abstract-specific relation that its children nodes are its specialization. LEAF node corresponds directly to primitive act.

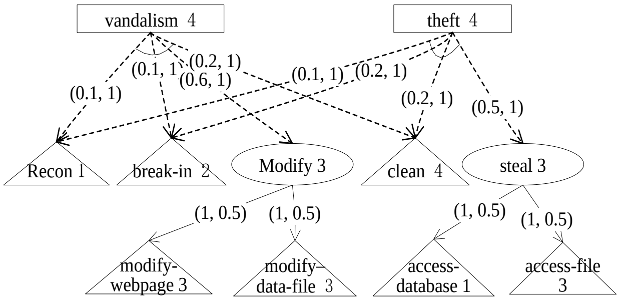

FIGURE 1 WPKG of simple intrusion domain. The numbers on the nodes represent time-slice of those nodes

The directed edges connect child nodes to their parents. The direction of arrows is from parents to their children. There are two types of edges, "AND" and "OR". The k-joint line points k subsequent child nodes from a parent node. It uses the k-joint line to present the whole-part relationships. In this paper, we add the weights on the edges to present the supporting degree, namely WPKG. These edges have two weights, child-support-parent and parent-support-child, which are depicted as (child-support-parent, parent-support-child) in Figure 1. In this paper, we use simple hierarchical plans, as most plan recognition work does. We assume that attackers have a plan library that provides recipes for achieving goals.

Figure 1 shows a WPKG for a "hacker" in a simplified computer network intrusion example. In the library, there are tow top-goals of attacker as theft and vandalism. The theft plan includes four steps: scan the system to determine vulnerabilities (recon), exploit the system's weaknesses to gain entry (break-in), export desired data (steal), and hide traces of presence on computer (clean). The orders of the four steps are partial ordered relations. Ordering constraints within a method are represented by time-slice. For example, the hacker must break-in before she can steal. The vandalism plan also includes four steps: (recon), (break-in), modify data in disk without authorization (modify), and (clean). The steal has the special children access-database and access-file. The modify has the special children modify-webpage and modify-data-file.

3.2 SUPPORTING DEGREE

In PKG, it only considers the influences of the children to their parents but not the parents to their children. We define the supporting degrees as follows.

3.2.1 Effects of the parts on the whole

In an availed plan, that if any part of a plan happens, the plan may take place. if all parts of a plan happen, the plan is sure to take place. So we add constraints that sum of all supporting degree of part plans to the whole plan equals *1*. It's same to PKG.

For example, if plans such as $A_1, A_2, \ldots, A_j, \ldots, A_n$ are parts of plan B, then the supporting degree of $A_i(1 \le i \le n)$ to B is P (B/A_i), the plan B has n parts of plans A_i, $i=1,2,\ldots,n$, then the supporting degree of the n parts of plans to B is:

$$\mathrm{P}'\left(B/\bigcap_{j=1}^{n} A_j\right)=\sum_{j=1}^{n} P\left(B/A_j\right)=1. \tag{1}$$

3.2.2 Effects of the whole on the parts

It is obvious that if a plan happens, all parts of the plan are sure to take place. It does not consider this in reference [13] and it is one of the main reasons why it cannot predict the unobserved actions. So we add supporting degree of part plans to the whole plan. Its value is equal 1.

For example, if plans such as $A_1, A_2, \ldots, A_j, \ldots, A_n$ are parts of plan B, then the supporting degree of B to $A_i(1 \le i \le n)$ is:

$$\mathrm{P}(A_i/B)=1. \tag{2}$$

3.2.3 Effects of the abstraction on the specialization

If an abstraction plan happens, it's all specialization plans may take place. It is not considered in [13]. So we add supporting degree of to the specialization. The constraint of this supporting degree is that sum of all supporting degree of one abstraction plans to all its specialization plan is equal 1.

For example, there is an abstraction plan N which has specialization notes $B_1, B_2, \ldots, B_j \ldots, B_n$ and its happening probability P(N) is already known, then the supporting degree of N to B_j is:

$$\mathrm{P}'(B_j)= max\ \{\mathrm{P}(B_j/N) \times \mathrm{P}(N), \mathrm{P}(B_j)\}. \tag{3}$$

3.2.4 Effects of the specialization on the abstraction

It is obvious that if a specialization plan happens, one of its abstract plans is sure to take place. So we define that supporting degree of specialization plans to the abstraction plan is equal 1. It's same to PKG.

For example, a specialization plans $A_1,A_2,...,A_i,...,A_n$ are prescribed, which A_n is specialization plans of abstraction plan *B*. The happening probability of abstraction plan *B* is:

$$\mathrm{P}'\left(B/\bigcap_{j=1}^{n}A_i\right)=\max\{\mathrm{P}(A_i)\,,\,\mathrm{P}(B_j)\}. \quad (4)$$

3.3 INTRUSION INTENTION RECOGNITION

The task of the intrusion intention recognition algorithm is to find the intrusion intention of an attacker. The intrusion intention recognition algorithm is described below: the inputs of the algorithm are WPKG and the set of observed actions. WPKG is build according the knowledge of network security experts. The set of observed actions is extracted from the network environment automatic perception mechanism. The output of the algorithm is Intrusion Intention Graph(IIG) whose nodes are the possible intrusion intentions of attackers. The basic idea of our intrusion intention recognition algorithm is searching WPKG by a bottom-up strategy starting at observed events nodes. The calculating probability of the intrusion intentions is synchronized with searching.

The intrusion intention recognition algorithm is as follow:

Input: *O*: a set of observed actions, G=<$\mathrm{V_{AND}}$,$\mathrm{V_{OR}}$,$\mathrm{V_{LEAF}}$, E>: a WPKG;

Output: *IIG*: a Intrusion Intention Graph;

```
Create the initial IIG:
 for all $o_i \in O$ do
       add $o_i$ to IIG， set P($o_i$)=1;
 for other nodes oj in G and $o_i \notin O$ do
       set P($o_j$)=0;
 for all nodes n in the IIG, do
    find all nodes' parent m of n from G;
       if m is n's abstraction parent node, then
         P(m)=max{P(m),P(n)};
       if m is n's whole parent node, then
         P(m)= P(m)+P(n)×P(n/m);
       if P(m)> $\psi$($\psi$ is threshold), then
         add m to IIG;
repeat this to get the IIG until the top level;
return IIG;
```

In this way the IIG is obtained.

Consider the following observations: (break-in, access-database), it indicates the intrusion intentions based on WPKG as Figure1 using intrusion intention recognize algorithm. The IIG is gotten as Figure 2. As Figure 2 shows, that the hacker is engaged in stealing information has very high probability (0.7). And both vandalism and steal intention have the low probability (0.1). Other intentions are impossible. The intrusion intentions recognized are same as Jiang's. These results in this step can explain observations but can't predict future actions.

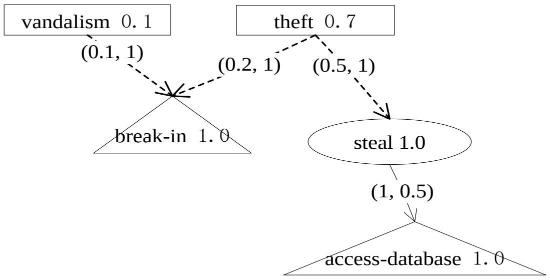

FIGURE 2 The IIG for our example in simple intrusion domain. The numbers on the nodes represent the probability of those nodes

3.4 FUTURE ACTION PREDICTION

The task of the future action predicting algorithm is to predict the future of attacker according its intrusion intention. The future action predicting algorithm is described below: the inputs of the algorithm are WPKG and IIG. IIG is constructed by intrusion intention recognition algorithm. The output of the algorithm is Intrusion Plan Graph (IPG), which includes the observed actions, intrusion intentions and the future attack actions. The basic idea of this algorithm is searching WPKG by a top-down strategy starting at top level.

The future action predicting algorithm is as follow:

Input: *IIG:* a Intrusion Intention Graph, G=<$\mathrm{V_{AND}}$,$\mathrm{V_{OR}}$, $\mathrm{V_{LEAF}}$, E>: a WPKG;

Output: *IPG:* a Intrusion Plan Graph;

```
Create the initial IPG intrusion intention graph as
  IIG:
 for all $I_i$ in IIG do
       add $I_i$ to IPG;
       P($I_i$) in IPG is equal to it in IIG ;
 for all nodes n in the IPG do
    find each part-child nodes $b_i$ of n from G;
       P($b_i$)=max{P($b_i$/n) ×P(n), P($b_i$)}=
           max{P(n), P($b_i$)};
       if P(bi)> $\psi$, then add m to IPG;
    find all special-child nodes bi of n from G;
       if all special-child nodes have the same
           probability, then
       P($b_i$)=max{P($b_i$/n) ×P(n), P($b_i$)};
    if P($b_i$)> $\psi$, then add m to IPG;
repeat this to get the IPG until the LEAF level;
return IPG;
```

In this way the IPG is obtained.

The future action predicting algorithm is used to predicting future attacks. The IGP is gotten as Figure 3 according IIG as Figure 2. As Figure 3 shows, the unobserved action recon is indicated with high probability (0.7). And the future action clean is also indicated with high probability (0.7). So our algorithm can not only indicate unobserved actions but also predict future action of attackers, which satisfies the requirements of network security environment. It lays a foundation for intrusion response and opposition.

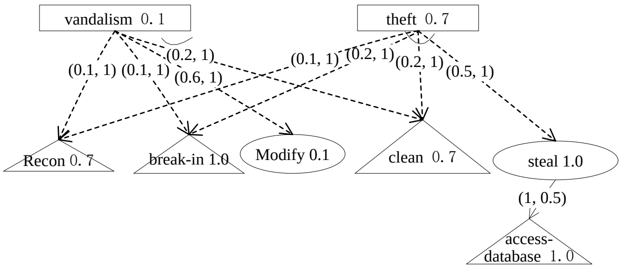

FIGURE 3 The IPG for our example in simple intrusion domain. The numbers on the nodes represent the probability of those nodes

3.5 INTRUSION RESPONSE

When future actions of attacker appear, go to the process of attack plan response. We use Adversarial Planning to prevent the implement of intrusion plan. The primitive action is similar to the actions used in a classical planning system that can be prevented directly using response function. We build their respective action sequences for each primitive action as the oppositional action and use ¬A to express the oppositional actions of action A. All the oppositional actions constitute the oppositional plan library. To find the oppositional action quickly, we use hash table to store oppositional plan library. The hash function is defined as follow.

$$\mathrm{add}(\neg A) = H（A）, \quad (5)$$

where A is a primitive action, ¬A is the oppositional action of A, add(¬A) is the entrance address of oppositional action ¬A, H is hash function which generate the entrance address of oppositional action according the name of a primitive action.

Firstly, it finds the primitive tasks. The future action predicting algorithm is used in this step. Then, oppositional actions are searched out from the hash table of oppositional actions.

The attack response algorithm is described as follows.

Input: *IPG*: a Intrusion Intention Graph, *G*=<V_{AND},V_{OR}, V_{LEAF}, E>: Weighed Plan Knowledge Graph, *OT*: Hash table of oppositional actions

Output: *FALSE* or *TRUE*;// opposition attempt is successful or failure.

```
(1) Create the initial intrusion action set $A$:
(2)     for all primitive actions $p_i$ in $IPG$ do add $p_i$ to set $A$;
(3) if set $A$ is nonempty then
(4)         select maximum probability action $a_j$;
(5)         remove $a_j$ from $A$;
(6) else return FALSE; \\ opposition failure
(7) Adversarial Planner search oppositional actions $\neg a_j$ from hash table of oppositional actions;
(8) if the response module execute the actions sequence of $\neg a_j$ successfully then
(9)     return TRUE; \\ opposition success
(10) else goto line (3).
```

4 Experimental results

To evaluate the effectiveness of our algorithms, we applied them in JAVA and tested it on a PC with a Intel Pentium E5300 processor at 2.6GHz, 2GB memory and Red Hat Linux 9.0 OS. The inputs of the algorithms are intrusion WPKG and events abstracted from real network environment. The empirical results are shown as follow.

4.1 TIME COMPLEXITY

The running time of our algorithms is too short to get the absolute value. It tacks theft for example. We got the running time by use the average value of running 10000 times. To clarify the influence of intrusion action to the recognition time, we gradually increased the number of WPKG observed actions. The CPU second per recognition is the average time it takes to process an observed action. As shown in Figure 4, there was a good linear relationship between CPU nanosecond per update and number of WPKG, that is, the relation between recognition time and number of nodes of knowledge graph is linear.

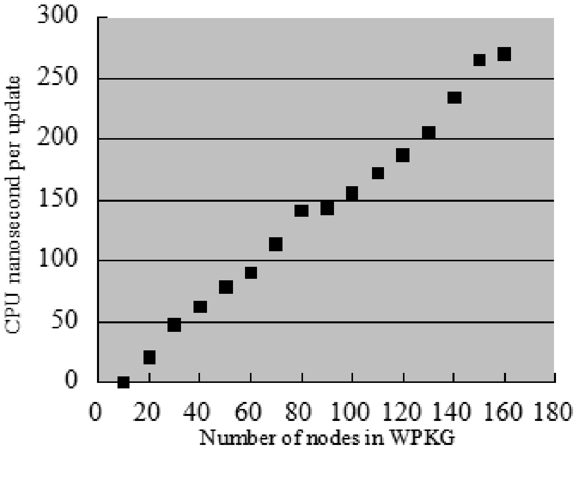

FIGURE 4 Experimental results of time complexity evaluation

4.2 PREDICTION ACCURACY

To evaluate the algorithm presented in this paper, we did the evaluation experiment on misreporting, failed reporting and recognizing. The definitions of basic concepts are as follow:

Rate of misreporting=number of no-attack actions recognized / total actions.

Rate of recognizing=number of attack actions recognized/ total attack actions.

Rate of failed reporting=*1*-Rate of recognizing.

The evaluation experiment is as follow. The intrusion detection method based on KPG and ours are respectively installed on two servers, which runs the services as HTTP, TELNET, FTP and so on. We also used another PC to attack the servers. In five days, we had sequentially attacked the two servers by same attack methods and times. The attack methods include Land, Telnet Flood, SYN Flood to FTP, NO-OPS Buffer Overflow, web-cgi-phf Scan, and NMAP Port Scan. The comparing experiment results of the tow algorithm are shown in Table1.

TABLE 1 Experiment results of prediction accuracy evaluation

	Rate of misreporting (%)	**Rate of recognizing (%)**	**Rate of failed reporting (%)**
KPG	6.3	91.2	8.8
Our algorithm	6.8	97.3	2.7

The results in Table 1 suggest that our algorithm can recognize the intrusion intention effectively. In the aspect of recognizing rate, our algorithm significantly outperforms PKG algorithm. But KPG algorithm is slightly do better at the rate of misreporting. This is due to the fact that our algorithm can predict future action and detect unobserved actions. As the unobserved and future actions are recognized, the recognizing rate is improved. It also leads to the increase of Rate of misreporting.

4.3 INTRUSION RESPONSE

To evaluate the intrusion response effect of our method, we did the evaluation experiment on knowledge of Figure 1. The inputs of the experiment are attack actions. Firstly, it recognized the intrusion intension and predicted the future actions according observed attacks. And then, the responses to attacks are generated using our response method. It uses ¬*A* to express the oppositional actions of action *A*. The plan threshold equals to *0.1*. The experiment results are show in Table 2.

As number observed actions increased, the intrusion intention recognized became gradually clear. And the probability of future actions predicted increased also. It is good for opposing the attacks. The results in Table 2 also suggest that our algorithm not only can recognize the attacker's intrusion intention and predict future attacks, but also it can generate oppositional actions to prevent intrusion. Because using hash table to search oppositional actions, it shortened the response time.

TABLE 2 Experiment results of intrusion response evaluation

Attack actions observed	**Intrusion intention recognized and its probability**	**Future actions predicted and its probability**	**Oppositional actions generated**
recon	(vandalism, 0.1) (Theft, 0.1)	break-in, 0.1 clean, 0.1	¬break-in ¬clean
recon, reak-in	(vandalism, 0.2) (Theft, 0.3)	clean,0.32 access-database,0.15 access-file,0.15	¬clean ¬access-database ¬access-file
recon, reak-in, access-database	(vandalism, 0.2) (Theft, 0.8)	Clean,0.8 access-database,0.15 access-file,0.15 modify-webpage,0.1 modify-data-file,0.1	¬clean ¬access-database ¬access-file

5 Conclusions

In this paper we construct the new intrusion detection and opposition algorithm based on the WPPG, which can recognize the attack intention in the complex network environment, predict the next action, and generate oppositional actions to prevent intrusion. It achieves combination between intrusion detection and intrusion response because of synthesize the advantage of intelligent planning and plan recognition successfully. In comparison with previous work on plan recognition, our algorithm has better prediction accuracy in network environment. Finally, unlike previous algorithm, ours can opposite attacks using oppositional actions. To using our plan recognition algorithm in complex network security domain, there are many details to consider deeply: How to construct the complete WPPG; How to build linkage with other safety equipment; How to handle misleading action, and so on.

Acknowledgments

This paper is supported by the National Science Foundation of China (61340059) and science and technology key projects of He'nan province (142102210081).

References

[1] Arunkumar R, Annalakshmi A 2014 *International Journal of Computer Applications* **85**(8) 8-15

[2] Bashir U, Chachoo M 2014 *2014 International Conference on Computing for Sustainable Global Development (INDIACom)* 806-9

[3] Geib C W, Goldman R P 2001 *Proceedings of DARPA Information Survivability Conference and Exposition (DISCEX) Anaheim CA* 46-55

[4] Wen-xiang G, Lei W, Yong-li L 2005 *Proceeding of 2005 International Conference on Machine Learning and Cybernetics* 1 225-30

[5] Geib C, Goldman R 2009 *Artificial Intelligence* **173**(11) 1101-32

[6] Ying L, Wen-Xiang G 2013 *Optik - International Journal for Light and Electron Optics* **124**(21) 4823-6

[7] Zengyu C, Yuan F, Jianwei Z, Baowei Z *2012 IEEE Symposium on Robotics and Applications (ISRA)* 961-3

[8] Ramanujan R, Sabharwal A, Selman B 2010 *Proceedings of 20th international conference on automated planning and scheduling (ICAPS-10) Toronto Ontario Canada* 242-5

[9] Brian K, Alan F, Jesse H 2013 *Proceedings of International Conference on Automated Planning and Scheduling (ICAPS-2013) Rome Italy* 322-6

[10] McDermott D, Handler J 1995 *Artificial Intelligence* **76**(1-2) 1-16

[11] Jensen R, Veloso M, Bowling M 2001 *Proceedings of the Sixth European Conference on Planning Toledo Spain* 136-42

[12] Willmott S, Richardson J, Bundy A2001 *Theoretical Computer Science* **252**(1) 45-82

[13] Jiang YF, Ma N 2002 Journal of Software **13**(4) 686-92 (*in Chinese*)

[14] Nau D S, Au T C, Ilghami O, et al. 2003 *J. Artif. Intell. Res.(JAIR)* **20**(1) 379-404

[15] Sohrabi S, Baier J A, McIlraith S A 2009 *Twenty-First International Joint Conference on Artificial Intelligence Pasadena California USA* 1790-997

Authors

Zengyu Cai, 1979, Xinyang of He'nan Province, China.

Current position, grades: lecturer at Zhengzhou University of Light Industry.
University studies: Master in computer application technology from Northeast Normal University in 2006.
Scientific interest: plan recognition and information security.
Publications: 2 papers.

Qikun Zhang, 1980, Xinyang of He'nan Province, China.

Current position, grades: lecturer at Zhengzhou University of Light Industry.
University studies: PhD in computer application technology from Beijing Institute of Technology in 2013.
Scientific interest: information security and cryptography.
Publications: 1 papers.

Ran Zhang, 1973, Wugang of He'nan Province, China.

Current position, grades: associate professor at Zhengzhou University of Light Industry.
University studies: PhD in computer architecture from Xi`an Jiaotong University in 2003.
Scientific interest: network security, distributed computing and privacy.
Publications: 2 papers.

Yong Gan, 1965, Zhuzhou of Hu'nan Province, China.

Current position, grades: professor at Zhengzhou University of Light Industry.
University studies: PhD in computer application technology from Xi`an Jiaotong University in 2013.
Scientific interest: information security, multimedia communications and network engineering.
Publications: 5 papers.