http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (14): 2712-2718, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.2712.2718

Research on Predicting Attack Paths Based on Bayesian Inference

YunFeng-Wang and Hui-Wang
College of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, 45400, Henan, China

Abstract: In order to eliminate path redundancy and improve the calculation accuracy of node belief in attack graphs, a novel model of attack feasibility is proposed; then a method for arriving at a calculation of node belief based on the Bayesian inference is designed. By analyzing the cost-benefit on child attack paths, this study presents an algorithm generating attack paths to eliminate path redundancy as far as possible and then puts forward an improved likelihood weighting algorithm on Bayesian inference to predict attack paths by computing node belief .The finally experimental results show that this method can effectively eliminate the path redundancy, evidently improve the calculation accuracy of node belief and consequently enhance the validity of prediction for attack paths.

Key words: Attack graph, path redundancy, attack feasibility, likelihood weight

INTRODUCTION

With the increasing complexity of network system, network attacks are gradually becoming automatic and complicated. On one hand, some attack tools can keep loopholes invasion disguised as part of the scanning system and make it run automatically. On the other hand, it has ranged from a single step to intricate multi-steps. However, passively defensive technologies, such as firewall, have been proved to be insufficient for them. Thus, to strengthen the security capability of network, some actively defensive technologies predicting network attacks have been developed.

Attack graphs can contain all of attack paths and has a good expansibility. Therefore, they can timely and effectively deal with complicate attacks (Ghosh and Ghosh, 2012) and are widely utilized in predicting complex network attacks in life. These advantages make attack graphs great tools to effectively predict all possible attack paths. In recent years, researches on them are particularly concerned with forecasting attack paths by calculating attack costs and node belief.

RELATED RESEARCH

Here, probabilistic approaches have been employed in performing such analysis. A quantitative model (Homer *et al*., 2009) is presented to objectively measure the probability of nodes which are shared by multiple attack paths and can be accomplished. This model may ensure that different paths have a proportional effect on the final calculation of the node belief by combining the Common Vulnerability Scoring System (CVSS) with probabilistic reasoning. However, the influence that path redundancy has on other nodes falls into negligence, thereby constituting an obstacle to calculating node belief.

An approximate Bayesian posterior inference algorithm (Zhang and Song, 2011) is improved to calculated node belief. To some extent, it can effectively predict attack paths. However, it only places its main emphasis on the observed dynamic data. Thus, path redundancy will be generated when some attacks in disguise are launched. In this case, above method will reduce the calculation accuracy of node belief.

For each attack, there is always some costs and benefits during its occurrence. According to this idea, A risk management framework (Poolsappasit *et al*., 2012) using Bayesian networks is presented to quantify the chances of network compromise at various levels. By comparing the total costs and overall benefits of all attacks in each attack path, it will finally choose paths that have the minimum costs and maximum benefits as the optimal ones. In spite of taking the dependency of "or" and "and" into account, it ignores the hybrid of them in some special circumstances. Therefore, it will generate path redundancy.

Hidden Markov Model (HMM) (Wang *et al*., 2013) is put forward to explore the relationship between system observations and states. The model runs quite well in eliminating the path redundancy by analyzing the probabilistic relationship among different types of nodes.

Corresponding Author: Hui Wang, College of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, 45400, Henan, China

Simultaneously, the limitation of the relationship among the same types of nodes exists. Therefore, when the relationship among the same types of nodes isn't considered, it will produce path redundancy and have a bad effect on the calculation accuracy of node belief.

To address above problems, this study makes some major contributions. Firstly, it proposes a model calculating values of attack feasibility and designs an algorithm eliminating all possible path redundancy. Then a likelihood weighting algorithm calculating node belief is improved.

MODEL OF NETWORK ATTACK GRAPH AND THE DEFINITION OF ATTACK PATH

To describe attack paths, definitions of attack graphs and attack paths become the cornerstone of this section.

Definition of network attack graph: Generally, the network attack is an intricate multi-steps process consisting of some basic attacks that are closely related. By running it, attackers may make a change in status of network resources for expected benefits. Simultaneously, they must pay the price (We call it Attack Cost) for their attacks. In some sense, it is also an economic behavior.

So, we can attempt to use network attack graphs to describe above three components (network resources, attacks, costs and benefits) and analyze their causal relationship. Based on the above analysis, an attack graph can formally be defined as.

Definition 1: An attack graph is a directed acyclic graph with one or more AND - OR nodes NAG = (V, V_0, V_G, A, E, W, I, P, II), where:

- $V = \{v_i | i = 1, 2, 3, YN_i\}$ is a set of nodes standing for network resources and it's a finite and non-empty set of AND-OR nodes. Here, v_i is a variable signifying resource nodes and the value of it represents status of resources. The value of each node variable v_i can be either True or False, denoting whether the node has been taken over by attackers;
- $V_0 \subseteq V$ is a subset of V. It denotes a set of nodes denote resources that attackers may take over initially. Graphically, it is the set of root nodes in NAG
- $V_G \subseteq V$ is a subset of V. It represents a set of target nodes that attackers are trying to hold ultimately
- $A = \{a_j | j = 1, 2, 3, YN_j\}$ is a set of nodes standing for attacks. It's also a finite and non-empty set of AND-OR nodes. Here, a_j is a variable representing attack nodes and the value of it represents status of attacks. The value of each node variable a_j can be either True or False, signifying whether attacks have been conducted by attackers
- $E = \{E_1 \cup E_2\}$ is a set of edges linking all nodes. $e = <n_1, n_2>$ stands for a directed edge that indicates the flow from n_1 to its child node n_2.E_1 = VHA = $\{e_{ij} | e_{ij} = <v_i, a_j>, i = 1, 2, 3,..N_i; j = 1, 2, 3, YN_j\}$ is a set of edges which remark attacks can only be conducted given that all the prerequisite resources are occupied by attackers; AHV = $\{e_{ji} | e_{ji} = <a_j, v_i>, i = 1, 2, 3, Yn_i; j = 1, 2, 3, YN_j\}$ is a set of edges which denote attacks may consequently let attackers own some other resources. In this study, we generally define Pre(n) and Con(n) as the superset and subset of node n
- W is a set of attack weight. w is a variable in W. Here, $\forall w \in W$ links all nodes together and is described as a two- tuple of (h, m). h represents attack costs from e_{ji} and m remarks the benefits corresponding to h
- I is a set of attack indicators that denote status of attacks and consists of a single variable f.f = {⊙, ◎} indicates the two status of e_{ji}. f_{ji} = ⊙ signifies that attackers may be likely to attack v_i by conducting a_j and f_{ji} = ◎ denotes that attackers may let go of attacking v_i by conducting a_j
- $P = (P_1 \cup P_2)$. P_1 is the conditional probability distribution that the attack a_j will be conducted. Only if some Pre(a_j) are taken over by attackers can a_j be conducted. So, $P_1 = \{p: (Pre(a_j), a_j) \rightarrow [0,1]\}$. Analogously, P_2 is the conditional probability distribution that a_j has been conducted successfully. In other words, it is equivalent to possessing its some prerequisite nodes. So $P_2 = \{P_2: (a_j, Con(a_j)) \rightarrow [0,1]\}$
- $II = \{V \cup A\} \rightarrow [0,1]$ is the probability distribution of node belief in NAG. $\pi(v_i)$ denotes the probability that attackers have occupied node v_i; $\pi(a_j)$ denotes the probability of that attack a_j has been conducted. In the study, we can expect $\pi(v_0) = 1.0$ and for all nodes n, there is always $\pi(n)0$. Because v_0 has been taken over by attackers initially

Definition 2: AND nodes specifically refers that the relationship is "AND" among all nodes in Pre(n).It means that only if all the nodes in Pre(n) are occupied successfully can node n be conducted.

Definition 3: OR nodes specifically refers that the relationships are "OR" among all nodes in Pre(n).It means that as long as one or more nodes in Pre(n) are occupied successfully can node n be conducted.

Based on above definitions, we can eventually get a network graph as Fig. 1. Firstly, directed edges are used

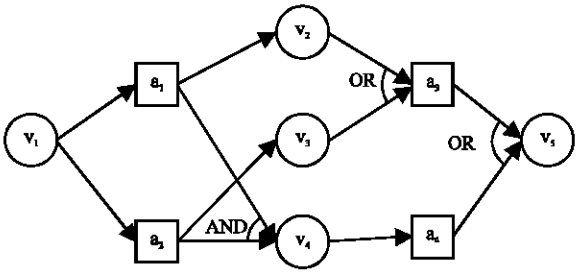

Fig. 1: A typical network attack graph

to denote the relationship between network resources and attacks. Then, the backbone structure of the target network is completed with taking 4 situations of AND-OR nodes into consideration. Finally, we will give values of P_1 and P_2 based on expert experience.

In Fig. 1, attackers initially occupies the node v_1 (with probability defined as $\pi(v_1)$). Then, attacks a_1, a_2 and a_3 will be conducted successively (with probabilities defined in P_1) and further may keep attackers taking over nodes v_2, v_3, v_4 (with probabilities defined in P_2).With all attacks conducted, the target node v_5 will be occupied finally.

Definition of attack path

Definition 4: We assume that there is an orderly sequence of nodes $v_0 \rightarrow a_1 \rightarrow v_1 \rightarrow a_2 Y a_j \rightarrow v_i Y v_G$ in attack graphs. For any two adjacent nodes in it, if they can make up an edge contained in the set of E. Then, above orderly sequence of nodes is one attack path.

Graphically, there are three attack paths in total if attackers want to occupy node v_5 (symbol indicates the relationship of AND among nodes) in Fig. 1:

Step 1: $\{<v_1, a_1>, <a_1, v_2>, <v_2, a_3>, <a_3, v_5>\}$
Step 2: $\{<v_1, a_2>, <a_2, v_3>, <v_3, a_3>, <a_3, v_5>\}$
Step 3: $\{<<v_1, a_1>, <v_1, a_2>, v_4>, <v_4, a_4>, <a_4, v_5>\}$

COST-BENEFIT ANALYSIS AND THE GENERATION PROCESS OF ATTACK PATH

Here, we analysis costs and benefits of one child attack path and propose a model calculating the value of attack feasibility.

Cost-benefit analysis of attack path

Cost of the child attack path: There are two components of attack cost (this study used $Cost(e_{ji})$ to represent it): Risky cost and operation cost. Here, a mathematical model (Wang and Liu, 2006) of operation cost is proposed.

$$Cost(e) = \alpha Hcost(Meta\text{-}operations)+\beta Hcost(Sepuence) \quad (1)$$

Operation cost can comprise cost(Meta-operations) and cost(Sepuence) and is the sum of them. Sequentially, we may use symbol e to denote the operation cost.

In fact, risk cost always correlates with risk coefficient (We shall denote it by θ), attackers experience and operation cost and is greatly used to measure the possibility that attacks may be conducted successfully. On one hand, it depends on the influence coefficient of attack targets (We shall denote it by M). The more important targets are, the more likely attacks are conducted and the more easily attacks are detected. On the other hand, it depends on the influence coefficient of attacks themselves (We shall denote it by Γ). The more complicated attacks are, the more likely targets can be taken over. So, the risk coefficient model is defined as follows:

$$\theta(e) = \Gamma(a_j)HM(v_i) \quad (2)$$

The more complicated the attack complexity (this study uses operation cost to represent it) is, the more time attacks will cost and the more likely attacks will be unsuccessful. With the increasingly accumulation of experience, attackers will cost lower risk cost. Because they will realize how to low the attack complexity and appetite for risk. So we can give the risk cost model as follows:

$$Cost = Cost(e)\times X(e)^{time-1}\times\theta(e) \quad (3)$$

In model 3, Cost(e) is the operation cost, $X(e)(X(e)<1)$is the experience coefficient, time is the number of attacks and $\theta(e)(\theta(e)>1)$ is the risk coefficient. From it, we can draw a conclusion that the variation of risk cost with changes of variables follows the same trends as the fact .So we can give a model as follows:

$$Cost(e_{ji}) = \varepsilon HCost(e)+\mu HCost(\varepsilon \text{ and } \mu \text{ denote attack weight}) \quad (4)$$

Model of attack feasibility: During the occurrence of attacks, attackers must pay the price for their attacks. Simultaneously, they can also get some expected benefits (m). Generally, only if the expected benefits are more than the actual costs will attackers launch attacks in comprehensive consideration. Hence, the definition of attack feasibility model is as follows:

From above definition, we realize that Δ is earnings yield of costs that attackers will spent. Only if the expected benefits are more than the actual costs, namely $\Delta > = 0$ will attackers launch attacks on the target networks.

Generation process of attack paths

Definition 5: For any two adjacent nodes m and n in attack graphs, if a directed edge that indicates the flow from m to n is between m and n, then there is a partial order relation between them and the partial order relation is recorded as <m, n>. Here, we can define the set of partial order relations as Partial Order Set (POS).

A comparison of STEP with POS obviously shows that the edge <m, n> is a common basic component of them. It means that we can gather all partially order relations into STEP by iterating through all elements in POS. The process of forming STEP is recorded as STEP⊲POS.

To demonstrate the path-forming process, this study will assign attack weight w to every edge eji based on above mathematical model and the expert experience at first and then cuts off the edge that its starting node is v0 and gathers it into POSi. And like the method above, we can gather all partially order relations into POS by the turns of the topological sort ψ. The attack graph assigned attack weight w is as Fig. 2 and its process is as follows (we assume that the topological sort):

$$\psi = \{v_1, a_1, a_2, v_2, v_3, v_4, a_3, a_4, v_5\}$$

A comparison of step 1, 2 and 3 with three attack paths forming based above method shows that the line sequence relations is equal and can low the complexity forming attack paths and that this method is valid and correct. So, the mentality will be expressed as the core of algorithm1.

Improved likelihood weighting algorithm based on bayesian inference: The network attack graph is a Bayesian network (Guo and Zhang, 2006) essentially. So, we can calculate node belief by running likelihood-weighting sampling. The specific operations obtaining samples are as follows:

Firstly, it provides ρ as the topological sort for sampling of each variable. Secondly, it will give sampling results to each variable: in the process of sampling, if one sampled variable is an evidence variable which has already known, the sampling results will be assigned by the distribution of p(X); if the sampled variable is not an evidence variable, then sampling results will be assigned by the distribution of p(x|pre(x)). Then, we will assign the sampling weight ω_e to evidence variables E = e and weight $\omega_{q,e}$ to the query variables Q=q and we will get the posteriori probability $p(Q = q|E = e) \approx \omega_{q,e}/\omega_e$.

Traditional algorithm calculating the probability of nodes has a great limitation. Here, we can take v_i as an example to specifically illustrate limitation. Generally, the

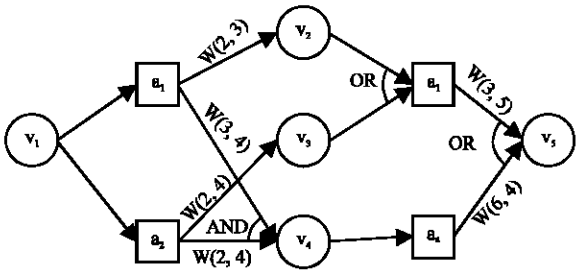

Fig. 2: Generating diagram of line sequence relations. Firstly, the edges $v_1 \rightarrow a_1$ and $v_1 \rightarrow a_2$ should be cut off. Recording, $POS_1 = \{<v_1, a_1>, <v_1, a2>\}$, The edge $a_1 \rightarrow v_2$ should be cut off. Recording: $POS_2 = POS_1 \cup \{<a_1, v_2>\}$, The value of attack feasibility $<a_1, v_2>$ is (3-2)/2>, so $f_{12} = \odot$, The edge $a_1 \rightarrow v_4$ should be cut off and the relationship between a_1 and a_2 is AND. Recording: $POS_3 = POS_2 \cup \{<a_1, v_4>\} \cup <a_1 \wedge a_2>$, The value of attack feasibility $<a_1, v_4>$ is (4-3)/3>0, so $f_{14} = \odot$, The edg $a_2 \rightarrow v_3$ should be cut off. Recording: $POS_4 = POS_3 \cup \{<a_2, v_3>\}$, The value of attack feasibility $<a_2, v_3>$ is (4-2)/2>0, so $f_{23} = \odot$, The edge $a_2 \rightarrow v_4$ should be cut off. Recording: $POS_5 = POS_3 \cup \{<a_2, v_4>\}$, The value of attack feasibility $<a_2, v_4>$ is (4-2)/2>0, so $f_{24} = \odot$, Secondly, the edges $v_2 \rightarrow a_3$, $v_3 \rightarrow a_3$ and $v_4 \rightarrow a_4$ should be cut off by the order of v_2, v_3, v_4. Recording: $POS_6 = POS_5 \cup \{<v_2, <v_3, a_3>, <v_4, a_4>\}$, The edge $a_3 \rightarrow v_5$ should be cut off. Recording: $POS_7 = POS_6 \cup \{<a_3, v_5>\}$, The value of attack feasibility $<a_3, v_5>$ is (5-3)/3>0, so f35 = ⊙, The edge $a_4 \rightarrow v_5$ should be cut off. Recording: $POS_8 = POS_7 \cup \{<a_4, v_5>\}$, The value of attack feasibility $<a_4, v_5>$ is(4-6)/6< 0, so $f_{45} = \circledcirc$, Thirdly, STEP can be obtained by iterating through all edges in POS, Iterating through all edges $<a_j, v_i>$ that value of attack indicator is ⊚, Implanting ⊚ to all edges $<a_j, v_i> \epsilon E_2$ where edge $<a_j, v_i>$ is in step 1, Outputting the set of line sequence relations step 1, Step 1: $\{<v_1, a_1>, <a_1, v_2>, <v_2, a_3>, <a_3, v_5>\}$, Step 2: $\{<v_1, a_2>, <a_2, v_3>, <v_3, a_3>, <a_3, v_5>\}$ and Step 3: $\{<<v_1, a_1> \wedge <v_1, a_2>, v_4>, <v_4, a_4>, <a_4, v_5>\}$

value of π(vi) is always equal to 0 when attackers let go of attacking vi. However, because of the randomness inherent to the sampling, the value of π(vi) is greater than zero in some cases and has a bad effect on the calculation accuracy of node belief. To solve it, this study improves the traditional algorithm. If the value of π(x) is always equal to 0, we will assign a fixed value False to X before sampling. It can ensure that the sampling process is

always carried out in accordance with the corresponding probability distribution. The improved algorithm is as algorithm 2.

Algorithm 1: Line Sequence Relations STEP(NAG,STEPi)

Input: NAG- a network attack graph
W-attack weight
STEP-a set of line sequence relations
E_2-a set of edges $<a_j, v_i>$
POS-a set of partial order relations
M-a set of relationship AND
Ψ-a topological sort in NAG
X, Y-any one of nodes

Output: STEPi-a set of line sequence relations in NAG

01. ψ←NAG
02. POSi←Ø, STEPi←Ø, M←Ø.
03. FOR (each node variable X inψ)
04. To find node variable Y that has a partial order relation with X.
05. IF (<X,Y>E2)
06. IF Δ(<X, Y>)0
07 I(<X,Y>)←
08. ELSE
09. I(<X,Y>)←
10. END IF
11. END IF
12. POS_i←POS_i←{<X, Y>}
13. IF (The relationship among the nodes Pre (X) is AND)
14. M←M←{Y_1, Y_2}
15. END IF
16. POS_{i+1}←POS_i←M
17.END FOR
18. STEP<POS
19. Iterating through all edges $<a_j, v_i>$ that the value of attack indicator is ;
20. IF (e_{ji}∈(E_2←step 1) and f_{ji} =)
21 e_{ji}∈(E_2→step 1), f_{ji} = ;
22 END IF
23.RETURN STEPi

The 3nd line of the pseudocodes is a loop control statement which will drive above algorithm to generate the partially order set POS in one run. In it, pseudocodes3~11 is to generate some edges which have line sequence relations and to add some values of attack indicator into them; Pseudocodes11~16 is to generate the partially order set POS. The pseudocodes18~23 is to return a set of STEP. Through running the process STEP POS, the STEP gathering all partial order relations will be returned.

Algorithm2: improved Likelihood Weighting (NAG, m, E, e, Q, q, ρ)

Input: NAG-a network attack graph;
m-effective sample number to generate;
E-a set of evidence nodes;
e-a value of evidence node
Ω-a set of query variable nodes
q-a value of query variable node
ρ-a topological sort in NAG

Output: The approximation of p(Q = q|E = e)

01.ρ→NAG
02.i→0, ω_e←0, ω_q, e<←0
03.WHILE (i<m)
04. Di←Ø
05. FOR (each node variable X in ρ)
06. EX←AbandonNodeSelect (step 1, E2, U, IU)
07. IF (X←Ex)
08. x←False
09. ELSE
10. IF (x←E) then
11. MarkX as sampled
12. ELSE
13. x←the sampling result according to p(X|Pre(X))
14. END IF
15. END IF
16. END FOR
17. D_i←D_i←{X = x}
18. $\omega_i \leftarrow \prod_{X \in \bar{E}} P(X \mid \pi(X))|_{D_i}$
19. ω_e←ω_e+ω_i
20. IF (Q = q←Di)then
21. $\omega_{q,e}$←$\omega_{q,e}$+ω_i
22. END IF
23. i←i+1
24.END WHILE
25.RETRUN $\omega_{q,e}/\omega_e$

By running algorithm 2, a probability distribution which contains all probabilities of nodes will be returned. Ulterior, this procedure can be divided into two stages:

- **Initialization (01-02):** In this stage, it will provide a topological sort ρ and initialize i, ω_e and $\omega_{q,e}$
- Nodes sampling (03-24). This stage is composed of 3 parts. The 3nd line is the first part and is an outside loop control statement which can drive the algorithm to sample all nodes in NAG in one run. Pseudocodes 5-16 is the second part. It's an inner loop. In it, pseudocodes 6~08 is to set the value False to the nodes which are from the algorithm 3 AbandonNodeSelect. Then, in line 9~11, root nodes are sampled according to the initial probability distribution. Finally, in line 12-13, other nodes are sampled by the distribution of p(x|pre(x)). Pseudocodes17-24 is the third part. It is used to compute all the weight of samples. In line 25, the result of $\omega_{q,e}/\omega_e$ will be returned eventually

Algorithm 3: Abandon node select (step 1, E2, U, IU)

Input: step 1-a set of line sequence relations in NAG
E2-a set of edge $<a_j, v_i>$
U-any one of variable in E_2
X-one resource status node corresponding with U(the relation between X and U is called X U)
IU-a status of U

Output: EX-a set of nodes given out

01.EX←Ø
02.FOR (each variable in E_2)
03. IF (IU = ? and X> U)
04. EX←EX←{X}
05. END FOR
06.RETURN EX

By running this function, we can get a set of nodes that attackers will abandon to select. It may ensure that the sampling process in algorithm 2 is always carried out in accordance with the corresponding probability distribution.

NODE BELIEF COMPUTATION EXAMPLES

Comparison with bayesian inference: Here, Java programs are designed and implemented to perform the following experiments:

Firstly, we use the traditional Bayesian inference algorithm to calculate the posterior probability of nodes illustrated in Fig. 1. The result is listed in Table 1 (i is the attack paths and the number of effective samples is 2000).

Then, we use algorithm 2 to calculate the posterior probability of nodes in Fig. 2. The result is listed in Table 2 (i is attack paths and the number of effective samples is 2000).

Table 1: Classical bayesian inference result

i	$\pi(v_1)$	$\pi(v_2)$	$\pi(v_3)$	$\pi(v_4)$	$\pi(v_5)$	$\pi(a_1)$	$\pi(a_2)$	$\pi(a_3)$	$\pi(a_4)$
1	1.000	0.633	--	--	0.655	0.792	--	0.697	--
2	1.000	--	0.650	--	0.655	--	0.800	0.697	--
3	1.000	--	--	0.507	0.655	--	0.800	--	0.400

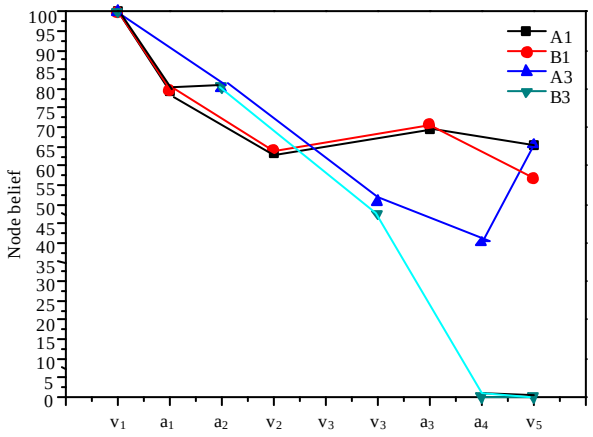

Fig. 3: Comparison diagram of node belief in step 1 and 3

The node belief calculated by algorithm 2 is different from the traditional one. To clearly observe the difference between them, we draw the comparison diagram of node belief in step 1 and 3 by the data of Table 1 and 2. The comparison diagram is as Fig. 3. (Here, letters A is the traditional algorithm and B is the improved algorithm and attack paths are represented in digital form.)

In Figure 2,Two basically experimental results can be seen: (1)Both π(v5) and π(a4) are equal to 0 in step 3 and (2) Node belief calculated by the improved algorithm is more than the traditional one in step 1. On one hand, all attack paths will not be abandoned in traditional algorithm because of not considering the path redundancy. In other words, none belief of nodes is 0.While it's not in the improved algorithm, so Fig. 2 shows the first experimental result. It means that the path redundancy has been eliminated effectively. On the other hand, in traditional algorithm, the influence that path redundancy has on other nodes falls into negligence. However, the negligence is not in the improved algorithm, so Fig. 2 shows the second experimental result. It means that the calculation accuracy of node belief is improved. So, we may draw conclusions: The path redundancy is eliminated effectively and the calculation accuracy of node belief is improved evidently.

Prediction about attack path: Here, the samples in STEP1 and STEP3 are analyzed according to the different sampling time. The probability of P(v5|v1) is the statistics of sampling results and P(v5 = True|v1 = True) is the searching probability. Θ_i is the sampling probability of nodes in STEPi and Mi is the sampling time. The sampling results are shown as Table 3 and the prediction of attack paths is shown as Fig. 4.

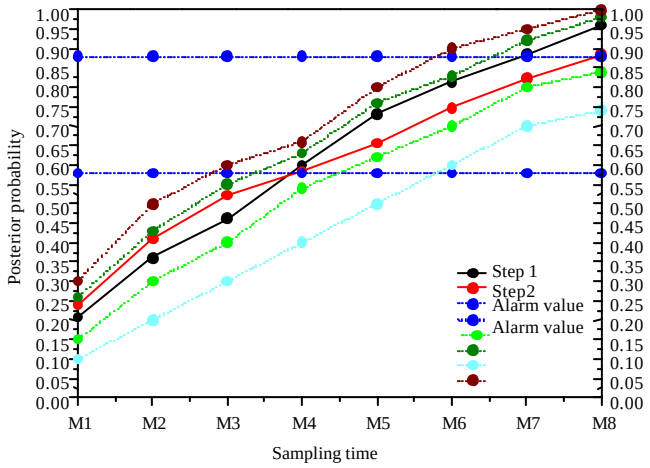

Fig. 4: Prediction diagram of attack path

Table 2: Improved likelihood weighting inference result

i	p(v1)	p(v2)	p(v3)	p(v4)	p(v5)	p(a1)	p(a2)	p(a3)	p(a4)
1	1.000	0.644	--	--	0.567	0.798	--	0.707	--
2	1.000	--	0.672	--	0.567	--	0.807	0.707	--
3	1.000	--	--	0.475	0.567	--	0.807	--	0

Table 3: Sampling result

	M1	M2	M3	M4	M5	M6	M7	M8
Θ1	0.2070	0.3600	0.4624	0.5996	0.7310	0.8145	0.8845	0.9606
Θ2	0.2401	0.4096	0.5220	0.5850	0.6561	0.7480	0.8237	0.8852

In Fig 4, it shows that the attack probability is gradually increased with the different sampling time. The reason is that the resources and experience attackers own will be much more with the advancement of attack process.

To predict attack paths that attackers will conduct, this study sets some alarm values. In Figure4, we assume that the alarm values are 0.88 and 0.58.When the attack probability is above 0.88, STEP1 will be the attack path which attackers select. When the attack probability is lower than 0.58, step 3 will be the attack path which attackers select. When the attack probability is between 0.88 and 0.58, both step 1 and 3 will be likely the attack paths which attackers select. Similarly, we can set more alert values to predict multiple attack paths.

CONCLUSION

Although, attack graphs are ideal tools predicting attack paths, it imperfectly exists some limitations (1) There may exist path redundancy in it and (2) There is lower accuracy of node belief for the prediction on attack paths. We address above limitations by proposing a model calculating values of attack feasibility and developing an improved likelihood weighting algorithm calculating node belief. The model is mainly used to eliminate path redundancy as far as possible and the improved algorithm is chiefly used to improve the accuracy of node belief. As a result, the path redundancy is eliminated effectively, the calculation accuracy of node belief is improved evidently and the validity of prediction for attack paths is enhanced consequently.

ACKNOWLEDGMENTS

This project is supported by the National Natural Science Foundation of China (No. 51174263), supported by Research Fund for the Doctoral Program of Higher Education of China (No. 20124116120004), supported by the Doctor Funds of Henan Polytechnic University (No. B2010-62) and supported by Educational Commission of Henan Province of China (No. 13A510325).

REFERENCES

Ghosh, N. and S.K. Ghosh, 2012. A planner-based approach to generate and analyze minimal attack graph. J. Applied Intell., 2: 369-390.

Guo, H.P. and L.W. Zhang, 2006. Introduction to Bayesian Networks. Science Press, Beijing, China, pp: 55-59.

Homer, J. X. Ou and D. Schmidt, 2009. A sound and practical approach to quantifying security risk in enterprise networks. J. Kansas State Univ. Tech. Rep., 3: 1-15.

Poolsappasit, N., R. Dewri and I. Ray, 2012. Dynamic security risk management using Bayesian attack graphs. IEEE Trans. J. Dependable Secure Comput., 1: 61-74.

Wang, H. and S.F. Liu, 2006. A scalable predicting model for insider threat. Chin. J. Comput. Chin. Edn., 08: 1346-1355.

Wang, S.Z., Z.H. Zhang and Y. Kadobashi, 2013. Exploring attack graph for cost-benefit security hardening: A probabilistic approach. Comput. Security, 32: 158-169.

Zhang, S.J. and S.S. Song, 2011. A novel attack graph posterior inference model based on Bayesian network. J. Inform. Security, 2: 8-27.