IOP Conf. Series: Materials Science and Engineering **719** (2020) 012065 doi:10.1088/1757-899X/719/1/012065

A Prediction Method of Node Attack Path based on Optimized-AG

Kunfu Wang, Wei Feng and Wanfeng Mao

System Engineering Research Institute of China State Shipbuilding Corporation, 1 Fengxiann East Road, Haidian District, Beijing P.R. China
Email:wangkunfu1166@163.com

Abstract. An optimization algorithm for reducing the scale of attack graphs and an attack path prediction method are presented herein to cover the difficulty of predicting the attack path in traditional attack graphs. Firstly, a new distance cost attack graph model is generated and converted into a marked attack graph via the algorithm. Secondly, the original attack graph is optimized by introducing an attack distance weight value and a path redundancy coefficient in order to determine the possible attack paths and calculate the cost of path attack behavior. Finally, the preferred attack path is determined based on an evaluation function. According to the experimental results, the above scheme eliminated the path redundancy of attack graphs and achieved comprehensive prediction of attack paths.

1. Introduction

As the internet industry advances rapidly, IT plays an important role in enterprises and the enterprise intranets are facing increasing threats. According to CSI/FBI’s Computer Crime and Security Survey [1] for consecutive five years, more than 85% of the damages resulting from computer security accidents are incurred by internal security vulnerabilities instead of viruses. The Development Status of China Internet Sites and Security Report in 2018 [2] illustrates a well-known case of the virus WannaCry specific to enterprises in May 2017. The attacker spread the virus via dangerous vulnerabilities. On this basis, future network attacks are likely to be more violent and have stronger impact on enterprises. In the Review of China’s Internet Network Security Situation in 2017 [3], the number of security vulnerabilities listed in China National Vulnerability Database (CNVD) rises continuously in recent years. Since 2013, the number of vulnerabilities listed in CNVD has grown by an average of 21.6% per year.

In recent years, researchers have built various attack graph models by analyzing vulnerability information of intranets [4] and created attack paths using the resource information and attack behavior in the network [5].

2. Related Research

Kaynar K[6] proposed an attack graph modeling method to simulate potential attack paths in the network. However, the method could only be used to obtain data through simulation experiments and was incapable of changing with the real environment. Muñozgonzález L[7] et al. used the Loopy Belief Propagation algorithm to perform static and dynamic analysis and drew a conclusion that the number of attack nodes expanded linearly. Even through this method could be applied to evaluate the risks of system damages in case of vulnerability interconnection, it was impossible to reduce the size of an attack and failed to provide any reference for analyzing the internets of larger enterprises.

Poolsappasit N[8] et al. were based on the Bayesian network and CVSS system and proposed to quantify the node confidence to costs and benefits. Then, they compared the cost and benefit values of different attack paths to optimize the Bayesian attack graph and select a preferred attack path. However, they failed to take into account the path redundancy in attack graphs, thus the path filtering in the attack graph was affected.

Man D[9] et al. developed a global attack graph generation algorithm based on the breadth-first search, which reduced the scale of an attack graph by limiting the attack steps and the probability of success of attack paths. Although they achieved some effects, attack costs and other factors were not included in the optimized attack graph and the predication of preferred attack paths was incomplete.

Wu Hongrun [10] et al. proposed an attack path prediction method based on attack costs in the analysis of network attacks. However, they ignored the fact that the attacker acted as an intelligent agent while launching a network attack and would integrate the factors of attack distance and cost of an attack path. Moreover, they only took the attack cost into account and could not predict the attack path accurately.

3. Establishment of Network Attack Graph Model NETAG

The existence of network resources (vulnerabilities) is the reason why an attacker launches attacks. Attackers gain permissions by exploiting the vulnerability information on the network and implement basic attacks to change the status of network resources, thereby occupying the desired resources. Then, they will attack the network again until they gets what they want. Therefore, the following definitions need to be considered while creating an attack graph model:

Definition 1: Attack vulnerabilities in the network system are defined as vulnerabilities by which an attacker obtains appropriate permissions.

Definition 2: The complexity of an attack is an measure of an attacker's ability to obtain appropriate permissions successfully by virtue of these vulnerabilities. Due to a variety of factors, the attack complexity cannot be calculated precisely for each of these vulnerabilities, but some quantitative indexes can be used to illustrate the difference of attack complexity between these vulnerabilities approximately. The authors herein referred to the study of Chen Xiaojun[12] et al. on the complexity of attacking vulnerabilities and developed a quantitative standard of complexity in percentages upon changes according to the actual situation. The vulnerability is less likely to be attacked successfully when the complexity is closer to 100. The quantitative standard value are shown in Table 1.

Table 1. Quantitative Standard of Attack Complexity

Grade	Complexity	Description
1	10	No attack tools required and detailed attack methods
2	30	Available attack tools and detailed attack methods
3	50	Without attack tools but with relatively detailed attack methods
4	70	Report vulnerabilities publicly and refer to attack methods roughly
5	90	Report vulnerabilities publicly but not refer to attack methods

3.1. Definition of Network Attack Graph

A network attack is a complex multi-step process that involves a series of closing related basic attack behavior. An attack graph is a directed graph of possible attacks from the perspective of the attacker as well as a dependency graph composed by the attacker implementing attacks with a vulnerable host in the target network. Based on the above analysis, the following definitions are given:

Definition 3: The network attack graph $NETAG = (V, A, E, D, W)$ represents a five-tuple directed acyclic graph with one or more OR nodes.

- $V = \{v_i \mid i = 1,2,3,\ldots N\}$ represents the set of resource nodes: The system has open network services (i.e., attack vulnerabilities), and these attack vulnerabilities are also defined as a set of resource nodes.
- $A = \{a_j \mid j = 0,1,2,3,\ldots N\}$ represents the set of attack nodes: The attacker occupies the next resource

node by initiating attacks via the attack nodes, and this is a nonempty finite set of OR nodes.

- $E=\{E_A \cup E_V\}$ represents the set of directed edges connecting nodes: The set $E_A \subseteq A\times V$ is a collection of attack behavior edges from attack nodes to resource nodes, in which $e_{ji} =< a_j, v_i >$ and $e_{ji} \in E_A$. The set $E_V \subseteq V\times A$ is a collection of edges that resource nodes triggers attack nodes, in which $e_{ji} =< a_j, v_i >$ and $e_{ji} \in E_A$.
- D represents the set of attack distance weight: $D=\left\{d_{ji} \mid j=0,1,2,...,N\ , i=1,2,...,N\right\}$, and $a_0 \to v_1 \to a_2 \dots a_j$ for the sequence of sequence points, in which d_{ji} is an attack distance weight value assigned to the attack behavior edges $<a_j,v_i>$ given the difficulty of attacking the resource nodes when the attack nodes a_j attacks against the resource nodes v_i.
- W represents the set of attack permissions: Refer to Definition 10 for details.

Definition 4: OR node means that all father nodes Pre(n) of node n are in an OR relationship, i.e., the node n is triggered if any node in the Pre(n) set is satisfactory.

3.2. Generation and Conversion of Network Attack Graph

3.2.1. Generation of network attack graph

From the attacker's initial access to attack permissions, the attack graph model NETAG is required to contain the status of all accessible networks of the attacker in order to analyze the status of each node in the network accurately.

For this purpose, the authors herein proposed an attack graph generation algorithm for forward search[13] and the specific steps are described as follows:

Step 1: collect service information in the network, i.e., vulnerabilities (resource nodes), attack nodes and network topologies and other information.

Step 2: construct a sequence of attack sequence points based on the network vulnerability information and the vulnerability exploit rules.

Step 3: find the network vulnerabilities v_i corresponding to the attack nodes a_j.

Step 4: create an attack behavior edge e_{ji} that a_j attacks against and occupies v_i with the exploit rules and store it to E_A; create a edge e_{ji} that a_j can be triggered through v_i and store it to E_V.

Step 5: construct a network attack graph utilizing the combination of E_A and E_V according to the association between network statuses.

In order to detail the above steps of generating an attack graph, the authors herein put forward an attack graph generation algorithm FRG-Alg for forward search as follows:

Algorithm 1: An attack graph generation algorithm FRG-Alg (a_0, A and V) for forward search

Description: An attack graph is generated by virtue of the directed relationship between attack nodes and resource nodes.

Input: the initial attack node a_0, the attack node set A and the resource node set V.

Output: NETAG..

```
For (a sequence of attack sequence points)
  While ($A \neq \phi, V \neq \phi$)
    For each $a_j \in A$
      Search $a_j \to v_i$ //find the open service information $v_i$ of $a_j$
      Relationship $a_j$ and $v_i$ to Edge $e_{ji}$ //compose the edge $e_{ji}$ based on the
      $E_V \leftarrow e_{ij}$
      $\forall e_{ij}$ the nodes are in an OR relationship
    End For
End For
$E \leftarrow (E_A \cup E_V)$ //store elements in $E_A$ and $E_V$ to the edge set $E$
Relationship $e_{ij}$ and $e_{ji}$ //the edge $e_{ji}$
```

```
    relationship between $a_j$ and $v_i$
6) $E_A \leftarrow e_{ji}$ //store $e_{ji}$ to the edge set $E_A$
7) For each $v_i \in V$
8) Search $v_i \rightarrow a_j$
9) Relationship $v_i$ and $a_j$ to Edge $e_{ij}$
    related to $e_{ij}$
16) Rank $e_{ij}$ and $e_{ji}$ //arrange the sequence of $e_{ij}$ and $e_{ji}$
17) NETAG $\leftarrow e_{ij}$, $e_{ji}$;
18) Return NETAG
```

The attack graph NETAG generated by the above algorithm is as shown in Figure 1:

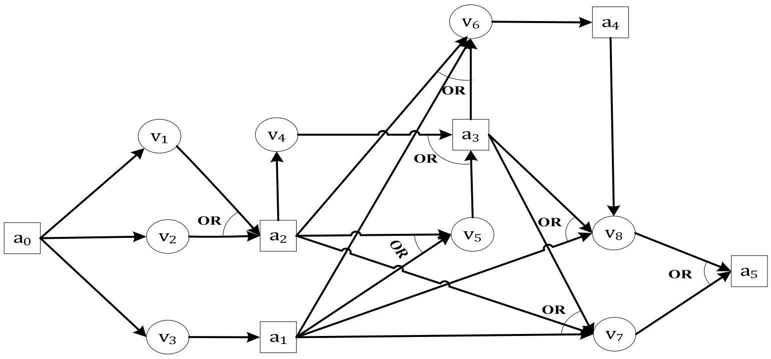

Figure 1. Initial Attack Graph NETAG

3.2.2. Description of attack path based on network attack graph

When an attacker attacks a network target, he/she attacks the resource node of the target first to change its status, and then attacks other resource nodes until the target node is occupied finally. The attack track of the attacker in this process is called an attack path. As shown in Figure 1, the attacker starts at the starting node a_0 and arrives at the destination node a_5 after passing through the nodes v_2, a_1, v_7, and the sequence composed by nodes a_0, v_2, a_1, v_7, a_5 in order is an attack path. Accordingly, an attack path is defined as follows:

Definition 5: If there is an ordered sequence of nodes $a_0, v_1, a_1, v_2, a_2 \cdots v_i, a_j$ from the starting node a_0 and any two neighbor nodes in the sequence meet $< v_i, a_j > \in E_V$ or $< a_j, v_i > \in E_A \left(0 < i < n, 0 \le j < n\right)$, the sequence of nodes $a_0, v_1, a_1, v_2, a_2 \cdots v_i, a_j$ is called an attack path of the network attack graph and expressed in $AStep$. All attack paths are set to $AStep_i \in AS$.

3.2.3. Conversion of network attack graph

In order to conduct further studies on the relationship between attack nodes and resource nodes in the attack graph and on the complexity of attack behavior (i.e., the assignment of distance weight of attack behavior edges), the authors herein converted the original attack graph and marked the converted attack behavior, thus different marks indicates different attack distance weight.

As v_i is a resource node and represents the open services (i.e., vulnerability information) of the host a_j, v_i is marked with a Bugtraq ID from the appropriate vulnerability database. d_{ji} refers to the attack distance weight value of the attack behavior edge $<a_j, v_i>$, i.e., the attack complexity of v_i. In this paper, the ID of vulnerability information v_i is assigned to the attack behavior edge $<a_j, v_i>$, and the ID corresponds to different attack distance weight d_{ji} by reference to the quantitative standard value of vulnerability information of the host. To describe the conversion process explicitly, a conversion algorithm AGC-Alg is designed for the original attack graph as detailed below:

Algorithm 2: A attack graph conversion algorithm **AGC-Alg** (NETAG, A, V, E_A and AS)

Input: NETAG, the attack node set A, and the resource node set V, the edge set E_A and the path set AS.

Output: the marked edge set $M\text{-}E_{\mathrm{A}}$, the attack permission set $D=\left\{d_{ji} \mid j=1,2,...,N, i=1,2,...,N\right\}$ and the marked attack graph M-NETAG.

```
$M\text{-}E_{\mathrm{A}} \leftarrow \emptyset$;
$D \leftarrow \emptyset$;
While( true) do
    For (each $AStep_i$ in $AS$)
        Search ID$\leftarrow v_i$; //Find the vulnerability ID corresponding to $v_i$
        Mark ID on $<a_j,v_i>$; //Mark edges with this ID
        $<a_j,v_i>' \leftarrow <a_j,v_i>$;
        Search Weight $\leftarrow$ID; //Find the attack distance weight corresponding to this ID
        $d_{ji} \leftarrow$ Weight;
        $D \leftarrow d_{ji}$; //Store the weight to the set $D$
        $<a_j,v_i>' \leftarrow d_{ji}$;
    End For
    $M\text{-}E_{\mathrm{A}} \leftarrow <a_j,v_i>'$; //Store the marked behavior edges to $M\text{-}E_{\mathrm{A}}$
    M-NETAG$\leftarrow M\text{-}E_{\mathrm{A}}$;
End While
Return M-NETAG
```

Upon the above algorithms, the converted attack graph M-NETAG is as shown below:

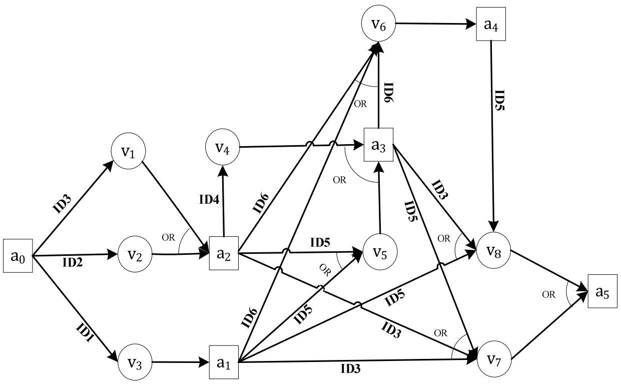

Figure 2. Marked Attack Graph M-NETAG

According to the converted attack graph, each attack behavior edge is assigned an ID to reflect the relationship between nodes. The value corresponding to the ID represents the distance weight from an attack node to a resource node.

3.3. Optimization based on M-NETAG

According to the experiment environment, attackers as an intelligent group prefer to select an easier attack path. But in the attack graph generated via the above algorithm FRG-Alg, there might be a large number of redundant attack paths. The authors herein analyzed the attack distance weight of accessible attack paths and introduced an attack path redundancy coefficient to optimize the above attack graph and reduce the scale of attack graph and the redundancy of attack paths. In this way, it is possible to analyze the optimized attack graph easily and prepare for the predication of the preferred attack path of attackers.

Definition 6: If d_{ji} is the attack distance weight of an attack behavior edge $<a_j,v_i>$ in the attack graph M-NETAG, the attack distance weight L_d represents the sum of weight of attack behavior edges in a single accessible path $AStep_i$ and is calculated as follows:

$$L_d = \sum d_{ji} \tag{1}$$

Definition 7: The maximum attack distance weight $L_{\max}$ represents the maximum attack distance weight among all attack paths $AStep_i$ from a_j to a_n if nodes a_j and a_n is accessible in the attack graph M-NETAG.

Definition 8: The minimum attack distance weight L_{min} represents the minimum attack distance weight among all attack paths $AStep_i$ from a_j to a_n if nodes a_j and a_n is accessible in the attack graph M-NETAG.

Definition 9: According to the Definitions 7 and 8, the attack path redundancy coefficient ξ represents the ratio of the maximum attack distance weight to the minimum attack distance weight among all attack paths $AStep_i$ from a_j to a_n and is calculated as follows:

$$\xi = \left(L_{\max} / L_{min}\right) \tag{2}$$

If larger ξ is obtained, there are more redundant paths in the attack graph, thus an attack distance weight threshold value γ is set to be divided by the redundant attack paths having an attack distance L_d greater than the threshold. The threshold γ is calculated as follows:

$$\gamma = \xi * \Delta + L_{min} \tag{3}$$

Δ represents the quantitative standard value of the path redundancy coefficient and is defined by an experienced expert.

According to the above definitions, the authors proposed an optimization algorithm AGO based on the attack graph M-NETAG. The optimized attack graph retains some effective attack paths for the reference of administrators. The attack graph optimization algorithm is detailed as follows:

Algorithm 3 An optimization algorithm **AGO** (d_{ji} and AS) for network attack graphs.

Description: The attack distance L_d is calculated for $AStep_i$ in the set AS of the attack graph M-NETAG and the attack paths with the L_d less than the threshold is sorted out. The L_d value of each path is marked using the vector $U = (U_1, U_2, \cdots U_i)$ and stored in the set ϖ in sequence.

Input: The distance weight d_{ji} of $<a_j, v_i>$ in the set D and the accessible attack path set AS

Output: the optimized attack path $OAStep_i$ and the path set OAS

```
$OAS \leftarrow \emptyset$
$\varpi \leftarrow \emptyset$
While (true) do
    For ( $AS$ each $AStep_i$ )
        $L_d = d_{ji}$++
        $U_i \leftarrow L_d$
        $\varpi \leftarrow L_d$
    End For
    Select $L_{\max}$, $L_{min}$ in $\varpi$
    Calculate $\xi = (L_{\max} / L_{min})$, $\gamma$
    Find $U_i \leftarrow ((L_d < \gamma)$ in $\varpi)$ //find the marks with weight less than the threshold
    $AStep_i \leftarrow U_i$ //find the corresponding path by the marks
    Reserve $v_i$ and $a_j$ in $AStep_i$ //reserve the attack and resource nodes in the attack path
    Store $V' \leftarrow v_i$, $A' \leftarrow a_j$
    Edge $a_j$ and $v_i$ to $e_{ij}$ and $e_{ji}$ //Compose an edge according to the relationship between $a_j$ and $v_i$
    Rank $e_{ij}$ and $e_{ji}$
    $OAStep_i \leftarrow (e_{ij}, e_{ji})$
    $OAS \leftarrow OAStep_i$
End While
Return $OAS$
```

According to the Algorithm 3 and all accessible attack paths from the initial attack node a_0 to the destination node a_5 in Figure 2, the minimum attack distance weight, maximum attack distance weight, path redundancy coefficient ξ and attack distance weight threshold are calculated. After the attack paths with the attack distance weight greater than the threshold is removed in M-NETAG, the optimized attack graph Optimized-AG reserves the effective attack paths for the reference of administrators, as shown in Figure 3.

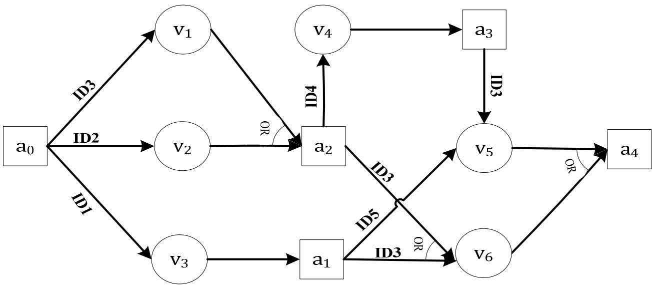

Figure 3. Optimized Attack Graph

4. Prediction of Attack Path Based on Optimized-AG.

According to the Optimized-AG, if there are attack paths with equal attack distance L_d, such effective attack path is defined as equivalent. Since launching a network attack is an economic behavior, attackers have to pay appropriate attack costs when they occupy the desired resources. Therefore, attack behavior cost is introduced in the paper and the preferred attack path of attackers among various effective attack paths is determined by calculating the total cost of the accessible paths and combining the analysis of L_d.

Definition 10: W means the set of attack permissions and $\forall w \in W$ is related to each node and expressed in two-tuple (d_{ji}, c). d_{ji} represents the attack distance weight of edges $<a_j, v_i>$ and c represents the attack costs paid when the attack behavior edges $<a_j, v_i>$ occurs.

Based on the above definition, each attack behavior edge $<a_j, v_i>$ is assigned a weight w and the current attack graph is shown as follows:

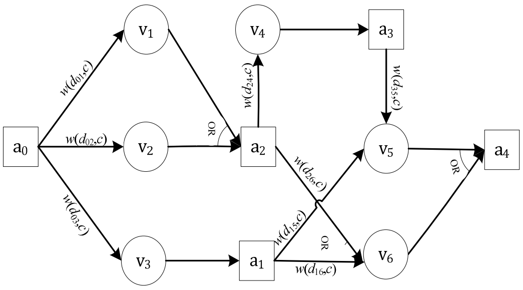

Figure 4. Optimized-AG with Weight

4.1. Cost Analysis of Node Attack Path Based on Optimized-AG

In this paper, the cost required to implement a single-step attack in the attack path of Optimized-AG is expressed as $Ecostep(e_{ji})$, i.e., the parameter c mentioned in Definition 10 above. Ignoring other aspects, the total cost of an attack sequence or an attack path is calculated as follows:

$$TCost(e_{ji}) = \sum_{e_{ji} \in E_A} Ecostep(e_{ji}) \tag{4}$$

Where, $Ecostep(e_{ji})$ consists of two parts, namely the operation cost $OCost(e_{ji})$ and the risk cost $RiskCost$. In the literature[14], a mathematical model of operation cost is developed in the process of analyzing the cost of attacks.

$$OCost\left(e_{ji}\right) = \mu * cost(Meta - operations) + \eta * cost(Sequence) \tag{5}$$

Where, $e_{ji} \in E_A$; and $cost(Meta - operations)$ is the meta-operation cost of a network node attack, so that the result depends on the meta-operations and the use of resources; $cost(Sequence)$ is the operation sequence cost of a network node attack; μ is the probability of a network node attack and η is path redundancy with low correlation in network node attack paths.

Moreover, the attacker has to take the risks of being discovered by the network administrators when he/she launches an attack, that is, he/she has to pay appropriate risk costs $RiskCost$. Therefore, the risk cost factor needs to be included in the cost analysis of an attack and $RiskCost$ is related to network security risk coefficient, attacker's experience and operation costs.

Network security risk coefficient is a measure of the likelihood of an attack being discovered and is calculated with the following formula:

$$Risk(e_{ji}) = \Psi(v_i) * \mathrm{I}(a_j) \tag{6}$$

In the formula (6), $Risk(e_{ji})$ is determined by the influence coefficient $\Psi(v_i)$ of a resource node on attack behavior as well as the influence coefficient $\mathrm{I}(a_j)$ of the attack node itself.

As the attack node proceeds and the attack distance L_d increases, the attacker needs to spend more time and takes higher risks of being exposed while attacking a resource node is increasingly complex. In this way, it is understood that the attacker will prefer a shorter attack distance to avoid risks as the cost of each further step grows exponentially. Therefore, attack distance is treated as one of the factors in the evaluation of risk cost in the paper. With the introduction of the attack path redundancy coefficient mentioned above, the risk cost is calculated as follows:

$$RiskCost = OCost(e_{ji}) * Risk(e_{ji}) * Ex(e_{ji})^{count-1} * \xi \tag{7}$$

Where, $OCost(e_{ji})$ is the operation cost; $Risk(e_{ji})$ $\left(Risk(e_{ji}) > 1\right)$ is security risk coefficient and is defined by an experience expert; $\left(0 < Ex(e_{ji}) < 1\right)$ is the attacker's empirical dependence coefficient; ξ is the attack path redundancy coefficient and $count$ is the number of attacks.

According to the definitions above, the cost $Ecostep(e_{ji})$ of attack behavior between attack nodes is calculated as follows:

$$Ecostep(e_{ji}) = \partial * OCost(e_{ji}) + \ell * RiskCost \tag{8}$$

Where, ∂、ℓ represent the attack cost weight coefficient. According to the definitions above, the total cost of a single attack path is calculated as follows:

$$TCost(e_{ji}) = \sum_{e_{ji} \in E_A} \left(\partial * OCost(e_{ji}) + \ell * RiskCost\right) \tag{9}$$

4.2. Prediction Method of Node Attack Path Based on Optimized-AG

The most possible effective attack paths are reserved in the Optimized-AG for the reference of administrators, and the paths might have different attack distance and attack cost, same attack distance and attack cost, same attack difference but different attack cost, or same attack cost but different attack distance. If only the attack path distance weight or attack cost is considered, locating the preferred attack path is not comprehensive and time-consuming.

To balance the role of the two factors in the process that the attacker prioritizes an attack path among all effective attack paths, a reasonable parameter-evaluation function needs to be designed as the criteria for determining the path selection. A path with a higher probability of being selected requires least attack path distance weight and attack cost, i.e., less evaluated value. Thus, the evaluation function is calculated as follows:

$$\begin{aligned}\oint(OAStep_i) &= \lambda * L_d + \gamma * TCost(e_{ji}) \\ &= \lambda * \sum d_{ji} + \gamma * \sum_{e_{ji}\in E_A} Ecostep(e_{ji})\end{aligned} \tag{10}$$

Where λ and γ represents attack distance and attack cost weight coefficient respectively ($0<\lambda<1$, $0<\gamma<1$, $\lambda+\gamma=1$ and $\lambda<\gamma$); L_d is the distance weight of an effective attack path and $TCost(e_{ji})$ is the attack behavior cost of an effective attack path. Based on the path prediction method above, a path prediction algorithm OASJ is designed. The preferred attack path of an attacker is sorted out by calculating the evaluated value of a path $OAStep_i$ for the reference of administrators. The algorithm is implemented as follows:

Algorithm 4 Attack path determination algorithm OASJ (OAS, d_{ji} and $Ecostep(e_{ji})$)

Description: The evaluated value of each path $OAStep_i$ is marked with a vector $\Theta=(\Theta_1,\Theta_2...\Theta_i)$, and the values are stored in the set Ω in sequence. The preferred attack path can be located by comparing the evaluated values in the set Ω.

Input: the set OAS in Optimized-AG, the attack distance weight d_{ji} of edges $<a_j,v_i>'$ and the attack behavior cost $Ecostep(e_{ji})$

Output: the evaluated value of a path $\oint(OAStep_i)$ and the preferred attack path $U\text{-}OAStep_i$

```
$\Theta \leftarrow \emptyset$, $\oint(OAStep_i) \leftarrow \emptyset$
While (true) do
    For ($OAS$ each $OAStep_i$)
        $L_d = d_{ji}$++
        $TCost(e_{ji}) = Ecostep(e_{ji})++$
        $c \leftarrow TCost(e_{ji})$
        $W \leftarrow w(d_{ji},c)$ //Store two-tuple values in the attack permission set
        Calculate $\oint(OAStep_i) = \lambda * L_d + \gamma * TCost(e_{ji})$
        $\Theta_i \leftarrow \oint(OAStep_i)$ //Mark the evaluated value with $\Theta_i$
        $\Omega \leftarrow \oint(OAStep_i)$
    End For
    Select $\oint(OAStep_i)_{\min}$ in $\Omega$
    $\Theta_i \leftarrow \oint(OAStep_i)_{\min}$ //find the corresponding marks by the evaluated value
    Through $\Theta_i$ finding $OAStep_i$
    $U\text{-}OAStep_i \leftarrow OAStep_i$
End While
Return $U\text{-}OAStep_i$
```

5. Experimental Verification

5.1. Experimental Network Configuration

To verify the feasibility of the attack path prediction method in this paper, a small Windows-based enterprise intranet is designed and has been subject to corresponding simulation experiment. The network typology applied in the experiment is as shown in Figure 5:

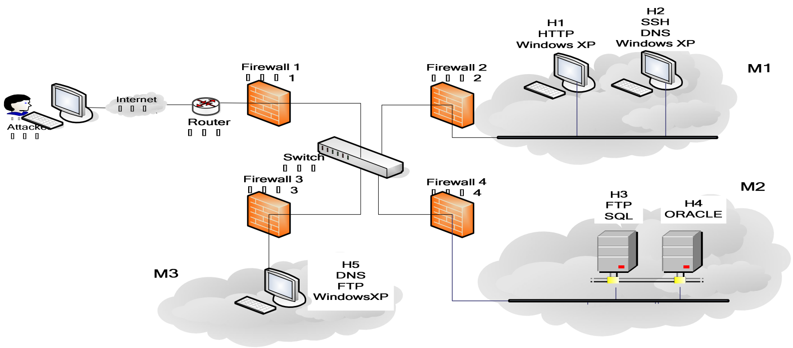

Figure 5. Topological Graph of Local Networks

In the experiment network, the safe zones M1, M2 and M3 in the Internet and LAN are separated from each other by firewalls, thus the attack host Attack (i.e. H0) assesses the LAN via Internet. In the zone M1 of LAN, the host H1 providing open HTTP services and the host H2 providing SSH and DNS services are separated from other networks by Firewall 2. In M2, a SQL server and an ORACLE server serves as the hosts H3 and H4 and are separated from other networks by Firewall 4. Moreover, H3 provides FTP services for hosts in M1 and M3. In M3, H5 provides DNS and FTP services and serves as a target host that stores important data. H5 is separated from other networks by Firewall 3. In the typological graph of local networks above, the firewalls will only allow the external host H0 to access the hosts H1 and H2, and any other external accesses are blocked. Moreover, the internal hosts H1 and H2 cannot access each other, but other hosts can access the open services of each host.

5.2. Experimental Data and Analysis

The experimental network is scanned with a scanner and the vulnerabilities and related information of each host are obtained as shown in Table 2. The quantitative value of the difficulty level of these vulnerabilities being exploited can be gained from querying the quantitative criteria of attack complexity of vulnerabilities in Table 1, and the results is shown in Table 3.

Table 2. Vulnerability information of each Host

Safe Zone	Host	Service	Bugtraq ID
M1	Host1	HTTP	ID1=CVE-2017-2666
	Host2	SSH/DNS	ID2=CVE-2018-0497
			ID3=CVE-2018-5538
M2	Host3	SQL/FTP	ID4=CVE-2018-0607
			ID5=CVE-2017-3329
	Host4	ORACLE	ID6=CVE-2018-3091
M3	Host5	DNS/FTP	ID3=CVE-2018-5538
			ID5=CVE-2017-3329

Table 3. Quantitative Value of Vulnerability Information in Each Host

Bugtraq ID	Distance Weight
CVE-2017-2666	10
CVE-2018-0497	10
CVE-2018-0607	50
CVE-2018-3091	70
CVE-2018-5538	30
CVE-2017-3329	90

According to the design of experiment and the information of hosts and vulnerabilities in the network, the generated network attack graph is shown in Figure 1. Based on the information in Table 2, the attack graph in Figure 1 is transformed into the attack graph marked with a vulnerability ID in Figure 2 by applying the conversion algorithm AGC-Alg, and the quantitative value of attack distance weight corresponding to the vulnerability ID is shown in Table 3. All accessible attack paths from the attack host H0 to the target host H5 are displayed in Figure 1, and the attack distance weight of all attack paths in Figure 1 is calculated with the attack graph optimization algorithm AGO mentioned above, thus the attack distance weight of each path goes through a gradual change process as shown in Figure 6 below:

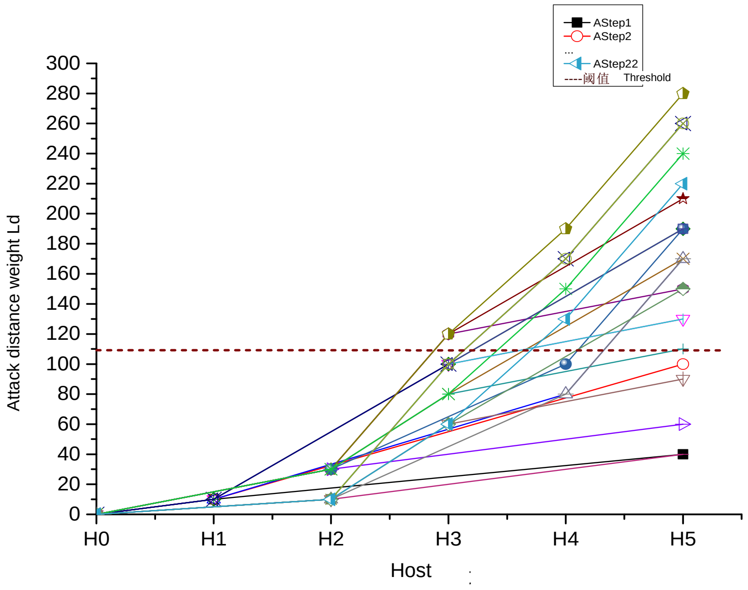

Figure 6. Weight graph of attack path

In Figure 6, each polygonal line represents an accessible attack path, and the fluctuation of a corresponding value indicates a process in which the distance weight of the attack path H0→…→ H5 overlaps. Among all attack paths in Figure 6, the minimum attack distance weight is 40 and the maximum attack distance weight is 280, thus the path redundancy coefficient ξ of the attack graph is 7. The quantitative standard value of the path redundancy coefficient is set to 10 by an experienced expert, and the calculated attack distance weight threshold γ is 110, as shown by the dotted line in the Figure. Therefore, the attack paths with the path weight over 110 are removed in Figure 6. The optimized attack paths are displayed in Figure 3 and the corresponding experimental network topological graph is shown in Figure 7:

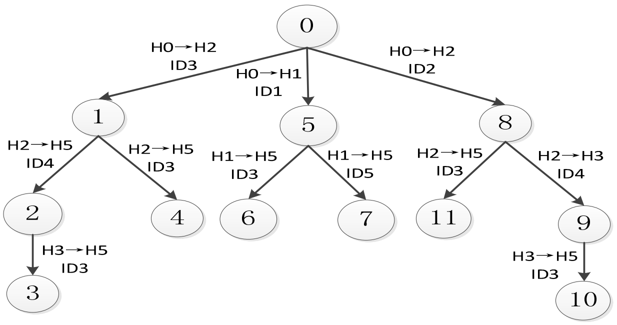

Figure 7. Optimized Experiment Topological Graph

Six accessible attack paths to the target host H5 are retained in Figure 7 and the attack behavior cost of each path is analyzed on this basis. According to the definitions of attack path behavior cost $TCost(e_{ji})$ above and based on the experiment of an expert, the appropriate parameters are set as follows: ∂、ℓ are set to 0.3 and 0.7 individually; $Risk(e_{ji})$ is set to 2 and indicates that the cost doubles for each further step; $Ex(e_{ji})$ is set to 0.5 and indicates that the second similar attack has a cost half of the original cost and ξ is set to 7 calculated above. The cost result analysis of attack paths is as shown in Table 4:

Table 4. Cost Information of Effective Attack Paths

$OAstep_i$	$Ecostep$(c)	$TCost(e_{ji})$
$OAstep_1$	$<a_0,v_1>$(2.1), $<a_2,v_6>$(4.6)	6.7
$OAstep_2$	$<a_0,v_1>$(0.7),$<a_2,v_4>$(1.9),$<a_3,v_5>$(4.0)	6.6
$OAstep_3$	$<a_0,v_2>$(3.2), $<a_2,v_6>$(6.7)	9.9
$OAstep_4$	$<a_0,v_2>$(0.3),$<a_2,v_4>$(0.7),$<a_3,v_5>$(1.4)	2.4
$OAstep_5$	$<a_0,v_3>$(2.1), $<a_1,v_5>$(4.5)	6.6
$OAstep_6$	$<a_0,v_3>$(1.2), $<a_1,v_6>$(2.5)	3.7

The preferred attack path is sorted out by analyzing the attack distance weight and cost and computing the evaluated value of each path in Figure 7 and Table 3. According to the experiment of an expert, λ is set to 0.45 and γ is set to 0.55 in $\oint(OAStep_i)$. The attack distance weight is calculated in a centesimal system and the attack cost is obtained on a score of ten, thus the path distance weigh need to be divided by 10 for uniformity and the final value $L_d{}'$ is substituted into the formula for L_d. The analysis of attack paths is shown in Figure 5.

Table 5. Effective Attack Path in Experimental Topological Graph

$OAstep_i$	Step Description	L_d	L_d'	$TCost$	$\oint$
$OAstep_1$	H0→(H2,DNS)→(H5,DNS)	60	6	6.7	6.385
$OAstep_2$	H0→(H2,HTTP)→(H4,SQL)→(H5,DNS)	110	11	6.6	8.58
$OAstep_3$	H0→(H2,SSH)→(H5,DNS)	40	4	9.9	7.245
$OAstep_4$	H0→(H2,SSH)→(H4,SQL)→(H5,DNS)	90	9	2.4	5.37
$OAstep_5$	H0→(H1,HTTP)→(H5,FTP)	40	4	6.6	5.43
$OAstep_6$	H0→(H1,HTTP)→(H5,DNS)	100	10	3.7	6.535

According to the simulation experiment, it can be seen from Table 5 that the path $OAStep_4$ is calculated to have a minimum evaluated value among all effective attack paths that may have same distance weight or same distance cost. If the attacker takes into account the attack distance weight and cost factors of the path and can access the path with less effort, he/she would select the path H0→(H2,SSH)→(H4,SQL)→(H5,DNS) as the attack path. The network administrators can prioritize the attacked path for reference and prepare security defense strategies in advance.

6. Conclusion

A new distance cost attack graph model NETAG is presented herein to cover the shortage of considerations in the existing attack path prediction methods of attack graphs. Firstly, the original model is transformed into a marked attack graph via the AGC-Alg algorithm and the attack distance weight and attack path redundancy coefficient are introduced to optimize the attack graph. Secondly, the Optimized-AG is subject to a cost calculation of attacking a path and the evaluation function is utilized to calculate the evaluated value of an attack path. Finally, the preferred attack path is located. The experiment suggests that the attack path prediction method of attack graphs presented herein eliminates path redundancy and the preferred attack path is determined based on the path distance weight and cost filtering. However, there still some deficiencies in the study. Although the authors provide the quantitative criteria of attack distance, the distance weight value is not detailed enough and needs to be assigned by an expert. Besides, the parameters are too complex in the mathematical model of attack cost and can only be obtained by an expert empirically. Therefore, further studies will focus on how to gain specific parameter values of distance weight and attack cost by conducting a large number of simulation experiments and referring to the experience of experts.

7. References

[1] *2017 Computer Crime and Security Survey* [EB/OL]. http://www.fbi.gov
[2] Development Status of China's Internet Sites and Safety Report (2018)[EB/OL]. http://www.cnii.com.cn/mobileinternet/2018-07/13/ 2082449.html
[3] National Computer Network Emergency Response Technical Team/Coordination Center of China [EB/OL]. http://www.cert.org.cn/publish/main/upload/File/situation.pdf
[4] Ni G, Ling G, Yiyue H E, et al. Dynamic Security Risk Assessment Model Based on Bayesian Attack Graph[J]. Journal of Sichuan University, 2016.
[5] Lee P P C, Misra V, Dan R. Distributed algorithms for secure multipath routing in attack-resistant networks[J]. IEEE/ACM Transactions on Networking, 2007, 15(6):1490-1501.
[6] Kaynar K. A taxonomy for attack graph generation and usage in network security[J]. Journal of Information Security & Applications, 2016, 29(C):27-56.
[7] Muñozgonzález L, Sgandurra D, Paudice A, et al. Efficient Attack Graph Analysis through Approximate Inference[J]. 2016, 20(3).
[8] Poolsappasit N, Dewri R, Ray I. Dynamic Security Risk Management Using Bayesian Attack Graphs[J]. IEEE Transactions on Dependable & Secure Computing, 2011, 9(1):61-74.
[9] Man D, Zhang B, Yang W, et al. A Method for Global Attack Graph Generation[C]// IEEE International Conference on Networking, Sensing and Control. IEEE, 2008:236-241.
[10] Wu Hongrun, Qin Jun, Zheng Bojin. Anti-attack Ability Based on Costs in Complex Networks[J]. Computer Science, 2012, 39(8):224-227.
[11] Jian S I, Chen P, Ningping G U, et al. Network attack graph backward depth-first building algorithm[J]. Computer Engineering & Applications, 2017.
[12] Chen Xiaojun, Shi Jinqiao, Xu Fei et al. Algorithm of Optimal Security Hardening Measures Against Insider Threat[J].Journal of Computer Research and Development, 2014,51(7):1565-1577.
[13] Qing Dapeng, Zhang Bing, Zhou Yuan et al. Depth-first Method for Attack Graph Generation [J].Journal of Jilin University (Engineering and Technology Edition), 2009, 39(2):446-452.
[14] Wang Hui, Liu Shufen. A Scalable Predicting Model of Insider Threats[J].Chinese Journal of Computers, 2006, 29(8):1346-1355.