A network security situation awareness method based on layered attack graph

ZHU Yu-hui[1,2], SONG Li-peng[1,2]

(1. *School of Data Science and Technology, North University of China, Taiyuan 030051, China*;
2. *Research Institute of Big Data and Security, North University of China, Taiyuan 030051, China*)

Abstract: The real-time of network security situation awareness (NSSA) is always affected by the state explosion problem. To solve this problem, a new NSSA method based on layered attack graph (LAG) is proposed. Firstly, network is divided into several logical subnets by community discovery algorithm. The logical subnets and connections between them constitute the logical network. Then, based on the original and logical networks, the selection of attack path is optimized according to the monotonic principle of attack behavior. The proposed method can sharply reduce the attack path scale and hence tackle the state explosion problem in NSSA. The experiments results show that the generation of attack paths by this method consumes 0.029 s while the counterparts by other methods are more than 56 s. Meanwhile, this method can give the same security strategy with other methods.

Key words: network security situation awareness (NSSA); layered attack graph (LAG); state explosion; community detection

CLD number: TP393.08 **Document code**: A

Article ID: 1674-8042(2019)02-0182-09 **doi**: 10.3969/j.issn.1674-8042.2019.02.011

0 Introduction

With the rapid development of the Internet, more and more sophisticated attacks are used by attackers. It becomes increasingly difficult to protect a network from the intrusions using traditional network security technology[1-2]. Thus, network security situation awareness (NSSA), which pays attention to comprehensive and active defense, is of great interest to the network security community.

NSSA is proposed by Bass, which is the perception of the elements in the network environment, the comprehension of their meanings and the projection of their status in the near future[3-4]. There are many models that have been used in NSSA, such as attack graph model[5-8], attack tree model[9], threat propagation model[10-14] and Markov model[2].

Although the attack-model-based approach can provide accurate security strategy, the task of developing a comprehensive and reliable NSSA model remains challenging. One key challenge is how to provide a defense strategy in real time[2]. Due to the massive scale and complex structure of the network, state explosion becomes a critical factor affecting the efficiency of NSSA model. To solve this problem, defending the attack path with maximum possible is proposed in Ref.[15]. However, the probability of bypassing the protected path will significantly increase when the number of attack paths grows large. Kaynar, et al. utilized distributed computation to attenuate this problem, but it needs extra hardware support[16]. Improving the real-time of NSSA without reducing the accuracy is one of the central challenges in the field of NSSA[4].

This study proposes a new NSSA method based on layered attack graph (LAG) to overcome the challenge. This method divides the target network into several logical subnets with various sizes in the light of the frequency of interaction between network nodes. In this division result, the nodes in the same logical subnet are linked closely and the nodes in different logical subnets are linked sparsely. The connections between logical subnets are generated in term of the physical network. These logical subnets and their connections constitute the logical network structure. In the actual attack process, the

Received date: 2019-01-15

Foundation items: National Natural Science Foundation of China (No. 61772478)

Corresponding author: ZHU Yu-hui (zhu425066454@foxmail.com)

attacker's behavior is monotonous[17]. That is to say, the attack path can be optimized by snooping logical network structure. Based on the above assumptions, the top-level attack graph on the basis of the logical structure of the network and the bottom-level attack graph on the basis of the connection relationship between nodes in each logical subnet are generated. When the attack target and the attack source are in the same logical subnet, the attack path is selected in term of the bottom-level attack graph; When the attack target and the attack source are in different logical subnets, the attack path is selected in term of the top-level attack graph. Based on two-layer attack graph, the reduction of attack path scale is accomplished, and it effectively solves the state explosion problem in NSSA. To validate the effectiveness of our method, an experiment is conducted in our lab. The experiment results show that this method can reduce more than 90% of the spatial-temporal consumption and give accurate strategy.

This paper is organized as follows. The framework of our NSSA model is present in Section 1. In Section 2, the proposed NSSA method is explicated including the generation of layered attack graph and attack path. Section 3 conducts experiments to evaluate the efficiency of the model. Finally, we end up our investigations with brief conclusion.

1 NSSA method

1.1 LAG

Attack graph model is used to depict the attack behavior in network, which shows the state transition process caused by multi-step attack. At the top-level, state nodes are generated according to the different attack states of each logical subnet, and the connections between state nodes in the top-level are generated in term of the basic information of the network; At the bottom-level, state nodes are generated according to the different attack states of each physical node, and the connections between state nodes in the bottom-level are generated in line with the basic information of the network. And the mapping relationship between top-level state nodes and bottom-level state nodes is generated in the light of the containment relationship between logical subnets and physical nodes. LAG can be used to generate attack path from source state node to target state node, and reduce the scale of attack path. The definition of LAG is given below.

Definition 1 LAG is represented by a five-tuple of $LAG=(S,V,L,E,\Delta)$. The meanings of each part are as follows:

1) S represents the set of state nodes. Each state node S_i contains four attribute values, $S_i=(SID, HID/DID, PL, LF)$, where SID is the number of state nodes, HID is the identity of target host, DID is the identity of target logical subnet, PL is the level of user privileges, LF is the level of state nodes, the state node is a top-level node if the value of LF is 1, and the state node is a bottom-level node if the value of LF is 0.

2) V represents the set of vulnerability nodes. Each vulnerability node V_i contains four attribute values, $V_i=(VID, HID, CVEID, P(VID))$, where VID is the vulnerability node number, HID is the host number of vulnerability, $CVEID$ is the only representation of vulnerability in the general vulnerability library, $P(VID)$ is the success probability of vulnerability utilization.

3) $L=\{L_i \mid i=1,2,\cdots,N\}$ is the set of hierarchical relationships of the state nodes, where L_i is the collection of the bottom-level nodes contained in the top-level state node S_i.

4) $E\subseteq S\times V\bigcup V\times S$ is the set of directed edges. The state node points to the vulnerability node, indicating that the vulnerability can be exploited in the current state. The vulnerability node points to the state node, indicating that the state can be reached by vulnerability attack.

5) $\forall\Delta(P_{i,j})\in\Delta$, $\Delta(P_{i,j})$ denotes the probability that an attacker uses vulnerability to change from state S_i to state S_j. The value of transfer probability depends on the vulnerability of state transition.

1.2 Situation awareness method based on LAG

This study bases NSSA on LAG, which contains the vulnerability information in the network system and the state transitions information caused by the exploit of vulnerabilities. And it is a two-layer graph with the information of logical network structure. Thus, the reduction of attack path scale can be realized by avoiding the generation of unreasonable attack path which is not coincident with the behavior of attacker.

The NSSA model based on LAG can be divided into three main parts, which are the generation of LAG, the generation of attack path and the situation assessment based on attack path, as shown in Fig. 1.

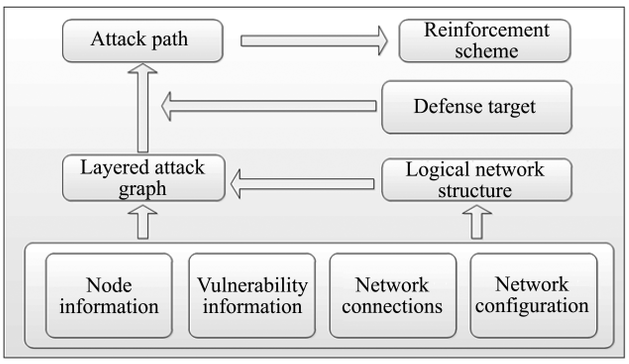

Fig. 1 Framework of situation awareness based on LAG

2 Realization of NSSA

2.1 Generation of LAG

The generation of LAG includes two key steps: dividing the network into several logical subnets by using a community discovery algorithm; generating LAG with relevant information, which contains logical network structure and basic network information such as node information, vulnerability information, network connection, network configuration, etc.

2.1.1 Division of network

In this paper, the logical network structure is gotten by using the Louvain algorithm[18] which is based on modularity[19]. Modularity is a common criterion to measure the intensity of network community structure. Higher value of modularity indicates closer community structure.

Network system $G=(V,E,W)$, where V is the node in the network system; E is the connection between nodes in the network system, indicating the communication connection between nodes in the network system; W is the weight of the edges in the network system, indicating the tightness degree of communication between nodes in the network system. The number of nodes in the network system $n=|V(G)|$. A network system with n nodes is represented by adjacent matrix $\boldsymbol{A}_{n\times n}$. If $A_{i,j}$ is 1, it represents that there is network communication between node i and node j. And if $A_{i,j}$ is 0, it represents that there is no network communication between node i and node j. The adjacency matrix $\boldsymbol{W}_{n\times n}$ is the weight of edge. Larger value of $W_{i,j}$ indicates the closer connection between node i and node j. $\boldsymbol{W}_{n\times n}$ is a directional matrix with weight. The modularity with direction and weight is[19]

$$Q=\frac{1}{w}\sum_{i=1}^{n}\sum_{j=1}^{n}\left[\boldsymbol{W}_{i,j}-\frac{w_i^{\text{out}}w_j^{\text{in}}}{w}\right]\delta(c_i,c_j),$$

$$w=\sum_{i=1}^{n}\sum_{j=1}^{n}W_{i,j}, \tag{1}$$

where W represents a network with direction and weight, $\boldsymbol{w}$ is the total weights of the edges of the network, w_j^{in} is the inbound weight of node j and w_i^{out} is the outbound weight of node i. In Ref. [20], a partition algorithm of network based on modularity is proposed.

Algorithm 1 Partition algorithm of network based on modularity

Do:
  Part1:
    Repeat
    For (V_i:V)
      Divided each node V_i in network G into a subnet
    End for
    For (V_i:V)
      For (V_j:V)
        Delete the node V_i from its subnet and add the node V_i to the subnet where the adjacent node V_j is located. Calculate the modularity increment $\triangle Q$ after joining.
    End for
    If $\triangle Q>0$ then
      Select the largest neighbor node of $\triangle Q$. Add node V_i to the subnet where node V_j is located.
    Else
      Node V_i remains unchanged.
    End if
  End for
  Until $\triangle Q$ no longer changes.
End part 1
  Part 2:
  The subnet partitioned by Part 1 is regarded as a new node, and the weights of the edges between new nodes are determined by the weights between the subnets. The sum of the weights of the edges in the subnet is the weight of the self-loop edge of the new node.
End part 2
While (network system cannot be divided).

2.1.2 LAG generation algorithm

LAG includes state nodes, vulnerable nodes, hierarchical relationship of state nodes, directed edge and probability of state transition. The state nodes are generated in line with the different vulnerability states of each physical node and logical subnet; The vulnerable nodes are generated on the basis of the

vulnerability information and configuration defects in the network; The hierarchical relationship of state nodes is generated in the light of the inclusion relationship between logical subnet and physical nodes; The directed edges are generated in line with the vulnerability nodes and state nodes; The probability of state transition is determined by the vulnerability degree given by the expert system. The LAG generation algorithm is given as follows.

Algorithm 2 LAG generation algorithm

Generating vulnerability node set V

Generating the probability of vulnerability V

For (H_i:H)

  According to the vulnerability V_i in host H_i, generate the bottom-level state node S_T.

End for

For (D_i:D)

  According to the vulnerability V_i in subnet H_i, generate the top-level state node S_T.

End for

For (S_i:S && LF=1)

  For (S_j:S && LF=0)

    If (S_j.HID∈S_i.DID)

      Recording the top-level state node S_i contains bottom-level state node S_j

    End if

  End for

End for

For (S_i:S)

  For (V_i:V)

    If the vulnerability V_i can be exploit in state S_i, generate edge $S_i \rightarrow V_i$.

  End for

End for

2.2 Generation of attack path

Attack path is an effective mean to describe the attack process. Analyzing attack path can discover the critical path of attacker's penetration and give pertinent defense strategy. The definition of attack path is given below.

Definition 2 Attack path r is the state transition process of an attacker from the initial state to the target state. R is the attack path set.

Definition 3 Attack success probability $P(r)$ refers to the probability that all state transitions in the attack path are successful. For attack paths $r=S_1 \rightarrow S_2 \rightarrow \cdots \rightarrow S_n$, $P(r)=P(S_2 \mid S_1) \times P(S_3 \mid S_2) \times \cdots \times P(S_n \mid S_{n-1})$.

For network defenders, because of the large number of network nodes, it is difficult to determine the target of the attacker. Usually, the important nodes in the network system are assumed the target of attack, such as the data server in the network.

The attack source is found through real-time collection and analysis of network data. The current mainstream way is to identify the threat subject in the network by using intrusion detection system (IDS) early warning events. Because IDS alarm information exist false and irrelevant alarms, this study uses the intrusion alarm verification algorithm based on environmental attributes proposed in Ref.[21] to identify attacks in the network as the source of attack. Note that the security event causing the alarm is t, and the existence possibility of the attack source t is

$$P(t)=\omega_1 \times R_{OS}+\omega_2 \times R_{Ser}+\omega_3 \times R_{Vul}+\omega_4 \times R_{Con}, \tag{2}$$

where R_{OS} is the correlation between alerts and target operating systems, R_{Ser} is the correlation between alerts and target network application services, R_{Vul} is the correlation between alerts and target network vulnerability information, R_{Con} is the correlation between alerts and target network security configuration information and $\boldsymbol{W}=[\omega_1,\omega_2,\omega_3,\omega_4]^T$ is a normalized weight vector. This paper uses $W=[0.11,0.23,0.28,0.38]^T$ provided by Xi, et al. The existence possibility of attack source t is $P(t)$, $P(t) \in [0,1]$, and higher value of $P(t)$ indicates higher possibility of security incidents occurring.

For the certain attack source and target, all attack paths can be found by searching the path between two state nodes in the LAG. At the same time, in order to reduce the scale of attack path, the method in Ref.[18] assumes that the attack behavior of attacker is monotonous, and the generation of attack path is constrained by attack rules. The attack behavior of attacker meets the following rules:

1) Attack action can penetrate the sub network area;

2) Attack action can reach a more important host;

3) Attack action can get higher permissions of the same host.

Based on the above rules, the unreasonable attack path can be avoided. Then, a generation algorithm of LAG attack path based on the depth first traversal is present.

Algorithm 3 Attack path generation algorithm

Create a stack theStack;

```
Start node start into stack;
Set node access state;
While (theStack is not empty) {
    S=theStack. peek (). Unvisited;
    If (stack top node has no access to neighbor nodes) {
        TheStack. pop;
    }
    Else{
        Set node access state;
        TheStack. push (S);
    }
    If (attack path does not conform to attack rules) {
        TheSack. pop ();
    }
    If (the top node of the stack is same as the end of the
     target node) {
        Paths. add (theStack);
        Set node access state;
        TheStack. pop;
    }
}
```

2.3 Node situation assessment based on attack path

Ref. [11] takes the node occurrence frequency in attack path as the evaluation index of node importance. Suppose that the set of attack sources detected in the network is $T=\{t_1,t_2,\cdots,t_n\}$. $P(t_i)$ is the existence probability of attack source t_i. For the state node S, N_i is the frequency of S appearing in partial attack paths with higher success probability, and the security situation $I(S)$ of the node is

$$I(S)=\sum_{i=1}^{n}P(t_i)\times N_i. \tag{3}$$

In the case of limited defense resources, the nodes with higher security situation should be defended.

3 Experiment analysis

3.1 Experiment environment

In this section, an experiment is conducted in our lab to validate the effectiveness of the proposed method. All program of the experimental operations are performed on the same system platform (processor of Intel® Core™ i7-6700 CPU 3.40 GHz, memory of 8.00 GB, operating system of Windows 10 Pro) and programming environment (Java 1.8).

The network topology in experiment is shown in Fig. 2.

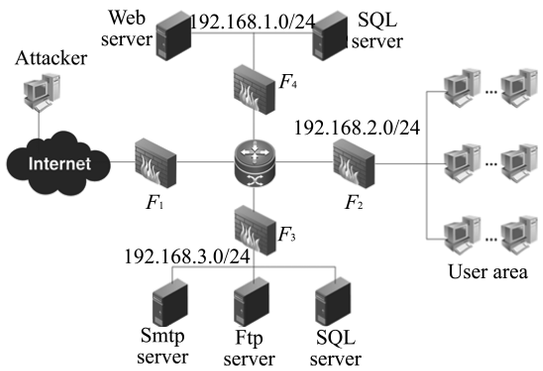

Fig. 2 Network topology in experiment

The network consists of three main areas: external server area, internal server area and user area. The external server area contains a web server and a SQL data server for providing external web page browsing and information storage. The network segment of external services is 192.168.1.0/24. The internal server includes SMTP server, FTP server and SQL server, for providing internal mail communications, file transfer and data storage services. The network segment of internal services is 192.168.2.0/24. The user area is in the segment of 192.168.3.0/24, which contains 20 hosts. The attacker is outside the network system and attacks the internal network system through remote network connection. Hosts information in network is shown in Table 1, and the access relationship between network segments is shown in Fig. 3.

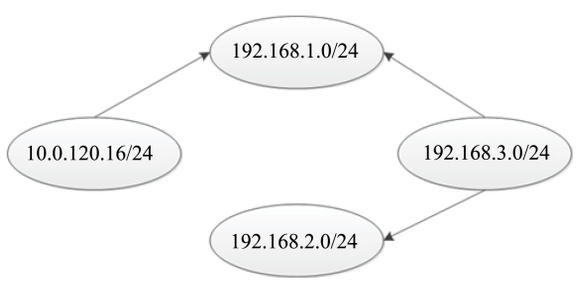

Fig. 3 Access relationship between network segments

Due to the access control list (ACL) in router and firewall, the access between each segments is limited. The external network can only directly access the external server area in the network system; the external server and the internal server cannot access other network areas; the user area can access the internal server area and the external server area. Using vulnerability scanning tools to scan each segments, the network vulnerabilities information can be obtained, which is show in Table 2.

Table 1 Hosts information in network

No.	Description	IP	No.	Description	IP
H_0	Remote Attacker	10. 0. 120. 16/24	H_{13}	Windows 10	192. 168. 3. 80/24
H_1	Web Server	192. 168. 1. 10/24	H_{14}	Windows 10	192. 168. 3. 90/24
H_2	SQL Serverv	192. 168. 1. 20/24	H_{15}	Windows 10	192. 168. 3. 100/24
H_3	Smtp Server	192. 168. 2. 10/24	H_{16}	Windows 10	192. 168. 3. 110/24
H_4	Ftp Server	192. 168. 2. 20/24	H_{17}	Windows 10	192. 168. 3. 120/24
H_5	SQL Server	192. 168. 2. 30/24	H_{18}	Windows 10	192. 168. 3. 130/24
H_6	Windows 7	192. 168. 3. 10/24	H_{19}	Windows 10	192. 168. 3. 140/24
H_7	Windows 10	192. 168. 3. 20/24	H_{20}	Windows 10	192. 168. 3. 150/24
H_8	Windows 10	192. 168. 3. 30/24	H_{21}	Windows 10	192. 168. 3. 160/24
H_9	Windows 10	192. 168. 3. 40/24	H_{22}	Windows 7	192. 168. 3. 170/24
H_{10}	Red Hat Linux 7. 2	192. 168. 3. 50/24	H_{23}	Windows 10	192. 168. 3. 180/24
H_{11}	Windows 10	192. 168. 3. 60/24	H_{24}	Windows 10	192. 168. 3. 190/24
H_{12}	Windows 7	192. 168. 3. 70/24	H_{25}	Windows 10	192. 168. 3. 200/24

Table 2 Vulnerabilities information in network

CVE No.	CVSS score	Description	Involved hosts
CVE-2011-0638	3. 4	Execute arbitrary programs via crafted USB data	H_6, H_7, H_9, H_{12}, H_{20}, H_{23}
CVE-2014-6271	10	Execute arbitrary code via a crafted environment	H_{10}
CVE-2013-2249	10	Mod_session_dbd module in the Appache HTTP Server	H_1
CVE-2018-8225	8. 6	A remote code execution vulnerability exists in DNS	H_{11}, H_{12}, H_{18}, H_{20}
CVE-2018-5703	10	A vulnerability in Linux kernel	H_4

3. 2 Division of experimental network

The network system has been divided into three parts by network configuration: external server area, internal server area and user area. The external server area and the internal server area are independent with fewer nodes, so they need not be divided; the user area that contains a amount of nodes is further divided in term of the frequency of traffic interaction between network nodes.

This paper collects the frequency of traffic interaction between network nodes from May to August in 2018, and gets the frequency of traffic interaction between network nodes in the network system. And the frequency of traffic interaction between the 20 hosts in the user area is stored in matrix $\boldsymbol{W}$ as Eq. (4).

$$\boldsymbol{W}=\begin{pmatrix}
0&0&0&17&2&14&0&0&3&0&0&1&1&0&0&0&0&5&0&1\\
0&0&0&5&3&1&12&0&1&0&17&2&9&5&3&0&0&0&0&1\\
0&0&0&0&1&0&0&0&0&0&0&0&6&0&0&0&2&0&0&5\\
17&5&0&0&0&16&5&0&5&0&6&0&6&5&4&0&0&0&0&0\\
5&3&1&1&0&19&1&1&4&7&1&9&8&2&6&11&12&11&7&6\\
33&1&0&18&18&0&0&0&2&5&0&5&3&0&2&7&4&13&0&11\\
0&11&0&6&1&0&0&0&11&0&23&2&5&10&7&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&7&0&2&0&3&0&0&17\\
2&1&0&5&4&2&9&0&0&4&8&4&7&0&6&6&6&5&1&15\\
0&0&0&0&6&6&0&0&4&0&0&2&9&0&0&4&4&6&1&1\\
0&17&0&6&1&0&23&0&9&0&0&0&8&23&26&0&0&0&2&0\\
2&1&0&0&8&7&2&0&4&2&0&2&6&0&0&15&18&1&6&4\\
1&3&1&3&2&1&2&2&4&3&2&2&0&4&4&3&4&2&6&2\\
2&4&0&4&0&0&5&0&0&0&9&0&6&0&5&15&18&1&6&4\\
0&1&0&4&5&2&5&1&6&0&21&0&8&5&0&0&0&1&0&0\\
0&0&0&0&12&9&0&0&6&4&0&16&7&1&0&0&15&5&5&0\\
0&0&3&0&12&4&0&3&7&4&0&19&8&2&0&15&0&8&8&22\\
8&0&0&0&11&13&0&0&6&7&0&1&4&0&1&5&8&0&2&1\\
0&0&0&0&7&0&0&0&1&1&2&6&13&1&0&5&8&2&2&6\\
1&1&20&0&6&11&1&23&19&1&0&3&7&0&0&0&22&1&6&2
\end{pmatrix}\tag{4}$$

The network is partitioned by algorithm 1, and the user area is further divided into four regions

according to the partition result. The overall network partition result is shown in Table 3, where D_0 represents the external area of the network system.

Table 3 Segmentation result of network

No.	Involved hosts
D_0	H_0
D_1	H_1, H_2
D_2	H_3, H_4, H_5
D_3	H_6, H_9, H_{11}
D_4	$H_7, H_{12}, H_{16}, H_{19}, H_{20}$
D_5	$H_{10}, H_{15}, H_{17}, H_{18}, H_{21}, H_{22}, H_{23}, H_{24}$
D_6	$H_8, H_{13}, H_{14}, H_{25}$

3.3 Spatial-temporal efficiency of attack path generation

In the attack experiment, the attack T is detected in the host H_0, and the probability $P(t)$ is 0.66. During the attack, the host H_6 and H_8 have connection with host H_1, and host H_{10} and H_{18} have connection with host H_4. In the network system, host H_4 stores important internal data, which is the main defense object of the network system. In order to verify the space-time efficiency of attack path generation under LAG, run the attack paths generation algorithm 1 000 times to generate the attack paths from attack source to defense object H_4, based on the attack graph in Ref. [8] and LAG. Table 4 shows the spatial-temporal efficiency of attack path generation based on different models.

Table 4 Spatial-temporal efficiency of attack path generation

Model	Time (s)	Space (piece)
Attack graph	56	34 641
LAG	0.029	16

From the data in the Table 4, it can be seen that in the experimental environment, the time and space consumption of the attack path generation based on the LAG is only five in ten thousands of the attack path generation based on the attack graph. And in this experimental environment, the network system only has 25 hosts, 5 different vulnerabilities and 13 vulnerable hosts. With the increase of network size and the number of vulnerabilities in the actual environment, the generation of attack path based on LAG is more efficient.

3.4 Security recommendations

In this experiment, the attack graph is shown in Fig. 4, and the LAG is shown in Fig. 5.

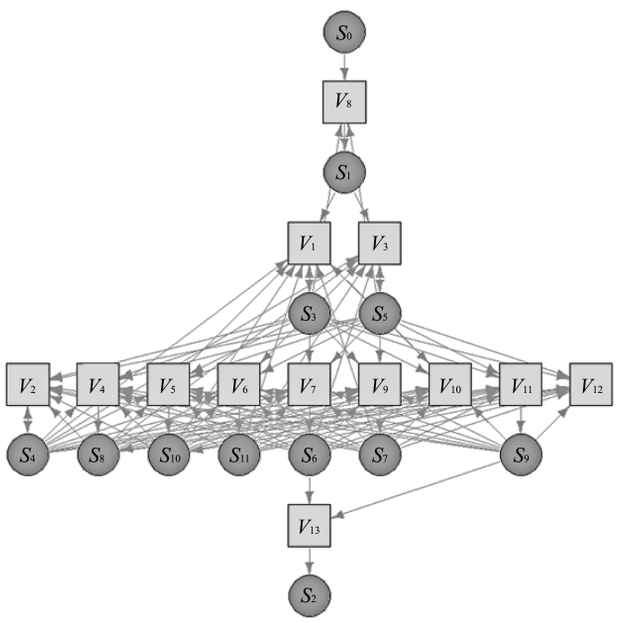

Fig. 4 Attack graph

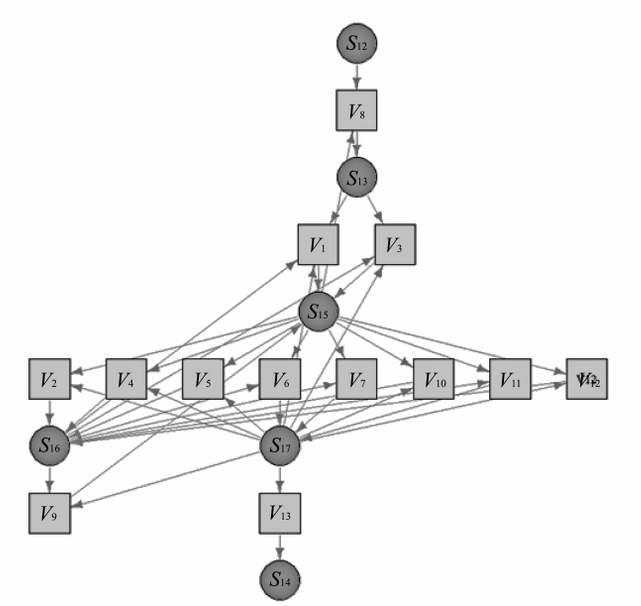

Fig. 5 Layered attack graph

The state nodes information is shown in Table 5, and the vulnerability nodes information is shown in Table 6. In Ref. [11], the frequency of nodes appearing in attack path set is taken as the evaluation index of node importance, and 10 attack paths with the highest success probability are selected for comparison analysis. The top 10 attack paths generated by LAG are shown in Table 7, and the top 10 attack paths generated by network attack graph are shown in Table 8. The statistical results of the

frequency of each node are shown in Fig. 6.

Table 5 Information of state nodes

No.	Description	Relationship	No.	Description	Relationship
S_0	$\langle S_0, H_0, \text{Root}, 1\rangle$	$L_0=\{\}$	S_9	$\langle S_9, H_{18}, \text{Root}, 0\rangle$	$L_9=\{\}$
S_1	$\langle S_1, H_1, \text{Root}, 1\rangle$	$L_1=\{\}$	S_{10}	$\langle S_{10}, H_{20}, \text{Root}, 0\rangle$	$L_{10}=\{\}$
S_2	$\langle S_2, H_4, \text{Root}, 1\rangle$	$L_2=\{\}$	S_{11}	$\langle S_{11}, H_{23}, \text{Root}, 0\rangle$	$L_{11}=\{\}$
S_3	$\langle S_3, H_6, \text{Root}, 1\rangle$	$L_3=\{\}$	S_{12}	$\langle S_{12}, D_0, \text{Root}, 1\rangle$	$L_{12}=\{S_0\}$
S_4	$\langle S_4, H_7, \text{Root}, 1\rangle$	$L_4=\{\}$	S_{13}	$\langle S_{13}, D_1, \text{Root}, 1\rangle$	$L_{13}=\{S_1\}$
S_5	$\langle S_5, H_9, \text{Root}, 1\rangle$	$L_5=\{\}$	S_{14}	$\langle S_{14}, D_2, \text{Root}, 1\rangle$	$L_{14}=\{S_2\}$
S_6	$\langle S_6, H_{10}, \text{Root}, 1\rangle$	$L_6=\{\}$	S_{15}	$\langle S_{15}, D_3, \text{Root}, 1\rangle$	$L_{15}=\{S_3, S_5, S_7\}$
S_7	$\langle S_7, H_{11}, \text{Root}, 0\rangle$	$L_7=\{\}$	S_{16}	$\langle S_{16}, D_4, \text{Root}, 1\rangle$	$L_{16}=\{S_4, S_8, S_{10}\}$
S_8	$\langle S_8, H_{12}, \text{Root}, 0\rangle$	$L_8=\{\}$	S_{17}	$\langle S_{17}, D_5, \text{Root}, 1\rangle$	$L_{17}=\{S_6, S_9, S_{11}\}$

Table 6 Information of vulnerability nodes

No.	Node information
V_1	$\langle V_1, H_6, \text{CVE-2011-0638}, 0.1\rangle$
V_2	$\langle V_2, H_7, \text{CVE-2011-0638}, 0.1\rangle$
V_3	$\langle V_3, H_9, \text{CVE-2011-0638}, 0.1\rangle$
V_4	$\langle V_4, H_{12}, \text{CVE-2011-0638}, 0.1\rangle$
V_5	$\langle V_5, H_{20}, \text{CVE-2011-0638}, 0.1\rangle$
V_6	$\langle V_6, H_{23}, \text{CVE-2011-0638}, 0.1\rangle$
V_7	$\langle V_7, H_{10}, \text{CVE-2014-6271}, 0.7\rangle$
V_8	$\langle V_8, H_1, \text{CVE-2013-2249}, 0.7\rangle$
V_9	$\langle V_9, H_{11}, \text{CVE-2018-8225}, 0.5\rangle$
V_{10}	$\langle V_{10}, H_{12}, \text{CVE-2018-8225}, 0.5\rangle$
V_{11}	$\langle V_{11}, H_{18}, \text{CVE-2018-8225}, 0.5\rangle$
V_{12}	$\langle V_{12}, H_{20}, \text{CVE-2018-8225}, 0.5\rangle$
V_{13}	$\langle V_{13}, H_4, \text{CVE-2018-5703}, 0.7\rangle$

Table 7 Attack paths with top 10 success probability by LAG

Attack path	Success probability
$S_0\to S_1\to S_3\to S_6\to S_2$	0.034
$S_0\to S_1\to S_5\to S_6\to S_2$	0.034
$S_0\to S_1\to S_3\to S_9\to S_2$	0.025
$S_0\to S_1\to S_5\to S_9\to S_2$	0.025
$S_0\to S_1\to S_3\to S_8\to S_6\to S_2$	0.019
$S_0\to S_1\to S_3\to S_{10}\to S_6\to S_2$	0.019
$S_0\to S_1\to S_5\to S_8\to S_6\to S_2$	0.019
$S_0\to S_1\to S_5\to S_{10}\to S_6\to S_2$	0.019
$S_0\to S_1\to S_3\to S_8\to S_9\to S_2$	0.013
$S_0\to S_1\to S_3\to S_{10}\to S_9\to S_2$	0.013

Table 8 Attack paths with top 10 success probability by attack graph

Attack path	Success probability
$S_0\to S_1\to S_3\to S_6\to S_2$	0.034
$S_0\to S_1\to S_5\to S_6\to S_2$	0.034
$S_0\to S_1\to S_3\to S_9\to S_2$	0.025
$S_0\to S_1\to S_5\to S_9\to S_2$	0.025
$S_0\to S_1\to S_3\to S_8\to S_6\to S_2$	0.019
$S_0\to S_1\to S_3\to S_{10}\to S_6\to S_2$	0.019
$S_0\to S_1\to S_5\to S_8\to S_6\to S_2$	0.019
$S_0\to S_1\to S_5\to S_{10}\to S_6\to S_2$	0.019
$S_0\to S_1\to S_3\to S_6\to S_9\to S_2$	0.017
$S_0\to S_1\to S_3\to S_7\to S_6\to S_2$	0.017

From the view of node occurrence frequency, except attack start state nodes and target state nodes, the state nodes with the highest frequency in the two attack paths are nodes $\{S_1, S_3, S_6\}$, and the corresponding hosts are $\{H_1, H_6, H_{10}\}$. If the defense resources are limited, the three hosts should be defended priority. In addition, compared with the LAG, the state node S_7 appears in attack graph. In the attack path which contains state node S_7, S_3 and S_7 belong to the same subnet with the same privilege. Attacker can reach the state node S_6 directly after reaching the state S_3, and it is unnecessary to attack S_7.

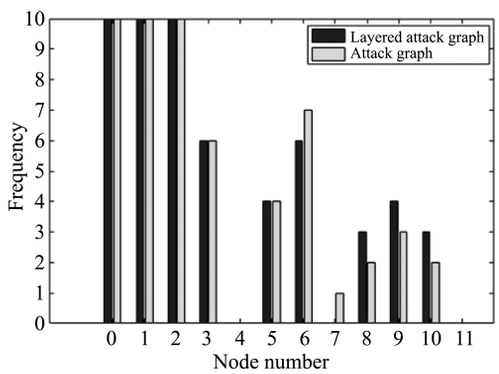

Fig. 6 Frequency of different nodes in different attack path sets

As whole, the frequency trend of each node in the two attack paths is concordant. If choosing different numbers of nodes for defense, we can provide the same security strategy. But the space-time efficiency of using LAG to generate attack path set is more effective than that of network attack graph. Therefore, the attack path generation based on the proposed LAG can effectively reduce attack path scale.

4 Conclusion

In this paper, by optimizing the selection of attack path based on the monotonic principle of attack behavior, the state explosion problem in NSSA has been tackled. Furthermore, experiment results have shown that our method can significantly improve the real-time of NSSA without reducing the accuracy.

There are still some problems in this method. The number and size of subnets will affect the performance of NSSA, which can be controlled in the community discovery algorithm. In the future, these problems will be focused in the light of defense demand.

References

[1] Liu X W, Wang H Q, Lü H W, et al. Fusion-based cognitive awareness-control model for network security situation. Journal of Software, 2016, 27(8): 2099-2114.

[2] Zhang Y, Tan X B, Cui X L, et al. Network security situation awareness approach based on markov game model. Journal of Software, 2011, 22(3): 495-508.

[3] Bass T. Intrusion detection systems and multisensor data fusion. Communications of the ACM, 2000, 43(4): 99-105.

[4] Zang X D, Su Q. Survey of network security situation awareness. Journal of Software, 2017, 28(4): 1010-1026.

[5] Chen X J, Fang B X, Tan Q F, et al. Inferring attack intent of malicious insider based on probabilistic attack graph model. Chinese Journal of Computers, 2014, 37(1): 62-72.

[6] Wang Y J, Xian M, Liu J, et al. Study of network security evaluation based on attack graph model. Journal on Communications, 2007, 28(3): 29-34.

[7] Liu Q, Yin J P, Cai Z P, et al. Uncertain-graph based method for network vulnerability analysis. Journal of Software, 2011, 22(6): 1398-1412.

[8] Zeng S W, Wen Z H, Dai L W, et al. Analysis of network security based on uncertain attack graph path. Computer Science, 2017, 44(S1): 351-355.

[9] Zhang C M, Chen T P, Zhang X Y, et al. A method of evaluating network system risk events probability based on attack tree. Fire Control & Command Control, 2010, 35(11): 17-19.

[10] Chen F, Liu D H, Zhang Y, et al. A hierarchical evaluation approach for network Security based on threat spread model. Journal of Computer Research and Development, 2011, 48(6): 945-954.

[11] Tian J W, Tian Z, Qi W H, et al. Threat propagation based security situation quantitative assessment in multi—node network. Journal of Computer Research and Development, 2017, 54(4): 731-741.

[12] Ma G, Du Y G, An B, et al. Risk evaluation of complex information system based on threat propagation sampling. Journal of Computer Research and Development, 2015, 52(7): 1642-1659.

[13] Lü H Y, Peng W, Wang R M, et al. A real-time network threat recognition and assessment method based on association analysis of time and space. Journal of Computer Research and Development, 2014, 51(5): 1039-1049.

[14] Liu Y L, Feng D G, Lian Y F, et al. Network situation prediction method based on spatial-time dimension analysis. Journal of Computer Research and Development, 2014, 51(8): 1681-1694.

[15] Wang S, Tang G M, Kou G, et al. Attack path prediction method based on causal knowledge net. Journal on Communications, 2016, 37(10): 188-198.

[16] Kaynar K, Sivrikaya F. Distributed attack graph generation. IEEE Transactions on Dependable & Secure Computing, 2016, 13(5): 519-532.

[17] Ammann P, Wijesekera D, Kaushik S. Scalable graph-based network vulnerability analysis. In: Proceedings of ACM Conference on Computer and Communications Security, Washington, 2002: 217-224.

[18] Blondel V D, Guillaume J L, Lambiotte R, et al. Fast unfolding of communities in large networks. Journal of Statistical Mechanics, 2008, (10): 155-168.

[19] Newman M E J. Fast algorithm for detecting community structure in networks. Physical review E, 2004, 69(6): 066133.

[20] Liu Y, Kang X H, Gao H, et al. A community detecting method based on the node intimacy and degree in social network. Journal of Computer Research and Development, 2015, 52(10): 2363-2372.

[21] Xi R R, Yun X C, Zhang Y Z. Quantitative threat situational assessment based on contextual information. Journal of Software, 2015, 26(7): 1638-1649.

基于层次攻击图的网络安全态势感知方法

朱宇辉[1,2]，宋礼鹏[1,2]

(1. 中北大学 大数据学院，山西 太原 030051；2. 中北大学 大数据与网络安全研究所，山西 太原 030051)

摘　要： 现有的网络安全态势感知算法通常会面临状态爆炸问题，严重影响算法的实时性。针对这一问题，提出了一个基于层次攻击图的网络安全态势感知方法。首先，利用社区发现算法将原网络划分为多个逻辑子网，所有逻辑子网及其之间的连接关系构成网络的逻辑结构。然后，基于原始网络和逻辑网络结构，优化攻击路径的选择，避免生成不符合攻击行为单调性原则的路径。该方法有效缩减了攻击路径规模，解决了状态爆炸问题。实验结果表明，基于层次攻击图的攻击路径生成用时仅为 0.029 s，与其它方法用时 56 s 相比提升显著。此外，该方法能提供与其它方法一致的防御策略。

关键词： 网络安全态势感知；层次攻击图；状态爆炸；社区发现

引用格式： ZHU Yu-hui, SONG Li-peng. A network security situation awareness method based on layered attack graph. Journal of Measurement Science and Instrumentation, 2019, 10(2): 182-190. [doi: 10.3969/j.issn.1674-8042.2019.02.011]