Journal of Computers Vol. 29 No. 3, 2018, pp. 56-72
doi:10.3966/199115992018062903006

An FCM-based Hierarchical Method for Evaluating Network Security Situation

Zhijie Fan[1,2*], Zhiping Tan[3], Chengxiang Tan[1], Xin Li[4]

[1] Electronics and Information Engineering School, Tongji University, Shanghai 201804, China

[2] The Third Research Institute of Ministry of Public Security, Shanghai 201204, China
{1310898, jerrytan}@tongji.edu.cn

[3] Huawei Technologies Co. Ltd, Shanghai 201206, China
aaronzptan@gmail.com

[4] College of Information Technology and Cyber Security, People's Public Security University of China, Beijing 100038, China
lixin1999@126.com

Received 11 December 2016; Revised 25 May 2017; Accepted 29 June 2017

Abstract. Network security situation assessment is an important research topic in the field of network security. In particular, the hierarchical analysis method is widely used in practice. However, the current assessment methods neglect common interrelation and restrictive correlation among security situation factors, and lack of security events backtracking capability. In this work, we proposed a new network security situation assessment method based on Fuzzy Cognitive Maps (FCM). Firstly, we created a structured description of the original security events. Secondly, we generated the FCM structure semi-automatically according to the original structured security events via the FCM build method we proposed. Thirdly, we classified the concept nodes into four types, i.e., vulnerability, service, host and system. Fourthly, we computed the security situation values of each type and the value of network security comprehensive situation. Fifthly, we assessed the network security comprehensive situation (NSCS) according to the network security state level table. At last, we introduced how to find the high risk events and trace the precondition. We used the DARPA2000 dataset which is developed by Lincoln Laboratory to verify and analyze our method and illustrated how to trace back the high risk events. The result shows that our method can model the network security situation accurately, and also has the security risk events backtracking capability.

Keywords: fuzzy cognitive maps, hierarchical analysis, network security situation, situation evaluation, tracing back

1 Introduction

In recent years, a huge number of different network security events are springing up. The traditional network security protection methods which mainly use the firewall and intrusion detection technology cannot meet the security needs of network in the reality. Network security situation awareness, as a new solution for network security protection, has drawn attractions of scholars all over the world. It is a supplement to the traditional solution for network security protection. It can enhance the ability of protecting the security of network systems.

Scholars from all over the world have done a lot of researches in the field of network security situation awareness and assessment. Endsley [1] and Bass [2] have made great contributions to the research of network security situation awareness. They proposed respectively the conceptual model and function

* Corresponding Author

model of situation awareness. Although a variety of network security situation awareness models have been proposed, they have the similar basically functions. Situation assessment is an important part of situation awareness, it is used to analyze network security situation according to quantitative analysis process of situation elements.

In this paper, we study the network security situation assessment. Scholars from all over the world have done a lot of researches in the field of network security situation assessment. The hierarchical analysis method is widely used in the practice. However, the methods neglect common interrelation and restrictive correlation among security situation factors, and lack of security events backtracking capability. To solve the above problems, we propose a new hierarchical network security situation assessment method based on Fuzzy Cognitive Maps (FCM). The method cannot only consider the common interrelation and restrictive correlation among security situation factors, but also has the security events backtracking capability.

The rest of this paper is organized as follows. In Section 2, we review the related work. Section 3 describes our hierarchical method based on FCM, in this section, the regional similarity degree calculation method is introduced. Section 4 presents our network security situation assessment method. Section 5 shows the simulations that describe experimental analysis. Finally, concluding remarks are made in Section 6.

2 Related Works

Hierarchical analysis framework of network security situation assessment has been proposed in [3]. It was used to synthetically analyze network security situation according to situation elements, and was most commonly used in practice. Another hierarchical analysis method was proposed in [4]. It was based on statistics about network alert frequency, alarm severity and network bandwidth consumption rate. The weights was assigned to each layer of vulnerability, service, host and system, then the value of network security situation was calculated according to the threat index. An improved framework of hierarchical analysis was proposed in [5]. It was based on a classical network security situation analysis (NSSA) model, and provided a standard flow for analyzing the security situation of information system. In [6], a conceptual framework and a method was proposed, it assessed the impact that network attacks might have to network assets, services, and missions. It described the model of network attack based on an extended conceptual graph.

A variety of network security situation assessment frameworks have been proposed. In [7], a complementary situation assessment method was proposed, it avoided one-sidedness and inaccuracy of individual, and shared situation assessment. In [8], a network security situation assessment method based on attack intention recognition was proposed. It was based on intruder, and discovered intrusion path to recognize every attack stages by using causal analysis of attack events. Then, it realized situation assessment based on the attack stages. In [9], the network security evaluation method based on attack intention guess was raised. It consisted of multi-source fusion decision, threat spread analysis.

Many kinds of specific network security situation assessment technology have been proposed. The ARMA method of security situation assessment was proposed based on information entropy and information fusion [10]. The method that fused multi-source alarm information through D-S evidence theory, and associated with nodes vulnerability information, integrates with the severity of threats [11]. The Hidden Semi-Markov Model (HsMM) was made use of to simulate the operation of network system, and the method was verified by experiment [12]. In [13], the authors gave a method of security information conversion from the low level to the high one based on STIX ontology, and then completed cyber security situation assessment by effectively identifying the high level security information. In [14], the authors utilized sliding time window mechanism to extract the observed value and hybrid multi-population genetic algorithm (MPGA) to train the HMM model parameters, so as to improve the reliability of parameters. It has improved the real-time performance and accuracy of the evaluation. In [15], the authors introduced a min-cut set and presented a new method to assess the network security situation under DDoS attacks. It has computed the influence value that attacks cause on network security situation according to the distance between the congested link and victim. Whether the link is in the min-cut set, the value is used for quantitative situation assessment. In [16], the authors studied the network security situation evaluation method relevantly. When a network includes multiple subnets, the network situation of the total network needs to be further aggregated. They proposed a new aggregation model and method.

In recent years, graph model has been widely used in the field of network security situation assessment. In particular, Bayesian Network Graph (BNG), Attack Graph (AG) and Fuzzy Cognitive Maps (FCM) have attracted much attention from scholars. A classifier based on Bayesian Network has been developed to analyze the network traffic in a network security situation in order to finish the network security situation assessment [17]. Another network security situation assessment system was developed by using attack graph model, it used Attack Graph (AG) and service dependencies to describe network security situation values over time [18]. Fuzzy Cognitive Maps (FCM) was first used to analyze risk impact factors, but it has not been used in the field of security situation assessment [19]. A lightweight method for security risk assessment based on fuzzy cognitive maps was proposed, but it was only fit for small scale network environment [20-21].

All these researches above solved the problems of assessing network security situation from different angles. However, the methods above neglect the analysis of common interrelation and restrictive correlation among security situation factors, and lack of security events backtracking capability.

In this paper, we proposes a new network security situation assessment method based on Fuzzy Cognitive Maps (FCM). Because FCM structure can quantitatively describe the causal relationship between any two concept nodes, our key research is that we use the feature of FCM to sufficiently considers the common interrelation and restrictive correlation among the different security factors of network security situation. In order to effectively use the feature of FCM to assess network security situation, we put forward a regional similarity calculation method for semi-automatically building FCM structure. Meanwhile, our network security situation assessment method has the ability of tracing back the high-risk events in network environment. Quantitative causal relationship between any two concept nodes in FCM structure provides a feasible way to trace back the high-risk events. The instance of security events backtracking is explained in experimental section.

3 An FCM-based Hierarchical Method

In this section, our hierarchical analysis method of network security situation assessment based on FCM is introduced in Fig. 1. We divide the processes of assessment into several phases. Firstly, a structured description of the original security events is created. Secondly, FCM structure is semi-automatically generated according to the original structured security events. Thirdly, the concept nodes are classified into four types, i.e., vulnerability, service, host and system. Fourthly, the security situation values of each type and the value of network security comprehensive situation (NSCS) are calculated. Fifthly, the NSCS is assessed according to the network security state level table. The last, we can find the high risk events and tracing the precondition.

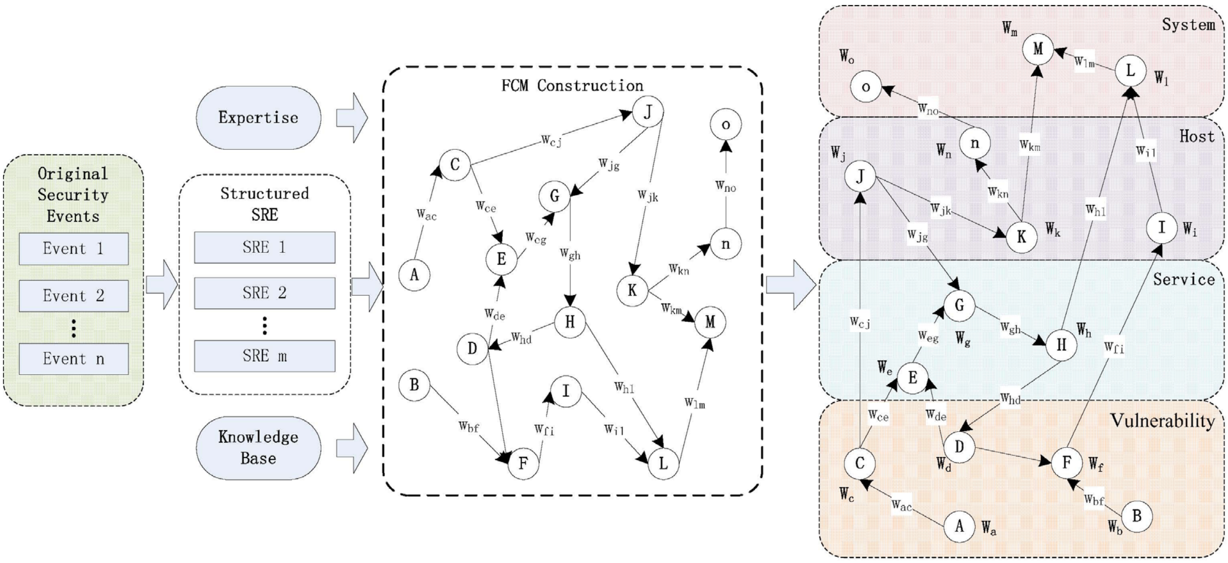

Fig. 1. Flow diagram of the FCM-based hierarchical method

Security Risk Event (SRE) is an activity that happens at a specific place and time caused by necessary security risk causes (such as system vulnerability and network attack) or unavoidable security risk consequences (such as service disruption and data leakage). In this work, the content of structural description of SRE contains three parts: basic information, correlation information and state information as shown in Table 1.

Table 1. The content of structural description of SRE

Description Set	Elements Set
Basic Information	Topic ID Date Summary
Correlation Information	Direct causes Direct consequences Context Parent nodes Child nodes
State Information	State level Monitor index

Some basic concepts in Table 1 are introduced as follows.

Definition 1. Basic Information is a description information set related to SRE itself, and it is represented as a multi-tuple: $BASIC\_INFO_i=\{Topic_i, ID_i, Date_i, Summary_i\}$, where $Topic_i$ refers to the name of SRE topic, ID_i refers to the identifier of the SRE, $Date_i$ refers to the time when the SRE takes place, and $Summary_i$ refers to the summary of the SRE.

Definition 2. Correlation Information is a set related to the description of relationship between two SREs, and it is represented as a multi-tuple: $COL\_INFO_i= \{Direct\_Causes_i, Direct\_Consequences_i, Context_i, Parent\_Nodes_i, Child\_Nodes_i\}$, where $Direct\_Causes_i$ refers to the possible direct causes of the SRE in the context, $Direct\_Consequences_i$ refers to the possible direct consequences of the SRE in the context, $Context_i$ refers to the context of the happening SRE, $Parent\_Nodes_i$ refers to the direct causes of happened SRE, and $Child\_Nodes_i$ refers to the direct consequences of the happened SRE. $Direct\_Causes_i$ is represented as a multi-tuple: $IC=\{IC_1, IC_2, ..., IC_i, ..., ICn\}$, and $Direct\_Consequences_i$ is represented as a multi-tuple: $DR=\{DR_1, DR_2, ..., DR_i, ..., DRn\}$.

Definition 3. State Information is a set related to the description of the SRE's state, and it is represented as a multi-tuple: $STATE\_INFO_i=\{State\_Level_i, Monitor\_Index_i\}$, where $State\_Level_i=f(Minitor\_Index)$ refers to the probability degree of the occurring event, and $Monitor\_Index_i$ refers to the index set monitoring whether a SRE has occurred, which can be formalized into $Minitor\_Index_i=\{MI_1,MI_2,...,MI_n\}$.

3.1 Fuzzy Cognitive Maps (FCM)

Fuzzy Cognitive Maps (FCM) was firstly proposed by professor Kosko [22], who combined cognitive map with fuzzy set theory. FCM model consists of node, directed edges and weights of directed edges, and it is a weighted directed graph that can describes causal relationship. The node in FCM is called concept node, which can describe the abstract things, concrete things, activities, system properties and system statuses according to the needs of system. The weights of directed edges in FCM structure are used to describe the causal relationship between any two concept nodes. Directed edges can be viewed as single layer neural network with feedbacks and the object-oriented concept. The knowledge is inside of concept nodes and weighted directed edges. FCM model uses weighted directed relationships to simulate fuzzy reasoning, in which the interrelationships among concept nodes are used to stimulate dynamic behavior of system.

Definition of FCM: all concept nodes: $c_1, c_2, ..., c_i, ..., c_n$ exist in FCM, the value's range of weighted directed edges is [-1, 1], e_{ij} is weight value of edge $<C_i, C_j>$, the matrix $E=f(e_{ij})$ is called an adjacent matrix or incidence matrix of FCM. After building the FCM model and obtaining the initial status values of all concept nodes, the status values of all concept nodes at any time can be calculated by the following formula:

$$A_i(t+1) = f\left(A_i(t) + \sum_{j=1, j \neq i}^{n} A_j(t) w_{ji} \right) \quad \textbf{(1)}$$

where suppose that $C=\{ c_1, c_2, \ldots, c_i, \ldots, c_n \}$ is a set of all concept nodes, $n=|C|$, and C_i is the value of the i-th concept node, recorded as A_i after mapping to range [0, 1], and it means the status value of concept node. $A_i(t)$ means the status value of i-th concept node at time t, and $A_i(t+1)$ means the status value of i-th concept node at time $t+1$. w_{ji} is the incidence matrix of concept nodes, also named as adjacent matrix. f is a function, the two or three valued step function and S-curve function are commonly used in practice.

3.2 FCM Structure Generation

FCM structure building is a key part of FCM model. It needs to determine the concept nodes and weighted interconnections using artificial method or semi-automatically method. Generally, the FCM structure is determined by artificial method, so the subjective intention features in the FCM model deviates from the actual situation. In this paper, we propose a regional similarity calculation method for FCM building semi-automatically. The flow diagram of FCM structure building is in Fig. 2. Firstly, we create a structured description of limited original concept nodes after deep analysis according to the combination of expert experience and historical data. Secondly, we carry out several continuous iteration based on the regional similarity calculation method we proposed. The final concept nodes of FCM will be calculated out. Thirdly, we complete the interconnections among concepts nodes automatically. Finally, we confirm the weights of directed edges by using typical machine learning technology.

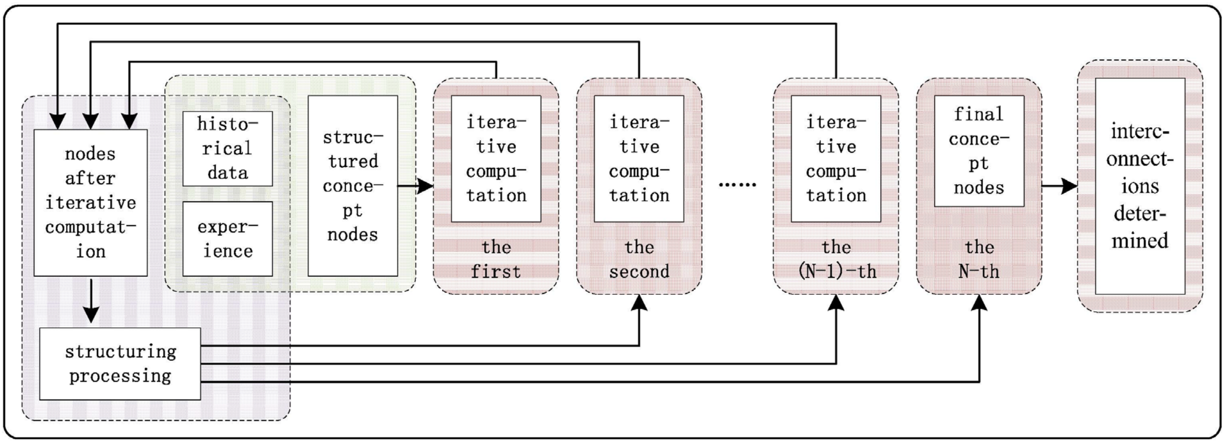

Fig. 2. Flow diagram of FCM structure building

3.3 Regional Similarity Degree Calculation Method

How to obtain all concept nodes of FCM by means of limited original concept nodes and how to determine interconnections among them will be introduced in this section. The definition of SRE similarity degree in this section is as follows.

Definition 4. SRE Similarity Degree describes the similarity between any two SRE concept nodes. It is represented as $sim(c_i, c_j)$, and meets the following conditions:

(1) $0 \le sim(c_i, c_j) \le 1$.

(2) $sim(c_i, c_j) \ne sim(c_j, c_i)$.

(3) $sim(c_i, c_j) = 1$ means that c_i is completely similar to c_j.

(4) $sim(c_i, c_j) = 0$ means that c_i is completely different from c_j.

Assuming there are two concept nodes c_i and c_j. They have respective sets of direct causes and direct consequences. In this paper, the concept nodes means SREs in real environment. Next, the calculation method of regional similarity degree between c_i and c_j is introduced.

R_i is the direct causes vector of c_i, and it is expressed as follows.

$R_i = \left(\left\langle IC_1^A, w_{IC1}^A \right\rangle, \cdots, \left\langle IC_i^A, w_{ICi}^A \right\rangle, \cdots, \left\langle IC_n^A, w_{ICn}^A \right\rangle\right)$, where IC_i^A is *i*-th direct causes of concept node c_i, w_{ICi}^A is the weight of IC_i^A in R_i.

O_i is the direct consequences vector of c_i, and it is expressed as follows.

$O_i = \left(\left\langle DR_1^A, w_{DR1}^A \right\rangle, \cdots, \left\langle DR_i^A, w_{DRi}^A \right\rangle, \cdots, \left\langle DR_n^A, w_{DRn}^A \right\rangle\right)$, where DR_i^A is *i*-th direct consequences of concept node c_i, w_{DRi}^A is the weight of DR_i^A in O_i.

R_j is The direct causes vector of c_j, and it is expressed as follows.

$R_j = \left(\left\langle IC_1^B, w_{IC1}^B \right\rangle, \cdots, \left\langle IC_j^B, w_{ICj}^B \right\rangle, \cdots, \left\langle IC_n^B, w_{ICn}^B \right\rangle\right)$, where IC_j^B is *j*-th direct causes of concept node c_j, w_{ICj}^B is the weight of IC_j^B in R_j.

O_j is the direct consequences vector of c_j, and it is expressed as follows.

$O_j = \left(\left\langle DR_1^B, w_{DR1}^B \right\rangle, \cdots, \left\langle DR_j^B, w_{DRj}^B \right\rangle, \cdots, \left\langle DR_n^B, w_{DRn}^B \right\rangle\right)$, where DR_j^B is *j*-th direct consequences of concept node c_j, w_{DRj}^B is the weight of DR_j^B in O_j.

Then, the regional similarity degree calculation method between c_i and c_j is expressed as $sim(c_i, c_j) = f(R_i, O_i, R_j, O_j, T_i, T_j)$, where T_i and T_j are the context of c_i and c_j, respectively. And the regional similarity degree calculation method is as follows:

(1) if $T_i \neq T_j$, then $sim(c_i, c_j) = 0$.

(2) if $T_i = T_j$, $R_i = R_j$, and $O_i = O_j$, then $sim(c_i, c_j) = 1$.

(3) if $T_i = T_j$, $R_i \neq R_j$, and $O_i \neq O_j$, then $sim(c_i, c_j) = f(O_i, R_j)$ and $sim(c_j, c_i) = f(O_j, R_i)$.

Next, we take regional similarity degree $sim(X, Y)$ of arbitrary vectors X and Y as an example to explain how to get the $sim(c_i, c_j)$ according to O_i and R_j. The calculation process of $sim(X, Y)$ is introduced as follow. The vectors are $X = \left(\left\langle x_1, w_{x1} \right\rangle, \cdots, \left\langle x_i, w_{xi} \right\rangle, \cdots, \left\langle x_n, w_{xn} \right\rangle\right)$ and $Y = \left(\left\langle y_1, w_{y1} \right\rangle, \cdots, \left\langle y_i, w_{yi} \right\rangle, \cdots, \left\langle y_m, w_{ym} \right\rangle\right)$. X' and Y' are in descending order according to the weights of X and Y, and then we get two new vectors: $X' = \left(\left\langle x'_1, w'_{x1} \right\rangle, \cdots, \left\langle x'_i, w'_{xi} \right\rangle, \cdots, \left\langle x'_n, w'_{xn} \right\rangle\right)$ and $Y' = \left(\left\langle y'_1, w'_{y1} \right\rangle, \cdots, \left\langle y'_i, w'_{yi} \right\rangle, \cdots, \left\langle y'_m, w'_{ym} \right\rangle\right)$.

We extract $X'' = \left(x'_1, x'_2, \cdots, x'_i, \cdots, x'_n\right)$ and $Y'' = \left(y'_1, y'_2, \cdots, y'_i, \cdots, y'_m\right)$ from X' and Y', then combine x'_i with x'_{i-1} and x'_{i+1} to make a new element $x'_{i-1}\, x'_i\, x'_{i+1}$, and the result is $X''' = \left(x'_1\, x'_2, x'_1\, x'_2\, x'_3, \cdots, x'_{i-1}\, x'_i\, x'_{i+1}, \cdots, x'_{n-1}\, x'_n\right)$. And combine y'_i with y'_{i-1} and y'_{i+1} to make a new element $y'_{i-1}\, y'_i\, y'_{i+1}$, and the result is $Y''' = \left(y'_1\, y'_2, y'_1\, y'_2\, y'_3, \cdots, y'_{i-1}\, y'_i\, y'_{i+1}, \cdots, y'_{m-1}\, y'_m\right)$.

The weights of element $w_{x'_{i-1} x'_i x'_{i+1}}$, $w_{x'_1 x'_2}$, and $w_{x'_{n-1} x'_n}$ are as follows:

$w_{x'_{i-1} x'_i x'_{i+1}} = (w'_{x'_{i-1}} + w'_{x'_i} + w'_{x'_{i+1}})/3$, $w_{x'_1 x'_2} = (w'_{x'_1} + w'_{x'_2} + 1)/3$, and $w_{x'_{n-1} x'_n} = (w'_{x'_{n-1}} + w'_{x'_n} + 0)/3$.

The weight vectors of X''' and Y''' are as follows:

$w_{X'''} = \left(w_{x'_1 x'_2}, w_{x'_1 x'_2 x'_3}, \cdots, w_{x'_{i-1} x'_i x'_{i+1}}, \cdots, w_{x'_{n-1} x'_n}\right)$ and $w_{Y'''} = \left(w_{y'_1 y'_2}, w_{y'_1 y'_2 y'_3}, \cdots, w_{y'_{i-1} y'_i y'_{i+1}}, \cdots, w_{y'_{m-1} y'_m}\right)$.

Thus we have the final weight from the following equation: $W = \left(w_1, w_2, \cdots, w_i, \cdots, w_{\min(n,m)}\right)$, where $w_i = (w_{x'_{i-1} x'_i x'_{i+1}} + w_{y'_{i-1} y'_i y'_{i+1}})/2$.

Next we use the logic function in Table 2 to calculate the value of $sim(x'_{i-1}\, x'_i\, x'_{i+1}, y'_{i-1}\, y'_i\, y'_{i+1})$. In the Table 2, $K = \left(k_0, k_1, \cdots, k_8\right)$ with $k_i = 1$ or $k_i = 0$. We take k_0 as an example. If $sim(x'_{i-1}, y'_{i-1}) > \theta$, then $k_0 = 1$, where θ is a threshold value. If $sim(x'_{i-1}, y'_{i-1}) \leq \theta$, then $k_0 = 0$. The value of $sim(x'_{i-1}, y'_{i-1})$ is obtained by Hamming distance function.

Table 2. Logic function table

	y'_{i-1}	y'_i	y'_{i+1}
x'_{i-1}	k_0	k_1	k_2
x'_i	k_3	k_4	k_5
x'_{i+1}	k_6	k_7	k_8

According to the value of $K=(k_0,k_1,\cdots,k_8)$, $sim(x'_{i-1}\,x'_i\,x'_{i+1},y'_{i-1}\,y'_i\,y'_{i+1})$ is computed as follow:

$$sim(x'_{i-1}\,x'_i\,x'_{i+1},y'_{i-1}\,y'_i\,y'_{i+1})=k_0w'_{x(i-1)}\,w'_{y(i-1)}+k_1w'_{x(i-1)}\,w'_{yi}+k_2w'_{x(i-1)}\,w'_{y(i+1)}+\cdots+k_8w'_{x(i+1)}\,w'_{y(i+1)}$$

It is important to note that because of the repeated calculated contribution of x'_i in $x'_{i-2}\,x'_{i-1}\,x'_i$, $x'_{i-1}\,x'_i\,x'_{i+1}$ and $x'_i\,x'_{i+1}\,x'_{i+2}$, we give priority to calculated contribution of x'_i in $x'_{i-2}\,x'_{i-1}\,x'_i$. If the calculated contribution of x'_i have been collected, we ignore the contribution of others. The same applies to *Y'''*. Finally, $sim(X,Y)$ is obtained from the following equation:

$$sim(X,Y)=sim(X''',Y''')=\sum_{i=1}^{\min(n,m)} w_i sim(x'_{i-1}\,x'_i\,x'_{i+1},y'_{i-1}\,y'_i\,y'_{i+1})$$

3.4 Operation Rules about Concept Nodes

According to the method in the previous section, we have the regional similarity degree matrix S:

$$S=\begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1j} & \cdots & s_{1n} \\ s_{21} & s_{22} & \cdots & s_{2j} & \cdots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ s_{i1} & s_{i2} & \vdots & s_{ij} & \vdots & \mathrm{s_{in}} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & \cdots & s_{nj} & \cdots & s_{nn} \end{pmatrix}$$

where s_{ij} is the regional similarity degree between c_i and c_j.

Assume *A* and *B* are two concept nodes. $sim(A,B)$ is the regional similarity degree between *A* and *B*, and $sim(B,A)$ is the regional similarity degree between *B* and *A*. We have the operation rules about *A* and *B* as follows.

Add concept node.

Rule 1. If $sim(A,B)<sim(B,A)$, a new concept node *C* will be added, which is the element of maximum weight in direct consequences of *A*. And another new concept node *D* will be added, which is the element of maximum weight in direct causes of *B*.

Rule 2. If $sim(A,B)>sim(B,A)$, a new concept node *C* will be added, which is the element of maximum weight in direct consequences of *B*, and another new concept node *D* will be added, which is the element of maximum weight in direct causes of *A*. $sim(A,B)=1$

Merge concept nodes.

Rule 3. If $sim(A,B)=1$, a new concept node *C* will be merged, which has the same direct causes and direct consequences with *A* and *B*. If $sim(A,B)\neq 1$, a new concept node *C* will be merged, which has the same direct causes with *A*, and the same direct consequences with *B*.

Rule 4. If $sim(B,A)=1$, a new concept node *C* will be merged, which has the same direct causes and direct consequences with *B* and *A*. If $sim(B,A)\neq 1$, a new concept node *C* will be merged, which has the same direct causes with *B*, and the same direct consequences with *A*.

Delete concept node.

Rule 5. If $sim(A,B)=0$, or $sim(B,A)=0$, the node will be deleted, which has different context between A and B.

Change concept nodes.

Rule 6. If $sim(A,B) > sim(B,A)$, the operation rule 3 and rule 2 will be carried out.

Rule 7. If $sim(A,B) < sim(B,A)$, the operation rule 4 and rule 1 will be carried out.

3.5 FCM Structure Building

In this section, the method of building FCM structure is introduced. We describe the FCM structure building steps by using four algorithm pseudo codes as follows.

In the algorithm pseudo code 1, the input C means set of concept nodes, and the output S means regional similarity degree matrix. We use the algorithm 1 to initialize/update the regional similarity degree matrix that is obtained according to our method in section 3.3 after semi-automatically filling the content of all concept nodes with basic information, correlation information and state information.

Algorithm pseudo code 1.

```
   Function: Initialize/Update Regional Similarity Degree Matrix
   Input:
     C (set of concept nodes)
   Output:
     S (regional similarity degree matrix)
   Initialize/Update_Matrix (C, S)
     {
       define object M the set of concept nodes
       int t <- num[M]
       create array S[i][j]
       for t <- 1 to length[M] do
         M[t] <- c_p
       end for
       for t <- 1 to length[M] do
         fill the content of all nodes with basic information,
correlation information and state information
       end for
       for i <- 1 to length[S] do
         for j<- 1 to length[S[i]] do
           use our method in section 3.3 to calculate s_ij according to
the information of concept nodes
           save as S
         end for
       end for
     }
   end.
```

In the algorithm pseudo code 2, the input S means regional similarity degree matrix that is obtained by using algorithm 1, the inputs δ and μ are lower and upper threshold, and the output C means set of concept nodes. We use the algorithm 2 to automatically create all concept nodes of FCM by using the operation rules about concept nodes in section 3.4.

Algorithm pseudo code 2.

```
   Function: Create all concept nodes of FCM
   Input:
     S (regional similarity degree matrix)
     δ (lower threshold), μ(upper threshold)
   Output:
     C' (set of concept nodes)
   Create_Nodes (S, δ, μ, C')
     {
       define object C' the set of concept nodes
       int t <- num[C']
       for ∀s_ij, s_ji ∈ S do
         if s_ij=s_ji=0 then add concept nodes between c_i and c_j
```

```
        else if sij=sji=1 then merge concept nodes between ci and cj
        else if sij•sji then
          if 0<sij• δ and δ<sji• μ then add concept nodes between ci and cj
          else if δ<sij<1 and μ<sji<1 then merge concept nodes between cj and ci
          else if 0<sij• δ and μ<sji<1 then change concept nodes between cj and ci
          end if
        else if sij>sji then
          if δ<sij• μ and 0<sji• δ then add concept nodes between cj and ci
          else if μ<sij<1 and δ<sji<1 then merge concept nodes between ci and cj
          else if μ<sij<1 and 0<sji• δ then change concept nodes between ci and cj
          end if
        else if δ<sij• μ and δ<sji• μ then
            delete repeated concept nodes
          for t <- 1 to length[M] do
              save as C'
          end for
        end if
      end for
    }
  end.
```

In the algorithm pseudo code 3, the input *C′* means set of concept nodes that is obtained by using algorithm 2, input *S* means regional similarity degree matrix that is obtained by using algorithm 1, the inputs θ and μ are middle and upper threshold, and the output *T* means the directed edges matrix of FCM. When all the concept nodes of FCM are created, we use algorithm 3 to create all directed edges of FCM.

Algorithm pseudo code 3.

```
  Function: Create all directed edges matrix of FCM
  Input:
    C' (set of concept nodes)
    S (regional similarity degree matrix)
    μ(upper threshold)
    θ(middle threshold)
  Output:
    T (directed edges matrix of FCM)
  Create_Edges (C', S, μ, θ, T)
    {
      create array T[i][j]
      for ∀ci ∈ C' do
        for i <- 1 to length[S] do
          for j <- 1 to length[S[i]] do
            if θ<sij• μ then
              tij <- 1
            else
              tij <- 0
            end if
          end for
        end for
      end for
      }
    }
  end.
```

In the algorithm pseudo code 4, the input *C* means the set of original concept nodes, the inputs δ, θ and μ are lower, middle and upper threshold, respectively, and the output *E* means the adjacent matrix of FCM. When the structure of FCM is built, we determine the weights of directed edges in FCM by using weights learning method in [23].

Algorithm pseudo code 4.

```
Function: Create adjacent matrix of FCM
Input:
  C (set of concept nodes)
  δ (lower threshold)
  μ (upper threshold)
  θ(middle threshold)
Output:
  E (adjacent matrix of FCM)
Create_Adjacent_Matrix (C, δ, μ, θ, E)
  {
    create arrays L[i][j], R[i][j], T[i][j] and E[i][j]
    repeat
      Initialize/Update_Matrix (C, L)
      Create_Nodes (L, δ, μ, R)
      Create_Edges (R, L, μ, θ, T)
    until all concept nodes are confirmed
    for i <- length[T] do
      for j <- length[T[i]] do
        determine weights by using learning method in [23]
        E[i][j] <- T[i][j]
      end for
    end for
   }
 end.
```

4 Network Security Situation Assessment

In this section, the method of network security situation assessment will be introduced based on the FCM. We choose the final SREs as the concept nodes to build the FCM model based the method that is proposed in the section 3. After building the FCM model of SRE, the status values of all concept nodes can be calculated by the following formula:

$$s_i(t) = f\left(s_i(t-1) + \sum_{i=1, j\neq i}^{n} s_j(t-1)w_{ji}\right) \tag{2}$$

where s_i denotes the status value of concept node. $s_i(t)$ denotes the status value of *i*-th concept node at time *t*, and $s_i(t-1)$ means the status value of *i*-th concept node at time *t*-1. w_{ji} is the adjacent matrix of FCM. *f* is an S-curve function:

$$f(x) = \frac{1}{1+e^{-cx}}$$

where *c* is constant 4. We classified the concept nodes into four types, i.e., vulnerability, service, host and system, then calculate the security situation value of each type using the follow equation:

$$Ev(t) = g\left(\sum_{i=1}^{n} w_i^a (s_1(t), \cdots, s_i(t), \cdots, s_n(t))\right) \tag{3}$$

where $Ev(t)$ denotes the security situation value in assigned type at time *t*, w_i^a denotes the weight of *i*-th concept nodes, $s_i(t)$ denotes the status value of *i*-th concept node in assigned type at time *t*, and $g(x)$ denotes the normalized function.

The NSCS can be calculated by follow equation:

$$E(t) = g\left(\sum_{i=1}^{n} w_i^b \times Ev_i(t)\right) \tag{4}$$

where $E(t)$ denotes the comprehensive security situation value at time *t*, w_i^b denotes the weight of *i*-th type, $Ev_i(t)$ denotes the security situation value of the *i*-th type at time *t*, and $g(x)$ denotes normalized function.

While the computing of cyber security situation is completed, we introduce cyber security state level table defined as in [24] to reflect quantitatively the cyber security situation. The security state level is divided into excellent, fine, middle, poor and danger with every level corresponding to a range, [0, 0.2], [0.2, 0.4], [0.4, 0.75], [0.75, 0.9] and [0.9, 1], respectively, and with the corresponding weight values are 0.06, 0.11, 0.21, 0.26 and 0.36, respectively. Then the value of NSCS is shown quantitatively in figures and tables.

5 Experimental Analysis

5.1 Introduction

This chapter records the analysis of experiment which used a data set named DARPA2000 Data Set from MIT Lincoln Laboratory [25]. DARPA2000 Data Set is an acknowledged data set in network security field, which includes comprehensive data and instructions documents. DARPA2000 Data Set has two DDos attack scenes, i.e., LLDOS1.0 and LLDOS2.0.2. We chose LLDOS1.0 as the target network system to analyze and assess the network security situation. The topological structure diagram of LLDOS1.0's attack scene is as shown in Fig. 3.

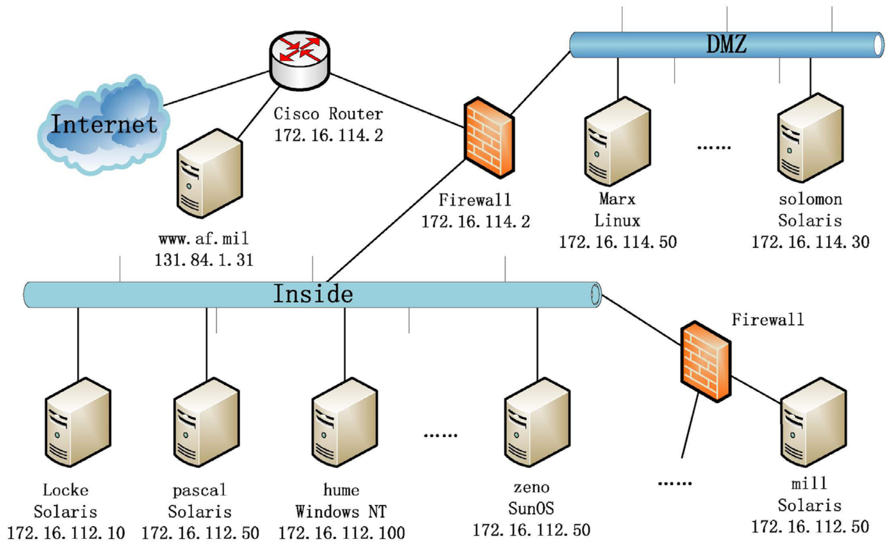

Fig. 3. Network topology graph

The vulnerability information in experiment system is as shown in Table 3. In the experiment, LLDOS1.0 is generated by a real multi-step attack which includes 5 steps. The first step is scanning the IP addresses form the network and attempting to find live hosts. The second step is checking all the live hosts and discovering the host which opened the *sadmind* service. The third step is launching a buffer overflow attack based on *Sadmind Buffer Overflow* Bug to the hosts (*Locke*, *Pascal* & *Mill*) which opened the *sadmind* service in order to get the permission of executing program on these hosts. The fourth step is installing *mstream* program to the hosts. The fifth step is launching a DDOS attack based on SYN FLOOD Bug to *www.af.mil* hosts by remotely operating the hosts which are installed *mstream* program.

Table 3. Vulnerability Information in our experiment

No.	Host	Vulnerability Information	Time
1	Mill, Locke, Pascal, Hume, Zeno	ICMP Incorrectly Configured	2000-03-07
2	Mill, Locke, Pascal	SunRPC Incorrectly Configured	2000-03-07
3	Mill, Locke, Pascal	Sadmind Buffer Overflow (CVE-1999-0977)	1999-12-10
4	Mill, Locke, Pascal	RCP Incorrectly Configured	2000-04-16
5	Mill	HINFO Query Incorrectly Configured	2000-04-16
6	www.af.mil	SYN Flood (CVE-1999-0116)	1996-09-19

5.2 FCM Structure Building

In the experiment, we extracted 14 original SREs as the original concept nodes according to the vulnerability list of experiment network system in Table 3. We constructed FCM model semi-automatically based on these 14 concept nodes by using the method proposed in section 3. Finally, we had 25 expanded concept nodes after iteration. The FCM model structure graph formed with all 39 concept nodes is described in Fig. 4. We used typical machine learning technology in [23] to determine the weights of directed edges. The weighted values $w(c_i, c_j)$ of adjacency matrix table in FCM model are showed in Fig. 4.

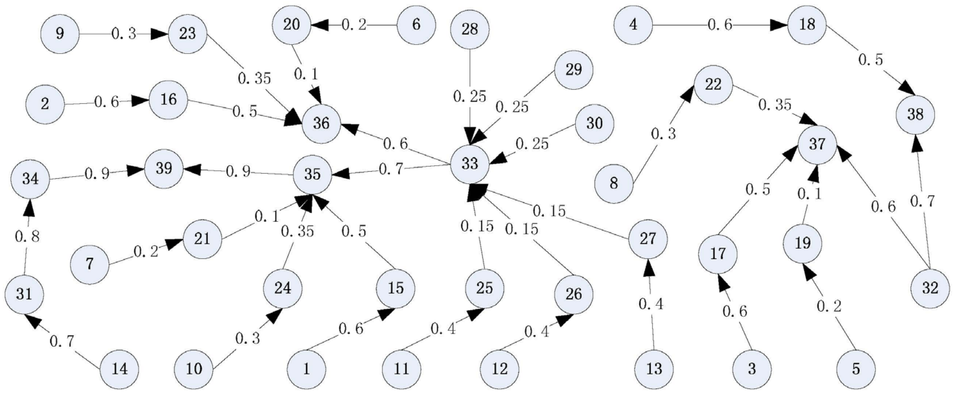

Fig. 4. FCM structure of experiment network SRE

5.3 Situation Assessment

Our experiment was carried out based on the 5 attack steps. The network security situation of the target network system was divided into 6 statuses which are the initial state and 5 attack states respectively. The initial state is the state before the attack happen, which is recorded as t_0. The 5 attack states was recorded as t_1, t_2, t_3, t_4 and t_5, respectively. With the attack progressing, we used our proposed methods to assess the state values of the SREs at t_1, t_2, t_3, t_4 and t_5. We divided all SREs concept nodes into four types: vulnerability, service, host and system, and then worked out the corresponding value of the comprehensive network security situation. Finally, we compared this result with ARMA methods, and gave the analysis results.

In order to assess the values of the comprehensive network security situation at t_1, t_2, t_3, t_4 and t_5, our process of operation in this experiment is given as follows.

Firstly, after building the SREs structure of FCM in section 5.2, we obtained all 39 concept nodes that means 39 SREs were created in experiment environment. Every SRE was the node of FCM structure, and the adjacency matrix of FCM was obtained by using our method in section 3. Then we calculated the all concept nodes values of SREs at different time according to equation (2), that can describe concept node situation at this time by considering the situation at last time and the interrelation between them. Until now, we could obtain all the network security situation values of SREs at t_1, t_2, t_3, t_4 and t_5, that are shown in Table 4.

Table 4. Information of SRE

No.	SRE	original/ extended	layer	weight in layer	state value at t_0	state value at t_1	state value at t_2	state value at t_3	state value at t_4	state value at t_5
1	Locke ICMP Incorrectly Configured	original	vulnerability	0.10	2.00	2.00	2.00	2.00	2.00	2.00
2	Pascal ICMP Incorrectly Configured	original	vulnerability	0.10	2.00	2.00	2.00	2.00	2.00	2.00
3	Mill ICMP Incorrectly Configured	original	vulnerability	0.07	1.00	1.00	1.00	1.00	1.00	1.00
…	…	…	…	…	…	…	…	…	…	…
14	SYN Flood existing in www.af.mil	original	System/ vulnerability	0.05/0.11	4.00	4.00	4.00	4.00	4.00	4.00
15	The risk of being malicious scanned in Locke	extended	service	0.06	0.00	1.20	2.40	3.60	4.80	6.00
…	…	…	…	…	…	…	…	…	…	…
25	RCP be malicious used in Mill	extended	service	0.02	0.00	0.80	1.60	2.40	3.20	4.00
26	RCP be malicious used in Pascal	extended	service	0.03	0.00	0.80	1.60	2.40	3.20	4.00
…	…	…	…	…	…	…	…	…	…	…
38	Security risk existing in Hume	extended	host	0.10	0.00	0.00	0.60	1.80	3.60	6.00
39	Security risk existing in www.af.mil	extended	system/ host	0.40/0.50	0.00	0.00	0.00	2.88	9.68	25.68

Secondly, all the SREs were classified into four types that is described in Fig. 1, and they are vulnerability, service, host and system, respectively. Every types of SREs have their own weights in different layers that were determined by human analyst. So, the security situation values of each SRE type at t_1, t_2, t_3, t_4 and t_5 can be calculated by using equation (3). And then, we used equation (4) to calculate the value of network security comprehensive situation (NSCS) at t_1, t_2, t_3, t_4 and t_5, that are shown in Table 5.

Table 5. Values of network security situation

	weight	t_0	t_1	t_2	t_3	t_4	t_5
vulnerability layer	0.13	0.16	0.16	0.16	0.16	0.16	0.16
service layer	0.18	0.00	0.02	0.07	0.12	0.22	0.35
host layer	0.23	0.00	0.00	0.08	0.52	1.45	3.36
system layer	0.46	0.03	0.06	0.16	0.48	1.19	2.60
non-normalized NSCS		0.035	0.052	0.125	0.383	0.941	2.052
normalized NSCS		0	0.001	0.045	0.173	0.449	1

Finally, all the values of NSCS at t_1, t_2, t_3, t_4 and t_5 were normalized to expressed quantitatively according to the network security state level table in [24]. The network security situation graph of each type and comprehensive network security situation graph of experiment network system are as shown in Fig. 5 and Fig. 6. Both figures reflect that our proposed method can correctly describe the change trend of network security situation.

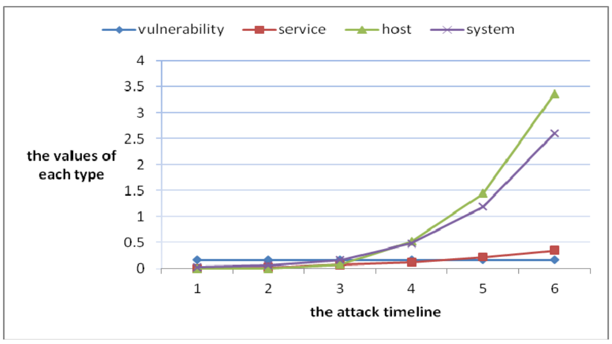

Fig. 5. Values of each type

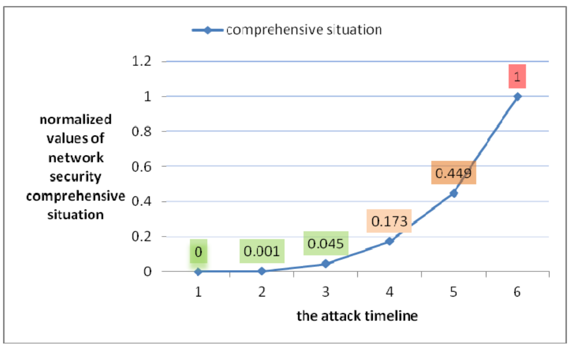

Fig. 6. Values of network security comprehensive situation

In our experiment, Fig. 5 showed that the values of service layer, host layer and system layer can accurately describe the security situation trend as attack time goes by. In Fig. 5, the graph quantitatively showed the values of NSCS as attack time goes by, and described different network security level at t_1, t_2, t_3, t_4 and t_5. According to the network security state level table in [24], we can found that the values of network security level marked with green color is lower security level, the one marked with brown color is middle security level, and the one marked with red color is high security level. It can quantitatively reflected the network security state levels in our experiment.

5.4 Comparative Analysis

During the experiment, we compared our method with a typical network security situation evaluation method named ARMA. ARMA model can analyze time sequence, mainly considers the impact of attack process on network security situation with time going by. Fig. 7 shows the difference between our proposed method and the ARMA model from time t_1 to t_5.

Then, we removed the Bug of *Sadmind Buffer Overflow* (CVE-1999-0977) of Locke in the following experiment, which caused the changing of the third and forth step of attack process, and after the change of security strategy, the results are shown in Fig. 8.

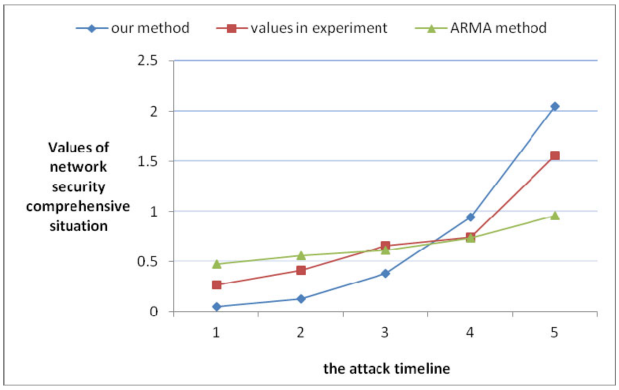

Fig. 7. The comparison with ARMA method

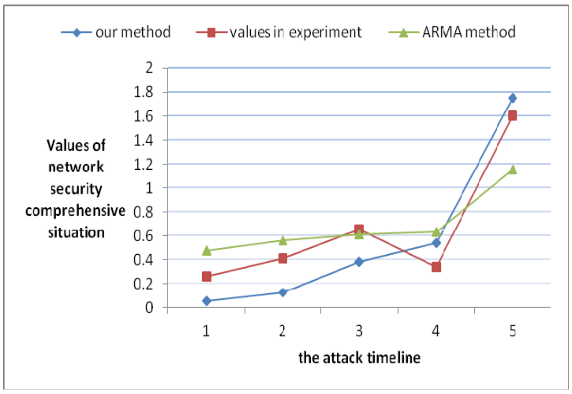

Fig. 8. The comparison with ARMA method after changing configuration

From the above figures, we can find that our method is more accurately than ARMA. The basic reason is that ARMA is a typical method for analyzing time sequence. It ignores the interrelationship of network security situation elements. Our proposed method can not only considers the impact of time sequence, but also takes into account the interrelationship among situation elements. Our method simulates SREs with FCM which is a dynamic deductive model that can describes the interrelationship among events and then evaluate the network security situation accurately.

5.5 SRE Tracing Back

The method we provided in this paper can not only evaluates the state of network security situation accurately, but also has the ability of tracing back the high-risk events. In the previous section, the final concept nodes of FCM is built semi-automatically according the original concept nodes, and all the values of concept nodes are calculated automatically.

In this section, we tried to find the outliers among the values of all concept nodes in FCM model to trace the abnormal concept nodes which represent the high-risk events. For example, we hashed the SRE values of network security situation risk at time t_1 and t_4 as shown in Fig. 9 and Fig. 10. We can see that events nodes with red circular mark are abnormal with high risk, so we confirm that SRE events nodes in red circles are high risk events at time t_1 and t_4.

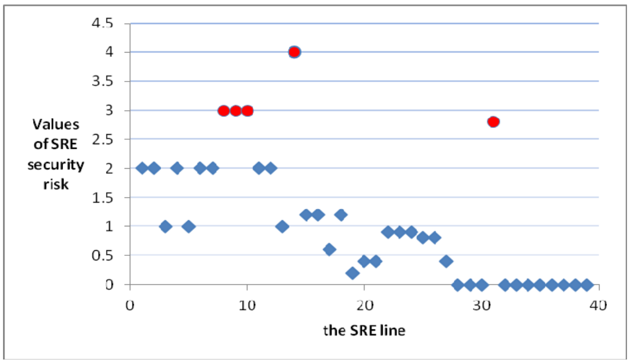

Fig. 9. Hash chart of SRE risk at time t_1

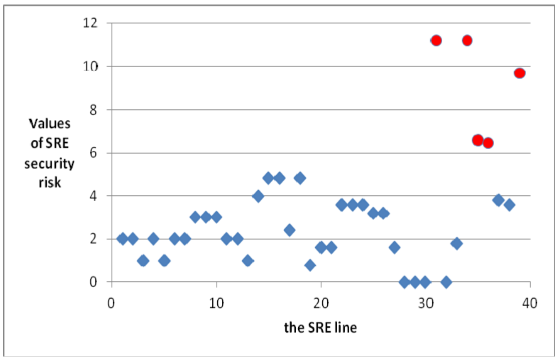

Fig. 10. Hash chart of SRE risk at time t_4

Because the content of structural description of SRE contains the item of direct causes, we can trace the precondition for the high risk events nodes. Furthermore, we can deeply trace the precondition for the precondition. The Fig. 11 shows an example for tracing deeply the precondition for the high risk events node, whose value is 9.68 at time t_4. In this instance, the node means *www.af.mil* has an high security risk, we can find the original reasons by deeply tracing. The original reasons for the high risk event are malicious use of *Rsh* and incorrect configuration of *RCP* in our experiment.

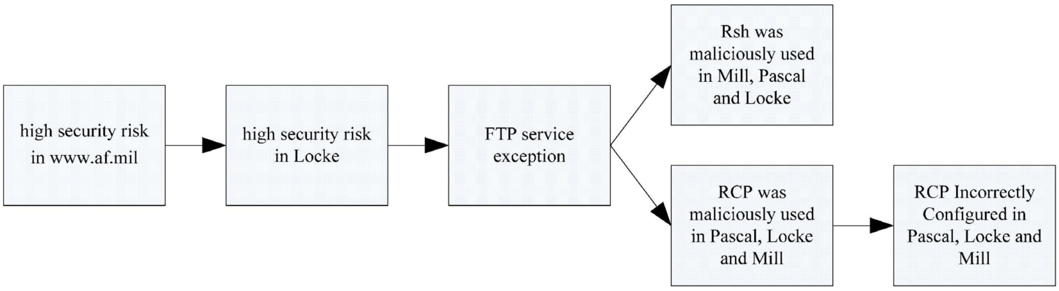

Fig. 11. The roadmap of tracing deeply

6 Conclusion

In this paper, the proposed method is to solve the two above problems. We proposed a new hierarchical assessment method based on FCM. First, we created a structured description of the original security events. Second, we semi-automatically generated the FCM structure according to the original structured security events. Third, we classified the concept nodes into four types, i.e., vulnerability, service, host and system. Fourth, we calculated the security situation values of each type and the value of network security comprehensive situation. Fifth, we assessed the network security comprehensive situation according to the network security state level table. Sixth, we found the high risk events and traced the precondition.

At last, we used DARPA2000 datasets to verify and analyze our new method by comparing other typical methods, and show how to trace back the high risk events. The results show that our method can not only reflect the trend of the network security situation accurately, but also has the security risk events backtracking capability.

When our method is used in a practical application, several problems should be considered. First, it is very important to choose more reasonable original SRE concept nodes according to the actual requirements in practical application. Second, it must be considered to find the appropriate threshold values in process of semi-automatically building FCM structure. Third, in order to effectively find the high-risk events in process of tracing back, it is necessary to combine with human analyst. In this paper, our experiment is based on the public DARPA2000 data set. Therefore, future research will include improving the model, and strengthening research on other practical applications to enhance the versatility of the model.

Acknowledgements

This work is supported in part by the National High Technology Research and Development Program of China under Grant 2015AA016009, the Key Projects in the National Science & Technology Pillar Program of China during the Twelfth Five-year Plan Period under Grant 2015BAK12B03-1 and the National Key R & D Program of China under Grant 2017YFB0802302 and 2017YFC0803702.

References

[1] M. Endsley, Situation awareness global assessment technique (SAGAT), in: Proc. the 88th National Aerospace and Electronics Conference, 1988.

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

[3] M. Dagdeviren, I. Yuksel, Developing a fuzzy analytic hierarchy process (AHP) model for behavior-based safety management, Information Sciences 178(6)(2008) 1717-1733.

[4] X.-Z. Chen, Q.-H. Zheng, X.-H. Guan, C.-G. Lin, Quantitative hierarchical threat evaluation model for network security, Journal of Software 17(4)(2006) 885-897.

[5] Y.-Y. Jia, H.-Y. Wu, D.-X. Jiang, A hierarchical framework of security situation assessment for information system, in: Proc. 2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2015.

[6] G. Jakobson, Mission cyber security situation assessment using impact dependency graphs, in: Proc. the 14th International Conference on Information Fusion (FUSION), 2011.

[7] A.A. Cain, D. Schuster, Applying measurement to complementary situation awareness, in: Proc. the 2016 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2016.

[8] K. Wang, H. Qiu, H.-P. Yang, D. Hou, Network security situation evaluation method based on attack intention recognition, in: Proc. 2015 4th International Conference on Computer Science and Network, 2015.

[9] X.-D. Cai, Network security threat situation evaluation based on fusion decision and spread analysis, International Journal of Security and ITS Applications 9(3)(2015) 383-388.

[10] Y. Wei, Y.-F. Lian, D.-G. Feng, A network security situational awareness model based on information fusion, Journal of Computer Research and Development 46(3)(2009) 353-362.

[11] Z.-Y. Qu, Y.-Y. Li, P. Li, A network security situation evaluation method based on D-S evidence theory, in: Proc. 2010 International Conference on Environmental Science and Information Application Technology (ESIAT), 2010.

[12] B.-Y. Zhang, A quantitative network situation assessment method based on stochastic model, Applied Mechanics and Materials 513-517(2014) 768-771.

[13] S. Lu, M.M. Kokar, A situation assessment framework for cyber security information relevance reasoning, in: Proc. the 18th International Conference on Information Fusion (FUSION), 2015.

[14] X.-Y. Li, H. Zhao, Network security situation assessment based on HMM-MPGA, in: Proc. 2016 2nd International Conference on Information Management (ICIM), 2016.

[15] F. Fang, X.-Y. Liang, J. Wang, X.-J. Tian, B. Zhang, J.-Y. Huang, Y. Sun, Network security situation evaluation method for distributed denial of service, in: Proc. 2012 International Conference on Instrumentation, Measurement, Computer, Communication and Control (IMCCC), 2012.

[16] J. Li, M. Cheng, J. Ni, F. Qian, Research on the aggregation model of network security situation awareness based on analytic hierarchy process, in: Proc. 2013 Fourth International Conference on Intelligent Systems Design and Engineering

Applications, 2013.

[17] M.A. Bode, S.A. Oluwadare, B.K. Alese, A.F. Thompson, Risk analysis in cyber situation awareness using Bayesian approach, in: Proc. 2015 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), 2015.

[18] I. Kotenko, E. Doynikova, Security evaluation for cyber situational awareness, in: Proc. 2014 IEEE Intl Conference on High Performance Computing and Communications, 2014.

[19] B. Lazzerini, L. Mkrtchyan, Analyzing risk impact factors using extended fuzzy cognitive maps, IEEE Systems Journal 5(2)(2011) 288-297.

[20] P. Szwed, P. Skrzyński, A new lightweight method for security risk assessment based on fuzzy cognitive maps, International Journal of Applied Mathematics and Computer Science 24(1)(2014) 213-225.

[21] P. Szwed, P. Skrzynski, P. Grodniewicz, Risk assessment for SWOP telemonitoring system based on fuzzy cognitive maps, in: A. Dziech, A. Czyżewski (Eds.), Multimedia Communications, Services and Security, Springer, Berlin, Heidelberg, 2013, pp. 233-247.

[22] B. Kosko, Fuzzy cognitive maps, International Journal of Man-Machine Studies 24(1996) 65-75.

[23] Y.-L. Zhang, X.-D. Liu, Weights learning of fuzzy cognitive maps, Journal of Chinese Computers Systems 34(5)(2013) 1147-1153.

[24] Z.-P. Wang, Research of network security situation evaluation based on index system, [dissertation] Changsha: College of Computer, National University of Defense Technology, 2010.

[25] DARPA2000 Data Sets, MIT Lincoln Lab. <http://www.ll.mit.edu/ideval/data/2000data.html>, 2000 (accessed 14.07.21).