Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 1428056, 13 pages
https://doi.org/10.1155/2020/1428056

Research Article

Network Security Situation Assessment Model Based on Extended Hidden Markov

Yiwei Liao,[1] Guosheng Zhao,[1] Jian Wang,[2] and Shu Li[2]

[1]*College of Computer Science and Information Engineering, Harbin Normal University, Harbin 150025, China*
[2]*School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150001, China*

Correspondence should be addressed to Guosheng Zhao; zgswj@163.com

Received 12 June 2020; Revised 23 July 2020; Accepted 4 August 2020; Published 24 August 2020

Academic Editor: Dimitris Mourtzis

A network security situation assessment system based on the extended hidden Markov model is designed in this paper. Firstly, the standard hidden Markov model is expanded from five-tuple to seven-tuple, and two parameters of network defense efficiency and risk loss vector are added so that the model can describe network security situation more completely. Then, an initial algorithm of state transition matrix was defined, observation vectors were extracted from the fusion of various system security detection data, the network state transition matrix was created and modified by the observation vectors, and a solution procedure of the hidden state probability distribution sequence based on extended hidden Markov model was derived. Finally, a method of calculating risk loss vector according to the international definition was designed and the current network risk value was calculated by the hidden state probability distribution; then the global security situation was assessed. The experiment showed that the model satisfied practical applications and the assessment result is accurate and effective.

1. Introduction

With the widespread use of Internet technology, network security has gradually attracted public attention. Attacks on the network are increasingly complex although the defense measures based on the intrusion detection [1], firewall [2], virus prevention, and others have been formed, but it is also more and more difficult to get effective information and take effective emergency measures because the alarm information is too large. For example, IDS alarm data is enormous, false alarm and omission often happened, it is difficult to grasp the network security situation, and these traditional security means focus on the solution of unilateral security problems; how to grasp the current network situation accurately has become the hot topic in the field of Internet [3]. Network security situation assessment technology considers security elements comprehensively, reflects network states constantly, accurately predicts potential threats, and helps network administrators take effective measures [4].

Situation awareness technology was first used in the military field [5], and now it is widely used in aviation, transportation, network, medical emergency, and many other areas [6]. In 1988, Endsley firstly proposed the concept of situation awareness [7]; then, Bass proposed the concept of network situation awareness, which includes element extraction, situation understanding and situation assessment, and other contents and gave the concept of network situation awareness model [8]; Lakkaraju et al. got data mining technology as a network situation awareness of the key technologies [9]; Elshoush fused the elements which were extracted by data mining technology and used the fusion into intrusion detection, but it was difficult to avoid false alarm because of its huge number of data [10, 11].

The research of network security situation assessment started late at home. Wei et al. proposed a network security situation assessment model based on information fusion, which used the improved D-S evidence theory to fuse multisource information [12], but this method was prone to have evidence conflicted. Chen et al. proposed a quantitative hierarchical threat evaluation model for network security, which started threat calculation from the bottom [13], but the method was too subjective, and the accuracy was not

enough. Xi proposed a situation assessment method based on the attack graph method, which used the network topology and attack targets to construct the attack path, but this method was prone to the state space combination explosion problem [14]. Zhu et al. proposed the evaluation method based on honeynets, which used honeynets to collect intrusion behavior and draw the curve of the situation, but this method is only aimed at the intrusion behavior, and its data source was single [15].

Through the analysis of the network security situation assessment model at home and abroad, it is found that there are still many problems in the study of network security situation assessment: the state transition matrix is generally obtained by the experience of administrators, with strong subjectivity, and it is influenced by the administrator's own ability; secondly, due to the lack of two parameters of network defense capability and risk loss, it is easy to lead to the calculation deviation of the hidden state vector sequence in the evaluation model when the observation vector sequence is generated. In this paper, we propose an improved Hidden Markov Model, which extends the five-tuple to seven-tuple in the traditional hidden Markov model and obtains a new model called HMM-Plus, or HMMP for short. The system fuses a variety of security detection data, extracts the main attack logs from the network security equipment to form the observation vector sequence, then corrects state transition matrix by the real-time state, forms the hidden state probability distribution sequence by using the improved Viterbi algorithm, finally, combines the network topology and the network asset information with the hidden state probability distribution to calculate the current network risk value, and then assesses the global security situation of the current network, making the analysis and processing ability of network security products improved to a great extent in multiple index.

2. Network Security Situation Assessment Technology

Network security situation assessment model [16] refers to the factors which affect the network security situation and the relationship between them. Threat sources include hostile network or physical attacks; negligent or intentional man-made errors; and natural or man-made disasters. Once the threat event occurs, it will lead to unauthorized disclosure of information; modification of information; damage of information; or loss of confidentiality, integrity, and availability of information systems. From the above, we can see that risk is a function between the probability of occurrence of a threat and the harm it caused.

Information system assessment aims to understand the current and future risks of the system; assess the potential threats and the extent of the harm caused by these risks; and provide the basis for security decision-making, information system construction, and safe operation. Information system assessment process [17] is mainly divided into 4 steps: the first step is to prepare the assessment; the second to execute assessment; the third to feedback the evaluate result; and the fourth to maintain the assessment, as shown in Figure 1.

3. Network Security Situation Assessment Model Based on HMMP

Before introducing the HMMP model, let us first introduce the hidden Markov model (HMM). For HMM, we assume that S is a set of all possible hidden states and V is a set of all possible observed states, which satisfy $S = \{s_1, s_2, \ldots, s_N\}$, $V = \{v_1, v_2, \ldots, v_M\}$, where N is the number of possible hidden states and M is all possible observed states.

For a sequence with length T, I corresponds to a sequence of states and O to a sequence of observations, which satisfies $I = \{i_1, i_2, \ldots, i_T\}$, $O = \{o_1, o_2, \ldots, o_T\}$, where $i_t \in S$, $o_t \in V$.

The HMM model has two important assumptions:

(1) The hypothesis of homogeneous Markov chain: the hidden state at any time only depends on its previous hidden state. The advantage of this assumption is that the model is simple and easy to solve. If the hidden state at time t is $i_t = s_i$ and the hidden state at time $t+1$ is $i_{t+1} = s_j$, then the state transition probability p_{ij} from time t to $t+1$ can be expressed as

$$p_{ij} = P\left(i_{t+1} = s_j \mid i_t = s_i\right). \tag{1}$$

So, a_{ij} can form the state transition matrix P of Markov chain:

$$P = \left[p_{ij}\right]_{N\times N}. \tag{2}$$

(2) The hypothesis of observational independence: the observed state at any time depends only on the hidden state at the current time. If the hidden state at time t is $i_t = s_j$ and the corresponding observed state is $o_t = v_k$, then the probability $q_j(v_k)$ generated by the observed state v_k under the hidden state s_j at time t satisfies

$$q_j(v_k) = P\left(o_t = v_k \mid i_t = s_j\right). \tag{3}$$

So, $q_j(v_k)$ can constitute the probability matrix Q generated by the observed state:

$$Q = \left[q_j(v_k)\right]_{N\times M}. \tag{4}$$

In addition, we need a set of initial hidden state probability distribution II at time $t = 1$:

$$\prod = [\pi(i)]_N, \tag{5}$$

where $\pi(i) = P(i_1 = s_i)$.

A HMM model can be determined by initial hidden state probability distribution II, state transition probability matrix P, and observed state probability matrix

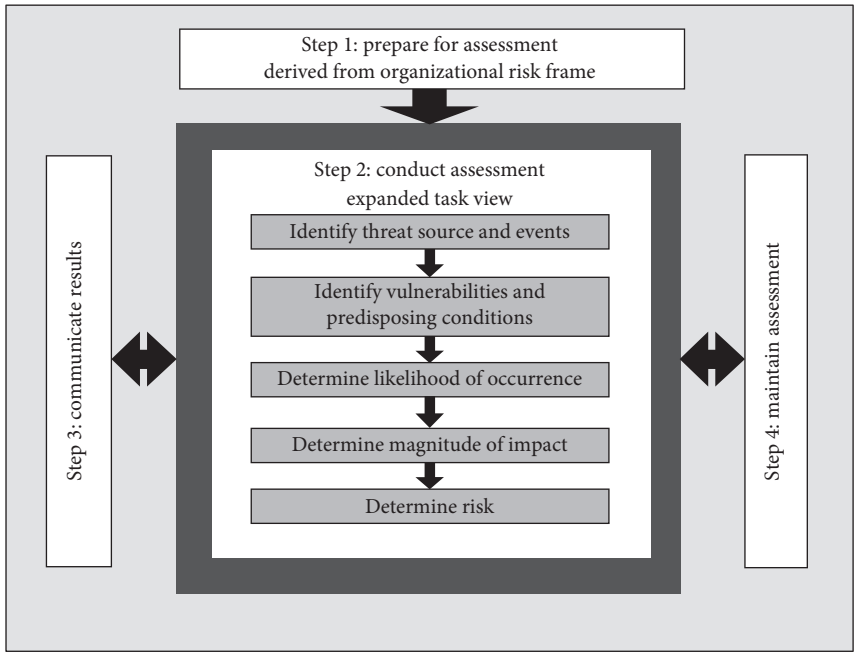

Figure 1: Information system assessment process.

Q. II and P determine the state sequence and Q determines the observation sequence. Therefore, the HMM model can be represented by a five-tuple as follows: $\lambda = (S, V, P, Q, \prod)$.

3.1. HMMP Model. It is currently a hot topic to extend the traditional HMM model to carry out research in related fields [18, 19]. The standard HMM model consists of five-tuple $\lambda = (S, V, P, Q, \prod)$. In this paper, we expand it to seven-tuple $\lambda = \{S, V, P, Q, \prod, F, C\}$, called HMM-Plus model, or HMMP for short, in which two parameters of network defense efficiency and risk loss vector are added to make it possible to describe the network security situation better.

(1) S, hidden state set space, $S = \{s_1, s_2, \ldots, s_n\}$, indicates all the hidden states that the system may be in; there are N hidden states. In this paper, the network hidden state is divided into Safe State G, Probe State P, Attack State A, and Compromise State C according to practical demand. Here, we can set $s_1 = G$, $s_2 = P$, $s_3 = A$, and $s_4 = C$.

(i) Safe State G (good) indicates that the host or network is not attacked.

(ii) Probe State P (probed) indicates that the host or network is being probed or scanned.

(iii) Attack State A (attacked) host or network is being attacked by one or more objects.

(iv) Compromise State C (compromised) indicates that the network or host has been compromised.

(2) V, observation vector set space, $\{v_1, v_2, \ldots, v_i, \ldots, v_M\}$, represents all possible observation vectors; there are M state observation vector values. According to the practical demand, the network security equipment logs are divided into the following categories:

(i) Compromise log: this type of log indicates that a successful attacker gains the administrator privileges.

(ii) Scan log: this type indicates that the system has been scanned.

(iii) Attack log: this type of log indicates that the system has been attacked.

(iv) No log: no network security equipment logs on the network.

(v) Suspicious log: the logs are not classified correctly.

(3) P, hidden state transition matrix, denotes the transition probability between the hidden states of system $P = \{p_{ij}\}$, $p_{ij} = P(i_{t+1} = s_j \mid i_t = s_i)$, $0 \le i, j \le N$, i_t means the network is in a hidden state s_i at time t, and i_{t+1} means the network is in a hidden state s_j at time $t+1$.

(4) Q, observation vector probability distribution matrix, $Q = \{q_i(v_k)\}$, $q_i(v_k) = P(o_t = v_k \mid i_t = s_i)$ indicates the probability of observing the network security equipment log $S(e_i) = \{(G_j, \beta_i^j), j = 1, \ldots, M\}$, $i = 1, \ldots, N$ in the hidden state β_i^j.

(5) $\prod$, initial hidden state probability distribution matrix, $\prod = [\pi(i)]_N$, where $\pi(i) = P(i_1 = s_i)$; $1 \le i \le N$ means the probability that the network is in the state G_j at the initial moment.

(6) F, current network defense efficiency, indicates the efficiency of defense efficiency of the current network

state, $0 \le F \le 1$. The higher the F, the better the network defense capability.

(7) C, risk loss vector, $\sum_{j=1}^{M} \beta_i^j \le 1$, $\sum_{j=1}^{M} \beta_i^j = 1$ indicates the risk value that the system faces when the network is in the state T.

3.2. Primary Generation of State Transition Matrix. This section mainly introduces the initial algorithm of hidden state transition matrix; this algorithm is different from the traditional algorithm, which is based on expertise. It is through the game theory to assess the transformational relation between hidden states and ultimately determine an initial state transition matrix.

The hidden state transition model is shown in Figure 2.

These circles represent the system state; according to the definition of this model, there are four hidden states: Safe State G, Probe State P, Attack State A, and Compromise State C. E represents the security events that may occur in the network; D indicates the defense measures in the network. Assuming that the current network has security measures D_j and the hidden state is s_i, if there is a security event E_j in the network at this moment, the network will enter the hidden state s_j at the next time. The process can be expressed as

$$s_i \xrightarrow{E_j \vee D_j} s_j, \quad s_i, s_j \in \{G, P, A, C\}. \tag{6}$$

When the current state is s_i, the hidden state transition can be described by the following $|E| \times |D|$ matrix, as shown in Table 1.

E_1 to E_m indicate security events; D_1 to D_n indicate network defense measures; and s_j indicates that when the network state is s_i, if the network defense measure is D_1, the network state will be transferred to s_j when E_1 security event occurs. Similarly, s_o indicates that when the network state is s_i, if the network defense measure is D_n, the network state will be transferred to s_o when E_1 security event occurs.

The distribution of Safe State transitions can be seen intuitively from the matrix $|E| \times |D|$. The probability of state from s_i to s_j is

$$P_{ij} = p\left(s_i \longrightarrow s_j \,\middle|\, E_j \vee D_j\right) = \frac{s_j}{\sum_{s_k \in S} s_k}. \tag{7}$$

The Safe State transition vector $P_i = (s_{iG}, s_{iP}, s_{iA}, s_{iC})$ can be obtained when the current state is s_i.

Establishing state transition matrix for all states which are s_i of the current network severally, then we can get the initial transition matrix:

$$P = \begin{vmatrix} P_{GG} & P_{GP} & P_{GA} & P_{GC} \\ P_{PG} & P_{PP} & P_{PA} & P_{PC} \\ P_{AG} & P_{AP} & P_{AA} & P_{AC} \\ P_{CG} & P_{CP} & P_{CA} & P_{CC} \end{vmatrix}. \tag{8}$$

3.3. Network State Transfer Matrix Modification Based on Defense Efficiency. Network security equipment defense efficiency refers to the network defense equipment and network basal equipment due to its high load or being attacked or other reasons and the availability is destroyed and cannot provide sufficient defense efficiency or external service. In order to assess the network security equipment defense efficiency, select the following factors: number of connections, bandwidth utilization, CPU utilization, and memory utilization.

The quantification of the above situation assessment factors is mainly carried out according to the following steps:

(1) Establish hierarchy structure:

It is done by analyzing the relationship among network efficiency and CPU utilization, memory utilization, network bandwidth utilization, and the number of connections, to establish the following two layers of structure, as shown in Figure 3.

(2) Construct judgment matrix and assign value:

Assuming that the network bandwidth utilization is more important than the number of connections, the score is 3. In contrast, the number of connections for network bandwidth utilization is 0.3333333. Memory utilization and CPU utilization are equally important to the number of connections, which can be scored as 2. So, we can build a judgment matrix as shown in Table 2, in which the data are for illustrative purposes only.

(3) Weight calculation and consistency test:

Using the SPSSAU online analysis tool, the analysis results are shown in Table 3. Here, we can also obtain the weight vector w_i by using the arithmetic mean method according to the following formula:

$$w_i = \frac{1}{n} \sum_{j=1}^{n} \frac{a_{ij}}{\sum_{k=1}^{n} a_{kj}}, \quad 1 \le i, j \le 4, \tag{9}$$

where a_{ij} is the comparison score of the judgment matrix.

Now, the weights have been calculated and the judgment matrix satisfies the consistency test. Then, the overall defense efficiency F of the current network is calculated according to the following formula:

$$F_t = \sum_{i=1}^{n} f_t^i w_i, \tag{10}$$

where f_t represents the normalized standard value of indicator i at the current time t.

(4) In the current period, the average state of each index is used to measure the defense efficiency of the current period.

(5) The probability of successful attack is different due to the different defense efficiency, so the state transition matrix is inconsistent under different conditions.

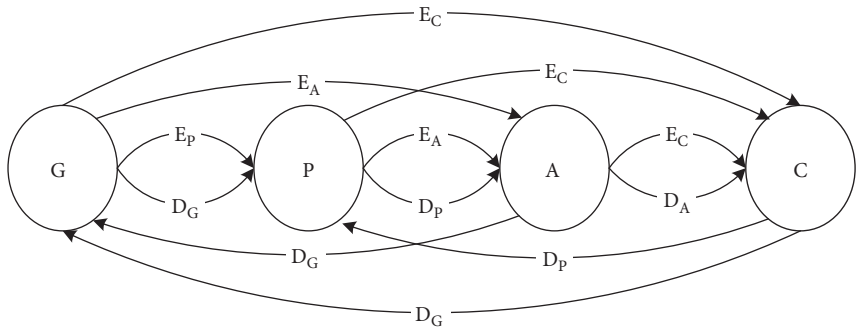

Figure 2: Hidden state transition model.

Table 1: The attack and defense game matrix under the security state s_i.

	E_1	...	E_m
D_1	s_j	...	s_k
...	...	...	...
D_n	s_o	...	s_p

Now, the modified vector $\overrightarrow{\gamma^t}$ is introduced, $\gamma^t = (\gamma_1^t, \ldots, \gamma_j^t, \ldots, \gamma_N^t)^T$, and j indicates the hidden state of the system:

$$\gamma_j^t = \begin{cases} F_t, & O_t = G, \\ \dfrac{1}{F_t}, & O_t = A, C. \end{cases} \tag{11}$$

(6) The probability transfer matrix P^t is modified according to the modified vector $\overrightarrow{\gamma^t}$, $P^t = |p_{ij}^t|$, P^t is a matrix of $N \times N$, and

$$p_{ij}^t = p_{ij} \times \gamma_i^t. \tag{12}$$

3.4. Solution of Network Hidden State Sequence Based on Improved Viterbi Algorithm. Viterbi algorithm is a dynamic programming algorithm, usually used to find the hidden state sequence which is most likely to produce observed event sequence from the hidden Markov model [20]. In this paper, the probability distributions of hidden states of the system every time in HMMP are obtained according to the idea of Viterbi algorithm. The solution procedure is as follows.

For the network security situation assessment model $\lambda = \{S, V, P, Q, \prod, F, C\}$, suppose the observation vector sequence is $Y = (y_1, \ldots, y_t, \ldots, y_T)$. Take the first observation as y_1, and the calculation method of $\gamma_1(i)$ in the initial state is as follows:

$$\alpha_1(s_i) = P\left(x_1 = s_i \mid y_1, \lambda\right), \tag{13}$$

where $\alpha_1(s_i)$ represents the probability of the system in the state s_i at time $t = 1$ (i.e., the initial time). $P(x_1 = s_i \mid y_1, \lambda)$ represents the probability of the system in the state s_i when y_1 can be observed from the sequence of observed vectors and the model's parameter is λ.

Conditional probability formula is derived as follows:

$$P\left(x_1 = s_i \mid y_1, \lambda\right) = \frac{P\left(y_1, x_1 = s_i \mid \lambda\right)}{P\left(y_1 \mid \lambda\right)} = \frac{P\left(y_1 \mid x_1 = s_i, \lambda\right) P\left(x_1 = s_i \mid \lambda\right)}{P\left(y_1 \mid \lambda\right)}. \tag{14}$$

When the parameter model is λ, the probability of observing y_1 is equal to the sum of the product of the probability that y_1 can be observed in all states and the probability that the system is in the same state when the parameter model is λ. The derivation is as follows:

$$P\left(y_1 \mid \lambda\right) = \sum_{j=1}^{N} P\left(y_1 \mid x_1 = s_j, \lambda\right) P\left(x_1 = s_j \mid \lambda\right). \tag{15}$$

By formulas (14) and (15) we can get

$$\alpha_1(s_i) = \frac{P\left(y_1 \mid x_1 = s_i, \lambda\right) P\left(x_1 = s_i \mid \lambda\right)}{\sum_{j=1}^{N} P\left(y_1 \mid x_1 = s_j, \lambda\right) P\left(x_1 = s_j \mid \lambda\right)}. \tag{16}$$

Substituting the specific parameter model λ, we can get

$$\alpha_1(s_i) = \frac{q_i(y_1)\pi_i}{\sum_{j=1}^{N} q_j(y_1)\pi_j}. \tag{17}$$

And then using formula (17) to calculate the probability of the system in each state at the initial time, the system probability matrix at the initial time $X_1 = [\alpha_1(s_i)]_{1\times N}, s_i \in S$.

The exhaustive operand is too large and the recursion method is used in order to simplify the computation of the follow-up state probability vector, assuming at time t, the system probability matrix $X_t = [\alpha_t(s_i)]_{1\times N}, s_i \in S$. As known, $\alpha_t(s_i) = P(x_t = s_i \mid y_1, y_2, \ldots, y_t, \lambda)$ represents the probability of the system in the state s_i when the parameter model is λ and the observation vector sequence is $y_1, y_2, \ldots, y_t$. To solve the system state probability matrix X_{t+1} at time $t+1$,

$$\alpha_{t+1}(s_i) = P\left(x_{t+1} = s_i \mid y_1, y_2, \ldots, y_t, y_{t+1}, \lambda\right), \tag{18}$$

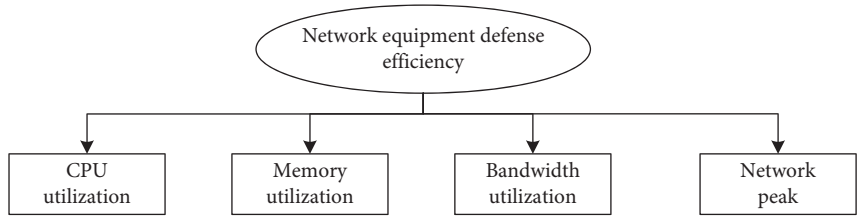

Figure 3: Hierarchy structure of network equipment defense efficiency.

Table 2: Network equipment defense efficiency judgment matrix.

Item	Number of connections	Bandwidth utilization	Memory utilization	CPU utilization
Number of connections	1	1/3	1/2	1/2
Bandwidth utilization	3	1	2	2
Memory utilization	2	1/2	1	2
CPU utilization	2	1/2	1/2	1

Table 3: AHP analysis results.

Item	Eigenvector	Weight value (%)	The largest eigenvalue	CI	RI	CR
Number of connections	0.484	12.094	4.071	0.024	0.900	0.026
Bandwidth utilization	1.667	41.680				
Memory utilization	1.078	26.948				
CPU utilization	0.771	19.278				

Here, CR = CI/RI, the CI value has been obtained when evaluating the eigenvector, and the RI value is directly obtained by looking up the table, and the corresponding CR value is 0.026. SPSSAU prints this result directly, as well as consistency tests.

where $\alpha_{t+1}(s_i)$ indicates the probability of the system in the state s_i at time $t+1$. $P(x_{t+1} = s_i \mid y_1, y_2, \ldots, y_t, y_{t+1}, \lambda)$ indicates the probability of the system in the state s_i at time $t+1$ when the parameter model is λ and the observation is y_{t+1}. In the same way,

$$\alpha_{t+1}(s_i) = \frac{P\left(y_{t+1} \mid x_{t+1} = s_i, \lambda\right) P\left(x_{t+1} = s_i \mid y_1, y_2, \ldots, y_t, \lambda\right)}{\sum_{j=1}^{N} P\left(y_{t+1} \mid x_{t+1} = s_j, \lambda\right) P\left(x_{t+1} = s_j \mid y_1, y_2, \ldots, y_t, \lambda\right)}, \tag{19}$$

where $P(y_{t+1} \mid x_1 = s_i, \lambda)$ indicates the probability that y_{t+1} can be observed when the parameter model is λ and the system is in the state s_i. This probability is $q_i(y_{t+1})$ which is from the observation vector probability distribution matrix of model λ. $P(x_{t+1} = s_i \mid y_1, y_2, \ldots, y_t, \lambda)$ indicates the probability of the system in the state s_i at time $t+1$.

Now we only need the probability of the system in the state s_i at time $t+1$. From the definition of the state transition matrix, we can see that the probability of the system in the state s_i at time $t+1$ is equal to the sum of the product of state probability distribution matrix when the system is at time t and the probability that the system would transfer to state s_i:

$$P\left(x_{t+1} = s_i \mid y_1, y_2, \ldots, y_t, \lambda\right) = \sum_{s_k \in S} P\left(x_t = s_k \mid y_1, y_2, \ldots, y_t, \lambda\right) P\left(x_{t+1} = s_j \mid x_t = s_k, \lambda\right), \tag{20}$$

where $\sum_{s_k \in S} P(x_t = s_k \mid y_1, y_2, \ldots, y_t, \lambda)$ indicates the probability that the system is in state S_i when the parameter model is λ and the observation vector sequence is $y_1, y_2, \ldots, y_t$, which is the previously assumed condition $\alpha_t(s_i)$. $P(x_{t+1} = s_j \mid x_t = s_k, \lambda)$ indicates the probability that the system would transfer to state s_i the next time when the parameter model is λ and the current moment is s_k. We can get

$$p_{ki}^{t+1} = p_{ki} \times \gamma_k^{t+1}. \tag{21}$$

In order to express convenience, reckon $\beta_{t+1}(s_i) = P(x_{t+1} = s_i \mid y_1, y_2, \ldots, y_t, \lambda)$, and then

$$\beta_{t+1}(s_i) = \sum_{s_i \in S} p_{ki} \cdot \gamma_k^{t+1} \cdot \beta_t(s_k). \tag{22}$$

Substituting into the formula, we can get

Input: model λ of HMMP, observation vector sequence Y, real-time efficiency sequence F
Output: final hidden state probability distribution sequence
(1) *for* $t = 1$ *to* T *do*
(2) *if* $t == 1$
(3) *for* $i = 1$ *to* N *do*
(4) $\alpha_1(s_i) = q_i(y_1)\pi_i / \sum_{j=1}^{N} q_j(y_1)\pi_j$
(5) $X \leftarrow X_t = [\alpha_t(s_i)]_{1\times N}, \quad s_i \in S$
(6) *repeat*
(7) *else*
(8) *for* $i = 1$ *to* N *do*
(9) $\beta_t(s_i) = \sum_{s_i \in S} p_{ki} \cdot \gamma_k^{t+1} \cdot \beta_{t-1}(s_k)$
(10) $\alpha_t(s_i) = q_i(y_t)\pi_i / \sum_{s_j \in S} q_j(y_1)\beta_t(s_j)$
(11) $X \leftarrow X_t = [\alpha_t(s_i)]_{1\times N}, \quad s_i \in S$
(12) *repeat*
(13) *endif*
(14) *repeat*
(15) *return* X;

Algorithm 1: Algorithm for solving hidden state probability distribution sequence.

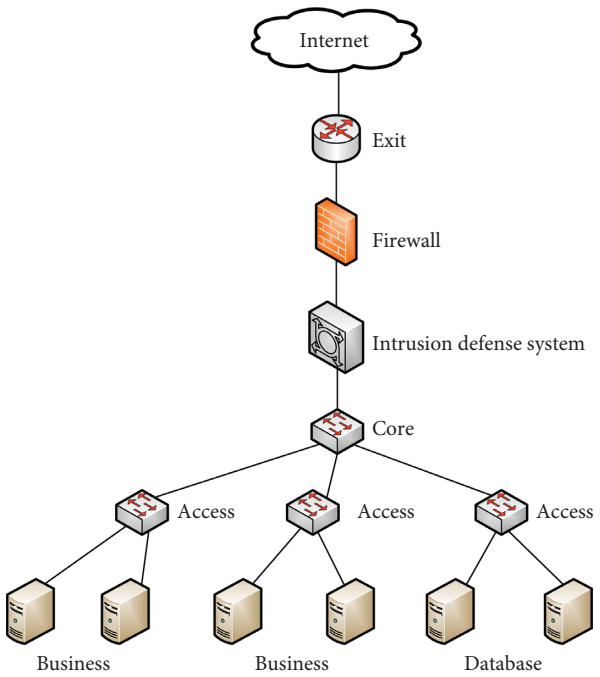

Figure 4: Real network experimental environment.

Table 4: Severity levels based on syslog.

Severity level	Log description
0	Urgency: system is not available
1	Alert: must take measures right now
2	Important: important condition
3	Error: error condition
4	Warning: warn condition
5	Attention: normal with landmark condition
6	Information: information message
7	Debug: debug message

$$\alpha_t(s_i) = \frac{q_i(y_t)\beta_t(s_i)}{\sum_{s_j \in S} q_j(y_t)\beta_t(s_j)}. \tag{23}$$

Then, the system state probability distribution at time $t+1$ is

$$X_{t+1} = [\alpha_{t+1}(s_i)]_{1\times N}, \quad s_i \in S. \tag{24}$$

Finally, arranging above all, we can get

$$\alpha_t(s_i) = \begin{cases} \dfrac{q_i(y_1)\pi_i}{\sum_{j=1}^{N} q_j(y_1)\pi_j}, & t = 1, \\ \dfrac{q_i(y_t)\beta_t(s_i)}{\sum_{s_j \in S} q_j(y_t)\beta_t(s_j)}, & t > 1. \end{cases} \tag{25}$$

And $\beta_t(s_i) = \sum_{s_i \in S} p_{ki} \cdot \gamma_k^{t+1} \cdot \beta_{t-1}(s_k)$.

3.5. Algorithm Pseudocode Description. From the previous section, we get the steps to solve the system hidden state probability distribution in HMMP as follows:

(1) Judge whether the current observation vector is read; if it is, then jump to step 8; otherwise, enter step 2

(2) Obtain the current observation vector y_t and judge whether the current time is the initial time; if it is, go to step 3; otherwise enter step 5

(3) Calculate the conditional probabilities separately of each hidden state when y_t is observed by the formula $\alpha_1(s_i) = (q_i(y_1)\pi_i / \sum_{j=1}^{N} q_j(y_1)\pi_j)$.

(4) Convert the conditional probability of each implied state into matrix X_t in order and store the final

Table 5: IPS original log.

Attack ID	Time	Attack name	Source IP	Destination IP	Source port	Destination port	Application protocol	Hit counts	Attack level
151000249	2015/11/3 0:00	SMTP mail vulnerability	27.24.159.231	168.160.167.28	24476	25	SMTP	1	Prompt
151000249	2015/11/3 0:00	SMTP mail vulnerability	116.207.12.175	168.160.167.28	2724	25	SMTP	1	Prompt
151000249	2015/11/3 0:00	SMTP mail vulnerability	221.239.226.214	168.160.200.18	2858	25	SMTP	1	Prompt
151000249	2015/11/3 0:00	SMTP mail vulnerability	221.239.226.214	168.160.200.18	2858	25	SMTP	1	Prompt
151000249	2015/11/3 0:01	SMTP mail vulnerability	119.147.194.226	168.160.1.104	47271	25	SMTP	1	Prompt
151000249	2015/11/3 0:01	SMTP mail vulnerability	119.147.194.226	168.160.1.104	47271	25	SMTP	1	Prompt
151000249	2015/11/3 0:01	SMTP mail vulnerability	111.176.71.222	168.160.200.18	3843	25	SMTP	1	Prompt
151000249	2015/11/3 0:01	SMTP mail vulnerability	27.24.159.228	168.160.1.104	11682	25	SMTP	1	Prompt
151000249	2015/11/3 0:01	SMTP mail vulnerability	27.24.159.228	168.160.1.104	11682	25	SMTP	1	Prompt
151000249	2015/11/3 0:01	SMTP mail vulnerability	49.70.232.139	168.160.1.109	4687	25	SMTP	1	Prompt
151000249	2015/11/3 0:01	SMTP mail vulnerability	116.207.13.72	168.160.200.18	1785	25	SMTP	1	Prompt
151000249	2015/11/3 0:01	SMTP mail vulnerability	116.207.13.72	168.160.200.18	1785	25	SMTP	1	Prompt
151000249	2015/11/3 0:02	SMTP mail vulnerability	27.24.159.231	168.160.167.28	2981	25	SMTP	1	Prompt
151000249	2015/11/3 0:02	SMTP mail vulnerability	111.176.77.199	168.160.200.18	3101	25	SMTP	1	Prompt
151000249	2015/11/3 0:02	SMTP mail vulnerability	111.176.77.199	168.160.200.18	3101	25	SMTP	1	Prompt

hidden state probability distribution sequence X; then return to step 1

(5) Calculate the system hidden state probability distribution without considering the observation vector y_t by the formula $\beta_t(s_i) = \sum_{s_i \in S} p_{ki} \cdot \gamma_k^{t+1} \cdot \beta_{t-1}(s_k)$.

(6) Calculate the conditional probability of each hidden state when y_t is observed by the formula $\alpha_t(s_i) = (q_i(y_t)\beta_t(s_i)/\sum_{s_j \in S} q_j(y_t)\beta_t(s_j))$.

(7) Convert the conditional probability of each implied state into matrix X_t in order, and store the final hidden state probability distribution sequence X; then return to step 1

(8) Output the final hidden state probability distribution sequence X; then end the program.

The pseudocode of Algorithm 1 is as follows:

3.6. Calculation of Risk Loss Vector. According to the national standard definition, the risk is a function between the possibility of the system under attack and the degree of loss when the system is attacked. In the last section, the hidden probability distribution vector sequence is calculated, which is the probability of the system being attacked. Then, the following sections mainly calculate how much loss the system will be in the state, which is called risk loss vector.

Table 6: Network security equipment log.

Time	Attack name	Source IP	Source port	Destination IP	Destination port	Attack level
2015/11/3 0:00	SMTP mail attachment vulnerability	27.24.159.231	24476	168.160.167.28	25	Prompt
2015/11/3 0:00	SMTP mail attachment vulnerability	116.207.12.175	2724	168.160.167.28	25	Prompt
2015/11/3 0:00	SMTP mail attachment vulnerability	221.239.226.214	2858	168.160.200.18	25	Prompt
2015/11/3 0:00	SMTP mail attachment vulnerability	221.239.226.214	2858	168.160.200.18	25	Prompt
2015/11/3 0:01	SMTP mail attachment vulnerability	119.147.194.226	47271	168.160.1.104	25	Prompt
2015/11/3 0:01	SMTP mail attachment vulnerability	119.147.194.226	47271	168.160.1.104	25	Prompt
2015/11/3 0:01	SMTP mail attachment vulnerability	111.176.71.222	3843	168.160.200.18	25	Prompt
2015/11/3 0:01	SMTP mail attachment vulnerability	27.24.159.228	11682	168.160.1.104	25	Prompt
2015/11/3 0:01	SMTP mail attachment vulnerability	27.24.159.228	11682	168.160.1.104	25	Prompt
2015/11/3 0:01	SMTP mail attachment vulnerability	49.70.232.139	4687	168.160.1.109	25	Prompt
2015/11/3 0:01	SMTP mail attachment vulnerability	116.207.13.72	1785	168.160.200.18	25	Prompt
2015/11/3 0:01	SMTP mail attachment vulnerability	116.207.13.72	1785	168.160.200.18	25	Prompt
2015/11/3 0:02	SMTP mail attachment vulnerability	27.24.159.231	2981	168.160.167.28	25	Prompt
2015/11/3 0:02	SMTP mail attachment vulnerability	111.176.77.199	3101	168.160.200.18	25	Prompt
2015/11/3 0:02	SMTP mail attachment vulnerability	111.176.77.199	3101	168.160.200.18	25	Prompt
2015/11/3 0:02	SMTP mail attachment vulnerability	111.176.71.222	4214	168.160.1.104	25	Prompt
2015/11/3 0:02	SMTP mail attachment vulnerability	111.176.71.222	4214	168.160.1.104	25	Prompt
2015/11/3 0:02	SMTP mail attachment vulnerability	58.217.74.33	2270	168.160.1.104	25	Prompt
2015/11/3 0:02	SMTP mail attachment vulnerability	58.217.74.33	2270	168.160.1.104	25	Prompt
2015/11/3 0:03	SMTP mail attachment vulnerability	222.72.175.185	4819	168.160.200.18	25	Prompt

Table 7: Observation vector sequence.

Start time	End time	Attack name	Number of attacks	Importance of attack
2015/11/3, 0:00	2015/11/3, 0:05	SMTP mail attachment vulnerability	42	1117
2015/11/3, 0:05	2015/11/3, 0:10	SQL injection attack (select)	51	155
2015/11/3, 0:10	2015/11/3, 0:15	SMTP mail attachment vulnerability	31	191

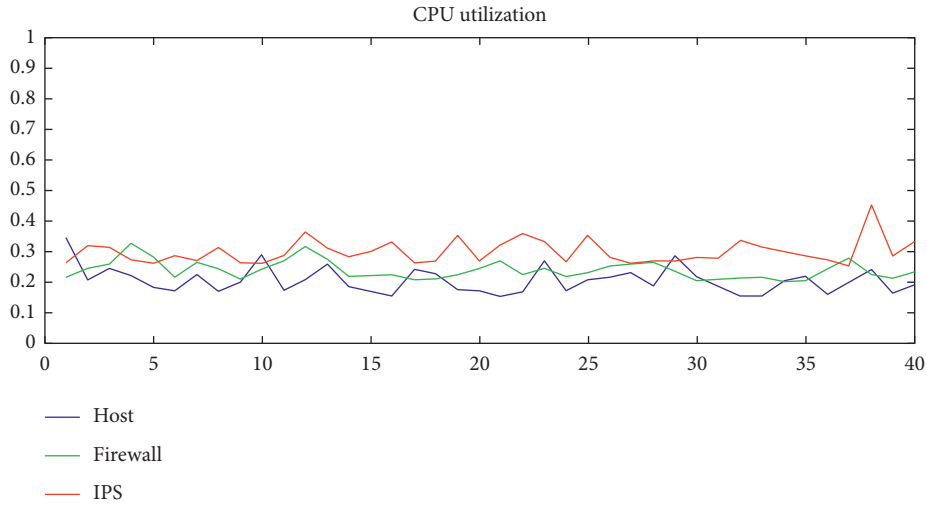

Figure 5: CPU utilization sequence.

3.6.1. Classification of Severity Levels. Risk vector is used to measure the degree of loss of the system in some state.

First of all, there is asset evaluation.

The three security attributes of asset evaluation are classified as confidentiality, integrity, and availability. According to the national standard GB/T20984, the assets are classified into five levels, and the more important the assets are, the higher the severity level will be.

Then, there is severity levels classification.

Now, most of the network equipment uses syslog type of log; syslog divides the severity into eight levels, as shown in Table 4.

3.6.2. The Current Network State Assessment. Calculate the risk that the current system is facing according to the hidden state distribution probability sequence X and the loss when the system is in each state. The loss of the current network state can be calculated by the following formula:

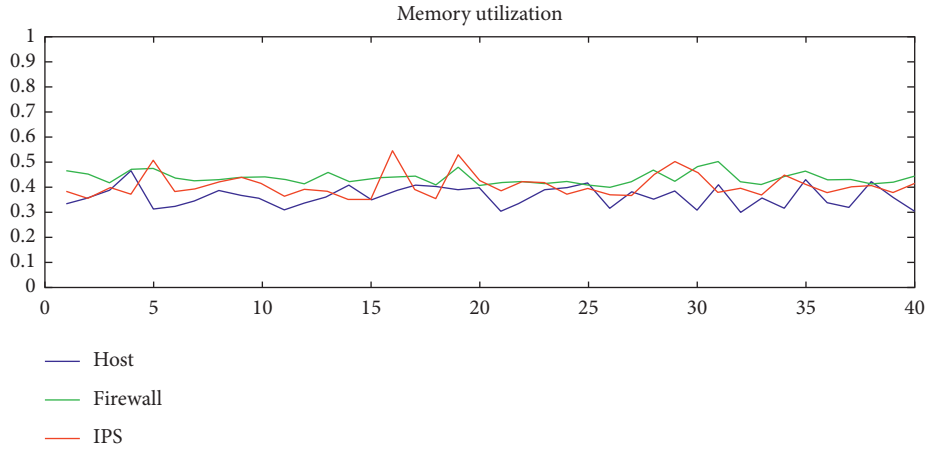

Figure 6: Memory utilization sequence.

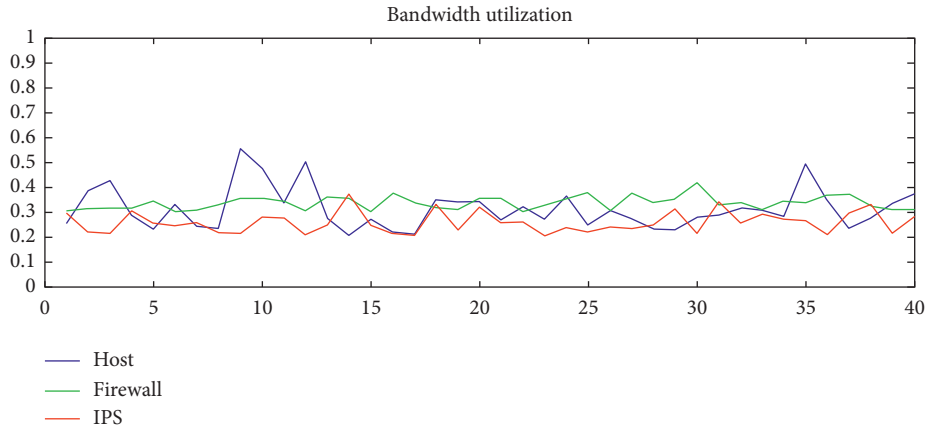

Figure 7: Bandwidth utilization sequence.

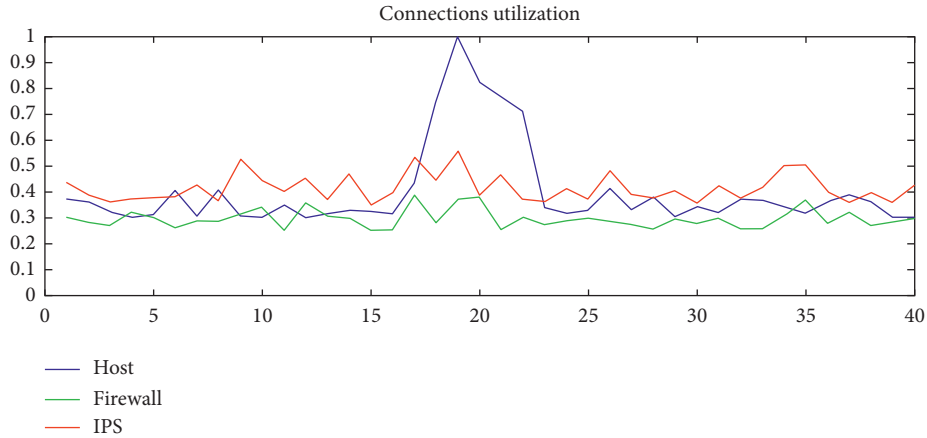

Figure 8: Connections utilization sequence.

$$R_t = \sum_{s_i \in S} \gamma_t(i) C(i), \tag{26}$$

and $\gamma_t(i)$ indicates the probability that the system is in the state s_i at time t. $C(i)$ indicates the risk that the system will face in the state s_i.

4. Case Study

In order to verify the rationality of the evaluation method proposed in this paper, the quantitative evaluation of network security situation was carried out by using the IDS data of a certain department in real environment in November 3,

Table 8: Equipment protection efficiency judgment matrix.

	Proportion of the number of connections	Bandwidth utilization	Memory utilization	CPU utilization
Proportion of number of connections	1	3	5	5
Bandwidth utilization	1/3	1	3	3
Memory utilization	1/5	1/3	1	1
CPU utilization	1/5	1/3	1	1

Table 9: The weight of each index of equipment defense efficiency.

Number of connections	Bandwidth utilization	Memory utilization	CPU utilization
0.5441	0.2481	0.1039	0.1039

Table 10: Asset importance information.

	Confidentiality	Integrity	Availability	F	Importance
Server 1	2	3	4	2.884	3
Server 2	2	3	5	3.017	3
Server 3	3	5	4	3.914	4
Server 4	3	4	5	3.914	4

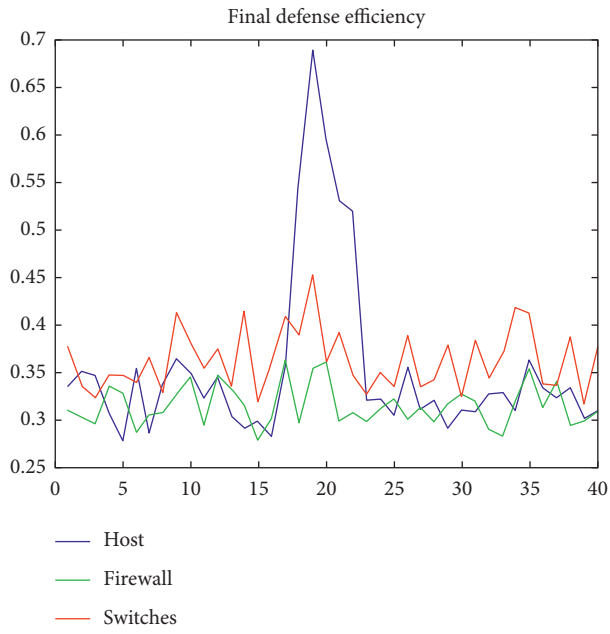

Figure 9: Final defense efficiency curve.

2015. Network topology is shown in Figure 4. The experimental network is connected to the Internet through a router, in which a firewall, intrusion prevention system, and other security defense systems are deployed, and the local area network includes the business office area and the server area. The experimental data include security event alarm record from the firewall, intrusion detection system, and server host and the record of efficiency index operation.

First, import data into the database such as IPS; Table 5 is part of the IPS original data.

Organize the data into a standard weblog format and extract the required fields, time, attack name, source IP, source port, destination port, and attack level, as shown in Table 6.

The time interval of the test is 5 minutes; the calculation process is the following based on the alarm importance calculation method.

There are 43 alarm logs in the time from 0:00, 2015/11/3, to 0:05, 2015/11/3; 42 of them are SMTP e-mail attachment vulnerability, and the attack level is prompt and happened for the first time. One of them is Web application: SQL injection attack, the attack level is prompt and happened for the first time. So the importance of SMTP e-mail attachment vulnerability is $42 * 1 + 42 * 25 + 1 * 25 = 1117$; the importance of SQL injection attack is $1 * 1 + 1 * 25 + 1 * 25 = 51$. So, the main alarm in this period is SMTP e-mail attachment vulnerability.

There are 56 alarm logs in the time from 0:05, 2015/11/3, to 0:10, 2015/11/3; 51 of them are SMTP e-mail attachment vulnerability, and the attack level is prompt and happened in the last period. Five of them is Web application: SQL injection attack, the attack level is prompt and happened for the first time. So, the importance of SMTP e-mail attachment vulnerability in this period is $51 * 1 + 51 * 1 + 1 * 1 = 103$; the importance of SQL injection attack is $5 * 1 + 5 * 25 + 1 * 25 = 155$. So, the main alarm in this period is Web application: SQL injection attack.

Obtain the main alarm vectors in each time period of 288 time periods in turn to form the alarm vector sequence, as shown in Table 7.

The next step is to model the Markov model. Firstly, the initial state transition matrix is constructed according to the security event and the defense strategy:

$$P = \begin{vmatrix} P_{GG} & P_{GP} & P_{GA} & P_{GC} \\ P_{PG} & P_{PP} & P_{PA} & P_{PC} \\ P_{AG} & P_{AP} & P_{AA} & P_{AC} \\ P_{CG} & P_{CP} & P_{CA} & P_{CC} \end{vmatrix} = \begin{vmatrix} 0.612 & 0.218 & 0.078 & 0.092 \\ 0.108 & 0.672 & 0.210 & 0.010 \\ 0.029 & 0.172 & 0.762 & 0.037 \\ 0.007 & 0.025 & 0.201 & 0.767 \end{vmatrix}. \quad (27)$$

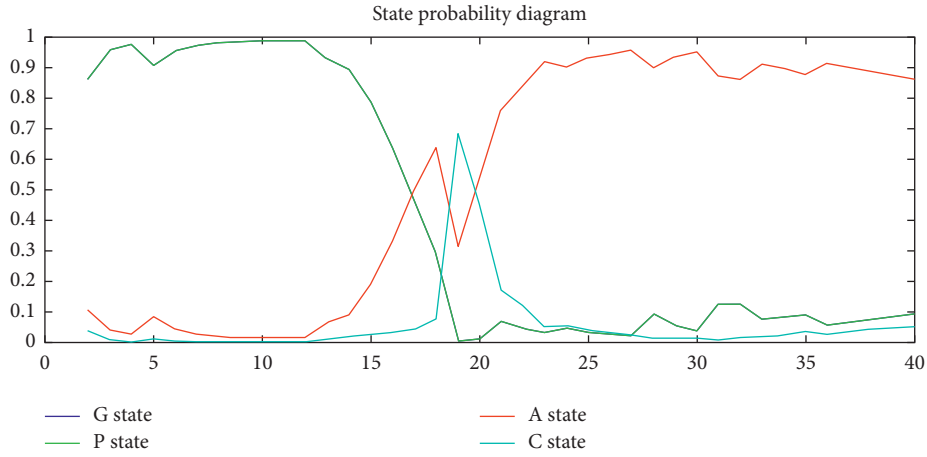

Figure 10: Hidden state distribution sequence.

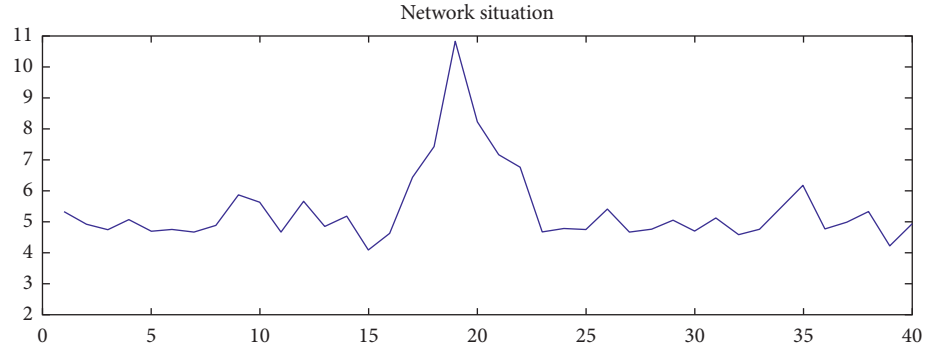

Figure 11: Final network situation diagram.

Extract the main efficiency indexes from the firewall, IPS, and the host in this period, including CPU utilization, memory utilization, bandwidth utilization, and connection utilization, as shown in Figures 5–8, respectively.

Analyze the weight of each parameter by the Analytic Hierarchy Process, the process is as follows:

CPU utilization is as important as memory utilization, as the main task of the network service is outside, so the network bandwidth utilization and the number of connections are more important than CPU utilization and memory utilization. The department prohibits the operation of large flow, but allowing more people to access the same time, so the number of connections is slightly more important than the bandwidth utilization. Finally, we can get the relation matrix, as shown in Table 8.

Input process, and get $CR = 0.0571 < 0.1$ which accord with the consistency standard. The weight is shown in Table 9.

Take each efficiency into the final defense efficiency curve, as shown in Figure 9.

The hidden state probability distribution curve, as shown in Figure 10.

And then, obtain the network risk vector in a similar way.

According to the confidentiality, integrity and availability of the national standard, the asset importance attribute is shown in Table 10.

Combined with the losses caused by security events, the risk loss vectors are obtained as follows:

$$C = (C_G, C_P, C_A, C_C) = (1, 2.6, 6.3, 14.7). \quad (28)$$

Finally, we can get the figure of network situation, as shown in Figure 11.

From Figure 11, we can see that in the 18 min to 20 min time period, the number of host connections is almost saturated, the service cannot provide services, and the host is actually in the capture state. And at this point, the network risk value is also in the highest state, in line with the actual situation. Through the above security events information and network situation diagram, network administrators can clearly understand the global network security event occurring at that time and control the network situation real-timely.

5. Conclusions

The network situation assessment technology based on HMMP is studied mainly in this paper, to solve the problem that network administrators can control the global network state real-timely in the face of multisource logs. In order to achieve this goal, state transition matrix generation method in HMMP is designed in this paper, the system can modify the state transition matrix automatically in real-time

according to the network state through this method and obtain the hidden state probability distribution sequence of the current system through the improved Viterbi algorithm. Finally, the final network risk value is obtained through the method of calculating the risk loss vector. The experiment shows that the security assessment based on HMMP in this paper can describe the current network state precisely and comprehensively.

Data Availability

The data set can be obtained free of charge from http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This present research work was supported by the National Natural Science Foundation of China (nos. 61202458 and 61403109), the Natural Science Foundation of Heilongjiang Province of China (no. F2017021), and the Harbin Science and Technology Innovation Research Funds (no. 2016RAQXJ036).

References

[1] P. Wang, L. Shi, B. Wang, Y. Liu, and Y. Wu, "A method for HMM-based system calls intrusion detection based on hybrid training algorithm," in *Proceedings of the International Conference on Information and Automation*, Shenzhen, China, June 2011.

[2] J. N. Davies, P. Comerford, and V. Grout, "Principles of eliminating access control lists within a domain," *Future Internet*, vol. 4, no. 2, pp. 413–429, 2012.

[3] C.-J. Chung, P. Khatkar, T. Xing, J. Lee, and D. Huang, "NICE: network intrusion detection and countermeasure selection in virtual network systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 10, no. 4, pp. 198–211, 2013.

[4] X. Y. Li, Q. X. Wang, and L. Yang, "Research of network security situation index system and visualization technology," *Journal on Communications*, vol. 32, no. 11, pp. 109–118, 2011.

[5] S. Mathew, S. Upadhyaya, M. Sudit, and A. Stotz, "Situation awareness of multistage cyber attacks by semantic event fusion," in *Proceedings of the Military Communications Conference*, pp. 1286–1291, San Jose, CA, USA, October 2010.

[6] F. Baiardi, F. Corò, F. Tonelli, and D. Sgandurra, "Automating the assessment of ICT risk," *Journal of Information Security and Applications*, vol. 19, no. 3, pp. 182–193, 2014.

[7] M. R. Endsley, "Design and evaluation for situation awareness enhancement," in *Proceedings of the Human Factors Society 32nd Annual Meeting*, pp. 97–101, Santa Monica, CA, USA, 1988.

[8] T. Bass, "Intrusion detection systems and multisensor data fusion," *Communications of the ACM*, vol. 43, no. 4, pp. 99–105, 2000.

[9] K. Lakkaraju, W. Yurcik, and A. J. Lee, "NVisionIP: net-flow visualisations of system state for security situational awareness," in *Proceedings of the 2004 ACM Workshop on Visualisation and Data Mining for Computer Security*, pp. 65–72, Washington, DC, USA, 2004.

[10] H. T. Elshoush and I. M. Osman, "Alert correlation in collaborative intelligent intrusion detection systems—a survey," *Applied Soft Computing*, vol. 12, no. 4, pp. 4349–4365, 2011.

[11] H. T. Elshoush, "An innovative framework for collaborative intrusion alert correlation," in *Proceedings of 2014 Science and Information Conference SAI*, pp. 607–614, London, UK, August 2014.

[12] Y. Wei, Y. F. Lian, and D. G. Feng, "A network security situational awareness model based on information fusion," *Journal of Computer Research and Development*, vol. 46, no. 3, pp. 353–362, 2009.

[13] X. Z. Chen, Q. H. Zheng, and X. H. Guan, "Quantitative hierarchical threat evaluation model for network security," *Journal of Software*, vol. 17, no. 4, pp. 885–897, 2006.

[14] R. Xi, "Quantitative assessment of network security situation based on fusion of multi-source," Master thesis, University of the Chinese Academy of Sciences, Beijing, China, 2013.

[15] S. Zhu, X. Chen, H. Xiao et al., "Network security design of ERP system IAM," *Electronic Design Engineering*, vol. 4, pp. 166–169, 2014.

[16] C. Xiuqing, Z. Yongping, and T. Jiutao, "HMM-based integration of multiple models for intrusion detection," in *Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, pp. 137–140, Chengdu, China, August 2010.

[17] C. Sharma and V. Kate, "ICARFAD: a novel framework for improved network security situation awareness," *International Journal of Computer Applications*, vol. 87, no. 19, pp. 26–31, 2014.

[18] Y. Wang, G. Zhuang, X. Chen, Z. Wang, and F. Chen, "Dynamic event-based finite-time mixed $H\infty$ and passive asynchronous filtering for T–S fuzzy singular Markov jump systems with general transition rates," *Nonlinear Analysis: Hybrid Systems*, vol. 36, p. 100874, 2020.

[19] Y. Wang, F. Chen, G. Zhuang, and G. Yang, "Dynamic event-based mixed $H\infty$ and dissipative asynchronous control for Markov jump singularly perturbed systems," *Applied Mathematics and Computation*, vol. 386, p. 125443, 2020.

[20] G. C. Kuang, X. F. Wang, and L. R. Yin, "A fuzzy forecast method for network security situation based on Markov," in *Proceedings of the 2012 International Conference on Computer Science and Information Processing (CSIP)*, pp. 785–789, Xi'an, Shaanxi, China, August 2012.