Network Situation Awareness Model Based on Incomplete Information Game

Hongbin Zhang[1,2], Yan Yin[1], Dongmei Zhao[2(✉)], Bin Liu[3,4], Yanxia Wang[5], and Zhen Liu[1]

[1] School of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang 050000, China

[2] Hebei Key Laboratory of Network and Information Security, Hebei Normal University, Shijiazhuang 050024, Hebei, China
zhaodongmei666@126.com

[3] School of Economics and Management, Hebei University of Science and Technology, Shijiazhuang 050000, China

[4] Research Center of Big Data and Social Computing, Hebei University of Science and Technology, Shijiazhuang 050000, China

[5] Hebei Geological Workers' University, Shijiazhuang 050081, China

Abstract. Game theory has been widely used in network security situational awareness. However, most of the currently proposed game-based offensive and defensive situational awareness methods are for traffic data, and there are fewer models or methods for analysis using vulnerability data. To overcome these issues, this paper proposes collecting periodic security vulnerability information in the network and utilizing the change in vulnerability status to achieve network security situational awareness. At this time, a network attack and defense game model based on incomplete information is proposed, which uses the state changes of the vulnerability life cycle to model the attack and defense behavior, calculates the benefits of both attack and defense through the evaluation of the exploitability of the vulnerability, and then quantifies the security situation value. We carried out the experiments using the vulnerability dataset, which was obtained by scanning the IP addresses of several enterprises in Hebei Province, China. The experimental results show that the approach of using network security vulnerabilities to assess network security status is feasible.

Keywords: Situation awareness · incomplete information · attack-defense game · vulnerability lifecycle · state transition matrix

1 Introduction

Nowadays, the network is moving towards large scale, big data, and multiple levels. At the same time, the types and number of attacks have increased dramatically. The number of security vulnerabilities released by the China National Vulnerability Database (CNVD) in 2021 was as high as 26,562, an increase of 24.2% compared with the previous year.

Q. Jiang et al. (Eds.): SPNCE 2022, LNICST 496, pp. 167–178, 2023.
https://doi.org/10.1007/978-3-031-30623-5_12

Due to the untimely discovery and patching of vulnerability information, users continue to suffer from attacks, resulting in the network status being unpromising. To address the many potential risks in the network, network security situational awareness (NSSA) has been created.

Network situation awareness was defined as the acquisition, understanding, and display of security elements that can bring about the network situation changes in a large-scale network environment, as well as the prediction of network development trends [1]. By extracting and comprehensively understanding many network security risk elements, situational awareness can evaluate the network security status and predict the impact of risks [2 3, 4, 5]. Therefore, using situational awareness to discover potential threats and respond has become a research priority in network security [6–8].

In 1999, game theory model theory gradually emerged in the field of network security and was applied to the assessment of network security status [9]. Game theory is the theory of strategy selection and confrontation between different game parties, and the process of network attack and defense is to use the limited resources in the network to select the appropriate strategy for confrontation, and the process is in line with the idea of game [10]. The literature [11] uses stochastic games to assess the network security posture and constructs an assessment model. In order to solve the problem that the power IoT is vulnerable to security threats due to the weak distributed open structure, the literature [12] proposes to construct a differential game model to model the interaction behavior of power IoT smart terminals and attackers, and gives the optimal defense strategy for the system by solving for the equilibrium value. In addition to this, network attack and defense games are combined with attack graphs [13, 14], Markov theory [15], and Bayesian networks [16] for situational awareness of the network.

In local area networks or small-scale networks, network security situational awareness is primarily based on the analysis of attack traffic data [17, 18]. However, in the context of large-scale networks, the amount of traffic data is huge and the workload on data processing is high. As a result, NSSA becomes more difficult and accuracy is greatly reduced. To overcome these issues, this paper collects information on security vulnerabilities by probing the network assets, analyses the state of the vulnerabilities, and then builds an attack-defense game model based on incomplete information according to the transfer of each state in the vulnerability lifecycle. Drawing on the Common Vulnerability Scoring System (CVSS) to evaluate the exploitability of vulnerabilities, this paper achieves a quantitative assessment of the network situation.

In summary, considering the characteristics of large-scale networks with complex topology, numerous network nodes, and difficulty in processing traffic information. The main contributions of this paper are divided into two points: 1) According to the periodic changes of vulnerabilities in the network, the vulnerability state transition matrix is determined. Combined with the attacker's ability, vulnerability availability, and the expected probability of vulnerability repair, the matrix is revised to improve the accuracy of the state transition matrix. 2) An incomplete information attack-defense game model is constructed, which uses CVSS to evaluate the exploitability of vulnerabilities, and quantifies the network security situation value.

2 State Transfer for Vulnerability

2.1 Vulnerability Lifecycle and State Transition Matrix

Vulnerabilities are flaws or errors in the specific implementation of hardware, software, protocols, or the customization of system security policies, which may be exploited intentionally or unintentionally, allowing outsiders to gain unauthorized access or destroy the system. The vulnerability lifecycle describes the entire process of a vulnerability from creation to remediation, and it is divided into 5 stages, as shown in Fig. 1.

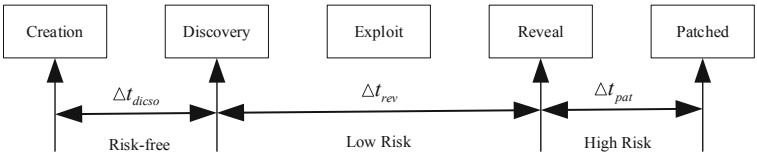


Fig. 1. Vulnerability lifecycle stages

2.2 State Transition Matrix

2.2.1 Determination of State Transition Matrix

As the life cycle of a vulnerability has different status characteristics at different times, this paper divides the life cycle status of the vulnerability into two categories: vulnerability disclosure and vulnerability undisclosed, and analyzes the life cycle of the vulnerability respectively.

- *Vulnerability disclosure*

After the vulnerability is disclosed, the vulnerability database records most of the information about the vulnerability such as the type, discovery time, collection time, and patch program. The life cycle is complete, and the state transition model is shown in Fig. 2. At this time, the vulnerability state transition matrix is given in Eq. (1).

$$Q_1 = \begin{bmatrix} P_{CC} & P_{CD} & 0 & 0 \\ 0 & P_{DD} & P_{DR} & P_{DE} \\ 0 & 0 & P_{RR} & P_{RE} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

In the formula, $State_a$ ($a = Cre, Dis, Rev, Exp$) is a tuple representation of each state of the vulnerability lifecycle. P_{ab} denotes the probability of a vulnerability moving from $State_a$ to $State_b$ in one step. The sum of the probabilities of transitioning from one state to another is 1.

The transition probability of each state is calculated as follows:

$$P_{CD} = \sum_{k=1}^{x} \left(\frac{NumAdd(State_{Cre \to Dis})_k}{Num(State_{Cre \to Dis})_{k-1}} \right) \Big/ x \tag{2}$$

$$P_{CC} = 1 - P_{CD} \tag{3}$$

Among: $NumAdd(State_{Cre \to Dis})_k$ represents the number of state transfers from $State_{\text{Cre}}$ to $State_{Dis}$, after the result of the *kth* scan. $Num(State_{Cre \to Dis})_{k-1}$ represents the number of identical state transfers, after the result of the *k-1th* scan. By analogy, other vulnerability state transfer probabilities are calculated. $P_{DR} \sim P_{EE}$ can be calculated using the same method.

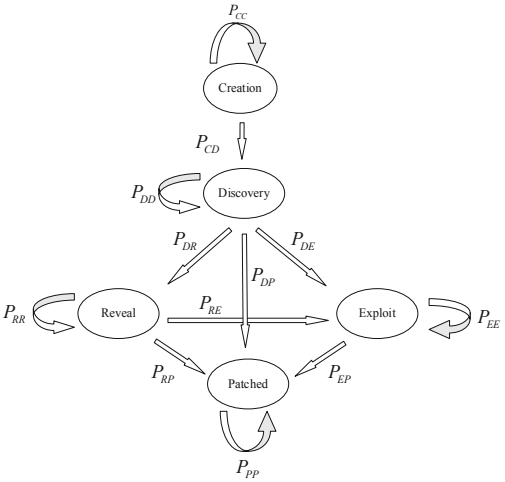


Fig. 2. State transition model

- *Vulnerability undisclosed*

When a vulnerability is not publicly disclosed, the exploitation of the vulnerability and the availability of patches have not yet been determined. Figure 3 illustrates the state transition model when the vulnerability is undisclosed.

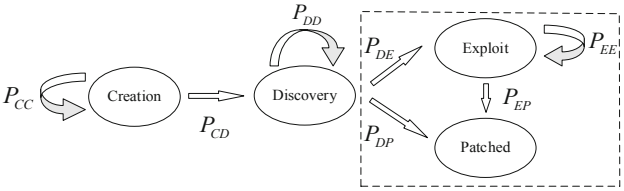


Fig. 3. State transition model when the vulnerability is undisclosed

The state transition matrix Q_2 is divided into 4 cases, namely, the vulnerability is not exploited and the security vendor has not provided a patch, the vulnerability is not

exploited but the patch has been provided, the vulnerability is exploited but the patch has not been provided, the vulnerability is exploited and the patch has been provided. Therefore, Q_2 is:

$$Q_{21} = \begin{bmatrix} P_{CC} & P_{CD} \\ 0 & 1 \end{bmatrix} Q_{22} = \begin{bmatrix} P_{CC} & P_{CD} & 0 \\ 0 & P_{DD} & P_{DE} \\ 0 & 0 & 1 \end{bmatrix} Q_{23} = \begin{bmatrix} P_{CC} & P_{CD} & 0 & 0 \\ 0 & P_{DD} & P_{DE} & P_{DP} \\ 0 & 0 & P_{DE} & P_{DP} \\ 0 & 0 & 0 & 1 \end{bmatrix} Q_{24} = \begin{bmatrix} P_{CC} & P_{CD} & 0 \\ 0 & P_{DD} & P_{DP} \\ 0 & 0 & 1 \end{bmatrix}$$

2.2.2 Correction of State Transition Matrix

The state transition probability of vulnerability is also related to the attacker's ability, the exploitability of the vulnerability, and the repair of the vulnerability. Therefore, define the correction function ρ of the vulnerability state transition matrix as Eq. (4).

$$\rho = \left(\sum_{\alpha=1}^{3} AB(\alpha) \cdot EA \cdot EP(Vuln) \right) \Big/ 3 \tag{4}$$

AB represents the ability of the attacker. According to the ability of the attacker, the attacker is divided into 3 levels: junior attacker, skilled attacker, and professional. The probability is set as 4/5, 4/25, and 1/25. *EA* represents the availability of vulnerabilities. According to the CVSS3.0, Table 1 lists the relevant indicators, description information, classification, and impact score of the vulnerability exploits. According to CVSS3.0, the calculation formula for *EA* is Eq. (5).

$$EA = 8.22 \times AV \times AC \times PR \times UI \tag{5}$$

Table 1. Vulnerability availability indicator score

Indicator	Classification	Score
Attack Vector (AV)	Network/Adjacent/Local/Physical	0.85/0.62/0.55/0.2
Attack Complexity (AC)	Low/High	0.77/0.44
Permission Requirement (PR)	None/Low/High	0.85/0.62/0.27
User Interaction (UI)	None/Require	0.85/0.62

EP(Vlun) is the expected probability of vulnerability repair.

$$EP(Vuln) = \sum_{k=1}^{x} \left(k_{Dis \to Pat} \; k_{\mathrm{Rev} \to Pat} \; k_{Exp \to Pat} \right) \begin{bmatrix} P_{DP} \\ P_{RP} \\ P_{EP} \end{bmatrix} \tag{6}$$

According to ρ, the correct result of the state transition matrix is $Q' = \rho \times Q$. When the probability of vulnerability transferring to Patched is greater than the probability of vulnerability transferring to Exploit, the network is in a safe state; otherwise, it is in a dangerous state.

3 Attack-Defense Game Based on Incomplete Information

1) **Model Definition**

Definition 1. The Incomplete Information Attack-defense Game Model (IIADGM) describes the network attack and defense behavior in the incomplete information scenario. *IIADGM = (N, T, S, P, U)*, the meaning of each element is as follows:

N: The participants in the game.
T: $T = (T_A, T_D)$ represents the set of types of players.
S: $S_A = (S_A^1, S_A^2, \cdots, S_A^i)$, , $(1 \leq i \leq n)$ denotes the set of attack strategies; the set of defense strategies is $S_D = (S_D^1, S_D^2, \cdots, S_D^j)$, , $(1 \leq j \leq m)$.
P: It denotes the set of a priori beliefs of the players.
U: $U = (U_A, U_D)$ is the set of the utility function.

2) **Situation Quantification**

The quantification of the benefits of both sides of the game is the basis of game analysis and the key to network situation assessment.

(1) Attack Reward (*AR*) *AR* is related to the probability that the vulnerability life cycle is in Exploit (P_{Exp}) and the impact of the vulnerability on the network (*IS*).

$$AR = P_{Exp} \times IS \tag{7}$$

Among: *IS* is related to Confidentiality (C), Integrity (I), and Availability (A). (*C,I,A*) are divided into 3 categories, namely None, Low, and High. The corresponding scores are 0, 0.22, and 0.56. P_{Exp} and IS are calculated as follows:

$$P_{Exp} = \sum_{a} P(State_{a \to Exp}), a = Cre, Dis, \text{Re}v, Exp \tag{8}$$

$$IS = 10.41 \times [1 - (1 - C) \times (1 - I) \times (1 - A)] \tag{9}$$

(2) Attack Cost (*AC*) The cost of launching an attack varies depending on the maturity of the attacker's exploitation of the code (*Pro*). The more proficient the code exploitation, the fewer resources and time it consumes. *Pro* is divided into 4 levels: Unproven that exploit exists, Proof of concept code, Functional exploit exists, and High, corresponding to the values of 0.91, 0.94, 0.97, and 1.

In addition, *AC* is also related to the perfection of the patch released by the vendor (*Pre*). The more complete the patch release, the higher the difficulty level for an attacker to exploit the vulnerability and the higher the cost required. *Pre* is also divided into 4 levels: Unavailable, Workaround, Temporary fix, and Official fix, corresponding to the values of 0.91, 0.94, 0.97, and 1. So

$$AC = Roundup(EA \times Pro \times Pre) \tag{10}$$

(3) Defense Reward (*DR*) The defender takes certain defensive measures that result in the attacker not getting the expected reward. Therefore, *DR* is numerically equal to the impact on the network if the attack is successful. So $DR = AR$.
(4) Defense Cost (*DC*) The cost incurred by a defender in employing some defensive measures to repel an attack. As real-life vulnerability remediation scenarios and techniques vary, costs are not easy to quantify. This paper considers that the cost of fixing a vulnerability is related to the base score of the vulnerability and the probability of the vulnerability being patched (P_{Pat}). The equation is shown below:

$$BaseScore = Roundup(IS + EA) \tag{11}$$

$$P_{Pat} = \sum_{a} P(State_{a \to Pat}), a = Cre, Dis, \mathrm{Rev}, Exp, Pat \tag{12}$$

$$DC = BaseScore \times P_{Pat} \tag{13}$$

(5) Attack Success Rate (θ) It reflects the probability that the attacker successfully exploits the vulnerability and has an impact, and is determined by the probability λ of the attack being detected and the probability β of the defense being successful.

Therefore, when the attacking and defending sides take strategies (S_A^i, S_D^j) to fight, the attack utility U_A and the defense utility U_D are:

$$U_A = \sum_{i=1}^{n} \sum_{j=1}^{m} \left\{ \left[1 - \lambda(S_D^j)\beta(S_D^j) \right] AR(S_A^i) - AC(S_A^i) \right\} \tag{14}$$

$$U_D = \sum_{j=1}^{m} \sum_{i=1}^{n} \left\{ \lambda(S_D^j)\beta(S_D^j) AR(S_A^i) - DC(S_D^j) \right\} \tag{15}$$

According to the utility value, the security situation of the target network is defined as $S = U_D - U_A$. When $S > 0$, the network is in a safe state. The larger the $|S|$, the more secure the network is. Instead, the network is in a dangerous state.

4 Experiments and Analyses

1) State Transition Matrix

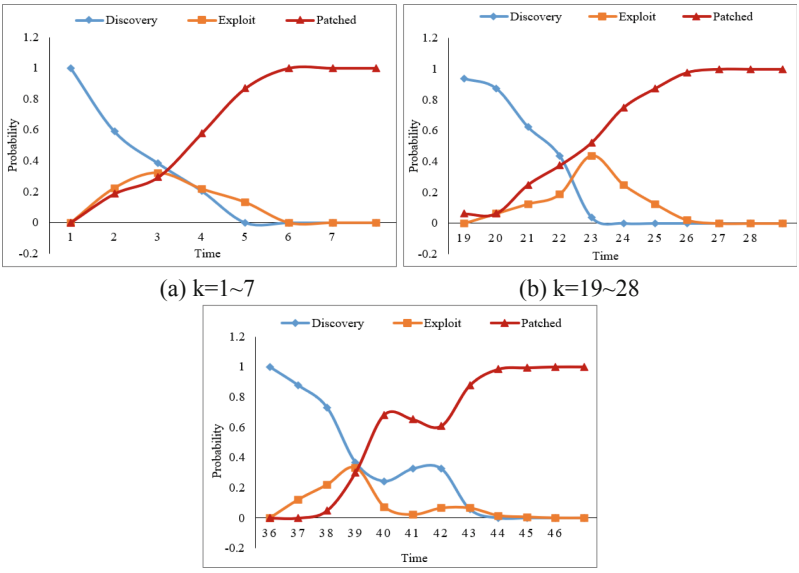
To determine the state transfer matrix, this paper performs vulnerability scans on the IP addresses of the websites of over 100 enterprises in Hebei Province. Since each round of vulnerability scanning takes a long time, this paper has conducted vulnerability scanning on a 7-day cycle. The scan started on January 25, 2021, and ended on January 19, 2022, going through 52 rounds of vulnerability scanning, with a total of 587 vulnerabilities scanned. These vulnerabilities are divided into 10 types. The vulnerability types and quantity statistics are shown in Table 2.

Table 2. Vulnerability type and number statistics

Type	Quantity	Type	Quantity
HTTP Request Smuggling	5	Command Injection	8
SQL Injection	5	Directory Traversal	34
SSRF	47	Arbitrary File Deletion	47
Safe Mode Bypass	79	Unauthorized Access	13
Command Execute	73	File Upload	29

As the time of the creation of a vulnerability is not recorded in vulnerability database, the time of discovery of a vulnerability is the time of exploitation of the vulnerability. Therefore, in the experiment, this paper only discusses the transition between the 3 states of Discovery, Exploit, and Patched.

Code execution vulnerabilities were selected for state transition analysis and the probability of state occurrence for this type of vulnerability in different vulnerability scanning cycles was calculated. As code execution vulnerabilities are not continuous throughout the scanning phase, this paper extracts three scanning cycles of k = 1–7, k = 19–28, and k = 36–46 to calculate the probability of state occurrence. The probability change curves are shown in Fig. 4.



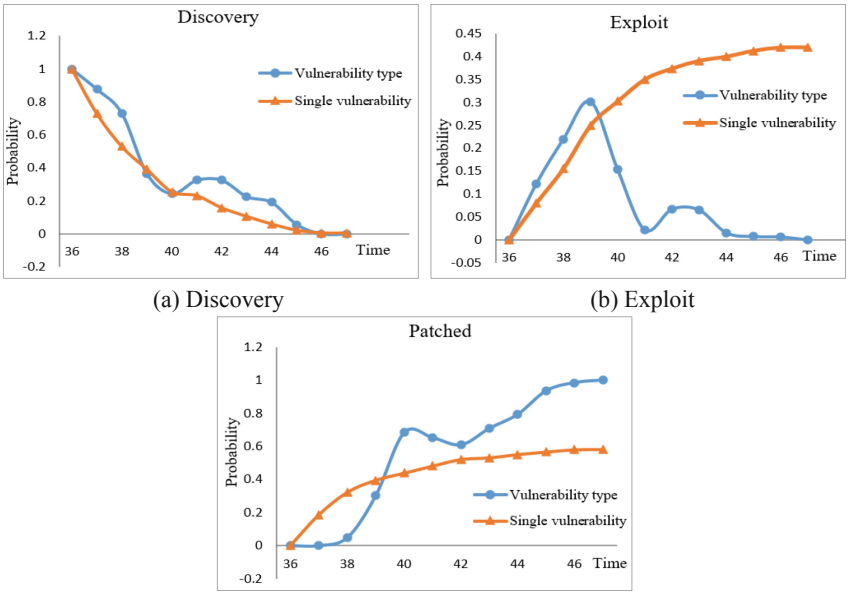
(a) k=1~7 (b) k=19~28

(c) k=36~46

Fig. 4. Probability curve

It can be seen from the Fig. 4 that: 1) In Discovery: When a vulnerability is first discovered, its probability value is 1. Over time, the vulnerability state gradually moves to the Exploit and Patched. 2) In Exploit: The Exploit state starts with a probability of 0. As the vulnerability is published and exploited, the probability of the Exploit state occurring increases and reaches the maximum value. However, after security vendors released vulnerability patches, the probability of Exploit gradually decreased. 3) In Patched: The initial probability is also 0. As the vulnerability was discovered, they began to be fixed. Over time, the probability of the Patched continues to increase, and eventually reaches a stable level.

To verify the accuracy of the state transfer method proposed in this paper, a comparative analysis with the single vulnerability state transition probability [19] method was performed. The comparison results are shown in Fig. 5.



(a) Discovery (b) Exploit

(c) Patched

Fig. 5. Comparative graphs of state occurrence probability

Figure 5 illustrates that the life cycle state transition method has 2 improvements compared to the state transition of a single vulnerability: 1) The method in this paper has a certain degree of recoverability. For example, the probability of Discovery appears to decrease, and then increase. This is because vulnerability patches are targeted, the same vulnerability patch does not fix all vulnerabilities of the same type, so vulnerabilities of that type may still be found in the next scan cycle. However, the state transition probabilities based on single vulnerabilities do not discuss this situation. 2) During the process of vulnerability state transfer, as the vulnerability is discovered and submitted, the probability of being in the Exploit increases. However, as patches are released and vulnerabilities are gradually fixed, the probability of Exploit decreases, and the probability of Patched

increases. However, based on the state transition process of single vulnerabilities, the probability of the Exploit keeps rising and eventually remains unchanged, ignoring the possibility of the above situation.

The modified state transfer matrix Q' is:

$$Q' = \rho \times Q = 0.9 \times \begin{bmatrix} 0.2858 & 0.1428 & 0.5714 \\ 0 & 0.0437 & 0.9563 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.2572 & 0.1285 & 0.5143 \\ 0 & 0.0393 & 0.8607 \\ 0 & 0 & 0.9 \end{bmatrix}$$

2) **Situation Assessment**

Combined with the state transition matrix, this paper analyzed the game process of the attack-defense of the Code Execution vulnerability in the period of k = 36–46 and calculated the attack-defense income and the situation value of the network. Figure 6. Illustrates the trend diagram of network security situation change.

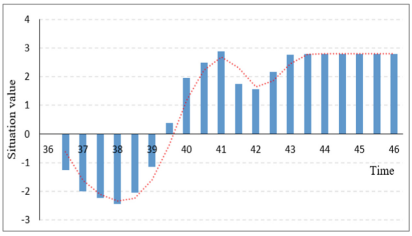


Fig. 6. Situation curve

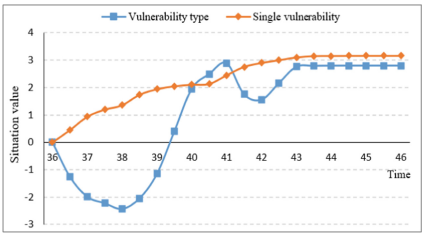


Fig. 7. Comparison of assessment results

It can be seen from Fig. 6: in k = 36–39 cycles, vulnerabilities are exploited and cause harm, and the network is in a dangerous state. In k = 38–39 cycles, security vendors release patches and some companies start to fix vulnerabilities, and the danger decreases. In k = 40–46 cycles, the situation value is positive and the network is in a safe state. k = 41–43 cycles, the situation value decreases somewhat as new Code Execution vulnerabilities are discovered and exploited. However, as time passes, the vulnerabilities are fixed, the game ends, and the situation values level off.

Figure 7 compares the assessment methods based on a single vulnerability and type of vulnerability. The single-vulnerability situational assessment method only considers the risk value of the vulnerability and does not consider the reduction of the risk value after the vulnerability has been fixed. This method cannot reflect the changes in the network situation in the long term, so the vulnerability type-based situational assessment method proposed is more in line with reality.

5 Conclusions

This paper proposes to use the transition of a vulnerability state to study the network security situation. In the time dimension, the security state of the network is analyzed

according to the state transition matrix of the vulnerability. Based on the process of vulnerability exploitation and repair, an incomplete information game model is constructed to assess the changes in the network situation. Through experimental analysis, the vulnerability type-based posture assessment method can provide an effective assessment of the network situation, and the assessment results are more realistic. The next work is to implement the prediction part of situational awareness and make accurate and reasonable prediction about unknown threats in the network.

References

1. Husák, M., Komárková, J., Harb, E.B., Čeleda, P.: Survey of attack projection, prediction, and forecasting in cyber security. IEEE Commun. Surv. Tutorials **21**(1), 640–660 (2018)
2. Park, M., Oh, H., Lee, K.: Security risk measurement for information leakage in IoT-based smart homes from a situational awareness perspective. Sensors **19**(9), 2148 (2019)
3. Zhang, H.B., Yi, Y.Z., Wang, J.S., Cao, N., Duan, Q.: Network security situation awareness framework based on threat intelligence. Comput. Mater. Continua **56**(3), 381–399 (2018)
4. Gong, J., Zang, X.D., Su, Q., Hu, X.Y., Xu, J.: Survey of network security situation awareness. J. Softw. **28**(4), 1010–1026 (2017)
5. Li, Y., Huang, G.-Q., Wang, C., Li, Y.-C.: Analysis framework of network security situational awareness and comparison of implementation methods. EURASIP J. Wirel. Commun. Netw. **2019**(1), 1–32 (2019). https://doi.org/10.1186/s13638-019-1506-1
6. Kou, G., Wang, S., Tang, G.: Research on key technologies of network security situational awareness for attack tracking prediction. Comput. Netw. Commun. **28**(1), 162–171 (2019)
7. Han, W.H., Tian, Z.H., Huang, Z.Z., Zhong, L., Jia, Y.: System architecture and key technologies of network security situation awareness system YHSAS. Comput. Mater. Continua **59**(1), 167–180 (2019)
8. Zhao, D., Liu, J.: Study on network security situation awareness based on particle swarm optimization algorithm. Comput. Ind. Eng. **2018**(125), 764–775 (2018)
9. Burke, D.A.: Towards a game theory model of information warfare. Airforce Institute of Technology (l999)
10. He, F.F., Zhang, Y.Q., Liu, H.Z., Zhou, W.: SCPN-based game model for security situational awareness in the internet of things. In: 2018 IEEE Conference on Communications and Network Security (CNS), pp. 1–5. IEEE, Beijing (2018)
11. Li, T.F., Li, Q., Yu, X., Wu, D.Y.: A topological vulnerability analysis-based network security situational awareness model. J. Comput. Sci. **38**(S2), 157–163+169 (2018)
12. Li, Z., Liu, Y.Z., Liu, D., Zhang, N., Lu, D.W., Huang, X.G.: A security defense model for ubiquitous electric internet of things based on game theory. In: 2020 IEEE 4th Conference on Energy Internet and Energy System Integration (EI2), pp. 3125–3128. IEEE, Wuhan (2020)
13. Jin, Z.G., Wang, J.X., Li, G., Yue, M.S.: A network defense policy generation method incorporating attack graphs and game models. Inf. Netw. Secur. **21**(01), 1–9 (2021)
14. J. Liu, Y. C. Zhang, H. Hu, J. L. Tan, Q. Len and C. W. Zhang: Efficient Defense Decision-Making Approach for Multistep Attacks Based on the Attack Graph and Game Theory. Mathematical Problems in Engineering (2020)
15. Li, X., Lu, Y., Liu, S., Nie, W.: Network security situation assessment method based on Markov game model. KSII Trans. Internet Inf. Syst. (TIIS) **12**(5), 2414–2428 (2018)
16. Hu, H., Liu, Y.L., Zhang, H.Q., Pan, R.X.: Optimal network defense strategy selection based on incomplete information evolutionary game. IEEE Access **6**, 29806–29821 (2018)
17. Liao, Y.W., Zhao, G., Wang, J., Li, S.: Network security situation assessment model based on extended hidden Markov. Math. Probl. Eng. 2020 (2020)

18. Shang, L., Zhao, W., Zhang, J., Fu, Q., Zhao, Q., Yang, Y.: Network security situation prediction based on long short-term memory network. In: 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), pp. 1–4. IEEE,Matsue (2019)
19. Hu, H., Ye, R.G., Zhang, H.Q., Chang, D.X., Liu, Y.L., Yang, Y.J.: Vulnerability life cycle oriented security risk metric method. J. Softw. **29**(5), 1213–1229 (2018)