International Journal of Digital Crime and Forensics
Volume 15 • Issue 1

Key Node Identification Based on Vulnerability Life Cycle and the Importance of Network Topology

Yuwen Zhu, State Key Laboratory of Mathematical Engineering and Advanced Computing, China*

Lei Yu, Chinese Academy of Sciences, China

ABSTRACT

The key network node identification technology plays an important role in comprehending unknown terrains and rapid action planning in network attack and defense confrontation. The conventional key node identification algorithm only takes one type of relationship into consideration; therefore, it is incapable of representing the characteristics of multiple relationships between nodes. Additionally, it typically disregards the periodic change law of network node vulnerability over time. In order to solve the above problems, this paper proposes a network key node identification method based on the vulnerability life cycle and the significance of the network topology. Based on the CVSS score, this paper proposes the calculation method of the vulnerability life cycle risk value, and identifies the key nodes of the network based on the importance of the network topology. Finally, it demonstrates the effectiveness of the method in the selection of key nodes through network instance analysis.

KEYWORDS

Importance of Topology, Key Network Nodes, Risk Value, Vulnerability Life Cycle

INTRODUCTION

With the highly complex nature of a network structure, the identification of key network nodes is an important method to analyze and master the complex network structure and function. The key nodes of the network refer to the nodes that play a decisive role in the structure and stability of the network. If a defender loses the authority of such nodes in the process of an attack and defense, it will lead to a rapid decline in network performance and even disrupt the connectivity of the entire network structure.

One of the important topics in network scientific research is how one can identify the influence of each node accurately and efficiently in a complex network. At present, network key node identification technology mainly refers to key node identification based on network topology and key node identification based on network node vulnerability.

DOI: 10.4018/IJDCF.317100

*Corresponding Author

However, the existing methods generally measure the influence of nodes from a single angle or a certain aspect, which is not comprehensive enough to consider all the problems. The traditional methods do not consider the aspect of attack and defense and ignore the impact of the network node's vulnerabilities in terms of network security and the difficulty of network attack and defense. Most of the key network nodes are identified by using static methods and the distribution law of the vulnerability utilization probability is not taken into consideration in the time dimension of vulnerability generation.

In order to provide a solution to the aforementioned problems, this paper studies the network key node identification method based on the vulnerability life cycle and the significance of the network topology. The network topology structure and the change of node vulnerability life cycle over time are comprehensively explained, thus dynamically reflect the changes of key network nodes in real-time.

The contributions of this paper are as follows:

- The authors propose a formal description of network key nodes based on vulnerability life cycle.
- The authors propose a calculation method of vulnerability life cycle risk value based on common vulnerability scoring system (CVSS) score.
- The authors propose a method for identifying key network nodes based on the vulnerability life cycle and the importance of network topology.
- The authors designed an example and perform a security analysis on a network abstract model, thereby proving rapid modeling, quantitative calculation, and the final key node identification of the target network.

The rest of this paper is structured as follows. The second section discusses the related work. The third section details the formal description of network key nodes based on the vulnerability life cycle. The fourth section calculates the vulnerability lifecycle risk based on CVSS score. The fifth section proposes the key node identification method based on the vulnerability life cycle and importance of network topology. The sixth section gives an example to illustrate the effectiveness of the method of identification of key network nodes. The seventh section gives a comparison of related work. Finally, the eighth section summarizes the paper and proposes future work.

RELATED WORK

Although a lot of research has been conducted in the fields of vulnerability life cycle, key network nodes, and multi-attribute analysis, a systematic theoretical method has not yet been proposed to incorporate the vulnerability life cycle into the analysis of key network nodes.

The concept of the vulnerability life cycle was first proposed by Arbaugh et al. (2000) of CERT (the Computer Emergency Response Team) in the United States. They analyzed several states that a vulnerability may experience from production to extinction. Combined with the security report issued by the coordination center of CERT, they reported the distribution of a number of vulnerabilities in different states over the years. Frei (200) used system dynamics to model and analyze the vulnerability life cycle, but the dataset used in the experiment was small, and the impact of software vendors was not analyzed. Combined with the open-source OSVDB vulnerability database, Kaaniche et al. (2013) analyzed the distribution of time length of Windows, Unix, and Mobile OS operating system vulnerabilities in different life cycle stages. The results show that the time distribution is related to specific operating system types. Mingqiu et al. (2011) calculated the vulnerability security risk value based on the Mamdani model by quantifying the attack frequency and technology of the time dimension of the vulnerability life cycle.

The most widely used methods for network key node identification based on network topology includes: degree centrality, betweenness centrality, proximity centrality, etc. Although the degree centrality (Freeman, 1977) algorithm is simple and efficient, it does not take the global structure of the network into consideration. Betweenness centrality (Freeman, 1978) and closeness centrality (Dangalchev, 2006; Salavati et al., 2018) do take the global structure of the network into consideration, but the algorithm has high time complexity and is not suitable for large-scale network applications. By taking the position of nodes in the network into consideration, Kitsak et al. (2010) proposed a k-shell decomposition method, which can be applied to large-scale networks. Since there are a large number of nodes with the same degree value in the network, the resolution of the k-shell (Newman, 2006) sorting method is not high. To solve the problem of coarse-grained division results of the k-shell sorting method, along with node degree and neighbor nodes, Kaixuan et al. (2006) proposed an improved k-shell method to identify key nodes according to the number of iterative layers in the process of node deletion.

FORMAL DESCRIPTION OF NETWORK KEY NODES BASED ON VULNERABILITY LIFE CYCLE

Vulnerability Life Cycle Stages

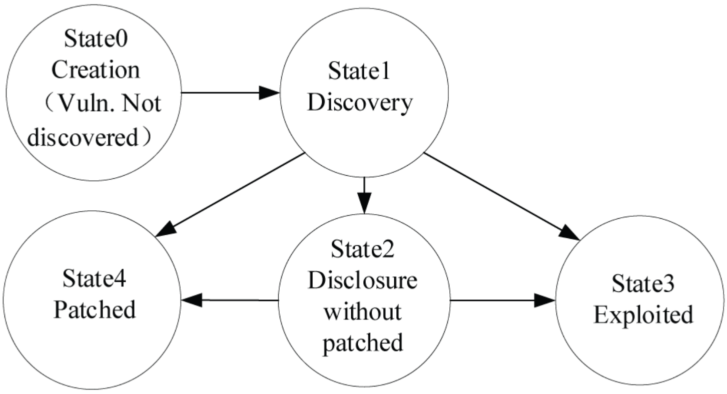
After much research by many scholars, it was found that almost all types of security vulnerabilities have the risk of being exploited by malicious users during their life cycle, but in different stages, the degree and type of risk are also different. The widely recognized concept of the vulnerability life cycle was proposed by Joh and Malaiya (2010), which mainly divides the vulnerability life cycle into five stages: creation, discovery, disclosure without patching, exploited, and patched. Figure 1 presents the vulnerability lifecycle state transition stages.

State 0 - Creation: Vulnerabilities usually exist at the beginning of the design of software or programs, and new vulnerabilities appear as the system is continuously updated. The vulnerability is in the “Creation” state during the period from the occurrence of the vulnerability until the time of its discovery.

State 1 - Discovery: The part where the vulnerability is located is discovered for the first time in the process of testing and operation. The discoverer of the vulnerability may be a software developer, a vulnerability research institution, or a hacker organization. Regardless of whether the discoverer is a black hat or a white hat, once such a defect is discovered, it marks the occurrence of a vulnerability, and the discoverer may not disclose the vulnerability in time. This is the period from vulnerability discovery until the time of vulnerability exploitation, vulnerability disclosure, or vulnerability repair. All internal vulnerabilities are included in the “Discovery” state.

State 2 - Disclosure Without Patched: Authoritative organizations (such as CERT, CVE, CNVD, etc.) issue vulnerability announcements to disclose detailed information about vulnerabilities.

Figure 1. Vulnerability lifecycle state transition diagram



The vulnerability is in the "Disclosure" state during the period from the time of disclosure of the vulnerability until the vulnerability is repaired.

State 3 - Exploited: This state means that the vulnerability is exploited by malicious attackers for the first time, such as a 0-day vulnerability, and the security of the system where the vulnerability is located is seriously threatened at this time. Vulnerabilities are in a critically dangerous "Exploited" state from the exploitation stage until the time of the bug fix stage.

State 4 - Patched: Refers to the first release of vulnerability patches by software developers. The vulnerability is in a "Patched" state after the network administrator fixes it.

Definition 1: Vulnerability state Vul_{state}^{i} represents that node i in the network has a vulnerability Vul, and the vulnerability is in state state, where:

$$state_{i} = \left(creation, discovery, disclosure, exploit, patched\right)$$

$$Vul_{state}^{i} = \left(Vul_{creation}^{i}, Vul_{discovery}^{i}, Vul_{disclosure}^{i}, Vul_{exploit}^{i}, Vul_{patched}^{i}\right)$$

Initially, the vulnerability starts with a state 0, when it has not yet been discovered. When it is in state 1, the vulnerability is discovered. There is no immediate risk when a white hat discovers it, and it has the potential to be exploited very quickly if a black hat discovers it. When in status 4, the vulnerability is disclosed and released with the patch, and the patch is applied to the software immediately. Therefore, the state is an absorbing state, meaning that the software is in a safe state. State 2 represents a situation where the vulnerability is disclosed but not patched. Both states 1 and 2 represent the exploitation of state 3 by exposing the system to an attacker.

Network Topology Representation

Due to the numerous vulnerability states of network nodes, the complex topology of the network is difficult to describe in real-time. For example, there are $2^{\frac{1}{2}\left(n-1\right)n^{2}}$ different combinations in the network structure of n nodes. Therefore, in the process of analysis, the formal description of the network attack and defense environment can greatly reduce the computational complexity (Zhu & He, 2021).

Definition 2: Network topology matrix. The network topology can generally be represented by a two-tuple $G = \left(V, E\right)$, where *V* represents the node set in the network $V = \left\{v_{1}, v_{2}, \ldots, v_{n}\right\}$, E represents the edge set $E = \left\{e_{1}, e_{2}, \ldots, e_{m}\right\}$, and the network topology can be represented by an n-order square matrix $A\left[n \times n\right]$:

$$A_{[i,j]} = \begin{cases} 1\left\langle v_{i}, v_{j}\right\rangle \in E \\ 0\left\langle v_{i}, v_{j}\right\rangle \notin E \end{cases}$$

Definition 3: The network vulnerability:

$$Vulnerability\left(v\right) = \left(Vul_{1}, Vul_{2}, Vul_{3}, Vul_{4}, \ldots, Vul_{n}\right)$$

represents that node *v* in the network has vulnerability $Vul_{1}, Vul_{2}, Vul_{3}, Vul_{4}, \ldots\ldots, Vul_{n}$.

Network Node Representation Method Based on Vulnerability Life Cycle

Based on the fact that most of the current network key nodes are identified by static methods, the network key node representation method based on vulnerability life cycle is proposed and used by considering the distribution law of node vulnerability utilization probability in the time dimension of vulnerability production.

Definition 4: Representation method of node v based on vulnerability life cycle. v_{state}^{vul} represents that there is vulnerability vul on node v, and the vulnerability is in state:

$$v_{state}^{vul} = \left(v_{creation}^{vul}, Vul_{discovery}^{vul}, v_{disclosure}^{vul}, v_{exploit}^{vul}, v_{patched}^{vul}\right)$$

For example, in an office network, the HP Officejet Pro printer on the intranet has a vulnerability CVE-2017-2741 code arbitrary execution vulnerability. At this time, the network administrator does not find and update the patch in time. In this case, the network node can be expressed as $v_{discovery}^{CVE-2017-2741}$.

CALCULATION OF VULNERABILITY LIFECYCLE RISK BASED ON CVSS SCORE

CVSS Vulnerability Scoring System

The common vulnerability scoring system, referred to as CVSS, is an open industry standard. It is designed to evaluate the severity of vulnerabilities (CVSS, 2018). As an open framework, it primarily provides users with standardized vulnerability scores and vulnerability risk severity levels. It can convert numerical scores into qualitative representations to help organizations properly assess and prioritize their security management operations (Jaquith, 2007). CVSS mainly evaluates the underlying characteristics of the vulnerabilities and their potential impact. Each aspect is composed of multiple constituents. The CVSS's score for vulnerabilities is mainly determined by quantifying the aforementioned components (Qiuyan & Yuqing, 2018). The calculation yields a value between 0 and 10 that represents the evaluation result of the vulnerability. The larger the value, the more detrimental the vulnerability. CVSS unifies the vulnerability assessment standards and makes various vulnerability assessment methods compatible with one another. It has become the standard of the network security industry (Ruyi, 2021). In this section, the authors only consider cases in which the vulnerability can be exploited after being disclosed in the NVD vulnerability library, and they disregard the timing uncertainty of 0-day exploits.

Vulnerability Life Cycle Risk Assessment Model

Definition 5: Value at risk (VaR) is usually expressed as the probability that an asset will suffer from a given negative impact event (Verdon & McGraw, 2004) or the likelihood of damage (C. Pfleeger & S. Pfleeger, 2003). Formally, risk can be expressed broadly by the following expression (NIST, 2010):

$$Risk = Likelihood\ \ of\ \ an\ \ adverse\ \ event \times Impact\ \ of\ \ the\ \ adverse\ \ event$$

This section defines risk from the perspective of the vulnerability life cycle, taking into account the probability of exploiting a vulnerability in the system and the impact of exploitation.

Definition 6: Based on the vulnerability lifecycle risk value, $Risk\left(v_{state,t}^{vul}\right)$ indicates that at time t, there is a vulnerability vul on node v, and the vulnerability is in the state state. It can be expressed as:

$$Risk\left(v_{state,t}^{vul}\right) = Attack\left(v, vul, t\right) \times Impact\left(vul\right)$$

where, $Attack\left(vul, t\right)$ represents the possibility that node v will be attacked after vulnerability vul is released t days later, and $Impact\left(vul\right)$ represents the impact of vulnerability vul being attacked.

Definition 7: State occurrence probability M indicates the probability of vulnerability in five different states described in Definition 1:

$$M\left(t\right) = \left(m_0\left(t\right), m_1\left(t\right), m_2\left(t\right), m_3\left(t\right), m_4\left(t\right)\right)$$

where:

$$i \in \left(creation, discovery, disclosure, exploit, patched\right)$$

$m_i\left(t\right)$ represents the probability that the vulnerability is in state i at time t.

In this paper, the authors only examine the probability of the vulnerability being exploited. As shown in Figure 1, according to Definition 6, the probability of the vulnerability vul on node v being attacked at time t can be expressed as:

$$Attack\left(vul, t\right) = \prod_{i=0}^{2} m_i\left(t\right)$$

State 0: In the process of vulnerability creation, because the modern software development and delivery process is extremely complex, and because of the involvement of the compilation environment and various class libraries, open-source code, public development kits, middleware, etc., the software delivery process involves complex support relationships. Meanwhile, the lack of transparency of software components and dependencies and the lack of security verification mechanism support make it difficult to trace the impact of software defects and hidden threats. Therefore, in this paper, the authors assume that all software has exploitable vulnerabilities in the development process, that is:

$$m_0\left(t\right) = 1$$

State 1: In the process of vulnerability discovery, with the current development of network security technology, vulnerability mining technology, methods, and tools are highly advanced. That is to say, it is believed that network security personnel can find vulnerabilities in software or programs at the beginning of use:

$$m_1\left(t\right)=1$$

State 2: After the time when the vulnerability is disclosed, the longer the vulnerability is exposed, the greater the possibility of the vulnerability is being successfully exploited. The time exploitability (TE) of a vulnerability is determined by the possibility of exploiting the vulnerability code, the degree of patch repairment, and the values satisfy the Pareto and Weibull distributions (Frei, 2006) namely:

$$m_2\left(t\right)=\left(1-\left(\frac{l}{t}\right)^r\right)\left(1-exp\left(-\frac{t}{q}\right)^u\right)$$

where, *l* and *r* represent Pareto distribution coefficients, with values of 0.260 and 0.00161, respectively; q and u represent Weibull distribution coefficients, with values of 0.209 and 4.04, respectively; t represents the number of days from the time of the disclosure date up until the evaluation date of a vulnerability.

Through the above analysis, the possibility of node v being attacked t days after the vulnerability vul is released, i.e., $Attack\left(vul,t\right)$ can be expressed as:

$$Attack\left(vul,t\right)=\left(1-\left(\frac{l}{t}\right)^r\right)\left(1-exp\left(-\frac{t}{q}\right)^u\right) \tag{1}$$

where $Impact\left(vul\right)$ represents the impact of the vulnerability on the system after being exploited, which can be expressed according to the impact factor in the CVSS basic metric equation, which includes three parts: confidentiality, integrity, and availability (2018):

$$ISS=1-\left[\left(1-Confidentiality\right)\left(1-Integrity\right)\left(1-Availability\right)\right]$$

$$Impact\left(vul\right)=6.42\times ISS \tag{2}$$

In this paper, the authors only consider the probability of the vulnerability being exploited, i.e., $state=disclosure$:

$$Risk\left(v_{disclosure,t}^{vul}\right)=Attack\left(vul,t\right)Impact\left(vul\right)=6.42\times ISS\left(1-\left(\frac{l}{t}\right)^r\right)\left(1-exp\left(-\frac{t}{q}\right)^u\right)$$

which is abbreviated as:

$$R\left(vul,t\right)=6.42\times ISS\left(1-\left(\frac{l}{t}\right)^r\right)\left(1-exp\left(-\frac{t}{q}\right)^u\right) \tag{3}$$

KEY NODE IDENTIFICATION METHOD BASED ON VULNERABILITY LIFE CYCLE AND IMPORTANCE OF NETWORK TOPOLOGY

Node Topology Significance Calculation

Effectively identifying key nodes in the process of network attack and defense has important theoretical and practical significance for improving the survival capacity of the network and maintaining the stability of the network structure.

In this section, the local and the whole are examined uniformly, and the method of combining the degree of centrality and the proximity centrality of network nodes is adopted. “Centrality” is an index for determining the significance of nodes in the network, and it is the quantification of node significance (Yang, 2013).

The degree of centrality is considered locally. The greater the node degree of a node, the higher the degree of centrality of the node, and the more important the node is in the network. In a network topology, centrality represents the relationship between each host and other hosts (David & Song, 2012):

$$DC\left(i\right)=\frac{\sum_{j\in G}a_{ij}}{n-1}$$

where $\sum_{j\in G}a_{ij}$ is the number of nodes or edges connected to node i in the network, and *n* is the number of nodes in the network.

Proximity centrality start is used to find nodes that can efficiently spread information through the network topology. Based on its calculation of the shortest path between all node pairs, the proximity centrality algorithm also calculates the sum of the distances from it to other nodes, and then calculates the reciprocal of the obtained sum. This method only uses local information to describe the significance of nodes, so it is suitable for the identification of key nodes in large-scale networks (Wenlan et al., 2017):

$$d_i=\frac{1}{n-1}\sum_{j=1}^{n}d_{ij}$$

$$CC\left(i\right)=\frac{1}{d_i}$$

where, d_i represents the average distance between node i and other points.

Definition 8: The importance of network node topology $Key\left(i\right)$ is composed of the degree of centrality and proximity centrality of nodes in the network:

$$Key\left(i\right)=\frac{DC\left(i\right)+CC\left(i\right)}{2} \tag{4}$$

Key Node Identification Method Based on Vulnerability Life Cycle and Network Structure

Definition 9: The key node $node$ based on the vulnerability life cycle satisfies:

$$node^{*} = \max_{node \in V}\left(\max_{vul \in Vulnerability} f\left(v, vul, t\right)\right)$$

where $f\left(v, vul, t\right)$ represents the risk value of the node v based on the vulnerability *vul* at time *t*, including the risk value of the vulnerability being attacked on the node and the importance based on the network topology structure, vulnerability is the set of all vulnerabilities of the node *v*, and *V* represents the set of all nodes in the network:

$$f\left(v, vul, t\right) = Key\left(v\right) R\left(vul, t\right)$$

where $Key\left(v\right)$ is the significance of the node based on the network structure, and $R\left(vul, t\right)$ is the risk value of the network node with vulnerability vul at time *t*.

EXPERIMENT AND ANALYSIS

In order to verify the effectiveness and applicability of the method in this paper, the authors built an actual network, and designed an experiment to test and analyze the method.

Network Environment

The experimental topology is shown in Figure 2. The network includes an attacking host, a firewall, a business host, a printer, and three servers.

First, use the Nmap tool to detect the above network topology and obtain the connectivity between the hosts, as shown in Table 1.

Host Vulnerability Life Risk Value Calculation

Use the vulnerability scanning tool Nessus to scan each network segment, and obtain the vulnerability information contained in each host by querying NVD, as shown in Table 2.

The impact value of each vulnerability is calculated according to formula (2), as shown in Table 3.

According to formula (1), Attack(vul,t) represents the possibility that the vulnerability vul will be attacked after t days of publication, and the time-varying curve is shown in Figure 3. With the increase of time, the possibility of the vulnerability being attacked gradually increases.

According to formula (3), the life cycle risk value of each vulnerability that changes with time can be obtained, as shown in Table 4.

From the perspective of time, it is assumed that May 20, May 30, June 20, June 30, and July 10 are the initial times, and the time until the vulnerability was announced is shown in Table 5.

According to the change of time t in Table 4, the exploitability of the vulnerability over time is shown in Figure 4. The horizontal axis represents the current time, and the vertical axis represents the time-varying vulnerability time exploitability. It can be seen from the figure that only the vulnerability vul_3 was released on May 20 at the initial moment, and then the vulnerabilities vul_1, vul_2, vul_4, *and* vul_5 were released one after another. As can be seen from Figure 4, at the same:

$$Attack\left(vul_3\right) \geq Attack\left(vul_2\right) \geq Attack\left(vul_4\right) \geq Attack\left(vul_5\right) \geq Attack\left(vul_1\right)$$

According to the data in Figure 4, using the algorithm in formula (3), the authors calculate the vulnerability life cycle risk value as shown in Figure 5. As can be seen from Figure 5, the risk value of each vulnerability increases as the time it is published increases.

Figure 2. Experimental Network Topology

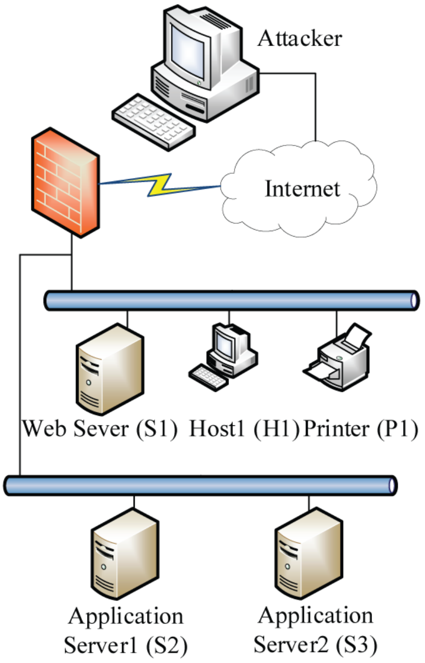


Table 1. Network Connection Relationship Between Hosts

Host	S1	S2	S3	H1	P1
S1	✓	--	--	✓	✓
S2	--	✓	✓	--	--
S3	--	✓	✓	--	--
H1	✓	--	--	✓	✓
P1	✓	--	--	✓	✓

Note: ▯ means host can be connected, -- means host cannot be connected.

Table 2. Host Configuration and Vulnerability Information Table

No.	Host Configuration	CVE	Release Time
S1	Web Sever Windows Sever 2008	CVE-2019-0620(vul_1)	June 12, 2019
S2	Application Server1 Windows Sever 2012	CVE-2019-7060(vul_2)	May 24, 2019
S3	Application Server2 Windows Sever 2012	CVE-2019-0708(vul_3)	May 24, 2019
H1	Host1 Red Hat Linux 8.0	CVE-2019-12381(vul_4)	May 27, 2019
P1	Printer1 HP LaserJet Managed	CVE-2019-6321(vul_5)	May 27, 2019

Table 3. Correspondence table of vulnerabilities and their impacts

No.	CVE	Impact
vul_1	CVE-2019-0620	6.3348
vul_2	CVE-2019-7060	6.3348
vul_3	CVE-2019-0708	6.3348
vul_4	CVE-2019-12381	1.8248
vul_5	CVE-2019-6321	6.3348

Figure 3. Vulnerability Time Exploitability Change Table

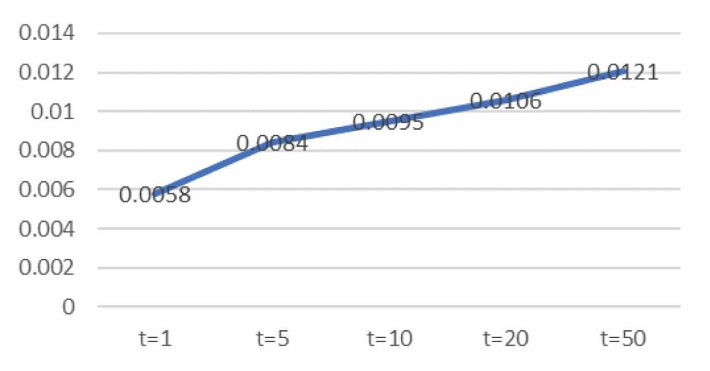


Table 4. Correspondence Table of Vulnerabilities and Their Impact Over Time

No.	t=1	t=5	t=10	t=20	t=50
vul_1	0.0367	0.0532	0.0602	0.0671	0.0767
vul_2	0.0367	0.0532	0.0602	0.0671	0.0767
vul_3	0.0367	0.0532	0.0602	0.0671	0.0767
vul_4	0.0105	0.0153	0.0173	0.0193	0.0221
vul_5	0.0367	0.0532	0.0602	0.0671	0.0767

Table 5. Corresponding Table of Vulnerability Announcement Duration

No.	May 20	May 30	June 20	June 30	July 10
vul_1	0	0	8	18	28
vul_2	0	6	27	37	47
vul_3	4	14	35	45	55
vul_4	0	3	24	34	44
vul_5	0	1	22	32	42

Figure 4. Exploitability of Vulnerabilities by Date

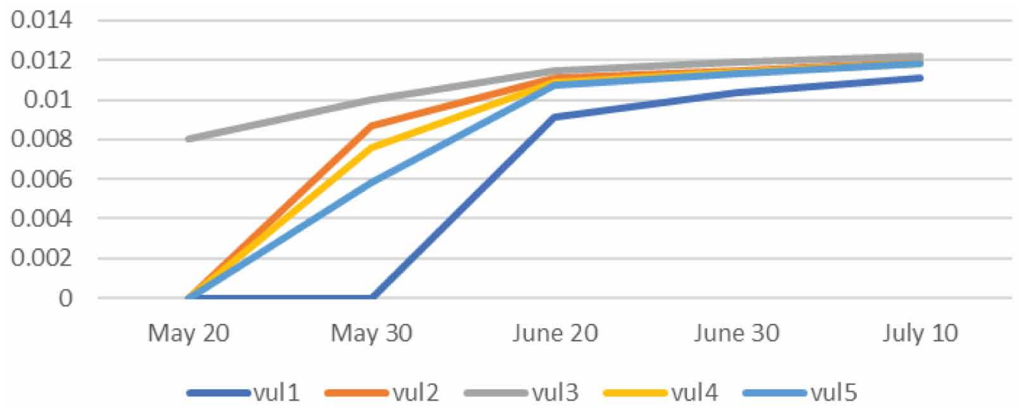
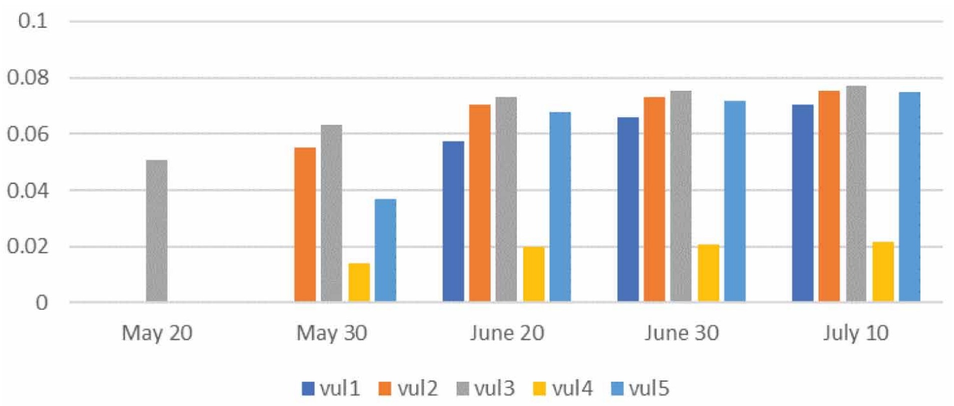


Figure 5. Histogram Corresponding to the Risk Value of the Vulnerability Life Cycle



Identification of Key Network Nodes Based on Vulnerability Life Cycle and the Importance of Network Topology

According to the method in section discussing node topology significance calculation, the importance of the topology structure of each host is obtained, and the results are shown in Table 6.

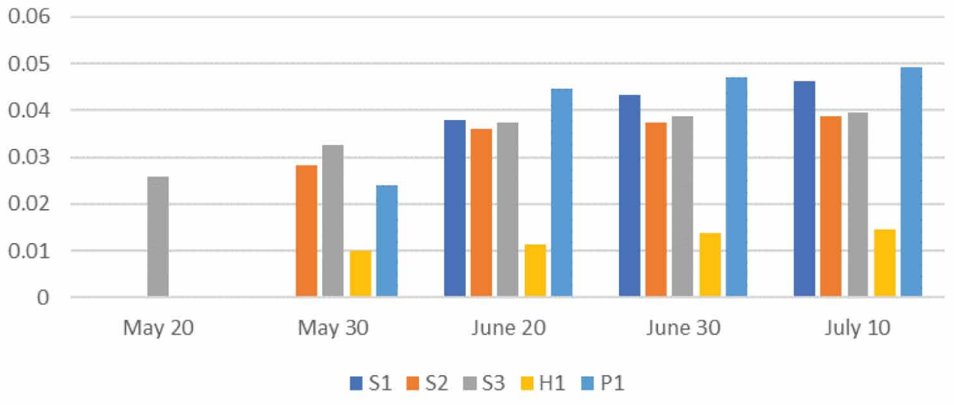
Figure 6 shows the risk value of node v based on vulnerability vul at different times, and on the initial time May 20, node S3 has the highest vulnerability risk value. The risk values of S2, S3, H1, and P1 gradually appeared and increased rapidly. By June 20, the risk value of node P1 began to be greater than S3 and surpassed other nodes. At the same time, after June 20, the risk value of P1 was far ahead. For other nodes, $R\left(S1\right) > R\left(S3\right) > R\left(S2\right) > R\left(H1\right)$.

Through the above analysis, after the vulnerability vul_5 was disclosed, P1 node is the key node of the above experimental network. This is because the vulnerability vul_5 on the P1 node was disclosed late, and the network administrator may neglect to fix it. At the same time, according to the data released by CVSS, the vul_5 vulnerability the influence range is wide and the P1 node is also the key node of the network topology, which verifies the validity and accuracy of the method in this paper. The calculation results provide a data reference for the risk control of network nodes.

Table 6. Importance of Host Topology

Host	Degree Centrality $BC(i)$	Closeness to Centrality $CC(i)$	Importance of Network Nodes $Key(i)$
S1	3/5	5/7	0.6571
S2	2/5	5/8	0.5125
S3	2/5	5/8	0.5125
H1	3/5	5/7	0.6571
P1	3/5	5/7	0.6571

Figure 6. Histogram Corresponding to Node Risk Value



RELATED WORK

The comparison between the method in this paper and the existing method is shown in Table 7. It can be seen from the table that the (Hongyu et al., 2022) comprehensively considers the importance of the host through the impact value of the host vulnerability, the importance of the host, and the probability of the host attack, but there is no analysis of the life cycle of the vulnerability:

Cox (2008) divides the vulnerability level by analyzing the possibility of vulnerability threat and exploitation, and uses the risk matrix to calculate the security risk level of the system, but the level

Table 7. Performance Comparison Between the Proposed Method and Other Methods

Features	Hongyu, 2022	Cox. 2008	Mingqiu, 2011	Our Method
Node Vulnerability	✓	✓	✓	✓
Vulnerability Lifecycle Analysis	×	×	✓	✓
Topology Analysis	✓	×	×	✓
Time Dimension Quantification	✓	×	✓	✓
Risk Quantification	✓	×	✓	✓

classification is highly subjective; Mingqiu et al. (2011) quantifies the vulnerability life cycle Period, the Mamdani model is used, and the fuzzy inference method is used to calculate the vulnerability security risk value, but the analysis of the network topology is lacking.

Compared with the related work, the method for identification key network nodes based on vulnerability life cycle and importance of network topology proposed in this paper has the following characteristics:

- The authors consider both the network topology and the vulnerability attributes of network nodes in the process of identifying key network nodes.
- The authors quantify the node risk value in the time dimension, combined with the evolution process of the vulnerability life cycle.

CONCLUSION AND FUTURE WORK

This paper proposed the identification method of key network nodes based on the vulnerability life cycle and the significance of network topology. Firstly, due to the absence of a relationship between network vulnerability and time in the existing network node description methods, the key nodes of the network were formally characterized in addition to the vulnerability life cycle. Meanwhile, a calculation method for the risk value of the vulnerability life cycle was proposed based on CVSS score. The key node identification method based on vulnerability life cycle and network structure was examined in conjunction with the significance of the host topology. Finally, by implementing the research results, the security of a genuine network environment was evaluated, the target network was rapidly modeled and quantitatively calculated, and the key nodes were identified.

Future research will focus on the following two points. Firstly, the authors will adopt observation and fields of fire, avenues of approach, key and decisive terrain, obstacles, and cover and concealment (OCOKA) to analyze the attributes and positions of each node in the network, and propose a cyberspace terrain analysis framework C-OCOKA based on offensive and defensive games. Secondly, the authors should use deep learning methods for complex networks to identify key nodes in the node topology, thereby reducing the computational complexity of the algorithm and enhancing the computational efficiency.

AUTHOR NOTE

Correspondence concerning this article should be addressed to Yuwen Zhu, State Key Laboratory of Mathematical Engineering and Advanced Computing, China.

COMPETING INTEREST STATEMENT

The authors declare that there are no conflicts of interest regarding the publication of this paper.

DATA AVAILABILITY

The data used to support the findings of this study are available from the corresponding author upon request.

FUNDING STATEMENT

This work was supported by the National Key R&D Program of China No. 2019QY1302 and the National Key R&D Program of China No. 2019QY1305.

REFERENCES

Arbaugh, W. A., Fithen, W. L., & Mchugh, J. (2000). Windows of vulnerability: A case study analysis. *Computer*, *33*(12), 52–59. doi:10.1109/2.889093

Cox, L. A. Jr. (2008). Some limitations of ”Risk=Threat×Vulnerability×Consequence” for risk analysis of terrorist attacks. *Risk Analysis*. *International Journal (Toronto, Ont.)*, *28*(6), 1749–1761. PMID:19000071

CVSS. (2018). *Common vulnerability scoring system v3.0: specification document*. CVSS. https://www.first.org/cvss/v3.0/cvss-v30-specification_v1.9.pdf

Dangalchev, C. (2006). Residual closeness in networks. *Physica A*, *2*(365), 556–564. doi:10.1016/j.physa.2005.12.020

David, N., & Yang, S. (2012). *Social Network Analysis* (2nd ed.). Shanghai People’s Publishing House.

Freeman, L. C. (1977). A set of measures of centrality based on betweenness. *Sociometry*, *40*(1), 35–41. doi:10.2307/3033543

Freeman, L. C. (1978). Centrality in social networks conceptual clarification. *Social Networks*, *1*(3), 215–239. doi:10.1016/0378-8733(78)90021-7

Frei, S. (2009). *Security econometrics: The dynamics of (in) security* (Vol. 93). ETH Zurich.

Frei, S., May, M., & Fiedler, U. (2006, September). Large-scale vulnerability analysis. In *Proceedings of the 2006 SIGCOMM workshop on Large-scale attack defense*. (pp. 131-138). ACM. doi:10.1145/1162666.1162671

Hongyu, Y., Haihang, Y., & Liang, Z. (2022). Host Security Assessment Method Based on Attack Graph. *Journal of Communication*, *43*(2), 89–99.

Jaquith, A. (2007). *Security metrics replacing fear, uncertainty, and doubt*. Pearson Education.

Joh, H., & Malaiya, Y. K. (2010). A framework for software security risk evaluation using the vulnerability lifecycle and CVSS metrics. CVSS metrics. In *Proc International Workshop on Risk And Trust in Extended Enterprises* (pp. 430-434). Research Gate.

Kaaniche, M., Marconato, G., & Nicomette, V. (2013). Security-related vulnerability life cycle analysis. In *2012 7th International Conference on Risks and Security of Internet and Systems (CRiSIS)* (pp. 1-8). IEEE.

Kaixuan, D., Hongchang, C., & Ruiyang, H. (2006). Method of node important ranking based on improved K-shell. *Computer Applied Research, 10*(3), 3017-3019.

Kitsak, M., Gallos, L., Havlin, S., Liljeros, F., Muchnik, L., Stanley, E., & Makse, H. (2010). Identification of influential spreaders in complex networks. *Nature Physics*, *6*(11), 888–893. doi:10.1038/nphys1746

Mingqiu, S., Leilei, W., & Yu, B. (2011). Research on time risk of security vulnerabilities based on life cycle theory. *Computer Engineering*, *37*(1), 131–133.

Newman, M. E. (2006). Finding community structure in networks using the eigenvectors of matrices. *Physical Review. E*, *74*(3), 036104. doi:10.1103/PhysRevE.74.036104 PMID:17025705

Pfleeger, C. P. & Pfleeger. (2003). *Security in Computing* (3rd ed.). Prentice Hall PTR.

Qiuyan, W., & Yuqing, Z. (2018). A general vulnerability rating method. *Computer Engineering*, *34*(19), 133–136.

Ruyi, W. (2021). *Research on vulnerability detection and security assessment technology based on association analysis*. Northwest University.

SP800, N. I. S. T. (n.d.). Risk management guide for information technology systems. *National Institute of Standards and Technology Special Publication*, 800-30.

Salavati, C., Abdollahpouri, A., & Manbari, Z. (2018). A novel fast centrality measure based on local structure of the network. *Physica A*, *469*, 635–653. doi:10.1016/j.physa.2017.12.087

Verdon, D., & McGraw, G. (2004). Risk analysis in software design. *IEEE Security and Privacy*, *2*(4), 79–84. doi:10.1109/MSP.2004.55

Wenlan, L., Ye, W., Li, L., & Caixue, Z. (2017). Research on network key node identification based on multi-attribute decision-making. *Intelligence Theory and Practice*, *40*(9), 95–100.

Yang, S. (2013). Networks An Introduction by MEJ Newman. Oxford University Press.

Zhu, Y. Yu., Yu, L., He, H., & Meng, Y. (2021). L., He, H., & Meng, Y. (2021). A defense strategy selection method based on the cyberspace wargame model. *Security and Communication Networks*, *2021*, 1–12. doi:10.1155/2021/4292670