IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 16, 2020, accepted August 5, 2020, date of publication August 10, 2020, date of current version August 21, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.3015551*

# A Vulnerability Risk Assessment Method Based on Heterogeneous Information Network

**WENRUI WANG, FAN SHI, MIN ZHANG, CHENGXI XU, AND JINGHUA ZHENG**
College of Electronic Engineering, National University of Defense Technology, Hefei 230037, China
Corresponding author: Min Zhang (dyzhangmin@163.com)

**ABSTRACT** Due to the increasing number of network security vulnerabilities, vulnerability risk assessment must be performed to prioritize the repair of high-risk vulnerabilities. Traditional vulnerability risk assessment is based primarily on the Common Vulnerability Scoring Systems (CVSS) and attack graphs. Nevertheless, the CVSS metrics ignore the impact of the vulnerability on the specific network, which accounts that the identical vulnerability exists in different network environments is assigned repeated values. Additionally, the attack graphs still suffer from scalability and readability issues. To solve the above problems, a ranking method based on the heterogeneous information network is innovatively proposed to assess the vulnerability risk in a specific network. It considers the exploitability of a vulnerability, the impact of a vulnerability on the network components, and the importance of the vulnerable components. First, a heterogeneous information network containing vulnerability and host and the relationships between host and host is constructed to compute the risk score for each vulnerability and implement the ranking process. Second, a model extension method is proposed to adapt to situations in which additional factors related to vulnerability risk assessment need to be considered. Finally, we explore two case studies to compare the proposed method with CVSS and attack graph-based methods. The simulation results show that the proposed method can accurately assess the risk of vulnerabilities in a specific network environment and that it has a lower computational complexity than other methods.

**INDEX TERMS** Common vulnerability scoring systems (CVSS), vulnerability, risk assessment, information fusion, heterogeneous information network.

## I. INTRODUCTION

With the rapid development of computer networks, the scale of networks is increasing, and a variety of network attacks and vulnerabilities have become increasingly common. The Community Emergency Response Team (CERT) found that the number of global network security incidents increased sharply from 2003 to 2019 [1]. It is difficult for network administrators to ensure that every vulnerability is fixed for each host. Notably, the process of remediating vulnerabilities can result in a loss of service quality, decreased performance, and it involves a high level of human effort. Therefore, vulnerability risk assessment is performed to select the vulnerability priorities with the highest corresponding risk for repair, which is conducive to effective network security reinforcement [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Luis Javier Garcia Villalba.

The traditional approach of assessing vulnerability risk is mainly through CVSS metrics [3], attack graphs, etc. CVSS metrics can provide quantitative risk scores for vulnerabilities and methods for eliminating vulnerabilities with the highest risk. However, there are several deficiencies in CVSS metrics. As a quantitative scoring system, objectivity and dispersion should be considered. Objectivity reflects how well the results of an assessment reproduce the nature of practical scenarios [4], while dispersion considers the degree of difference and distribution of the results. For example, the Access Vector is a submetric of CVSS that has three possible values: Local_(L), Adjacent Network_(A), and Network_(N). One survey reported that for all known vulnerabilities, the Network_(N) value accounts for 85.69% of the three possible values [4], which can lead to a situation in which several vulnerabilities are assigned the same risk CVSS score in a network. However, that is an unreasonable result, as that will not able to distinguish which vulnerability possess a higher

risk score. Meanwhile, CVSS metrics do not consider the specific network environment. Therefore, the CVSS scores does not objectively reflect the vulnerability risk in diverse network environments.

The attack graph-based method represents attack scenarios by showing possible attack paths from the attackers to the target. Many researchers have assessed network risks and modeled network threats based on attack graphs [5], [6]. The analysis of an attack graph will facilitate identifying critical exploitations of vulnerabilities, assets, and vulnerable configurations. This will help administrators of the network strengthen network security. However, there are still several problems with the attack graph-based approach. Attack graph generation can involve up to polynomial complexity, and the evaluation and analysis of attack graphs to determine all possible attack paths suffer from scalability issues [7]. Meanwhile, a large-scale attack graph is complicated, making it difficult for humans to digest all the dependency relations and specify the key problems in a limited amount of time. Moreover, the attack graph model needs improvement because it currently only considers the relationships among vulnerabilities; it cannot model high-level attack Tactics, Techniques, and Procedures (TTPs) [8].

Given the problems mentioned above, this paper proposes a ranking method based on the Heterogeneous Information Network (HIN) [9] for vulnerability risk assessment. First, the proposed approach considers both the exploitability and the impact of vulnerabilities in a specific network; this approach avoids assigning the same risk score for the same vulnerability in different networks. Second, utilizing a heterogeneous information network-based method can model multiple types of objects and the relationships among them as well as different types of semantic information [9]. Additionally, such models possess sufficient representation ability and extensibility when more factors need to be considered due to the changes in the network environment. Finally, The proposed approach utilizes the graph-based ranking method, which promises an acceptable computational complexity.

The main contributions of this paper are summarized as follows:

1) **A comprehensive semantic represent model.** The vulnerability risk assessment is conducted from a new perspective based on the heterogeneous information network that can fuse more information and introduce higher-level semantics. The proposed approach may help to model the higher-level threat such as the Tactics, Techniques, and Procedures. Furthermore, our work will facilitate the ontology-based quantitative risk assessment.
2) **HIN-based ranking method for vulnerability risk assessment.** This paper proposes a vulnerability risk assessment method based on the heterogeneous information network for a specific network environment; First, a heterogeneous information network containing vulnerabilities, hosts and the relationships between hosts is constructed. Then, a ranking algorithm for vulnerability risk ranking is designed that considers the exploitability of a vulnerability, the impact of a vulnerability on the network components and the importance of vulnerable components. Finally, to accommodate changes in factors that need to be considered, the solution of extending the model and the calculation method of vulnerability risk scores are proposed.
3) **Practicality comparison.** To demonstrate the advantages and disadvantages of the proposed approach, two practical case studies (include three comparisons) are conducted and the results of the proposed model are compared with the CVSS metric-based method and attack graph-based methods. We constructed two small enterprise network environments to test the methods.The source code and input file for the attack graph generation tool are available online. The results show that the proposed method can produce sufficiently precise results, with an acceptable level of computational complexity.

The remainder of this paper is organized as follows:

In Section 2, the related work is introduced. In Section 3, first, we introduce the system model, which contains a brief review of the heterogeneous information network, CVSS metrics. Second, the model we proposed, the computing method, and the solution of model extension are introduced. The two case studies (including three comparisons) are described in Section 4 for the constructed network environment, and the results are presented. Finally, Section 5 concludes the paper.

## II. RELATED WORKS

### A. THE CVSS-BASED APPROACHES

Many studies have been conducted to improve the usability of CVSS by optimizing objectivity and dispersion. To improve the objectivity of CVSS, [10], [11] performed the CVSS research and proposed several attributes as novel metrics. The heterogeneity of diversity and vulnerability distributions was considered in [12]. The authors of [13] used 3000 vulnerabilities from the National Vulnerability Database (NVD) [14] to validate the objectivity of CVSS. The approach in [15] proposed combining a scoring system with the CVSS to measure the severity cost of hosts. In [16], a temporal feature was added to CVSS. The authors of [17] conducted research on the dispersion of the CVSS. In [4], a CVSS-based vulnerability scoring system was proposed to improve dispersion.

However, the above studies ignored the influences of different environments on vulnerability risk [18]. For example, the same vulnerability may have different risk levels for different devices or environments; thus, the risk level should be determined based on the importance of devices to the network, the relevant security requirements and other associated factors. Notably, [19] researched vulnerability risk in the OSs of tablets and smartphones. The risk assessment formula for the CVSS was optimized to adapt to vulnerability

risk assessments for IoT systems [20]. A vulnerability risk assessment method based on the CVSS was proposed to assess the vulnerability risk of a cloud service [21]. The above studies targeted vulnerability risk assessments in specific environments. Nevertheless, some deficiencies remain in the above studies. When the factors that should be considered change, the above models cannot be extended to adapt to the new environment.

### B. THE ATTACK GRAPH-BASED METHODS

Attack graph-based methods are widely used in network security risk analysis, threat mitigation, decision making, etc. A review of attack graphs in cybersecurity is given in [22]. To perform network security risk analysis, reference [23] combined CVSS metrics with an attack graph to provide precise assessments of the risk of a vulnerability. The authors of [24], [25] combined a Bayesian network with the CVSS to quantify the possibility of network compromise and strengthen the network. In [26], the attack costs and benefits were quantified and integrated with different metrics to evaluate countermeasures for security issues based on an attack graph. The authors of [2] integrated the idea of dynamic defense into attack graph analysis and proposed a probability-based approach to perform a quantitative network risk assessment. In [27], [28], PageRank [29] was utilized to evaluate the importance of nodes or states in an attack graph, which will improve the readability of the attack graph. In [5], the maximum reachable possibility of nodes based on the graph-based inference algorithm [30] were evaluated. The above studies conducted detailed analyses based on attack graphs. Nevertheless, the generation and analysis of attack graphs still suffer from scalability issues. Meanwhile, a lack of standards, prescriptive methodologies and common approaches in terms of visual syntax lead to another important issue, which is the lack of the sufficient readability [22].

## III. VULNERABILITY RISK ASSESSMENT MODEL

In this section, we first briefly review the heterogeneous information network and construct the Device-Vulnerability bi-type graph. Second, we propose weighted ranking rules based on the heterogeneous information network for vulnerability risk assessment and obtain the corresponding vulnerability risk score and ranking. Finally, we consider the requirement of model extension and propose a solution.

### A. PRELIMINARIES

A heterogeneous information network contains multiple types of objects, the relationships between objects and different semantic information.

*Definition 1 (Information Network [31]):* An information network is a directed graph $G = (V, E)$ with an object-type mapping function $\tau : V \rightarrow A$ and a link-type mapping function $\phi : E \rightarrow R$, where each object $v \in V$ is associated with a particular object type $\phi(v) \in A$ and each link $e \in E$ is associated with a specific relation $\phi(e) \in R$. If the number of object types $|A| > 1$ or the number of relationship types $|R| > 1$, then the information network is a heterogeneous information network; otherwise, the network is a homogeneous information network.

For a given complex heterogeneous information network, a meta-description must be provided to fully understand the object types and link types in the network. Therefore, a network pattern is defined that describes the structure of the network.

*Definition 2 (Network Schema [31]:)* A network schema is denoted as $T_G = (A, R)$, which is a meta-template for an information network. A schema includes object type mapping $\phi(v) \in A$ and link mapping $\phi(e) \in R$. A directed graph is defined based on object type $A$ and link type $R$. Relation $R$ maps from type $A$ to type $B$, denoted as $A \xrightarrow{R} B$. $A$ and $B$ are the source type and target type, denoted as $R.S$ and $R.T$, respectively. The inverse relationship $R^{-1}$ is called $A \xrightarrow{R^{-1}} B$.

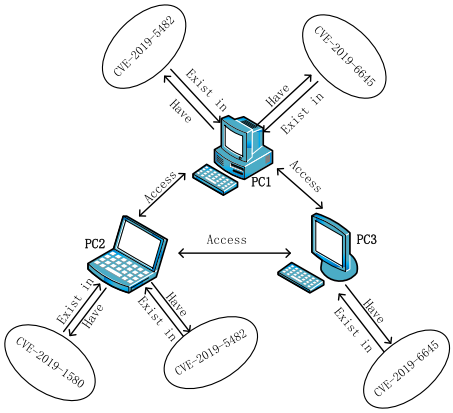



**FIGURE 1.** A simple heterogeneous information network.

*Example 1:* Figure 1 is an information network that contains two type of objects: the Device (PC1, PC2, PC3) and the Vulnerability (CVE-2019-5482, CVE-2019-6645, CVE-2019-1580). There are multiple relationship types between different types of objects. For example, the relationship between Device and Vulnerability is that a vulnerability exists in the device. The relationship between a device and another device is that one device accesses the other device. A device can be regarded as a PC, Server, etc. The numbers of object types and connection types in the figure are greater than one, so this is a typical **heterogeneous information network**. For the heterogeneous information network in Figure 1, Figure 2 shows the network schema.

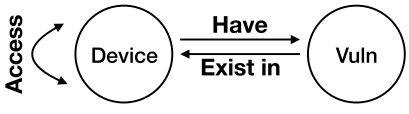



**FIGURE 2.** The corresponding network schema.

*Definition 3 (CVSS [3]):* The Common Vulnerability Scoring System (CVSS) [3] consists of three measurement

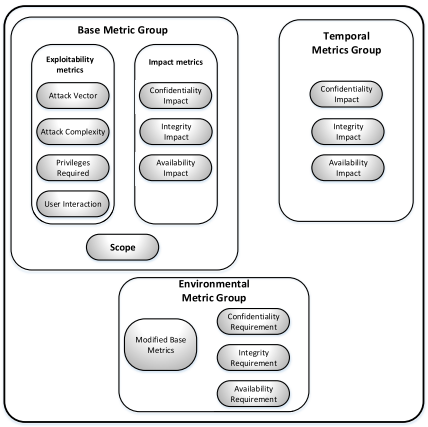



**FIGURE 3.** CVSS v3,0 metric groups.

groups: Base, Temporal, and Environmental (v3.0). The details are shown in Figure 3. The Base group contains three main metrics: Exploitability, Scope, and Impact metrics. The Base group metrics reflect the severity of a vulnerability based on its intrinsic characteristics, which remain constant over time and assume a reasonable worst-case impact across different deployment environments. This paper primarily uses the Exploitability and Impact metrics of the Base group.

Exploitability Score (ES): The exploitability index reflects the characteristics of vulnerable entities which are called vulnerable components. Each of the Exploitability metrics is scored relative to a vulnerable component and reflects the vulnerability attributes that lead to a successful attack. The availability metrics mainly include Attack Vector (AV), Attack Complexity (AC), Privileges Required (PR), and User Interaction (UI).

Impact Score (IS): The impact metrics are scored according to the component that suffers the worst outcome that is most directly and predictably associated with a successful attack. Impact metrics include Confidentiality Impact (C), Integrity Impact (I), and Availability Impact (A).

### B. VULNERABILITY RISK ASSESSMENT BASED ON THE HETEROGENEOUS INFORMATION NETWORK

In a specific network environment, the risk score of a vulnerability is related not only to its own attributes, such as exploitability and component impact, but also to the network environment in which the vulnerability is located. Vulnerability risk levels should vary among different network environments. Therefore, we recalculated the IS and combined it with the ES to assess the risk of a vulnerability.

The PageRank [29] gives the statement: ''a page has a high rank if the sum of the ranks of the backlinks is high''. Inspired by this statement, we considered this situation to be similar to that of the vulnerability on a host, i.e. The host accrues a higher risk score when it is accessible by many high-risk hosts. This is reasonable because when intruders who successfully penetrate one host subsequently use it as a springboard; thus, all hosts that can be accessed by the compromised host in some way are themselves more likely to be compromised. Therefore, the risk score of this host should be higher. The situation mentioned above involves only the host and the accessible relationship, i.e. one type of object (host) and relationship (access/accessed).

**TABLE 1.** Notations and descriptions.

| Notation | Description |
|---|---|
| $ES(i)$ | The Exploitability Score of vulnerability $i$ |
| $PN(i,j)$ | The number of ports that device $i$ can access device $j$ |
| $W_{DV}$ | The adjacency matrix of objects Device (D) and Vulnerability (V) |
| $W_{VD}$ | The transpose matrix of $W_{DV}$ |
| $W_{DD}$ | The adjacency matrix of objects Device (D) and Device (D) |
| $W_{DV}(i,j)$ | The matrix element in row $i$ and column $j$ of $W_{DV}$ |
| $W_{VD}(i,j)$ | The matrix element in row $i$ and column $j$ of $W_{VD}$ |
| $W(i,j)$ | The matrix element in row $i$ and column $j$ of matrix$W$ |
| $\sum W_{\bullet j}$ | The sum of column $j$ of matrix $W$ |
| $\vec{r}_V^{(k+1)}(i)$ | The risk score of the vulnerability $i$ after the $(k+1)$th iteration |
| $\vec{r}_D^{(k+1)}(i)$ | The risk score of the device $i$ after the $(k+1)$th iteration |
| $m$ | The total number of vulnerabilities in the network |
| $n$ | The total number of devices in the network |

In a heterogeneous information network, although multiple types of objects and relationships exist, a similar penetration scenario can occur. PopRank [32] proposed a framework to rank multiple objects in a heterogeneous information network based on the idea that the popularity of different types of objects affect other objects. Inspired by PopRank, we believe that the risk score should be influenced by multiple types of objects; therefore, the popularity in PopRank is regarded as the risk score in this paper. For example, in a network, we believe that a host should be considered high-risk if it exposes multiple high-risk vulnerabilities. Similarly, when a vulnerability affects multiple high-risk hosts, that vulnerability should be assigned a high risk value. We can imagine that a portion of the vertex's value (risk score) ''flows'' to its out-neighbors (whether that node represent a host or a vulnerability); at the meantime, the vertex assembles the value from its in-neighbors. Therefore, calculating the rank for the Device (D) and Vulnerability (V) respectively will reflect the risk of the host and vulnerability in a network in some extent.

As a consequence, given the heterogeneous graph shown in Figure 2, we can use the above heuristic relations to rank vulnerabilities and perform risk assessment. According to the above analysis, the risk score of each vulnerability is obtained through Formulas 1 and 2. The specific notations and their descriptions are listed in Table 1.

$$\vec{r}_D^{(k+1)}(i) = \alpha \sum_{j=1}^{m} W_{DV}(i,j) * \vec{r}_V^{(k)}(j) + (1-\alpha) \sum_{j=1}^{m} W_{DD}(i,j) * \vec{r}_D^{(k)}(j) \quad (1)$$

$$\vec{r}_V^{(k+1)}(j) = \sum_{i=1}^{n} W_{VD}(j,i) * \vec{r}_D^{(k)}(i) \quad (2)$$

The parameter $\alpha$ is used to control the weight of risk for different types of nodes, and we set it to 0.5. When a vulnerability has a high ES, it will be easy to exploit. Therefore, to assess the risk precisely, we construct the adjacency matrix $W_{DV}$ by setting the $W_{DV}(i,j)$ as $ES(j)$ if device $i$ has vulnerability $j$. Similarly, the $PN(i,j)$ is used as the weight between device $i$ and $j$.

To make the formula work, we should initialize the vector $\vec{r}_V^{\,(0)}$ and $\vec{r}_D^{\,(0)}$. In this paper, we use the the normalized IS of vulnerability to initialize the vector $\vec{r}_V^{\,(0)}$, and use $\frac{1}{n}$ to initialize the $\vec{r}_D^{\,(0)}$. Afterwards, the first iteration will work, which produce two vectors $\vec{r}_D^{\,(1)}$ and $\vec{r}_V^{\,(1)}$ respectively. Keeping the process of iteration, two sequences will be produced: $C_1 = \{\vec{r}_V^{\,(0)}, \vec{r}_V^{\,(1)}, \vec{r}_V^{\,(2)}, \vec{r}_V^{\,(3)} \cdots\cdots\}$ and $C_2 = \{\vec{r}_D^{\,(0)}, \vec{r}_D^{\,(1)}, \vec{r}_D^{\,(2)}, \vec{r}_D^{\,(3)} \cdots\cdots\}$.

According to [33], $C_1$ and $C_2$ will converge to the primary eigenvector of $(\alpha W_{VD}(I-(1-\alpha)W_{DD})^{-1}W_{DV})$ and $\alpha W_{DV}W_{VD}+(1-\alpha)W_{DD}$ respectively. The iterative method is a power approach [34] used to compute eigenvectors. Therefore, we will calculate the risk score with an iterative method. Before we calculate the risk score using equations 1 and 2, we should be normalized by the column:

$$W_{ij} = \frac{W_{ij}}{\sum W_{\cdot j}} \tag{3}$$

Furthermore, the more exploitable a vulnerability is, the greater the risk value of that vulnerability that will be passed to the host. Therefore, to conduct a reasonable risk assessment, hosts that have incoming edges only and no outgoing edges, we distribute their risk values evenly among the other hosts. Therefore, we set a constant value $a_p\vec{r}_p^{\,(0)}$ to disperse the risk value from the host which has no out neighbours to other hosts, as shown in the extension Formula 4.

In this paper, we calculate the results to six decimal places. Therefore, when the difference between two iterations is less than $10^{-7}$, we consider the result convergent.

### C. MODEL EXTENSION

In Section 2, to perform quantitative vulnerability risk assessment, we consider not only the risk from a vulnerability itself but also the risk score propagated from others devices in the network. However, the factors that need to be considered may change as the study progresses or the network environment changes. When an assessment needs to include additional factors, we can identify the relationships between the new factors and the existing nodes in the network schema. Then, the weights and an appropriate adjacency matrix should be constructed.

For example, when a vulnerability risk assessment needs to additionally consider the life cycle of vulnerabilities, as shown in Figure 4, we can add a node that represents vulnerability life cycle and build a reasonable adjacency matrix $W_{VL}$. When a vulnerability $v_i$ is in the life cycle of $L_j$, $W_{VL} = weight$, where $weight$ can be defined flexibly; however, it should be ensured that the larger the $weight$ is, the higher the risk of that vulnerability is. When a risk assessment based on multiple types should be conducted, the following formula can be used.

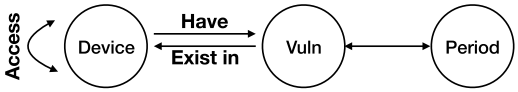



**FIGURE 4.** The extended network schema.

$$\vec{r}_p^{\,(k+1)}(i) = \frac{\sum_{q=1}^{m}\lambda_{pq}\sum_{j=1}^{n_q}W_{pq}(i,j)*\vec{r}_q^{\,(k)}(j) + a_p\vec{r}_p^{\,(0)}(i)}{\sum_{q=1}^{m}\lambda_{pq}+a_p} \tag{4}$$

where $\vec{r}_p^{\,(k+1)}(i)$ represents the quantitative score of instance i of type $p$ after the $k+1$-th iteration. Here, $m$ is the total number of types in the heterogeneous information network, and $n_q$ is the number of instances of type $q$. The parameter $\lambda_{pq}$ is the weight of the type $p$ and $q$, and $\alpha_p$ is the weight of the initial value of type $p$. Before calculating the risk score, we normalize all the adjacency matrixes by column, as in Section 3.2 (Formula 3). The equation will then converge according to reference [35].

## IV. EXPERIMENT

We conduct two case studies to compare our method with the CVSS metrics and the attack graph-based methods. The source files involved in the experiment part are available online.[1]

### A. COMPARISON WITH CVSS METRICS

In this section, we compare our proposed method with CVSS metrics. We construct a typical enterprise LAN, which was used in [36], as shown in Figure 5.

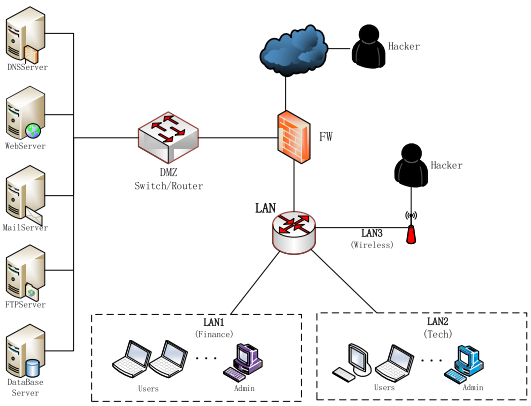



**FIGURE 5.** The experimental environment network.

The network includes two internal LANs (one for finance and one for technicians), a wireless LAN open to visitors (if the network is penetrated, an intruder can enter the internal network) and a DMZ hosting the servers, including a DNS

[1] https://github.com/stanwwr/HIN-based-vulnerability-risk-assessment

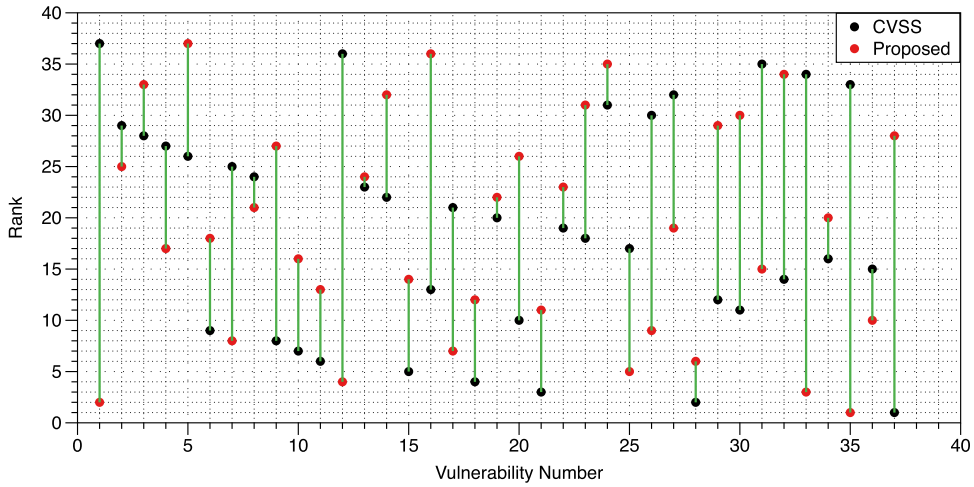



**FIGURE 6.** The ranking comparison with CVSS-based method.

server to provide DNS services, a web server to provide web services, a mail server to provide mail services, an FTP server to store and transfer files and a database server to store data. There are 19 hosts in the network. The financial department has seven hosts (No. 1-7); six of them are user PCs (No. 1-6), and one is an administrator PC (No. 7). The technical department has seven hosts (No. 8-14), among which six are user PCs and one is an administrator PC. The DNSServer, WebServer, MailServer, FTPServer, and DatabaseServer are numbered 15-19, respectively.

In this paper, the device include firewalls, switches and routers will not counting as one of the hosts. Because, the functions of the above devices are to control the accessible relationships, and the modifications of these devices are mainly involve the accessible relationships between other devices. Therefore, we simplify the property of these devices as the accessible relationships in the adjacent matrix $W_{DD}$. The more detailed consideration will be performed in the future.

For the acquisition of the knowledge of experimental network environment, the network scanning tools, such as Nessus [37] or OpenVAS [38] can be used. The network metadata, such as a list of hosts, services, ports, vulnerabilities, etc., can be extracted by OpenVAS and Nessus. The detailed description of known vulnerabilities can be obtained from standard data sources, such as the National Vulnerability Database (NVD) [14]. A detailed description of the experimental data can be found in Appendix.

To make the formula work, we should initialize the vector $\vec{r}_V^{(0)}$ and $\vec{r}_D^{(0)}$. In this paper, we use the the normalized IS of each vulnerability to initialize the vector $\vec{r}_V^{(0)}$, and use $\frac{1}{n}$ to initialize the $\vec{r}_D^{(0)}$.

Table 2 shows the vulnerability risk assessment results of the approach proposed in this paper and the CVSS metrics. Notably, ''Risk Score'' is the quantitative score of the vulnerability calculated by the method proposed in this paper, while ''Impact Score (IS)'' is a metric of the CVSS Base Score (BS). Here, ''Combination'' is the risk assessment score, which combined the Risk Score with ES and ''Base Score (BS)'' is the CVSS Base Score of the vulnerability (for comparison, the scores are normalized).

Additionally, because IS does not consider the vulnerable extent or the importance of components in a specific network environment, the IS cannot be used to accurately assess vulnerability risk in a specific network environment. For example, in Table 2, the IS values of vulnerabilities 35 and 31 are equal and rank relatively low. However, in the results of the algorithm proposed in this paper, vulnerability 35 has the highest risk score. From the perspective of CVSS, the IS of vulnerability 35 is 3.6, and its ES is 3.9. Therefore, if only the IS of a vulnerability is considered, the risk of a vulnerability may not be appropriately defined. However, in this network, vulnerability 35 exists in hosts 7, 13, 14, 15, and 19 (Table 6 in the Appendix), among which host 7 and host 14 are administrator PCs, which can access all the hosts and servers through various ports. Similarly, hosts 15 and 19 are servers in the LAN. All the hosts in the network can access these servers through the relevant ports; thus they are relatively important devices. Although the IS of vulnerability 35 is relatively low, the vulnerability should still be considered high risk because it exists on several important devices. Therefore, the method proposed in this paper can more reasonably assesses the risk of a vulnerability. The ''Combination'' result considers both the impact of a vulnerability on a specific network and the exploitability of that vulnerability. Compared with the CVSS metrics approach, the vulnerability risk assessment method proposed in this paper provides a more reasonable and accurate vulnerability risk value and can clearly distinguish among vulnerability risk values to provide a high-quality vulnerability repair strategy.

Figure 6 shows a ranking comparison between risk score and IS. Different risk scores can be obtained by different

**TABLE 2.** The comparison with CVSS-based method.

| Rank | Risk Score | | Impact Score (IS) | | Combination | | Base Score (BS) | |
|---|---|---|---|---|---|---|---|---|
| | Score | Num | Score | Num | Score | Num | Score | Num |
| 1 | 0.151763 | 35 | 0.041528 | 37 | 0.171753 | 35 | 0.031857 | 18 |
| 2 | 0.126802 | 1 | 0.041528 | 28 | 0.146791 | 1 | 0.031857 | 10 |
| 3 | 0.110537 | 33 | 0.041528 | 21 | 0.142452 | 12 | 0.031857 | 6 |
| 4 | 0.091196 | 12 | 0.041528 | 18 | 0.130527 | 33 | 0.031539 | 32 |
| 5 | 0.087685 | 25 | 0.041528 | 15 | 0.111339 | 28 | 0.03122 | 25 |
| 6 | 0.086224 | 28 | 0.041528 | 11 | 0.107674 | 25 | 0.03122 | 23 |
| 7 | 0.048351 | 17 | 0.041528 | 10 | 0.073344 | 18 | 0.03122 | 22 |
| 8 | 0.040352 | 7 | 0.041528 | 9 | 0.06834 | 17 | 0.03122 | 19 |
| 9 | 0.028344 | 26 | 0.041528 | 6 | 0.066372 | 21 | 0.03122 | 17 |
| 10 | 0.027471 | 36 | 0.028654 | 20 | 0.063452 | 11 | 0.03122 | 8 |
| 11 | 0.022292 | 21 | 0.026578 | 30 | 0.063315 | 10 | 0.03122 | 7 |
| 12 | 0.022088 | 18 | 0.026578 | 29 | 0.060878 | 6 | 0.03122 | 5 |
| 13 | 0.019372 | 11 | 0.026578 | 16 | 0.060571 | 15 | 0.03122 | 4 |
| 14 | 0.016491 | 15 | 0.024917 | 32 | 0.060342 | 7 | 0.03122 | 3 |
| 15 | 0.013807 | 31 | 0.024502 | 36 | 0.055571 | 20 | 0.03122 | 2 |
| 16 | 0.012059 | 10 | 0.024502 | 34 | 0.054411 | 29 | 0.029627 | 21 |
| 17 | 0.010642 | 4 | 0.024502 | 25 | 0.048395 | 9 | 0.029627 | 15 |
| 18 | 0.009622 | 6 | 0.024502 | 23 | 0.048334 | 26 | 0.029627 | 11 |
| 19 | 0.008366 | 27 | 0.024502 | 22 | 0.046075 | 30 | 0.029627 | 9 |
| 20 | 0.007596 | 34 | 0.024502 | 19 | 0.04163 | 16 | 0.02899 | 26 |
| 21 | 0.006743 | 8 | 0.024502 | 17 | 0.038748 | 36 | 0.025804 | 36 |
| 22 | 0.006545 | 19 | 0.024502 | 14 | 0.030632 | 4 | 0.025804 | 34 |
| 23 | 0.006395 | 22 | 0.024502 | 13 | 0.026733 | 8 | 0.025804 | 13 |
| 24 | 0.006395 | 13 | 0.024502 | 8 | 0.026535 | 19 | 0.024849 | 20 |
| 25 | 0.006395 | 2 | 0.024502 | 7 | 0.026385 | 22 | 0.024212 | 28 |
| 26 | 0.004315 | 20 | 0.024502 | 5 | 0.026385 | 2 | 0.023893 | 35 |
| 27 | 0.004315 | 9 | 0.024502 | 4 | 0.022008 | 31 | 0.023893 | 33 |
| 28 | 0.004195 | 37 | 0.024502 | 3 | 0.021985 | 23 | 0.023893 | 29 |
| 29 | 0.003155 | 29 | 0.024502 | 2 | 0.021985 | 3 | 0.023893 | 14 |
| 30 | 0.001995 | 30 | 0.021595 | 26 | 0.021622 | 37 | 0.023574 | 24 |
| 31 | 0.001995 | 23 | 0.021595 | 24 | 0.020615 | 5 | 0.0223 | 27 |
| 32 | 0.001995 | 14 | 0.019518 | 27 | 0.019642 | 27 | 0.021982 | 37 |
| 33 | 0.001995 | 3 | 0.01495 | 35 | 0.018872 | 34 | 0.021663 | 30 |
| 34 | 0.000626 | 32 | 0.01495 | 33 | 0.017672 | 13 | 0.020707 | 16 |
| 35 | 0.000626 | 24 | 0.01495 | 31 | 0.016515 | 32 | 0.016884 | 31 |
| 36 | 0.000626 | 16 | 0.012043 | 12 | 0.011902 | 24 | 0.016884 | 1 |
| 37 | 0.000626 | 5 | 0.005814 | 1 | 0.010196 | 14 | 0.015929 | 12 |

methods for the same vulnerability. The longer the green line is, the greater the difference between the two methods. There is a clear difference among vulnerabilities 1, 35, 12, and 37. The IS of vulnerability 1 is 1.4, and its ES is 3.9. Therefore, from the perspective of CVSS, vulnerability 1 is not a high-risk vulnerability. However, in the specific network environment, vulnerability 1 exists in hosts 7, 13, 14, 15, 16, and 19 (Table 6 in the Appendix), among which host 7 and host 14 are administrator PCs that can access all hosts and servers through relevant ports; additionally, hosts 15, 16, and 19 are servers in the LAN. All the hosts in the network can access these servers through the relevant ports; thus, the servers are relatively important devices. Therefore, the risk score of vulnerability 1 needs to be adjusted. Vulnerabilities 35,12, and 37 are in the same situation as vulnerability 1.

Figure 9 in the Appendix shows how the risk score of each vulnerability changes over an increasing number of iterations. As shown, the risk score of each vulnerability eventually

**TABLE 3.** The results of BAG-based static risk assessment and vulnerability risk ranking from the approach proposed.

| No. | Vulnerability | Probability | Vulnerability Ranking |
|---|---|---|---|
| 1 | SQL Injection | 80.54% | MS Video ActiveX Stack BOF |
| 2 | Heap corruption in OpenSSH | 77.22% | LICQ Buffer Overflow (BOF) |
| 3 | Error message information leakage | 77.22% | Remote Login |
| 4 | DNS Cache Poisoning | 77.22% | SQL Injection |
| 5 | Improper cookies handler in OpenSSH | 69.00% | IIS vulnerability in WebDAV service |
| 6 | Remote code execution in SMTP | 62.50% | DNS Cache Poisoning |
| 7 | Remote Login | 58.68% | MS SMV service Stack BOF |
| 8 | OpenSSL uses predictable random | 57.00% | Squid port scan vulnerability |
| 9 | MS Video ActiveX Stack BOF | 57.00% | Error message information leakage |
| 10 | IIS vulnerability in WebDAV service | 53.00% | Remote code execution in SMTP |
| 11 | Squid port scan vulnerability | 52.19% | Improper cookies handler in OpenSSH |
| 12 | LICQ Buffer Overflow (BOF) | 43.29% | Heap corruption in OpenSSH |
| 13 | MS SMV service Stack BOF | 38.11% | OpenSSL uses predictable random |

converges. For most of these vulnerabilities, the number of iterations required to achieve convergence is approximately 10 to 20. For vulnerabilities 1, 12, 25, 28, 33, and 35, the final risk score are higher than the initial values; for others, the final scores are lower than the initial values. Thus, when a specific network environment is considered, a different risk score should be obtained to more precisely assess the vulnerability risk.

### B. COMPARISON WITH ATTACK GRAPH-BASED METHODS

In this section, we conduct two case studies to specify the advantages and disadvantages of the proposed method and two attack graph-based methods.

#### 1) COMPARISON WITH BAG-BASED RISK ASSESSMENT

The Bayesian Attack Graph (BAG)-based risk assessment method was proposed in [2], and it conducts an in-depth exploration to determine static and dynamic risk assessments and perform risk mitigation analysis. To perform the risk assessment, the BAG is constricted for this experimental network, and the prior probabilities are initialized. Subsequently, the unconditional or conditional probability for each node is calculated based on Bayesian inference to assess the probability of risk occurrence. To perform the risk mitigation analysis, the cost and benefit of 13 security controls for the test network are quantified and those that maximizes the benefit and minimizes the cost are selected. The static risk assessment and risk mitigation in [2] are closest to those in our work; therefore, we will mainly consider them.

The probability of each vulnerability in [2] for static risk assessment is shown in Table 3, where ''Vulnerability'' and ''Probability'' denote the 13 vulnerabilities and their corresponding exploitation probability (from [2]). The ''Vulnerability Ranking'' shows the ranking results of the proposed method. We initialize $W_{VD}$, $W_{DD}$, $\vec{r}_V^{(0)}$ and $\vec{r}_D^{(0)}$ and use Formulas 1 and 2 to acquire the ranking. The parameter $\alpha$ is set to 0.5. As shown, the ''SQL Injection'' vulnerability has the largest exploitation probability in the experimental network. However, in the ranking list of the proposed method, the ''SQL Injection'' is ranked fourth, while ''MS Video ActiveX Stack BOF'' vulnerability is ranked first.

The result is not unexpected. The outcome of the static risk assessment in [2] is the probability of exploitation, which only considers the relationships between vulnerabilities and their exploitability but dose not consider the impact of those vulnerabilities. Therefore, in the risk mitigation part, the BAG-based method considers the cost and benefit of performing more reasonable decision making support. For ''SQL Injection'', as shown in Table 4, the corresponding security control is ''query restriction'', which has a relatively high cost but provides only low benefits. Therefore, the ''SQL Injection'' will not usually be the first vulnerability to be patched.

Table 4 shows the results of the proposed method and the risk mitigation part in [2]. The ''Cost ($A$)'', ''Outcome ($B$)'', and ''Net Benefit ($B - A - 622.0$)'' denote the cost, outcome, and benefit of using each security control individually. The ''Vulnerability Ranking'' shows the ranking results of the proposed method. Because the cost of exploiting a vulnerability is not considered in our work, we compare the results based on the ''Outcome'' shown in Table 4. The vulnerability ''MS Video ActiveX Stack BOF'' has the highest ranking, and its corresponding remediation action is ''apply an MS workaround''; this control is ranked third in [2]. The corresponding remediations for the vulnerabilities ''LICQ Buffer Overflow (BOF)'' and ''Remote Login'' are ''filtering external traffic'', which ranked second. The third vulnerability in the ranking list ''SQL Injection'' can also be fixed by ''filtering external traffic''. The ''IIS vulnerability in WebDAV service'' can be mitigated by the ''filtering external traffic'' and ''disable WebDAV'', which are ranked second and fourth, respectively. The results obtained in this paper are basically consistent with those in [2], which means that when using the order of the vulnerability ranking to perform

**TABLE 4.** The cost and benefit of security control in BAG-based method and the vulnerability risk ranking from the proposed method.

| Security Control | Cost ($A$) | Outcome ($B$) | Net Benefit ($B-A-622.0$) | Vulnerability Ranking |
|---|---|---|---|---|
| apply OpenSSH security patch | 63 | 1407.89 | 722.89 | MS Video ActiveX Stack BOF |
| filtering external traffic | 70 | 1208.84 | 516.84 | LICQ Buffer Overflow (BOF) |
| apply MS work around | 14 | 1202.45 | 566.45 | Remote Login |
| disable WebDAV | 250 | 1095.90 | 223.90 | SQL Injection |
| limit DNS access | 53 | 1000.65 | 325.65 | IIS vulnerability in WebDAV service |
| add Firewall | 205 | 881.15 | 54.15 | DNS Cache Poisoning |
| disable portscan | 11 | 875.44 | 242.44 | MS SMV service Stack BOF |
| apply MS09-004 work around | 31 | 861.10 | 208.10 | Squid port scan vulnerability |
| add Network IDS | 102 | 858.91 | 134.91 | Error message information leakage |
| use POP3 instead | 153 | 704.67 | -70.33 | Remote code execution in SMTP |
| encryption | 34 | 681.75 | 25.75 | Improper cookies handler in OpenSSH |
| query restriction | 84 | 681.00 | -25.00 | Heap corruption in OpenSSH |
| digital signature | 33 | 673.281 | 8.28 | OpenSSL uses predictable random |

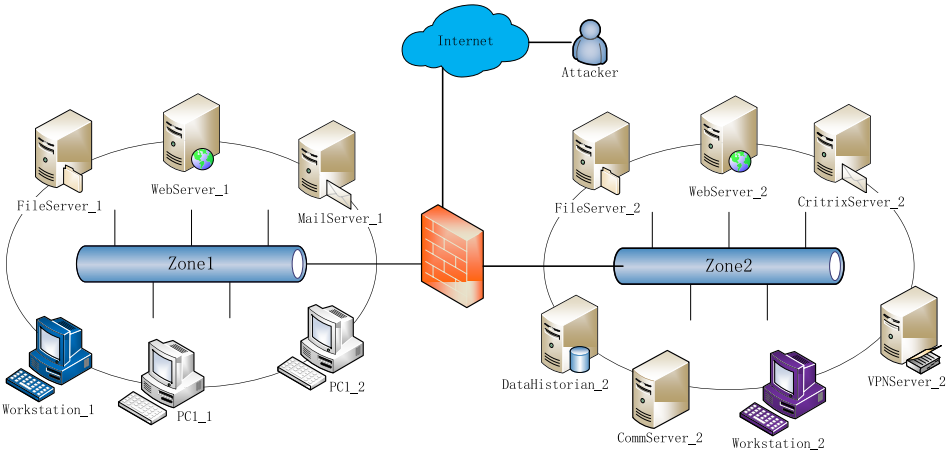



**FIGURE 7.** The experimental network.

risk mitigation, the outcome will be maximized. Therefore, the proposed method can produce a high-quality handbook for vulnerability risk mitigation. Although the vulnerability ranking does not strictly correspond to the security control ranking, that does not influence the effectiveness of repairing the vulnerability. Generally, a system update process will fix several prioritized vulnerabilities. Therefore, as long as several of the vulnerabilities with the highest risk scores are fixed, network security will be effectively strengthened.

### 2) COMPARISON WITH ASSETRANK FOR RISK ASSESSMENT

AssetRank was proposed in [28] and ranks the nodes in an attack graph using a PageRank-based method. To perform a comparison with AssetRank, we constructed a middle-sized network, which containing 13 hosts and 7 vulnerabilities. Figure 7 depicts the experimental network. We used the attack graph generation tool MulVAL [39] to generate the attack graph and we reproduced the AssetRank method and constructed the proposed method to target this network.

Table 5 shows the results of AssetRank and the proposed method. The columns titled "Vertex" and the "Rank$\times 10^2$" denote the nodes in the attack graph and the results of AssetRank, respectively, while the columns titled "Vulnerability Ranking" and "Score" indicate the ranking and score of the proposed method, respectively.

As Table 5 shows, AssetRank produces several repeated values that represent less precise results. For example, for the vulnerabilities "CVE-2010-0483", "CVE-2002-0392" and "CVE-2010-0812",the value should be different. Figure 8 shows the partial attack graph generated by MulVAL(for clarity, we simplifies the attack graph). The number before the colon in each node represents the node number in the full attack graph generated(available online). The diamonds denote the "OR" vertices, ellipses indicate the "AND" vertices, and the boxes denote the sink

**TABLE 5.** The results of AssetRank and proposed approach.

| Vertex | Rank$\times 10^2$ | Vulnerability Ranking | Score |
|---|---|---|---|
| vulExists(fileServer_1,CVE-2010-0812,windows_2003_server,remoteClient,privEscalation) | 1.395 | CVE-2010-0492 | 0.414 |
| vulExists(webServer_1,'CVE-2002-0392',httpd,remoteExploit,privEscalation) | 1.395 | CVE-2002-0392 | 0.178 |
| vulExists(commServer_2,'CVE-2010-0483',windows_2000,remoteClient,privEscalation) | 1.395 | CVE-2010-0483 | 0.105 |
| vulExists(vpnServer_2,'CVE-2010-0492',openvpn,remoteClient,privEscalation) | 1.231 | CVE-2010-0812 | 0.094 |
| vulExists(subnet_1_2,'CVE-2010-0490',ie,remoteClient,privEscalation) | 1.231 | CVE-2010-0490 | 0.080 |
| vulExists(subnet_1_1,'CVE-2010-0483',windows_2000,remoteClient,privEscalation) | 1.231 | CVE-2010-0491 | 0.065 |
| vulExists(citrixServer_2,'CVE-2010-0490',ie,remoteClient,privEscalation) | 1.231 | CVE-2010-0494 | 0.064 |
| vulExists(dataHistorian_2,'CVE-2010-0494',mountd,remoteExploit,privEscalation) | 1.171 | ———- | ——— |

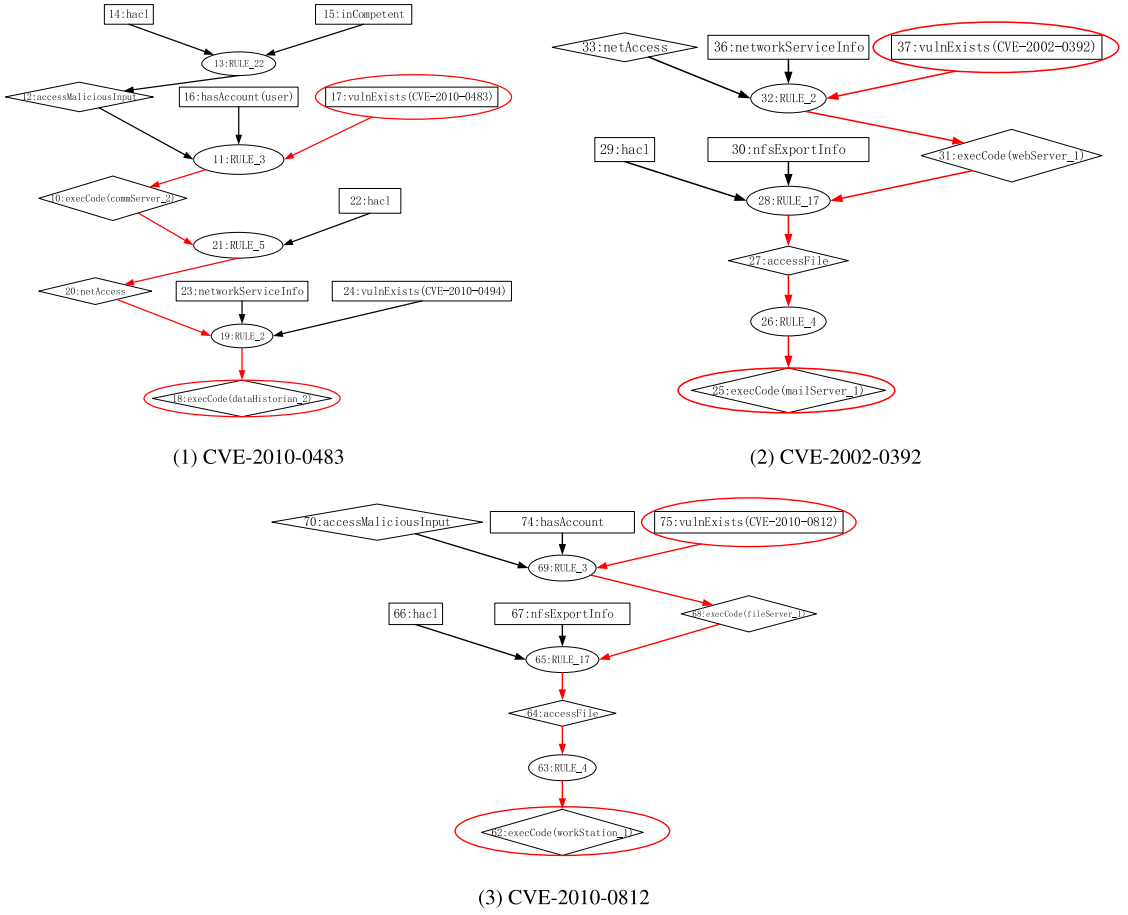



**FIGURE 8.** The partial attack graph of the vulnerability.

vertices. As shown in the attack graph, these three vulnerabilities account for the "execCode" on "DataHistorian_2", "MailServer_1" and "Workstation_1", and patching the three vulnerabilities will eliminate the three "execCode" threats. However, these three machines are distinctively different. In this network, "workstation_1" can not access any other devices; therefore, it cannot spread the risk by accessing other devices. The "MailServer" can be accessed by most devices in the test environment and can also access many other devices; therefore, the vulnerability rating for "CVE-2002-0392" should be more important than that of "CVE-2010-0812". The "CVE-2010-0483" is in the same situation. Therefore, the proposed method assesses vulnerability risk more precisely than does AssetRank and provides high-quality remediation suggestions.

### 3) COMPUTATIONAL COMPLEXITY ANALYSIS

The proposed method processes each link in the heterogeneous graph and performs iterative computations. Let $C$ denote the number of nodes of different types in the

**TABLE 6.** The detailed description of vulnerabilities.

| Num | Host | CVE | Impact Score (IS) | Exploitability Score (ES) | Base Score (BS) |
|---|---|---|---|---|---|
| 1 | 7,13,14,15,16,19 | CVE-2018-15473 | 1.4 | 3.9 | 5.3 |
| 2 | 15 | CVE-2016-10134 | 5.9 | 3.9 | 9.8 |
| 3 | 16 | CVE-2017-16943 | 5.9 | 3.9 | 9.8 |
| 4 | 1,3,4,5,11,12 | CVE-2019-0708 | 5.9 | 3.9 | 9.8 |
| 5 | 17 | CVE-2018-2893 | 5.9 | 3.9 | 9.8 |
| 6 | 1,11 | CVE-2013-4730 | 10.0 | 10.0 | 10.0 |
| 7 | 1,4,5,8 | CVE-2017-15376 | 5.9 | 3.9 | 9.8 |
| 8 | 3,11 | CVE-2017-7494 | 5.9 | 3.9 | 9.8 |
| 9 | 18 | CVE-2015-6125 | 10.0 | 8.6 | 9.3 |
| 10 | 1,3,10,11,12 | CVE-2015-0313 | 10.0 | 10.0 | 10.0 |
| 11 | 1,2,3,4,5,6 | CVE-2013-2551 | 10.0 | 8.6 | 9.3 |
| 12 | 7,9,15,16,18,19 | CVE-2010-5107 | 2.9 | 10.0 | 5.0 |
| 13 | 15 | CVE-2016-6617 | 5.9 | 2.2 | 8.1 |
| 14 | 16 | CVE-2018-19518 | 5.9 | 1.6 | 7.5 |
| 15 | 1,2,4,5,10,11 | CVE-2012-0002 | 10.0 | 8.6 | 9.3 |
| 16 | 17 | CVE-2010-4015 | 6.4 | 8.0 | 6.5 |
| 17 | 1,6,8,9,11 | CVE-2017-15222 | 5.9 | 3.9 | 9.8 |
| 18 | 1,2,5,6 | CVE-2015-0014 | 10.0 | 10.0 | 10.0 |
| 19 | 3,11,12 | CVE-2017-8589 | 5.9 | 3.9 | 9.8 |
| 20 | 18 | CVE-2015-5477 | 6.9 | 10.0 | 7.8 |
| 21 | 9,17,18 | CVE-2006-5051 | 10.0 | 8.6 | 9.3 |
| 22 | 15 | CVE-2017-7668 | 5.9 | 3.9 | 9.8 |
| 23 | 16 | CVE-2018-8302 | 5.9 | 3.9 | 9.8 |
| 24 | 17 | CVE-2017-3506 | 5.2 | 2.2 | 7.4 |
| 25 | 6,10,13,14 | CVE-2019-9760 | 5.9 | 3.9 | 9.8 |
| 26 | 2,4,6 | CVE-2014-5415 | 5.2 | 3.9 | 9.1 |
| 27 | 5,12 | CVE-2017-11780 | 4.7 | 2.2 | 7.0 |
| 28 | 9,13 | CVE-2003-1562 | 10.0 | 4.9 | 7.6 |
| 29 | 16,18 | CVE-2012-1823 | 6.4 | 10.0 | 7.5 |
| 30 | 16 | CVE-2014-2957 | 6.4 | 8.6 | 6.8 |
| 31 | 2,3,4,12 | CVE-2018-0976 | 3.6 | 1.6 | 5.3 |
| 32 | 17 | CVE-2018-3110 | 6.0 | 3.1 | 9.9 |
| 33 | 6,8,10,14,19 | CVE-2019-9809 | 3.6 | 3.9 | 7.5 |
| 34 | 5,11,12 | CVE-2017-0146 | 5.9 | 2.2 | 8.1 |
| 35 | 7,13,14,15,19 | CVE-2018-5314 | 3.6 | 3.9 | 7.5 |
| 36 | 10 | CVE-2016-0036 | 5.9 | 2.2 | 8.1 |
| 37 | 15,16 | CVE-2015-6564 | 10.0 | 3.4 | 6.9 |

heterogeneous information network, $K$ denote the total number of links in the heterogeneous graph, and $V$ denote the total number of objects. Let $t$ denote the number of iterations. Then, the complexity of the calculation task requires $O(t * C * K)$. Additionally, $K <= V^2$; therefore, the maximal complexity for performing the calculations is $O(t * C * V^2)$. The approach in [2], has a complexity of $O(N^3)$ to generate the attack graph, where $N = A * M$, $A$ is the number of attributes in the attack graph, and $M$ is the number of machines in the system. The probability-based analysis of the attack graph has a complexity of $O(2^n)$, where $n$ is the number of variables. For the method in [28], first, the time to generate the attack graph will be consumed; then, the time consumption of the ranking part is equal to that of the proposed method. In summary, the proposed approach provides sufficient precision at an acceptable computational complexity.

## V. CONCLUSION

In this paper, we innovatively propose a vulnerability risk assessment method based on the heterogeneous information network. First, we briefly reviewed the heterogeneous

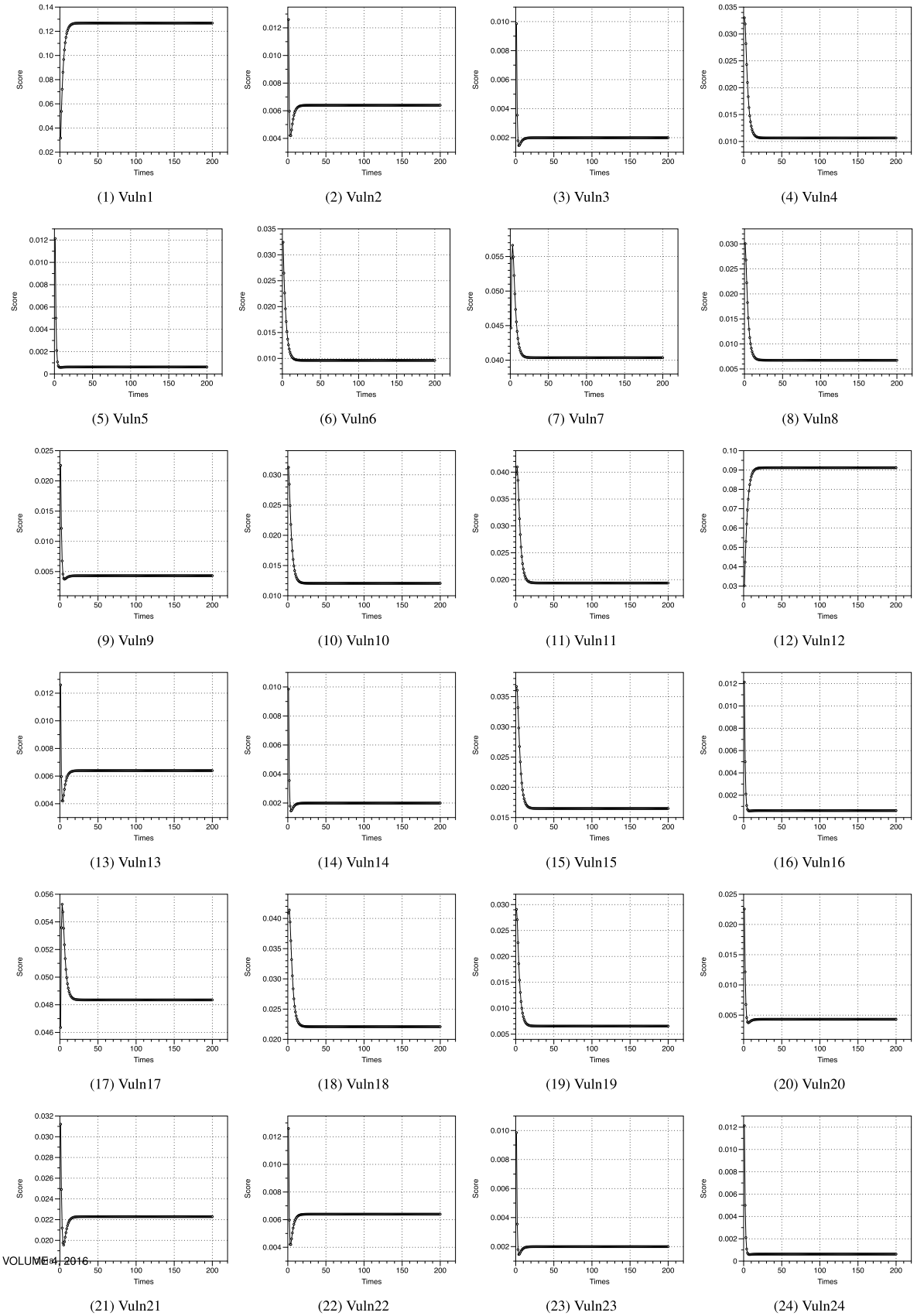
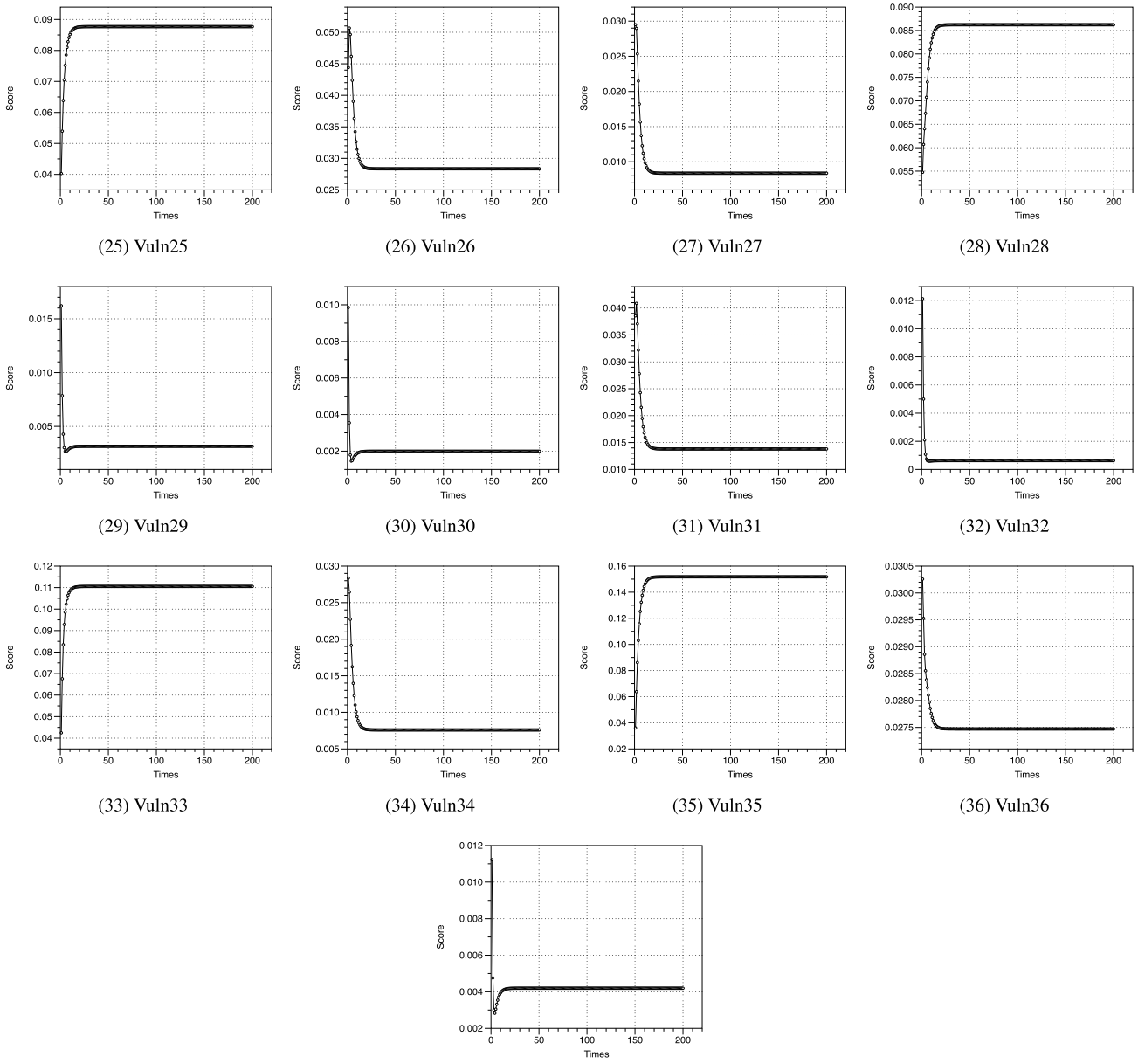



**FIGURE 9.** The risk score of each vulnerability changes with increasing number of iterations.




(25) Vuln25 (26) Vuln26 (27) Vuln27 (28) Vuln28

(29) Vuln29 (30) Vuln30 (31) Vuln31 (32) Vuln32

(33) Vuln33 (34) Vuln34 (35) Vuln35 (36) Vuln36

(37) Vuln37

**FIGURE 9.** ***(Continued.)*** **The risk score of each vulnerability changes with increasing number of iterations.**

information network and then introduced a ranking method based on the proposed heterogeneous information network, which mainly includes the establishment of the heterogeneous network model and the calculation method, which considers not only the exploitability of a vulnerability and its corresponding impact on the related components but also the impact of that vulnerability on those components in a specific network environment. Second, a method for extending the model is proposed, allowing the model to easily be adapted to changes in the network environment. Finally, a comparison with the CVSS metrics method and two attack graph-based methods were performed to demonstrate the advantages and disadvantages of the proposed method. The experiment simulation results show that the proposed method can more precisely assess the vulnerability risk in a specific network than can CVSS metric-based methods or AssetRank, and our approach provides precision equivalent to the BAG-based method but at a lower computational complexity. Nevertheless, the proposed approach still has several deficiencies. For example, the model does not fuse the rich semantic information to construct a higher-level model, and the vulnerabilities assessed in the test network are not completely up to date. In future works, we plan to perform more empirical and theoretical research to eliminate these drawbacks in this paper;

**TABLE 7.** Accessibility information.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| (1,1,3389) | (1,1,21) | (1,1,23) | (1,2,3389) | (1,2,23) | (1,3,3389) | (1,3,445) | (1,4,23) |
| (1,4,3389) | (1,5,23) | (1,5,3389) | (1,5,445) | (1,6,23) | (1,6,21) | (1,7,22) | (1,8,21) |
| (1,9,22) | (1,9,21) | (1,10,21) | (1,11,21) | (1,13,22) | (1,13,21) | (1,15,80) | (1,15,443) |
| (1,16,143) | (1,16,443) | (1,17,1521) | (1,18,53) | (1,18,443) | (1,19,21) | (2,1,3389) | (2,1,21) |
| (2,1,23) | (2,2,3389) | (2,2,23) | (2,3,3389) | (2,3,445) | (2,4,23) | (2,4,3389) | (2,5,23) |
| (2,5,3389) | (2,5,445) | (2,6,23) | (2,6,21) | (2,7,22) | (2,8,21) | (2,9,22) | (2,9,21) |
| (2,10,21) | (2,11,21) | (2,13,22) | (2,13,21) | (2,15,80) | (2,15,443) | (2,16,143) | (2,16,443) |
| (2,17,1521) | (2,18,53) | (2,18,443) | (2,19,21) | (3,1,3389) | (3,1,21) | (3,1,23) | (3,2,3389) |
| (3,2,23) | (3,3,3389) | (3,3,445) | (3,4,23) | (3,4,3389) | (3,5,23) | (3,5,3389) | (3,5,445) |
| (3,6,23) | (3,6,21) | (3,7,22) | (3,8,21) | (3,9,22) | (3,9,21) | (3,10,21) | (3,11,21) |
| (3,13,22) | (3,13,21) | (3,15,80) | (3,15,443) | (3,16,143) | (3,16,443) | (3,17,1521) | (3,18,53) |
| (3,18,443) | (3,19,21) | (4,1,3389) | (4,1,21) | (4,1,23) | (4,2,3389) | (4,2,23) | (4,3,3389) |
| (4,3,445) | (4,4,23) | (4,4,3389) | (4,5,23) | (4,5,3389) | (4,5,445) | (4,6,23) | (4,6,21) |
| (4,7,22) | (4,8,21) | (4,9,22) | (4,9,21) | (4,10,21) | (4,11,21) | (4,13,22) | (4,13,21) |
| (4,15,80) | (4,15,443) | (4,16,143) | (4,16,443) | (4,17,1521) | (4,18,53) | (4,18,443) | (4,19,21) |
| (5,1,3389) | (5,1,21) | (5,1,23) | (5,2,3389) | (5,2,23) | (5,3,3389) | (5,3,445) | (5,4,23) |
| (5,4,3389) | (5,5,23) | (5,5,3389) | (5,5,445) | (5,6,23) | (5,6,21) | (5,7,22) | (5,8,21) |
| (5,9,22) | (5,9,21) | (5,10,21) | (5,11,21) | (5,13,22) | (5,13,21) | (5,15,80) | (5,15,443) |
| (5,16,143) | (5,16,443) | (5,17,1521) | (5,18,53) | (5,18,443) | (5,19,21) | (6,1,3389) | (6,1,21) |
| (6,1,23) | (6,2,3389) | (6,2,23) | (6,3,3389) | (6,3,445) | (6,4,23) | (6,4,3389) | (6,5,23) |
| (6,5,3389) | (6,5,445) | (6,6,23) | (6,6,21) | (6,7,22) | (6,8,21) | (6,9,22) | (6,9,21) |
| (6,10,21) | (6,11,21) | (6,13,22) | (6,13,21) | (6,15,80) | (6,15,443) | (6,16,143) | (6,16,443) |
| (6,17,1521) | (6,18,53) | (6,18,443) | (6,19,21) | (7,1,3389) | (7,1,21) | (7,1,23) | (7,2,3389) |
| (7,2,23) | (7,3,3389) | (7,3,445) | (7,4,23) | (7,4,3389) | (7,5,23) | (7,5,3389) | (7,5,445) |
| (7,6,23) | (7,6,21) | (7,7,22) | (7,8,23) | (7,8,21) | (7,9,22) | (7,9,21) | (7,10,21) |
| (7,11,21) | (7,13,22) | (7,13,21) | (7,14,21) | (7,15,22) | (7,15,80) | (7,15,443) | (7,16,143) |
| (7,16,22) | (7,16,443) | (7,17,1521) | (7,18,53) | (7,18,443) | (7,19,22) | (7,19,21) | (8,1,21) |
| (8,1,23) | (8,2,23) | (8,4,23) | (8,5,23) | (8,6,23) | (8,6,21) | (8,7,22) | (8,8,23) |
| (8,8,21) | (8,9,22) | (8,9,21) | (8,10,3389) | (8,10,21) | (8,11,3389) | (8,11,21) | (8,11,445) |
| (8,12,3389) | (8,12,445) | (8,13,22) | (8,13,21) | (8,14,21) | (8,15,80) | (8,15,443) | (8,16,143) |
| (8,16,443) | (8,17,1521) | (8,18,53) | (8,18,443) | (8,19,21) | (9,1,21) | (9,1,23) | (9,2,23) |
| (9,4,23) | (9,5,23) | (9,6,23) | (9,6,21) | (9,7,22) | (9,8,23) | (9,8,21) | (9,9,22) |
| (9,9,21) | (9,10,3389) | (9,10,21) | (9,11,3389) | (9,11,21) | (9,11,445) | (9,12,3389) | (9,12,445) |
| (9,13,22) | (9,13,21) | (9,14,21) | (9,15,80) | (9,15,443) | (9,16,143) | (9,16,443) | (9,17,1521) |
| (9,18,53) | (9,18,443) | (9,19,21) | (10,1,21) | (10,1,23) | (10,2,23) | (10,4,23) | (10,5,23) |
| (10,6,23) | (10,6,21) | (10,7,22) | (10,8,23) | (10,8,21) | (10,9,22) | (10,9,21) | (10,10,3389) |
| (10,10,21) | (10,11,3389) | (10,11,21) | (10,11,445) | (10,12,3389) | (10,12,445) | (10,13,22) | (10,13,21) |
| (10,14,21) | (10,15,80) | (10,15,443) | (10,16,143) | (10,16,443) | (10,17,1521) | (10,18,53) | (10,18,443) |
| (10,19,21) | (11,1,21) | (11,1,23) | (11,2,23) | (11,4,23) | (11,5,23) | (11,6,23) | (11,6,21) |
| (11,7,22) | (11,8,23) | (11,8,21) | (11,9,22) | (11,9,21) | (11,10,3389) | (11,10,21) | (11,11,3389) |
| (11,11,21) | (11,11,445) | (11,12,3389) | (11,12,445) | (11,13,22) | (11,13,21) | (11,14,21) | (11,15,80) |
| (11,15,443) | (11,16,143) | (11,16,443) | (11,17,1521) | (11,18,53) | (11,18,443) | (11,19,21) | (12,1,21) |
| (12,1,23) | (12,2,23) | (12,4,23) | (12,5,23) | (12,6,23) | (12,6,21) | (12,7,22) | (12,8,23) |
| (12,8,21) | (12,9,22) | (12,9,21) | (12,10,3389) | (12,10,21) | (12,11,3389) | (12,11,21) | (12,11,445) |
| (12,12,3389) | (12,12,445) | (12,13,22) | (12,13,21) | (12,14,21) | (12,15,80) | (12,15,443) | (12,16,143) |
| (12,16,443) | (12,17,1521) | (12,18,53) | (12,18,443) | (12,19,21) | (13,1,21) | (13,1,23) | (13,2,23) |
| (13,4,23) | (13,5,23) | (13,6,23) | (13,6,21) | (13,7,22) | (13,8,23) | (13,8,21) | (13,9,22) |
| (13,9,21) | (13,10,3389) | (13,10,21) | (13,11,3389) | (13,11,21) | (13,11,445) | (13,12,3389) | (13,12,445) |
| (13,13,22) | (13,13,21) | (13,14,21) | (13,15,80) | (13,15,443) | (13,16,143) | (13,16,443) | (13,17,1521) |
| (13,18,53) | (13,18,443) | (13,19,21) | (14,1,3389) | (14,1,21) | (14,1,23) | (14,2,3389) | (14,2,23) |
| (14,3,3389) | (14,3,445) | (14,4,23) | (14,4,3389) | (14,5,23) | (14,5,3389) | (14,5,445) | (14,6,23) |
| (14,6,21) | (14,7,22) | (14,8,23) | (14,8,21) | (14,9,22) | (14,9,21) | (14,10,3389) | (14,10,21) |
| (14,11,3389) | (14,11,21) | (14,11,445) | (14,12,3389) | (14,12,445) | (14,13,22) | (14,13,21) | (14,14,22) |
| (14,14,21) | (14,15,22) | (14,15,80) | (14,15,443) | (14,16,143) | (14,16,22) | (14,16,443) | (14,17,22) |
| (14,17,1521) | (14,18,22) | (14,18,53) | (14,18,443) | (14,19,22) | (14,19,21) | (15,15,22) | (15,15,80) |
| (15,15,443) | (15,16,143) | (15,16,443) | (15,17,1521) | (15,18,53) | (15,18,443) | (15,19,21) | (16,15,80) |
| (16,15,443) | (16,16,143) | (16,16,22) | (16,16,443) | (16,17,1521) | (16,18,53) | (16,18,443) | (16,19,21) |
| (17,15,80) | (17,15,443) | (17,16,143) | (17,16,443) | (17,17,22) | (17,17,1521) | (17,18,53) | (17,18,443) |
| (17,19,21) | (18,15,80) | (18,15,443) | (18,16,143) | (18,16,443) | (18,17,22) | (18,17,1521) | (18,18,22) |
| (18,18,53) | (18,18,443) | (18,19,21) | | | | | |

meanwhile, we will continue to concentrate on integrating our method into ontology-based knowledge system and other semantic models for conducting quantitative risk analysis and threat modeling.

## APPENDIX

Table 6 shows the relevant information of vulnerabilities contained in the data set. The column ''Num'' means the sequence number of vulnerabilities. The column ''Host'' means the hosts where the vulnerability exists. The column ''CVE'' means the CVE number of the vulnerability. Column ''Impact Score (IS)'', ''Exploitability Score (ES)'', and ''Base Score (BS)'' means the CVSS metric of the vulnerability.

Figure 9 shows how the risk score of each vulnerability changes with the increase of the number of iterations.

Table 7 shows the accessible ports of devices. The tuple $(a, b, c)$ in the Table 7 means the host $a$ can access the host $b$ through port $c$.

## ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

## REFERENCES

[1] (2019). *National Internet Emergency Response Center(CNCERT-CC)*. [Online]. Available: https://www.cert.org.cn/publish/main/17/index.html

[2] D. López, O. Pastor, and L. G. Villalba, ''Dynamic risk assessment in information systems: State-of-the-art,'' in *Proc. 6th Int. Conf. Inf. Technol.*, Amman, Jordan, 2013, pp. 8–10.

[3] *Common Vulnerability Scoring System V3.0*. Accessed: Mar. 5, 2020. [Online]. Available: https://www.first.org/cvss/v3.0/specification-document

[4] C. Wu, T. Wen, and Y. Zhang, ''A revised CVSS-based system to improve the dispersion of vulnerability risk scores,'' *Sci. China Inf. Sci.*, vol. 62, no. 3, p. 39102, Mar. 2019.

[5] Y. Zheng, K. Lv, and C. Hu, ''A quantitative method for evaluating network security based on attack graph,'' in *Proc. Int. Conf. Netw. Syst. Secur.*, 2017, pp. 349–358.

[6] H. Wang, Z. Chen, J. Zhao, X. Di, and D. Liu, ''A vulnerability assessment method in industrial Internet of Things based on attack graph and maximum flow,'' *IEEE Access*, vol. 6, pp. 8599–8609, 2018.

[7] J. Zeng, S. Wu, Y. Chen, R. Zeng, and C. Wu, ''Survey of attack graph analysis methods from the perspective of data and knowledge processing,'' *Secur. Commun. Netw.*, vol. 2019, pp. 1–16, Dec. 2019.

[8] J. B. Hong, D. S. Kim, C.-J. Chung, and D. Huang, ''A survey on the usability and practical applications of graphical security models,'' *Comput. Sci. Rev.*, vol. 26, pp. 1–16, Nov. 2017.

[9] Y. Sun and J. Han, ''Mining heterogeneous information networks: Principles and methodologies,'' *Synth. Lectures Data Mining Knowl. Discovery*, vol. 3, no. 2, pp. 1–159, Jul. 2012.

[10] H. Ghani, J. Luna, and N. Suri, ''Quantitative assessment of software vulnerabilities based on economic-driven security metrics,'' in *Proc. Int. Conf. Risks Secur. Internet Syst. (CRiSIS)*, Oct. 2013, pp. 1–8.

[11] M. Keramati and M. Keramati, ''Novel security metrics for ranking vulnerabilities in computer networks,'' in *Proc. 7th Int. Symp. Telecommun. (IST)*, Sep. 2014, pp. 883–888.

[12] Y. Wang and Y. Yang, ''PVL: A novel metric for single vulnerability rating and its application in IMS,'' *J. Comput. Inf. Syst.*, vol. 8, no. 2, pp. 579–590, 2012.

[13] H. Holm and K. K. Afridi, ''An expert-based investigation of the common vulnerability scoring system,'' *Comput. Secur.*, vol. 53, pp. 18–30, Sep. 2015.

[14] *National Vulnerability Database*. Accessed: Feb. 15, 2020. [Online]. Available: https://nvd.nist.gov/vuln/search

[15] L. Wang, S. Jajodia, A. Singhal, P. Cheng, and S. Noel, ''K-zero day safety: A network security metric for measuring the risk of unknown vulnerabilities,'' *IEEE Trans. Dependable Secure Comput.*, vol. 11, no. 1, pp. 30–44, Jan. 2014.

[16] M. Keramati, ''New vulnerability scoring system for dynamic security evaluation,'' in *Proc. 8th Int. Symp. Telecommun. (IST)*, Sep. 2016, pp. 746–751.

[17] A. A. Younis, Y. K. Malaiya, and I. Ray, ''Using attack surface entry points and reachability analysis to assess the risk of software vulnerability exploitability,'' in *Proc. IEEE 15th Int. Symp. High-Assurance Syst. Eng.*, Jan. 2014, pp. 1–8.

[18] V. R. Kebande, I. Kigwana, H. S. Venter, N. M. Karie, and R. D. Wario, ''CVSS metric-based analysis, classification and assessment of computer network threats and vulnerabilities,'' in *Proc. Int. Conf. Adv. Big Data, Comput. Data Commun. Syst. (icABCD)*, Aug. 2018, pp. 1–10.

[19] A. Coşkun and U. Bostanci, ''Vulnerability analysis of smart phone and tablet operating systems,'' *Tehnički Vjesnik*, vol. 25, no. 6, pp. 1860–1866, 2018.

[20] A. Ur-Rehman, I. Gondal, J. Kamruzzuman, and A. Jolfaei, ''Vulnerability modelling for hybrid IT systems,'' in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Feb. 2019, pp. 1186–1191.

[21] Z. Li, C. Tang, J. Hu, and Z. Chen, ''Vulnerabilities scoring approach for cloud SaaS,'' in *Proc. IEEE 12th Int. Conf Ubiquitous Intell. Comput. IEEE 12th Int. Conf. Autonomic Trusted Comput. IEEE 15th Int. Conf. Scalable Comput. Commun. Associated Workshops (UIC-ATC-ScalCom)*, Aug. 2015, pp. 1339–1347.

[22] H. S. Lallie, K. Debattista, and J. Bal, ''A review of attack graph and attack tree visual syntax in cyber security,'' *Comput. Sci. Rev.*, vol. 35, Feb. 2020, Art. no. 100219.

[23] L. Gallon and J. J. Bascou, ''Using CVSS in attack graphs,'' in *Proc. 6th Int. Conf. Availability, Rel. Secur.*, Aug. 2011, pp. 59–66.

[24] J. Wang, M. Neil, and N. Fenton, ''A Bayesian network approach for cybersecurity risk assessment implementing and extending the FAIR model,'' *Comput. Secur.*, vol. 89, Feb. 2020, Art. no. 101659.

[25] B. Asvija, R. Eswari, and M. B. Bijoy, ''Bayesian attack graphs for platform virtualized infrastructures in clouds,'' *J. Inf. Secur. Appl.*, vol. 51, Apr. 2020, Art. no. 102455.

[26] M. S. Barik, A. Sengupta, and C. Mazumdar, ''Attack graph generation and analysis techniques,'' *Defence Sci. J.*, vol. 66, no. 6, p. 559, Oct. 2016.

[27] V. Mehta, C. Bartzis, H. Zhu, E. M. Clarke, and J. M. Wing, ''Ranking attack graphs,'' in *Proc. Int. Workshop Recent Adv. Intrusion Detection* 2006, pp. 127–144.

[28] R. Sawilla and X. M. Ou, ''Googling attack graphs,'' Defence R&D, Ottawa, ON, Canada, Tech. Rep. TM 2007-205, 2007.

[29] L. Page, S. Brin, R. Motwani, and T. Winograd, ''The PageRank citation ranking: Bringing order to the Web,'' Stanford InfoLab, Stanford, CA, USA, Tech. Rep. 1999-66, Nov. 1999.

[30] Z. Gyongyi, H. Garcia-Molina, and J. Pedersen, ''Combating Web spam with trustrank,'' in *Proc. 30th Int. Conf. Very Large Data Bases (VLDB)*, 2004, pp. 1–12

[31] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, ''PathSim: Meta path-based top-K similarity search in heterogeneous information networks,'' *Proc. VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, Aug. 2011.

[32] Z. Nie, Y. Zhang, J.-R. Wen, and W.-Y. Ma, ''Object-level ranking: Bringing order to Web objects,'' in *Proc. 14th Int. Conf. World Wide Web (WWW)*, 2005, pp. 567–574.

[33] Y. Sun, J. Han, P. Zhao, Z. Yin, C. Hong, and T. Wu, ''RankClus: Integrating clustering with ranking for heterogeneous information network analysis,'' in *Proc. 12th Int. Conf. Extending Database Technol., Adv. Database Technol. (EDBT)*, Saint Petersburg, Russia. New York, NY, USA: Association for Computing Machinery, 2009, pp. 565–576. [Online]. Available: https://doi.org/10.1145/1516360.1516426, doi: 10.1145/1516360.1516426.

[34] C. N. Bouza, ''Handbook of computational statistics: Concepts and methods,'' *Technometrics*, vol. 47, no. 3, pp. 383–384, 2012.

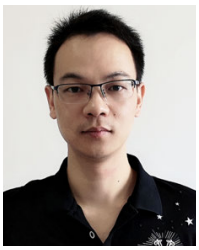
[35] M. Ji, J. Han, and M. Danilevsky, ''Ranking-based classification of heterogeneous information networks,'' in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2011, pp. 1298–1306.

[36] L. Munoz-Gonzalez, D. Sgandurra, M. Barrere, and E. C. Lupu, ''Exact inference techniques for the analysis of Bayesian attack graphs,'' *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 2, pp. 231–244, Mar. 2019.

[37] *Nessus Vulnerability Scanner*. Accessed: Apr. 20, 2020. [Online]. Available: http://www.nessus.org

[38] *Openvas*. Accessed: Apr. 25, 2020. [Online]. Available: http://www.openvas.org
[39] X. Ou, S. Govindavajhala, and A. W. Appel, "Mulval: A logic-based network security analyzer.," in *Proc. USENIX Secur. Symp.*, Baltimore, MD, USA, vol. 8, 2005, pp. 113–128.

**WENRUI WANG** was born in China, in January 1996. He received the B.Sc. degree (Hons.) in computer science and technology from Xi'an Jiaotong University, China, in 2018. He is currently pursuing the master's degree with the National University of Defense Technology, Hefei, China. His research interests include cyber security, machine learning, and network vulnerability analysis.



**FAN SHI** was born in China, in 1983. He received the B.Sc. degree (Hons.) in network engineering and the M.Sc. degree in information security from the National University of Defense Technology, Hefei, China, in 2004 and 2007, respectively. He is currently an Associate Professor with the National University of Defense Technology. His research interests include information security and network security situation awareness.

**MIN ZHANG** was born in 1966. He received the Ph.D. degree from Anhui University, China. He is currently a Professor with the National University of Defense Technology. His research interests include communication network security, intelligent computation, and vulnerability analysis.

**CHENGXI XU** was born in China, in 1989. He received the B.Sc. degree (Hons.) in automation and the M.Sc. degree in computer application technology from the National University of Defense Technology, Hefei, China, in 2007 and 2010, respectively, where he is currently pursuing the Ph.D. degree in cyberspace security. His research interests include internet infrastructure security and network measurement.

**JINGHUA ZHENG** was born in China, in 1976. She received the M.Sc. and Ph.D. degrees in communication engineering from the National University of Defense Technology, Hefei, China, in 2005 and 2018, respectively. She is currently an Associate Professor with the National University of Defense Technology. Her research interests include cyber security and communication network security.