Advances in Artificial Intelligence, Big Data and Algorithms
G. Grigoras and P. Lorenz (Eds.)

doi:10.3233/FAIA230850

A Vulnerability Threat Assessment Model Based on Bayesian Networks

Wenming LIU[a,b], Rui ZHAI[a1], Fang ZUO[a]
[a]*School of Software, Henan University,*
Kaifeng, 475001, China;
[b]*Institute of Information Engineering,*
Chinese Academy of Sciences,
Beijing, China

Abstract. A The integration of information and network technology into industries has led to the widespread use of software. Unfortunately, these software designs may contain defects and vulnerabilities pose risks that could endanger safety. To ensure safety, it is essential to assess and analyze software vulnerabilities and implement adequate security measures. However, existing vulnerability assessment models cannot measure the severity of industrial software vulnerabilities which pose a significant challenge. To overcome this issue, a vulnerability threat assessment model for industrial software is proposed based on the Common Vulnerability Scoring System (CVSS 3.1). The proposed model is tailored to the specific characteristics of the scenarios of industrial software. The proposed model introduces three new indicator factors: device security, information security, and life safety which were inferred using a Bayesian network model. These factors are adjusted according to the lifecycle of industrial software vulnerabilities. The proposed model accurately scores vulnerabilities in industrial software examples and provides a basis for effective vulnerability repair and reinforcement in the industrial software field.

Keywords. Industrial software, Common vulnerability model, Bayesian network model, Risk assessment

1. Introduction

Industrial software is extensively utilized in various fields, including manufacturing, energy, power, communication, sensing devices, and aerospace, among others. According to recent statistics, more than 80% of critical infrastructure related to national security and people's livelihood depends on industrial software for automation[1]. Stability and security are two critical indicators for industrial software. However, the open-source nature of software development poses significant challenges to the security of industrial software since it relies on third-party components (TPC)[2], which may introduce vulnerabilities into industrial software (Jones, 2018). The large amount of data collection, transmission, analysis, and sharing involved in industrial software poses a significant risk of data leakage and tampering since data lacks encryption and authentication. Moreover, the growing number of security vulnerabilities, coupled with

[1] Corresponding author: Rui ZHAI, School of Software, Henan University;
e-mail: 15539622281@163.com

highly sophisticated and persistent targeted attacks, requires more advanced security measures than traditional protection mechanisms such as intrusion detection and firewalls (Wang, 2020). Therefore, countries worldwide have launched active responses in policies, standards, technologies, and solutions to address the importance of the industrial sector, its vulnerable security situation, and increasingly severe attack threats.The security threats faced by the industrial sector have surpassed the effectiveness of traditional protection measures such as intrusion detection and firewalls. Consequently, countries worldwide have shifted their focus towards the importance of the industrial sector and its vulnerable security situation, leading to the development of active responses in policies, standards, technologies, and solutions. Unlike traditional software, industrial software operates in a relatively closed environment. As a result, during the development process, industrial software places more emphasis on functional implementation rather than security. This differs from traditional IT software, which follows strict security software development standards and security testing processes. Consequently, industrial software is more prone to having security flaws.

Assessing the security and vulnerability of industrial software is of utmost importance for both its producers and users. A scientifically reliable approach is therefore crucial to understanding the overall security status of industrial software. This necessitates the use of information security vulnerability assessment technology[3].

In the traditional information security field, the Common Vulnerability Scoring System (CVSS) is widely accepted for assessing vulnerabilities. Threat assessment of vulnerabilities is an effective means of ensuring the stable operation of industrial software. A comprehensive assessment and analysis of vulnerabilities can help better understand potential threats and impacts, and corresponding security measures can be taken to ensure the safety and stable operation of industrial software. Vulnerability threat assessment not only helps users discover and deal with vulnerabilities in a timely manner but also improves the security and reliability of industrial software, effectively protecting the normal operation of industrial systems and equipment. However, due to the particularity of industrial software scenarios, the focus of vulnerability rating is different. Vulnerabilities in industrial software generally affect sensors, energy, transportation, chemicals, and other infrastructure production or use units that use industrial products. These devices prioritize high availability of functions as well as equipment safety. The CVSS model measures vulnerability risk level based on the inherent characteristics of vulnerabilities. However, unlike traditional vulnerabilities, once industrial vulnerabilities are successfully exploited, they may disrupt industrial equipment operations and even threaten the operator's life. Therefore, the current model cannot reasonably evaluate the severity of vulnerabilities in the industrial field. Consequently, it is necessary to design a vulnerability threat assessment model that is suitable for industrial software.

This paper proposes a vulnerability threat assessment model for industrial software based on the concept of active defense, which takes into account the severity of security events after vulnerabilities are exploited. The model evaluates industrial software vulnerabilities using the same calculation process as CVSS and weighting and scoring calculation methods, making it highly usable. Additionally, the model incorporates security-related events to effectively characterize the impact of vulnerabilities on industrial software security, ensuring the model's effectiveness. By combining the concept of active defense, this model provides a scientific and reasonable evaluation of the severity of vulnerabilities in industrial software and supports industrial software security assessment, vulnerability repair, and system reinforcement.

2. Related Work

In the field of system security vulnerability assessment, security experts and scholars at home and abroad have explored and researched various methods from qualitative, quantitative, and combined qualitative and quantitative perspectives. The following are some related research works on vulnerability assessment.

Liu et al. [4] proposed a novel quantitative and qualitative vulnerability scoring system, which modified the CVSS vulnerability scoring method for generating and calculating vulnerability impact. They used a rule-based vulnerability impact assessment method, provided a rating table for rule-based vulnerability impact, and further determined the vulnerability impact based on the rating table. The quantitative method used the vulnerability rating as a scoring value, and combined it with the vulnerability feasibility score to arrive at the scoring formula. The vulnerability feasibility score had the same calculation method as the CVSS feasibility score, and the model achieved good verification results on the vulnerability dataset.

Spanos et al. [5] proposed a novel quantitative scoring system WIVSS, based on CVSS. Their impact calculation was based on the CVSS impact calculation method with improved weighting. Using a rule-based impact indicator weighting generation method, they analyzed the indicator relationships to obtain corresponding rules. The weight values of the evaluation indicators were approximately determined using the rules, the impact indicator values were added together to obtain the impact score, and the vulnerability feasibility score was entered into the CVSS 2.0 formula for the final vulnerability score calculation.

Chao et al. [6] proposed a novel Android application risk assessment method that could provide both quantitative and qualitative assessments. Their method merged multiple risk factors, such as system permissions, API calls, Intent Filter action attributes, and data flow. They assigned risk values based on factor-based risk classification and addition, and allocated weights of factor subsets based on hierarchical clustering. Experiments show that the evaluation results can effectively reflect the real security risks of Android applications.

Zhou et al. [7] proposed a network attack surface risk assessment method based on Bayesian attack graphs. By constructing a Bayesian attack graph of resources, vulnerability vulnerabilities, and their dependencies in the network system, the method considers the dependency between nodes, the correlation between resource exploits, and the impact of attack behavior on attack paths. It infers the probability of an attacker reaching each state and the most probable attack path. Experimental results demonstrate the feasibility and effectiveness of the proposed method, which can provide good support for the selection of dynamic defense measures for attack surface.

Ur-Rehman et al. [8] proposed an enhanced CVSS-based complex information system vulnerability scoring model that modifies the attack vector and attack complexity indicators based on the differences between IoT and traditional networks. Additionally, it adds a personal safety indicator that reflects the characteristics of IoT to better evaluate IoT vulnerabilities. The weights of each added indicator are determined based on laboratory analysis and past experience, and the vulnerability is scored by simple expansion of the CVSS formula.

Therefore, after conducting extensive research and analyzing in-depth reports on the industrial software industry, and taking into account the working principles of general vulnerability assessment models, we have identified the following factors to evaluate the threat of vulnerabilities in industrial software:

1. Industrial software security risks are distinct from those found in general operating software, as ordinary vulnerabilities in traditional software could only result in system crashes or occasional malfunctions. Conversely, industrial software's vulnerabilities could cause severe outcomes such as production interruptions, system failure, and even personal injuries. Consequently, assessing the severity of industrial software vulnerabilities accurately demands consideration of their potential influence on equipment safety and physical well-being.

2. The lifecycle of vulnerabilities in industrial settings differs from those in traditional ones. Due to the unique nature of goals and poor system patch compatibility within industrial environments, when traditional and industrial vulnerabilities coexist in the same environment, industrial vulnerabilities pose a more significant threat and impact.

3. Inevitably, existing industrial systems may contain vulnerabilities or configuration issues that could disadvantage system security. Threat actors could exploit these vulnerabilities and weaknesses, resulting in minor system outages, confidential data theft, or more severe security incidents with the possibility of information leakages in industrial equipment[9].

3. Vulnerability Threat Assessment Model

The vulnerability threat assessment model for industrial software is presented in Figure 1, and it comprises three dimensions in its scoring model: basic indicators, time indicators, and environmental indicators[10]. The basic indicators consist of the vulnerability's access vector, attack complexity, authentication requirements, confidentiality, integrity, availability impact, life safety, equipment safety, and information security. Time indicators include the level of confidence in the report, patch level, vulnerability lifecycle weight, and code maturity. Environmental indicators include factors such as attack complexity, authentication requirements, and availability impact[11]. The final score for the severity of the vulnerability impact is computed by combining the three dimensions.

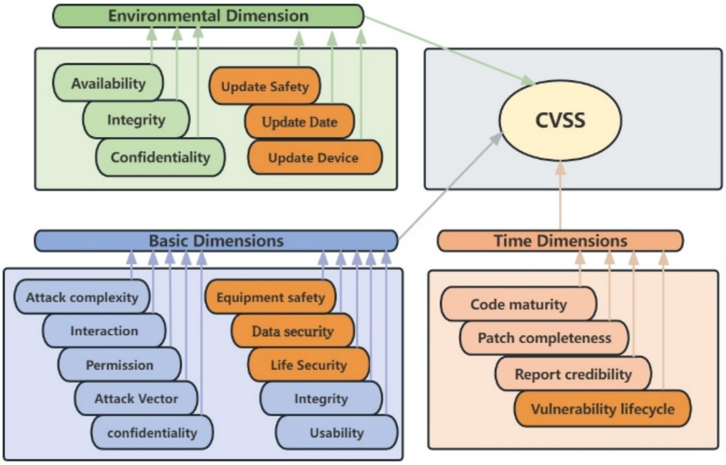

Figure 1. Vulnerability Threat Assessment Model for Industrial Software

3.1. Algorithm Design

The Vulnerability Assessment Model algorithm consists of three steps.

Step 1 The basic score is calculated by combining the feasibility score and impact score. The vulnerability impact score S_i and vulnerability feasibility score S_e are used to calculate the basic score, while scope is used to evaluate whether the vulnerability affects resources outside its security scope. The formula for calculating the basic score is shown in Eq. (3). If S_i is less than or equal to 0, the vulnerability does not pose a threat under the current evaluation criteria, and the final score is 0. If S_i is greater than 0 and the impact range remains the same, the final score for the severity of the vulnerability impact is the minimum value between 10 plus the sum of S_i, S_e, and a deviation constant of 0.2.

Step 2 The Time Score is used to describe the current availability of exploit techniques or code, the existence of any patches or solutions, or the credibility of the vulnerability report. It is calculated based on the maturity of the exploit code, patch level, report credibility, and vulnerability lifecycle weight. The specific formula for calculating the temporal score is shown in Eq. (2).

Step 3 The Environmental Score S_{en} is calculated by adjusting the Temporal Score to account for environmental factors that may impact the vulnerability, such as security measures, network topology, and attacker privileges. The Environmental Score provides a more accurate vulnerability assessment method, as the same vulnerability may have different impacts in different environments, and the Environmental Score can reflect these differences. The specific formula for calculating the Environmental Score is shown in Eq. (3).

$$S_{base} = \begin{cases} 0, S_i \leq 0 \\ \text{Round}\left(\text{Min}\left[(S_i + S_{\text{e}}) + 0.2, 10\right]\right), S_i > 0, \\ \text{socpe} = \text{Unchanged} \\ \text{Round}\left(\text{Min}\left[1.08 \times (S_i + S_{\text{e}}) + 0.2, 10)\right)\right), S_i > 0, \\ \text{scope} = C\text{hanged} \end{cases} \tag{1}$$

$$S_{time} = 7.57 \times E_{av} \times E_{ac} \times E_{pr} \times E_{ui} \tag{2}$$

$$S_{en} = \begin{cases} \text{socpe} = \text{Unchanged} \\ \text{Round}\left(\text{Min}\left[(I_m + I_e)\right]\right) \times S_{time} \\ \text{scope} = C\text{hanged} \\ \text{Round}\left(\text{Min}\left[1.08 \times (I_m + I_e)\right]\right) \times S_{time} \end{cases} \tag{3}$$

3.2. Bayesian Networks Infer Parameters

Quantifying the impact of a vulnerability on personnel, device, and information security is essential when the vulnerability is exploited. Nonlinear regression has been used in existing research to assess automotive system security, but this method may produce inaccurate results due to the complex and varying severity levels of industrial software faults. Bayesian networks can accurately model risk analysis by combining prior knowledge with sample data and calculating node occurrence importance. In this study, we propose a Bayesian network model to infer the security event factors introduced into the CVSS 3.1 model, as shown in Figure 2.

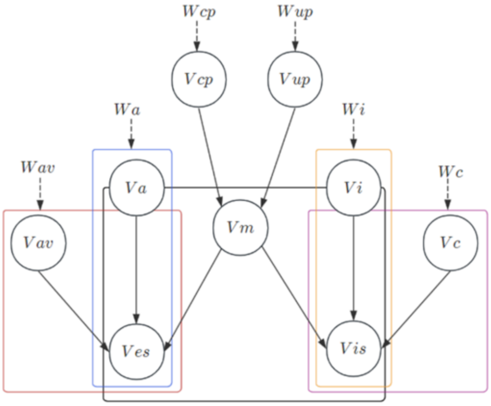

Figure 2. Structure diagram of Bayesian model derivation

The proposed Bayesian network model structure is shown in Figure 2, with nodes for attack vector, privileged access, availability, personnel safety, and information security. The model accounts for the direct impact of the attack vector, privileged access, and availability on device security, as well as the impact of privileged access, confidentiality, and integrity on information security. Successful exploitation of the vulnerability may pose a threat to personnel safety. Table 1 shows the inferred indicators and factors.

Table 1. Scoring indicator weight

Indicator Name	Indicator Grading	Possible Values
Equipment safety	**[H,L,N]**	[0.69,0.31,0]
Personnel safety	**[H,L,N]**	[0.69,0.31,0]
Information security	**[H,L,N]**	[0.62,0.38,0]

4. Experimental Results and Analysis

To evaluate the effectiveness of the proposed vulnerability threat assessment model, we selected and analyzed ten typical vulnerabilities, including Siemens industrial software vulnerabilities and common security vulnerabilities. We assessed the effectiveness of the proposed model by evaluating the rationality of vulnerability scores.

4.1. Vulnerability Rating Analysis

Table 2 present the vulnerability indicators' values for the selected vulnerabilities under the $CVSS_{IS}$ model. By applying the weights provided in the table to the $CVSS_{IS}$ model, the score for each vulnerability can be calculated. The score comprises not only the vulnerability score obtained through $CVSS_{IS}$ scoring but also the general vulnerability assessment of CVSS 3.1 and the Vehicle Security Model for Intelligent Evaluation (VSMIV) based on CVSS. VSMIV is a representative model applicable in different environments[12].

Table 2. Comparison of CVSS 3.1, VSMIV, and $CVSS_{IS}$ scores

Vulnerability number	CVSS 3.1	VSMIV	$CVSS_{IS}$
CVE-2020-15782	8.8	8.2	9.5
CVE-2018-14791	7.8	7.6	8.1
CVE-2015-5374	7.5	7.6	7.9
CVE-2017-16728	6.9	7.1	8.2
CVE-2020-15783	7.2	6.3	7.8
CVE-2019-10915	7.8	7.2	7.9
CVE-2019-9493	6.5	6.4	6.3
CVE-2019-9977	8.8	9.3	7.9
CVE-2018-18203	6.4	7.2	6.3
CVE-2018-9318	9.2	9.8	8.8

Table 2 shows that the Common Vulnerability Scoring System Industrial Control Systems $CVSS_{IS}$ scores for traditional security vulnerabilities are generally lower but still reasonable. For example, the vulnerability CVE-2019-9977 decreased from 8.8 to 7.9 due to the added evaluation indicators that do not pose a threat to personal safety or cause significant device losses. Moreover, as a critical system vulnerability in the vehicle's infotainment system, the score for CVE-2019-9977 increased to 9.3 in VSMIV compared to 8.8 in CVSS 3.1.

The overall vulnerability score for industrial facilities increased under the Common Vulnerability Scoring System Industrial Control Systems model. For instance, the exploitation of CVE-2018-14791 can allow an attacker to remotely crash the controller system or execute malicious code by transmitting malicious data packets. Attackers can leverage this vulnerability to manipulate industrial automation production lines, resulting in significant impacts on industrial equipment. CVE-2017-16728 is another vulnerability affecting Schneider Electric's EcoStruxure Control Expert industrial software product, owing to defects in the software that allow attackers to execute arbitrary code, thereby endangering personnel safety by disrupting normal operation of industrial equipment. As such, vulnerabilities in industrial facilities can affect both the industrial equipment and personal safety of workforce resulting in greater harm, as indicated by higher scores in the proposed model compared to CVSS 3.1 and VSMIV.

Aside from traditional security vulnerabilities, VSMIV assigns higher scores to the remaining vulnerabilities than CVSS 3.1. CVE-2015-5374, which poses a threat to personnel safety in industrial operations, also impacts industrial equipment operation, leading to higher VSMIV scores relative to CVSS 3.1. This is due to the $CVSS_{IS}$ model adding three indicators to the vulnerability assessment: device security, information security, and life safety, providing a more comprehensive depiction of the potential effects of the vulnerability on industrial software. Device security is particularly important as it considers controllability, availability, and the potential impact on production and manufacturing processes, including threats to personal safety. The composite score of these indicators can help users better assess the severity and impact of vulnerabilities and take appropriate security measures.

5. Conclusions and Future Directions

This paper proposes a vulnerability threat assessment model, the $CVSS_{IS}$, for industrial software. The model is based on device security, life safety, and information security. The corresponding rating indicators primarily focus on device controllability, information confidentiality and integrity, and the potential impact on life safety posed by threats. The formula weights under the time dimension have been optimized based on the characteristics of the vulnerability lifecycle in industrial software to determine vulnerability assessment severity reasonably. In contrast with the typical CVSS 3.1 model, the $CVSS_{IS}$ model evaluates the impact of vulnerabilities on industrial software, identifies whether they can damage industrial equipment, and determines the severity of life safety threats. Additionally, information security-related indicators have been added, allowing the vulnerability score to reflect the potential information leakage of industrial software to a certain degree. The Bayesian network model proposed in this paper infers new security indicators based on existing indicator factors, enabling the model to better represent and depict vulnerabilities in industrial software. Experimental test results demonstrate that the $CVSS_{IS}$ and VSMIV models adopt more comprehensive rating indicators for vulnerability assessment of the same industrial vulnerability. Therefore, the $CVSS_{IS}$ model can perform vulnerability threat assessment more reasonably.

Acknowledgements

This work is supported in part by Kaifeng Science and Technology R&D (Grant No.2201009), in part by Postgraduate Education Reform and Quality Improvement Project of Henan Province (Grant No. YJS2022JD26), in part by Postgraduate Education Innovation Training Base (Grant No. SYLJD2022008), and in part by Postgraduate Teaching Reform Project of Henan University (Grant No. SYLAL2022030. No. YJSJG2023XJ021 and No. YJSJG20 23XJ040).

References

[1] High Fei, Wang Zheng, Wang Li. The Connotation, Key Features, and Implementation Mechanism of the New National System in the Modern Era. [J]. China Technology Forum,2023(01):1-9.

[2] Gao Qing, Chen Jing, Xu Ping, Zhang Shikun. Research on Security Vulnerability Patterns in Industrial Embedded Software Development. [J].Research on Information Security,2022,8(06):595-604.

[3] LI Z, TANG C, HU J B, et al. Vulnerabilities scoring approach for cloud SaaS[J]. Journal on Communications, 2016, 37(8): 157-166.

[4] LIU Q, ZHANG Y. VRSS: a new system for rating and scoring vulnerabilities[J]. Computer Communications, 2011, 34(3): 264-273.

[5] SPANOS G, SIOZIOU A, ANGELIS L. WIVSS: a new methodology for scoring information systems vulnerabilities[C]. 2013: 83-90.

[6] Fan CHAO, Zhi YANG, Xuehui DU, Bing HAN. Classified risk assessment method of Android application based on multi-factor clustering selection[J]. Chinese Journal of Network and Information Security, 2021, 7(2): 161-173.

[7] Yuyang ZHOU, Guang CHENG, Chunsheng GUO. Risk assessment method for network attack surface based on Bayesian attack graph[J]. Chinese Journal of Network and Information Security, 2018, 4(6): 11-22.

[8] UR-REHMAN A， GONDAL I,KAMRUZZAMAN J.Vulnerability modelling for hybrid industrial control system networks[J]. Journal of Grid Computing, 2020,18(4): 863-878.

[9] Li Xinge, Hu Xiaoya, Zhou Chunjie, et al. Multidimensional Collaborative Analysis of Vulnerabilities throughout the Life Cycle of Industrial Control Systems. Control and Decision[J].Controland Decision,2022,37(11):2827-2838.

[10] CHEN X Z,WU Y,LI J H.System and approach of security testing and evaluation for invehicle information systems[J]. Journal of Cyber Security, 2017,2(2): 15-23.

[11] Frigault M , Wang L , Jajodia S ,et al.Measuring the Overall Network Security by Combining CVSS Scores Based on Attack Graphs and Bayesian Networks[J]. 2017.

[12] Haiyang YU, Xiuzhen CHEN, Jin MA, Zhihong ZHOU, Shuning HOU. Information security vulnerability scoring model for intelligent vehicles[J]. Chinese Journal of Network and Information Security, 2022, 8(1): 167-179.