Vulnerability Risk Assessment and Mitigation Methods Combining Dynamic and Static Features (Postprint)

Authors: Ye Ziwei, Guo Yuanbo, Ju Ankang

Date: 2019-01-28T00:00:00+00:00

Abstract

This study investigates how to improve the accuracy of vulnerability risk assessment and proposes a vulnerability risk assessment and mitigation method that combines dynamic and static features. By employing fixed attributes commonly used in traditional risk assessment methods and derived from the CVSS scoring system—such as attack complexity, impact level, and attack vector—as static features, and treating attributes that may change over time—such as defense capability, vulnerability remediation status, and attacker capability—as dynamic features, the combination of both enables a more comprehensive evaluation of vulnerability risk levels. The paper presents quantitative calculation methods for each feature in practical applications, along with a recommendation method for vulnerability remediation strategies. Using the risk assessment process for a single vulnerability and the assessment results for multiple vulnerabilities as examples, comparative experiments are conducted between the evaluation results and CVSS scores. The results demonstrate that the proposed method can provide more accurate vulnerability risk assessment results and reasonable remediation strategies by considering specific network environments, thereby validating the feasibility and effectiveness of the method.

Full Text

Vulnerability Risk Assessment and Mitigation Method Combining Dynamic and Static Features

Ye Ziwei, Guo Yuanbo, Ju Ankang
(Information Engineering University, Zhengzhou 450000, China)

Abstract

This paper addresses the challenge of improving the accuracy of vulnerability risk assessment by proposing a novel method that combines dynamic and static features. Traditional risk assessment methods typically rely on static attributes from the CVSS scoring system, such as attack complexity, impact degree, and attack vector. Our approach incorporates these as static features while introducing dynamic features that may change over time, including defense capabilities, vulnerability remediation status, and attacker capabilities. By integrating both types of features, we achieve a more comprehensive evaluation of vulnerability risk. We present quantitative calculation methods for each feature in practical applications and propose a method for recommending vulnerability remediation strategies. Using both single-vulnerability assessment processes and multi-vulnerability assessment results as examples, we conduct comparative experiments with CVSS scores. The results demonstrate that our method can provide more accurate vulnerability risk assessments and reasonable remediation strategies tailored to specific network environments, thereby validating its feasibility and effectiveness.

Key words: vulnerability; risk assessment; dynamic feature; static feature

0 Introduction

With the development of specialized network technologies such as traditional Internet, industrial control networks, Internet of Things, and mobile Internet, the severity of cybersecurity issues is escalating. Network attacks based on vulnerability exploitation pose significant threats to various network systems. Consequently, researching vulnerability origins, categorizing vulnerability types, and assessing vulnerability risk levels are crucial for enhancing network resilience and self-healing capabilities.

In theoretical research on vulnerability risk assessment, Fu et al. [**?**] proposed a method based on rough set theory to reduce reliance on expert experience when selecting attributes that may affect vulnerability risk. Tang et al. [**?**] introduced a genetic fuzzy analytic hierarchy process that calculates software vulnerability risk values through fuzzy judgment matrices and optimal solutions to nonlinear optimization problems. Huang et al. [**?**] developed a security vulnerability assessment framework based on the FAHP algorithm, integrating multiple common vulnerability attributes to meet diverse assessment scenarios. Additionally, researchers have applied existing methods to mobile Internet [**?**], telecommunications equipment [**?**], and power control industrial systems [**?**, **?**].

A review of existing research reveals that current vulnerability risk assessment primarily relies on a priori knowledge of vulnerability attributes. However, in specific network environments, dynamically changing attributes such as vulnerability patchability, existing defense measures, and attacker capabilities significantly impact actual risk levels. For instance, vulnerabilities with official patches can have their risks eliminated through patch installation; deploying

targeted defense measures can reduce risks for specific vulnerability types; and higher attacker capabilities increase vulnerability risks. Since existing studies cannot incorporate these dynamic features for more precise risk assessment, network administrators may misjudge network security status and vulnerability remediation priorities.

To address these limitations, this paper proposes a vulnerability risk assessment and mitigation method that combines dynamic and static features. We treat attack complexity, exploitation consequences, and attack vector (local/adjacent/network) as static features, while patch status, existing defense measures, and attack capabilities serve as dynamic features. Together, these features assess vulnerability risks in target networks and prioritize remediation efforts accordingly.

The main contributions of this paper are:
a) Analysis and selection of vulnerability dynamic and static features;
b) A vulnerability risk assessment method combining both feature types to inform further security measures;
c) A vulnerability remediation strategy recommendation method based on assessment results and vulnerability types;
d) Validation of the method' s practicality and effectiveness through case studies.

1.1 CVSS Standard

The core of vulnerability risk assessment lies in the quantification of risk-related features, as the quality of quantification methods directly affects assessment accuracy and applicability. Since vulnerabilities exist across diverse software and hardware in various application scenarios, different vendors and organizations prioritize different assessment aspects, resulting in no unified, universal standard.

The CVSS standard currently offers the best general applicability and widest adoption, supported by vulnerability databases including the U.S. National Vulnerability Database (NVD), China National Vulnerability Database (CNVD), and vendors such as Symantec and Oracle. In August 2015, CVSS version 3.0 was released, updating certain metrics and evaluation systems from version 2.0. Version 3.0 primarily adds analysis of inter-vulnerability effects, employs more granular attack difficulty metrics, and can analyze exploit chains among multiple vulnerabilities on the same component, thereby improving objectivity and compatibility.

Numerous studies have discussed and improved the CVSS standard. Allodi and Younis et al. [**?**, **?**] examined whether CVSS' s pattern-based classification and scoring accurately assess security risks. Johnson et al. [**?**] discussed CVSS' s universal applicability across databases and scenarios. Liu et al. [**?**] investigated finer-grained vulnerability security classification based on CVSS. Ruohonen et al. [**?**] analyzed costs throughout the vulnerability lifecycle from discovery to

inclusion and their influencing factors. Other scholars have applied CVSS to telecommunications networks [?] and industrial control networks [?].

1.2 Attack Capability Analysis

Attacker capabilities directly influence network risk levels. Different target networks face potential attackers with different objectives, which correlate with attack capabilities. For example, attackers motivated by political demands or financial gain typically possess higher capabilities, while those testing skills or playing pranks usually have lower capabilities. Implementing different defenses for different attacker capabilities can effectively reduce defense costs while meeting security requirements.

Two primary approaches exist for assessing attacker capabilities. The first involves pre-attack assessment based on network type, data confidentiality levels, and likely attack objectives. In this direction, Holsteijn [?] employed attack tree techniques to determine attacker intentions and capabilities, recognizing that attackers seeking financial gain versus technical challenges evaluate attack outcomes differently and exhibit different capabilities. Jaafor et al. [?] constructed a multi-layered graph model for social engineering attacks, analyzing potential attacks and attacker capabilities in social networks, forums, or blogs by decomposing social engineering into multiple attack phases, environments, and elements including attackers, victims, required resources (techniques), and specific attack behaviors. Durkota et al. [?, ?] decomposed network behaviors into natural (normal user), defender, and attacker behaviors, assuming attackers could obtain current network status but could not predict defender actions, and used game theory to evaluate potential attacker capabilities and behaviors.

The second approach assesses capabilities during attacks based on attack methods, intensity, and targets. Fadlallah et al. [?] experimentally validated that combining attack graph techniques with intrusion detection enables dynamic analysis of ongoing attacks to infer attacker objectives and capabilities. Pieters et al. [?] proposed a framework that evaluates attacker capabilities and intensity in real-time during attacks based on attacker investment, thereby adjusting vulnerability exploitation probabilities.

2 Method Overview

Our proposed vulnerability risk assessment and mitigation method comprises two phases: vulnerability risk assessment and vulnerability remediation strategy recommendation.

2.1 Vulnerability Risk Assessment

Traditional CVSS-based vulnerability risk assessment only considers static features such as attack complexity, attack vector, and impact degree, resulting in many vulnerabilities receiving identical or similar scores that fail to differentiate

risks precisely. Moreover, vulnerability risk levels change with conditions such as official patch releases, exploit code publication, and new defense capabilities. To address these issues, we propose a method combining dynamic and static features for more granular risk assessment and more reasonable remediation strategies.

2.1.1 Feature Selection for Risk Assessment

As previously described, we categorize vulnerability features into static and dynamic types. Static features include attack complexity, attack vector, and impact degree. In the CVSS scoring system, these features have the value ranges shown in Table 1. Attack complexity has three values indicating increasing complexity from low to high. Attack vector has three values indicating whether attackers can exploit the vulnerability remotely, from hosts on the same physical/logical network, or only locally. Impact degree quantifies effects on confidentiality, integrity, and availability, with values typically provided in vulnerability databases.

Table 1 Value Range of Static Features

Feature	Value Range
Attack Complexity	Low/medium/high
Attack Vector	Network/adjacent/local
Impact	(0,10]

Dynamic features include defense capability, vulnerability repair status, and attack capability. Defense capability represents the network' s ability to defend against attacks exploiting the vulnerability, indicating resistance to vulnerability-induced risks. Vulnerability repair status indicates vendor patch release situations, representing mitigation measures. Attack capability represents the probability of successful exploitation, indicating how attacker skills affect risk. Based on static feature ranges, we define dynamic feature ranges as shown in Table 2.

Table 2 Value Range of Dynamic Features

Feature	Value Range
Defense Capability	[0,1]
Vulnerability Repair	False/part/true
Attack Capability	[0,1]

Feature value specifications are as follows:

a) Defense capability is a probability value representing the network' s defense success rate against attacks exploiting the vulnerability. A value of 0

indicates defense mechanisms cannot detect or block attacks, while 1 indicates successful detection and blocking every time. In practice, this value can be quantified by statistically analyzing historical logs to determine defense success rates against specific vulnerability types.

b) Vulnerability repair status has three categories indicating: no solution available, third-party mitigation available, or official patch available. In practice, if an official patch exists but cannot be installed due to compatibility issues or introduces new vulnerabilities, it should be treated as no official patch (False or Part).

c) Attack capability is a probability value representing the likelihood of successful exploitation. A value of 0 indicates insufficient attacker capability, while 1 indicates guaranteed success. Since risk assessment primarily guides pre-attack remediation, we evaluate potential attacker capabilities based on network characteristics rather than real-time attack analysis. Section 1.2 cited relevant research on practical attack capability assessment.

Following reference [**?**], we assign quantitative values to static features as shown in Table 3 . Impact values directly use NVD-provided scores.

Table 3 Quantization Value of Static Features

Feature	Quantization Value
Attack Complexity	medium: 0.71, high: 0.35
Attack Vector	network: 1.0, adjacent: 0.646, local: 0.395
Impact	slight: 2.9, complete: 10.0

Dynamic feature values for defense capability and attack capability are already quantitative. Vulnerability repair status values (false/part/true) correspond to 0/0.6/1.

Based on these assignments, we propose the vulnerability risk scoring formula:

$$\text{RiskScore} = \text{StaticScore} + \lambda \times \text{DynamicScore}$$

where StaticScore and DynamicScore represent comprehensive scores for static and dynamic features, respectively, and λ is a proportionality coefficient. The calculations are:

$$\text{StaticScore} = \frac{\text{AttackVector} \times \text{Impact}}{\text{AttackComplexity}}$$

$$\text{DynamicScore} = (1-\text{DefenceCapability}) \times (1-\text{VulnerabilityRepair}) \times \text{AttackCapability}$$

Given the value ranges, StaticScore ranges from 0.135 to 2.857, and DynamicScore ranges from 0 to 1, making RiskScore range from 0.135 to 3.857. To adjust RiskScore to a 0-10 range (matching CVSS' s upper bound for conventional use), we calculate $\lambda = 2.592$, yielding:

$$\text{RiskScore} = 2.592 \times (\text{StaticScore} + \text{DynamicScore})$$

The average impact for vulnerability type t is calculated as:

$$I_t = \frac{1}{n_t} \sum_{j=1}^{n_t} I_{ji}$$

where n_t is the total number of vulnerabilities of type t.

2.2 Vulnerability Remediation Strategy Recommendation

After calculating risk scores for all vulnerabilities in a target network, multiple vulnerabilities may still have identical or very similar scores. Traditional remediation strategies assign the same priority to such vulnerabilities, making it difficult to determine repair order. To address this, we propose a vulnerability type-based comparison method that prioritizes vulnerabilities with identical risk scores according to their types, enabling fine-grained remediation strategies. This method statistically analyzes average impact degrees across vulnerability types to establish type priorities. When multiple vulnerabilities share the same risk score, those with higher average impact for their type should be repaired first.

Method 1: Vulnerability Type-Based Risk Comparison and Remediation Strategy Recommendation

a) Collect vulnerability types, quantities, and impact scores from NVD over a specified period.

b) Calculate the average impact degree for each vulnerability type.

c) Rank vulnerability types by average impact degree for prioritizing vulnerabilities with identical risk scores.

d) Use a vulnerability scanner to identify all known vulnerabilities in the target network, with host set H and vulnerability set V.

e) Calculate each vulnerability' s risk score RiskScore.

f) Sort all vulnerabilities by risk score in descending order. For vulnerabilities with score differences less than 0.5, sort by type average impact

degree; if types are identical, treat as equal priority.

g) Output the vulnerability remediation strategy based on the ranking.

The algorithm is represented as follows:

Algorithm 1: Vulnerability Remediation Strategy Recommendation

Input: NVD vulnerability collection V_{re} (containing vulnerability type T and impact I), target network vulnerability collection V.

Output: Vulnerability remediation strategy Seq(V).

```
1. for (t = 1; t   n; t++)
2.   calculate it = (1/nt) * Σ(Iji) for j=1 to nt;
3.   rank It;
4. for (s = 1; s   m; s++)
5.   calculate RiskScore(vs);
6. rank RiskScore(V);
7. PRI(V) = count(RiskScore(V));
8. for (a = 1; a   m; a++)
9.   for (b = a + 1; b   m; b++)
10.    if (|RiskScore(va) - RiskScore(vb)| < 0.5)
11.      if (ita = itb)
12.        set pria = prib;
13.      else exchange(pria, prib);
14.    end if
15.   end for
16. end for
17. return Seq(V);
```

Lines (1)-(3) calculate and rank average impact degrees by vulnerability type, where n is the total number of vulnerability types, i_t is the average impact for type t, n_t is the count of type t vulnerabilities, and I_t is the set of i_t values. Lines (4)-(5) calculate risks for all target network vulnerabilities, where m is the total number of vulnerabilities and v is a vulnerability element. Line (6) ranks all vulnerabilities by risk score, and line (7) establishes the remediation priority set PRI(V). Lines (8)-(16) compare adjacent vulnerabilities: if score differences are less than 0.5 (non-zero), the vulnerability with higher type average impact receives higher priority; if scores are identical, priorities are equal. i_{ta} and i_{tb} represent average impact degrees for vulnerabilities a and b' s types, while pri_a and pri_b are their remediation priorities. Line (17) outputs the final remediation strategy Seq(V) as a recommended repair sequence, with earlier positions indicating higher priority.

3.1 Single Vulnerability Risk Assessment Case

We illustrate our method using CVE-2018-14359 (a buffer overflow vulnerability in Mutt and NeoMutt enabling arbitrary code execution). Table 4 shows partial

information from NVD.

Table 4 Information of CVE-2018-14395

Feature	CVSS 2.0	CVSS 3.0
Attack Complexity	medium	low
Attack Vector	network	network
Impact	complete	complete
CVSS Score	7.5	9.8

CVSS 3.0 significantly reduced the attack complexity compared to 2.0, slightly increased impact, and maintained the same attack vector, resulting in a substantially higher comprehensive score.

Assume this vulnerability exists in a target network without official patches or mitigation solutions, where attackers can successfully exploit it every time. Historical log analysis shows the network' s defense mechanisms have a 0.8 success probability against this vulnerability type.

1) Using CVSS 2.0:
   StaticScore = 0.63 × 1.0 / 0.71 = 0.887
   DynamicScore = (1 - 0.8) × (1 - 0) × 1 = 0.2
   RiskScore = 2.592 × (0.887 + 0.2) = 2.818
2) Using CVSS 3.0:
   StaticScore = 0.63 × 1.0 / 0.35 = 1.800
   DynamicScore = (1 - 0.8) × (1 - 0) × 1 = 0.2
   RiskScore = 2.592 × (1.800 + 0.2) = 5.184

Compared to CVSS scores, our method yields 2.818 (vs. CVSS 2.0' s 7.5) and 5.184 (vs. CVSS 3.0' s 9.8). Despite being a high-risk vulnerability, the network' s effective defense mechanisms significantly reduce its actual risk level below CVSS ratings.

3.2 Multiple Vulnerability Assessment and Remediation Strategy Experiment

To validate our method' s accuracy and effectiveness for multiple vulnerabilities, we used Nessus to scan a host in our experimental network, discovering five vulnerabilities with static features shown in Table 5 .

Table 5 Vulnerabilities on Experimental Host (based on CVSS 2.0 standard)

CVE ID	Attack Complexity	Attack Vector	Impact	CVSS Score
CVE-1999-0499	low	network	partial	5.0

CVE ID	Attack Complexity	Attack Vector	Impact	CVSS Score
CVE-1999-0517	low	network	partial	5.0
CVE-2016-0128	medium	network	partial	4.3
CVE-2017-0143	medium	network	complete	9.3
CVE-2017-0267	medium	network	partial	4.3

Successful exploitation of CVE-1999-0499, CVE-1999-0517, and CVE-2017-0267 causes information disclosure; CVE-2016-0128 enables man-in-the-middle attacks; CVE-2017-0143 allows arbitrary code execution. Based on disclosure and patch dates, we assume official patches are available for CVE-1999-0499, CVE-1999-0517, and CVE-2016-0128; CVE-2017-0143's patch has compatibility issues requiring third-party mitigation; CVE-2017-0267 has no solution. Attackers can successfully exploit all vulnerabilities, with defense probabilities of 0.8 against information disclosure, 0.5 against man-in-the-middle attacks, and 0.2 against arbitrary code execution.

Table 6 Results of Vulnerability Risk Assessment (based on CVSS 2.0 standard)

CVE ID	StaticScore	DynamicScore	RiskScore
CVE-1999-0517	1.268	0.2	3.795
CVE-1999-0499	1.268	0.2	3.795
CVE-2016-0128	0.887	0.08	2.508
CVE-2017-0143	2.007	0.8	7.261
CVE-2017-0267	0.887	1.0	4.894

The results show that despite CVE-2017-0143' s high CVSS 2.0 score of 9.3, its risk level in our target network is only 5.078 due to effective defenses against arbitrary code execution. CVE-1999-0499 and CVE-1999-0517, while causing lower-risk information disclosure, should be repaired next due to low attack complexity and moderate impact, sharing equal priority as they are the same type. CVE-2016-0128 and CVE-2017-0267 pose minimal risk and can be repaired last. The final recommended remediation strategy is: CVE-2017-0143 > CVE-1999-0499 = CVE-1999-0517 > CVE-2016-0128 > CVE-2017-0267.

Additionally, CVE-1999-0499 and CVE-1999-0517 lack CVSS 3.0 information for comparison. Tables 7 and 8 show the CVSS 3.0-based data and assessment results for the remaining three vulnerabilities.

Table 7 Vulnerabilities on Experimental Host (based on CVSS 3.0 standard)

CVE ID	Attack Complexity	Attack Vector	Impact	CVSS Score
CVE-2016-0128	medium	network	low	5.9
CVE-2017-0143	low	network	high	8.1
CVE-2017-0267	medium	network	high	7.5

Table 8 Results of Vulnerability Risk Assessment (based on CVSS 3.0 standard)

CVE ID	StaticScore	DynamicScore	RiskScore
CVE-2016-0128	0.63	0.08	1.838
CVE-2017-0143	2.857	0.8	9.456
CVE-2017-0267	1.8	1.0	7.254

4 Conclusion

The risk level of specific vulnerabilities in particular networks varies with patch availability, defense mechanisms, and attacker capabilities. Considering these dynamic factors directly affects security implementation priorities, this paper proposes a vulnerability risk assessment and mitigation method combining dynamic and static features. By integrating relatively stable features (attack complexity, impact, attack vector) as static features and variable features as dynamic features, we achieve more accurate risk assessment. The results inform network security analysis and vulnerability impact evaluation, enabling prioritized remediation strategy recommendations to guide network hardening.

Future work will investigate additional dynamic features not yet considered and explore their integration to further improve assessment accuracy.

References

[1] Fu Zhiyao, Gao Ling, Sun Qian, et al. Evaluation of vulnerability severity based on rough sets and attributes reduction [J]. Journal of Computer Research and Development, 2016, 53(5): 1009-1017.

[2] Tang Chenghua, Tian Jilong, Tang Shensheng, et al. A risk assessment method for software vulnerability based on GA-FAHP [J]. Computer Science, 2015, 42(9): 134-138.

[3] Huang Chiencheng, Lin Fengyu, Lin Yeongsung, et al. A novel approach to evaluate software vulnerability prioritization [J]. Journal of Systems & Software, 2013, 86(11): 2822-2840.

[4] Li Shancang, Tryfonas T, Russell G, et al. Risk assessment for mobile systems through a multilayered hierarchical Bayesian Network [J]. IEEE Trans on Cybernetics, 2016, 46(8): 1749-1759.

[5] Wiik J, Gonzalez J J, Lipson H F, et al. Dynamics of vulnerability-modeling the life cycle of software vulnerabilities [C]//Proc of the 22th International System Dynamics Conference. 2004.

[6] Ciapessoni E, Cirio D, Kjølle G, et al. Probabilistic risk-based security assessment of power systems considering incumbent threats and uncertainties [J]. IEEE Trans on Smart Grid, 2016, 7(6): 2890-2903.

[7] Yang Guotai, Wang Yufei, Luo Jianbo, et al. Vulnerability of power CPS information network and its evaluation method [J]. Electric Power, 2018, 51(1): 83-89.

[8] Allodi L, Massacci F. Comparing vulnerability severity and exploits using case-control studies [J]. ACM Trans on Information & System Security, 2014, 17(1): 1-20.

[9] Younis A, Malaiya Y K, Ray I. Evaluating CVSS base score using vulnerability rewards programs [C]//Proc of International Information Security and Privacy Conference. Switzerland: Springer International Publishing, 2016: 62-75.

[10] Johnson P, Lagerstrom R, Ekstedt M, et al. Can the common vulnerability scoring system be trusted? a Bayesian analysis [J]. IEEE Trans on Dependable & Secure Computing, 2016, PP(99): 1-1.

[11] Ruohonen J, Holvitie J, Hyrynsalmi S, et al. Exploring the clustering of software vulnerability disclosure notifications across software vendors [C]//Proc of the 13th IEEE/ACS International Conference of Computer Systems and Applications. Piscataway, NJ: IEEE Press, 2016.

[12] Li Wei. Application of vulnerability quantification scoring method in Telecom security strategy [J]. Network Security Technology and Application, 2015, 2(2): 27-28.

[13] Wang Zuoguang, Wei Qiang, Liu Wenwen. Quantitative risk assessment of industrial control systems based on attack-tree and CVSS [J]. Application Research of Computers, 2016, 33(12): 3785-3790.

[14] Rick V H. The motivation of attackers in attack tree analysis [D]. Delft, Holland: Delft University of Technology, 2015.

[15] Jaafor O, Birregah B. Multi-layered graph-based model for social engineering vulnerability assessment [C]//Proc of IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. New York: ACM Press, 2016: 1480-1488.

[16] Durkota K, Lisý V, Bošanský B, et al. Approximate solutions for attack graph games with imperfect information [C]//Proc of International Conference on Decision and Game Theory for Security. Switzerland: Springer, 2015: 228-249.

[17] Durkota K, Lisý V, Kiekintveld C, et al. Case studies of network defense with attack graph games [J]. IEEE Intelligent Systems, 2016, 31(5): 24-30.

[18] Fadlallah A, Sbeity H, Malli M, et al. Application of attack graphs in intrusion detection systems: an implementation [J]. International Journal of Computer Networks, 2016, 8(1): 1-12.

[19] Pieters W, Davarynejad M. Calculating adversarial risk from attack trees: Control strength and probabilistic attackers [M]//Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance. Switzerland: Springer, 2015: 201-215.

[20] Liu Qixu, Zhang Chongbin, Zhang Yuqing, et al. Research on key technology of vulnerability threat classification [J]. Journal on Communications, 2012, 33(S1): 79-87.

Note: Figure translations are in progress. See original paper for figures.