Towards Vulnerabilities Caused by Application Configuring: A Meta Analysis of the National Vulnerability Database

Richard May, Xenia Marlene Zerweck

Harz University of Applied Sciences, Department of Automation and Computer Science

Abstract

Configuring applications might lead to diverse security issues. In this study, 263 vulnerabilities of the National Vulnerability Database were analyzed, which were caused by application configuring. Seven key characteristics and three research opportunities were proposed. In most cases, such vulnerabilities emerge in clients of web applications and their network components with a medium-high severity, typically leading to access, data manipulation, and resulting system errors.

1. Introduction

Due to an increasing variety of customer demands regarding customized software and associated features, the amount of configurable software is constantly growing. However, configurability leads to a higher software complexity, which in turn results in diverse security risks, e.g., due to feature interactions or configuration errors [1,2]. In recent years, the number of cyber attacks increased significantly, ranging from gaining access to protected data, e.g., brute force attacks, to spying and extorting, e.g., ransomware [3,4]. Cyber attacks are often based on the exploitation of system vulnerabilities, usually as a result of any kind of fault during a system's life-cycle [5]. Exploiting vulnerabilities can cause serious consequences, such as unauthorized data access or fatal system errors [6,7].

Information related to vulnerabilities is collected in a variety of sources, e.g., vulnerability and exploit databases or security advisories [8,9]. One of the most common vulnerability databases is the *National Vulnerabilty Database (NVD)* provided by the US National Institute of Standards and Technology [10,11]. The NVD lists thousands of Common Vulnerability and Exposures (CVE), which are unique identifiers for indexing and characterizing vulnerabilities [12]. These CVE are categorized by Common Weakness Enumerations (CWE) and rated regarding their severity by a numerical score (0–10), called Common Vulnerability Scoring System (CVSS) [13,14]. Referring to the NVD a prime body of knowledge on security-related information is aggregated and publicly available, providing a highly valuable basis for analyses to explore trends and relationships [15].

2. Methodology

Problem statement and research objectives. In recent years, vulnerabilities and their exploitation have been extensively investigated in literature, e.g., in terms of a trend analysis of the NVD [16]. However, despite the existing research, there is currently no comparable study focusing solely on application configuring-related vulnerabilities of any database. We argue that it is highly relevant to analyze and understand especially the relationships between vulnerabilities and configuration issues due to the current trend towards increasing configurability and the growing number of cyber attacks, e.g., in manufacturing-related machine-learning systems [17–20]. We conducted a meta analysis of the NVD, focusing on CWE-16 as category for vulnerabilities primarily caused by application configuring. Our research objective is *to characterize these vulnerabilities to explore the relationships between vulnerabilities and application configuring*.

Analysis design and conduct. First, we conducted an automated search for vulnerabilities tagged with CWE-16, i.e., vulnerabilities caused by application configuring. Our focus was on all CVE listed in the NVD. The search was performed by using the NVD API and resulted in 263 CVE without duplicates. We argue that this number of CVE is sufficient to derive deep insights and valuable conclusions regarding common characteristics. Second, the fetched data was qualitatively analyzed to extract and classify trends, characteristics, and interdependencies. Specifically, we developed a Python 3.9 tool to create a word cloud and determine relationships between 1) CVSS score, 2) release date, and 3) date of last modification. In addition, the CVE were classified in a collaborative Excel spreadsheet based on the authors' expertise. More specifically, the descriptions were analyzed regarding 1) the location or device (e.g., mobile device), 2) the affected application field (e.g., operating system), and 3) the major party related to the vulnerability (e.g., host or client).

3. Results

Time and severity analysis. Referring to the publishing dates, most CVE were created between 2007 and 2013 (81%). However, since CVE characteristics usually change over time, we argue that especially the last modification is more relevant. Most modifications were made in 2017 and 2018 (62%). Precisely, CVE descriptions changed about seven years (6.96) after their publication. The considered CVE have an average CVSS score of 6.1, showing a medium-high severity. The lowest CVSS score is 1.9 (2 times, e.g., pop-up message reading) and the highest is 10.0 (20 times, e.g., cross-site scripting).

Description analysis. As shown in Figure 1, most CVE (71%) refer to a computer, e.g., a personal computer. 22% of the CVE are related to a network architecture, e.g., servers. Only 5% focus on routers and 3% on mobile devices. The affected application fields are more diverse, ranging from desktop applications (30%, e.g., Adobe Acrobat Reader) over web applications (23%, e.g., Mozilla Firefox) to operating systems (21%, e.g., Microsoft Windows), servers (17%, e.g., Apache Tomcat), as well as more network and communication-related features (9%, e.g., proxy settings). Most CVE (73%) refer to an affected client, while all others are related to the host.

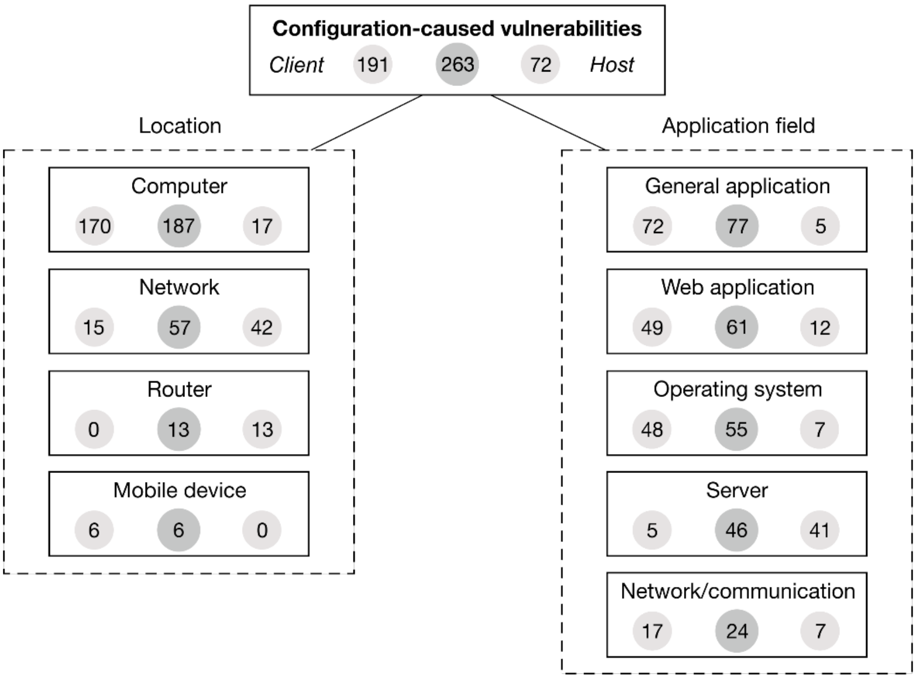

Figure 1: Overview of the vulnerability location, application field, and the affected parties (numbers indicate amount of corresponding CVE with: left client, middle total, and right host).

These observations are emphasized and extended by the word cloud (see Figure 2). Precisely, most common terms are related to network access (e.g., remote (attacker), port, HTTP) as well as to the affected party (e.g., local, user). Interestingly, relevant application fields are also mentioned (e.g., code, file, OS). Others refer more to the CWE-related properties of attacks (e.g., arbitrary, request) or configuration-related characteristics (e.g., configuration, default, version).

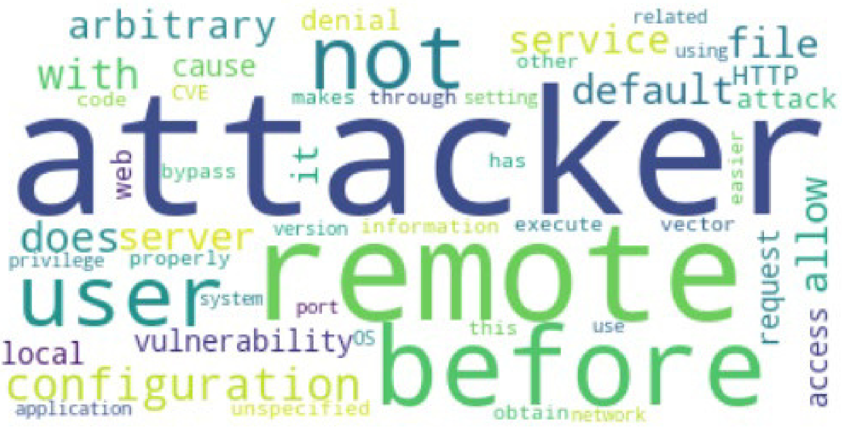

Figure 2: Word cloud consisting of the most common terms of CVE tagged with CWE-16.

4. Challenges and Research Opportunities

Based on the extracted data, we characterize vulnerabilities caused by configuring as follows:

- The **affected systems** are usually applications running on the local computer that provide an Internet connection (e.g., for updates) or web applications running on an internal or external network (e.g., web browser or content management systems). Applications running on local systems are especially sensitive to interactions with their underlying operating system.

- The **vulnerable system components** refer mainly to network components, which can be exploited by remote access due to invalid application configurations, often even default configurations, or inter-dependencies between existing features and newly added features by version updates.

- The **affected parties** are primarily clients, including local applications on a computer or mobile device, password-protected accounts, or web interfaces.

- The **types of cyber attack**s relate to highly diverse and dynamic attacks, ranging from brute-force attacks over side-channel attacks to session hijacking. The identification of trends is not possible.

- The **vulnerability severity** is medium-high, implying that cyber attacks exploiting vulnerabilities caused by configuring have a significant impact on affected systems.

- The **impact of the vulnerability exploits** is highly diverse but can categorized by common impacts. Most vulnerabilities refer to application and data access, manipulation of data, or system errors. Note that data-related impacts depend on successful system-related access by the attacker.

- The **timely relevance** is high as implied by last modifications of the vulnerabilities after an average of seven years.

Based on the findings of the analysis and the characteristics identified, there arise three important research opportunities (RO):

RO1 Since vulnerabilities, cyber attacks, and exploit impacts are highly diverse, security solutions are needed that cover as much diversity as possible. System components that are particularly susceptible to application configuration errors (e.g., evolutionary issues) should be considered in particular. So, research is needed to develop dynamic security patterns, which address configurability and its interactions during the systems' life-cycle.

RO2 The NVD data does not allow conclusions regarding the actual CVE solution, as the last modification does not represent the closure of a CVE. Furthermore, the modification peak in 2017 and 2018 is not explainable within the scope of this study. However, we assume that CWE-16 was increasingly used for tagging at this time, whereas today it may only be used as a general classifier for CWE representing different facets of configurability. Thus, it is recommended to extend our study, including other (partly) configuration-related CWE.

RO3 To analyze more facets and derive more significant characteristics, a higher amount of vulnerability databases should be considered. However, existing databases do not provide a uniform basis of the data and associated classification systems [5]. Therefore, we strongly recommend the development of a taxonomy for the unform synthesis of different databases.

5. Conclusion

In our study we analyzed the NVD to characterize vulnerabilities caused by application configuring. More specifically, we extracted seven essential characteristics and proposed three relevant opportunities for future research. We argue that our results provide a highly valuable basis for both researchers and practitioners especially regarding future research in the context of identifying concrete relationships and impacts of configuration-related aspects, software vulnerabilities, and their exploitation. Further studies are planned, especially in the context of conducting uniform database analyses.

6. Contact

Friedrichstraße 57 – 59, 38855 Wernigerode
Richard May
rmay@hs-harz.de / 0000-0001-7186-404X
https://richardmay.de/

7. References

[1] Apel, S., Batory, D., Kästner, C., & Saake, G. (2016): Feature-oriented software product lines, Heidelberg: Springer.

[2] May, R., Biermann, C., Krüger, J., Saake, G., & Leich, T. (2022): A systematic mapping study of security concepts for configurable data storages, International Systems and Software Product Line Conference, 108–119.

[3] Bendovschi, A. (2015): Cyber-attacks–trends, patterns and security countermeasures, in: Procedia Economics and Finance, 28, 24–31.

[4] Conti, M., Dargahi, T., & Dehghantanha, A. (2018): Cyber threat intelligence: challenges and opportunities, in: Cyber Threat Intelligence, 1–6.

[5] Kenner, A. (2020): Model-Based Evaluation of Vulnerabilities in Software Systems, International Systems and Software Product Line Conference, 112–119.

[6] Elhakeem, Y. F. G. M., & Barry, B. I. (2013): Developing a security model to protect websites from cross-site scripting attacks using ZEND framework application, in: International Conference on Computing, in: Electrical and Electronic Engineering, 624–629.

[7] Bhattacharjee, S., Salimitari, M., Chatterjee, M., Kwiat, K., & Kamhoua, C. (2017): Preserving data integrity in iot networks under opportunistic data manipulation, International Conference on Dependable, Autonomic and Secure Computing, 446–453.

[8] Mulwad, V., Li, W., Joshi, A., Finin, T., & Viswanathan, K. (2011): Extracting information about security vulnerabilities from web text, International Conferences on Web Intelligence and Intelligent Agent Technology, 257–260.

[9] Kenner, A., Dassow, S., Lausberger, C., Krüger, J., & Leich, T. (2020): Using variability modeling to support security evaluations: virtualizing the right attack scenarios, International Working Conference on Variability Modelling of Software-Intensive Systems, 1–9.

[10] Booth, H., Rike, D., & Witte, G. A. (2013): The national vulnerability database (NVD): Overview, in: ITL Bulletin, 1–3.

[11] Zhang, S., Ou, X., & Caragea, D. (2015): Predicting cyber risks through national vulnerability database, in: Information Security Journal: A Global Perspective, 24(4-6), 194–206.

[12] Neuhaus, S., & Zimmermann, T. (2010): Security trend analysis with cve topic models, International Symposium on Software Reliability Engineering, 111–120.

[13] Wang, J. A., Xia, M., & Zhang, F. (2008): Metrics for information security vulnerabilities, in: Journal of Applied Global Research, 1(1), 48–58.

[14] Kanakogi, K., Washizaki, H., Fukazawa, Y., Ogata, S., Okubo, T., Kato, T., & Yoshioka, N. (2021): Tracing cve vulnerability information to capec attack patterns using natural language processing techniques, in: Information, 12(8), 2098–3013.

[15] Williams, M. A., Dey, S., Barranco, R. C., Naim, S. M., Hossain, M. S., & Akbar, M. (2018): Analyzing evolving trends of vulnerabilities in national vulnerability database, International Conference on Big Data, 3011–3020.

[16] Kuhn, D. R., Raunak, M. S., & Kacker, R. (2017): An Analysis of Vulnerability Trends 2008-2016, International Conference on Software Quality, Reliability and Security Companion, 587–588.

[17] Kenner, A., May, R., Krüger, J., Saake, G., & Leich, T. (2021): Safety, security, and configurable software systems: a systematic mapping study, International Systems and Software Product Line Conference, 148–159.

[18] May, R. (2022): Security and configurable storage systems in Industry 4.0 environments: A systematic literature study, in: Open Conference Proceedings, 151–156.

[19] May, R., Biermann, C., Zerweck, X. M., Ludwig, K., Krüger, J., & Leich, T. (2024): Vulnerably (mis)configured? Exploring 10 years of developers' Q&As on Stack Overflow, in: International Working Conference on Variability Modelling of Software-Intensive Systems, 112–122.

[20] May, R., Alex, A. J., Suresh, R., & Leich, T. (2024): Product-line engineering for smart manufacturing: A systematic mapping study on security concepts, in: International Conference on Software Technologies, 1–8.