Educational Administration: Theory and Practice
2022, 28(04), 718 - 724
ISSN: 2148-2403
https://kuey.net/

Research Article

Vulnerability Management in Computer Systems: Challenges and Approaches

Nayan Goel*

*Senior Application Security Engineer, Upgrade, Inc., Foster City, California, USA.

Citation: *Nayan Goel, Vulnerability Management in Computer Systems: Challenges and Approaches , Educational Administration: Theory and Practice, 28(04) 718 - 724*
Doi: 10.53555/kuey.v28i4.11607

ARTICLE INFO

ABSTRACT

Vulnerability management is an important part of computer system security, which tends to discover physical vulnerabilities, evaluate them, and address them to limit them to be exploited by harmful actors. With the increase in the complexity, interconnection, and distribution of computer systems in the cloud and hybrid environments, vulnerabilities have become difficult to manage. The timely remediation efforts are complicated by the fact that the new threats emerge quickly, the vulnerabilities are being disclosed more, the organizational resources are quite limited, and the operations are constrained. This paper explores the notion of vulnerability management in computer systems, its fundamental processes, significant issues about the matter and current solutions applicable in managing security vulnerabilities. It puts a focus on vulnerability management lifecycle with structured vulnerability management, prioritization based on risk, automation and alignment with organizational security practices. The abstract also emphasizes the need to have proactive and ongoing vulnerability management approaches to make the system more resilient, minimize exposure to cyber threats, and benefit the overall information security goals.

Keywords: Vulnerability management, computer systems security, risk assessment, patch management, cybersecurity, threat mitigation

I. Introduction

The growing reliance on computer systems to facilitate important organizational, economical and social processes has elevated cybersecurity to the center of interest in the contemporary computing facilities. Vulnerability management is one of the several aspects of cybersecurity, but it is a critical component since it offers a methodical approach to the process of detecting, evaluating, ranking, and addressing the vulnerabilities to security that may be used by hackers. They can be caused by software bugs, system setups, and flaws in design, and when exploited, lead to information breaches, service failures, and massive financial and reputational damages (Pfleeger and Pfleeger, 2012; Foreman, 2019).

With the maturity up to complexity in computing infrastructures, brought about by cloud computing, Internet of Things (IoT) deployments, industrial controls, and large-scale enterprises, vulnerability management has become more difficult. The present systems are extremely interconnected and dynamic, and it is hard to have a comprehensive visibility and timely corrective action on all assets. It has been found out that due to the increasing vulnerabilities reported, as well as the lack of security resources and operational constraints, mitigation efforts are often delayed or marginal (Kotha, 2015; Walkowski et al., 2021). This state of affairs adds to the attack surface and exposes the systems to existing and emerging threats.

Vulnerability management is not just limited to mere vulnerability detection but must be carried out in a way that informed risk assessment and the related decisions are made in order to find a balance between the security requirement and system availability and performance. Different models and frameworks are suggested to facilitate this process, and these are standardized scoring systems, such as the Common Vulnerability Scoring System (CVSS), that contribute to the prioritization of remediation actions (Walkowski et al., 2021). As well, vulnerabilities mitigation strategies can be subdivided into technical, organizational, and procedural steps, such as the development of programs secure at code level and automatic patches or the implementation of policies and constant monitoring (Shahriar and Zulkernine, 2012; Ye et al., 2021).

The relevance of vulnerability management is further emphasized in specialized domains such as cloud computing, IoT, and critical infrastructure systems, where vulnerabilities can have cascading and large-scale

consequences. Studies on cloud environments, IoT ecosystems, and supervisory control and data acquisition (SCADA) systems highlight unique vulnerability characteristics and the need for tailored management strategies (Grobauer et al., 2010; Ten et al., 2008; Anand et al., 2020). Consequently, vulnerability management is increasingly viewed as a continuous, proactive, and integral component of overall cybersecurity strategy rather than a one-time or reactive activity (Lai & Hsia, 2007).

In this context, understanding the challenges and approaches associated with vulnerability management in computer systems is essential for improving security posture and resilience. This section introduces the foundational concepts of vulnerability management, setting the stage for a detailed examination of its challenges and the methods employed to address them.

II. Types of Vulnerabilities

Vulnerabilities in computer systems represent weaknesses that may be exploited to compromise system security, functionality, or data integrity. These vulnerabilities arise from diverse sources, including design flaws, implementation errors, misconfigurations, and operational practices. Understanding their types is essential for effective vulnerability management, as it enables systematic identification, assessment, and remediation strategies (Pfleeger & Pfleeger, 2012; Foreman, 2019).

2.1 Software Vulnerabilities

Software vulnerabilities originate from defects in application code, operating systems, and supporting libraries. Common examples include buffer overflows, injection flaws, insecure authentication mechanisms, and improper error handling. Such weaknesses often stem from poor coding practices or insufficient testing and may persist throughout the software lifecycle if not adequately addressed (Shahriar & Zulkernine, 2012). According to Kotha (2015), software vulnerabilities constitute a significant proportion of reported security issues due to the widespread reliance on complex and rapidly evolving software systems.

2.2 Hardware and Firmware Vulnerabilities

Hardware and firmware vulnerabilities arise from flaws embedded in physical components or low-level system software such as BIOS and device firmware. These vulnerabilities are particularly critical because they operate below the operating system level, making detection and remediation more challenging. Exploitation of such weaknesses can lead to persistent threats that bypass traditional security controls (Foreman, 2019). Lai and Hsia (2007) emphasize that hardware-related vulnerabilities can undermine network security even when higher-level protections are in place.

2.3 Network Vulnerabilities

Network vulnerabilities are associated with weaknesses in network architecture, protocols, and configurations. Examples include unsecured communication channels, weak encryption, improper firewall rules, and exposed services. These vulnerabilities can be exploited to intercept data, launch denial-of-service attacks, or gain unauthorized access to systems (Pfleeger & Pfleeger, 2012). Effective network vulnerability assessment is therefore a cornerstone of enterprise security management (Kotha, 2015).

2.4 Cloud Computing Vulnerabilities

Cloud environments introduce unique vulnerabilities due to shared resources, virtualization technologies, and multi-tenancy models. Issues such as insecure APIs, data leakage, improper access controls, and hypervisor vulnerabilities are common concerns in cloud computing systems (Grobauer et al., 2010). The dynamic and scalable nature of cloud infrastructures further complicates visibility and control, increasing the likelihood of misconfigurations (Foreman, 2019).

2.5 Internet of Things (IoT) Vulnerabilities

IoT vulnerabilities arise from constrained device resources, weak authentication, lack of regular updates, and heterogeneous system architectures. Many IoT devices are deployed with minimal security controls, making them attractive targets for attackers (Anand et al., 2020). These vulnerabilities can have cascading effects, particularly in smart environments and critical infrastructures.

2.6 Critical Infrastructure and SCADA Vulnerabilities

Supervisory Control and Data Acquisition (SCADA) and other industrial control systems face vulnerabilities related to legacy technologies, limited patching capabilities, and real-time operational requirements. Exploitation of such vulnerabilities can lead to severe physical and economic consequences (Ten et al., 2008). As noted by Walkowski et al. (2021), standardized scoring and assessment mechanisms are essential for prioritizing risks in these environments.

2.7 Summary of Vulnerability Types

Table 1 provides a consolidated overview of major vulnerability types, their sources, and typical impacts.

Table 1: Major Types of Vulnerabilities in Computer Systems

Vulnerability Type	Primary Source	Typical Impact
Software	Coding flaws, insecure libraries	Data breaches, privilege escalation
Hardware/Firmware	Design flaws, outdated firmware	Persistent system compromise
Network	Misconfigurations, weak protocols	Unauthorized access, service disruption
Cloud Computing	Insecure APIs, multi-tenancy risks	Data leakage, account hijacking
IoT	Weak authentication, lack of updates	Large-scale attacks, privacy loss
SCADA/Critical Systems	Legacy systems, limited patching	Operational failure, physical damage

Overall, vulnerabilities manifest across all layers of computer systems, from hardware to applications and networks. A comprehensive vulnerability management program must therefore account for these diverse vulnerability types to ensure effective risk mitigation and system resilience (Kotha, 2015; Walkowski et al., 2021).

III. Vulnerability Management Lifecycle

The vulnerability management lifecycle represents a structured and continuous process through which organizations identify, evaluate, and remediate security weaknesses within computer systems. Rather than a one-time activity, it is an ongoing cycle designed to adapt to evolving threats, system changes, and operational requirements. An effective lifecycle ensures that vulnerabilities are addressed systematically and prioritized based on risk and impact (Kotha, 2015; Foreman, 2019).

Asset Identification and Classification.

The lifecycle begins with identifying and cataloguing all assets within the computing environment, including hardware, software, networks, applications, and data repositories. Accurate asset classification allows organizations to understand system criticality and exposure, forming the foundation for effective vulnerability assessment. Without comprehensive visibility into system components, vulnerabilities may remain undetected or improperly prioritized (Pfleeger & Pfleeger, 2012; Lai & Hsia, 2007).

Vulnerability Discovery and Assessment.

Once assets are identified, vulnerabilities are discovered through techniques such as automated scanning, configuration analysis, penetration testing, and threat intelligence review. These activities help detect known and emerging weaknesses across operating systems, applications, networks, and cloud environments. Standardized frameworks and scoring systems, such as the Common Vulnerability Scoring System (CVSS), are often employed to assess severity and exploitability, enabling consistent evaluation across systems (Walkowski et al., 2021; Grobauer et al., 2010).

Risk Analysis and Prioritization.

Not all vulnerabilities pose the same level of risk; therefore, effective prioritization is essential. This phase involves analyzing vulnerabilities based on factors such as severity, likelihood of exploitation, asset value, and potential business impact. Risk-based prioritization supports efficient allocation of limited security resources and ensures that the most critical vulnerabilities are addressed first (Kotha, 2015; Foreman, 2019). In specialized environments such as IoT and SCADA systems, contextual risk considerations are particularly important due to safety and operational constraints (Ten et al., 2008; Anand et al., 2020).

Remediation and Mitigation.

Remediation involves eliminating vulnerabilities through actions such as patch deployment, code correction, system reconfiguration, or component replacement. Where immediate remediation is not feasible, mitigation controls such as access restrictions, monitoring, or compensating security measures are applied to reduce risk. Patch management remains a central challenge, particularly in large-scale and legacy systems, prompting increasing interest in automated and scalable patch assessment techniques (Shahriar & Zulkernine, 2012; Ye et al., 2021).

Verification, Monitoring, and Reporting.

After remediation or mitigation, systems must be re-evaluated to verify that vulnerabilities have been effectively addressed. Continuous monitoring ensures that new vulnerabilities are promptly detected as systems evolve or new threats emerge. Documentation and reporting provide visibility into vulnerability

trends, remediation effectiveness, and compliance with security policies, supporting informed decision-making and continuous improvement (Foreman, 2019; Pfleeger & Pfleeger, 2012).

Overall, the vulnerability management lifecycle provides a disciplined and repeatable approach to reducing security risks in computer systems. By integrating identification, assessment, prioritization, remediation, and continuous monitoring, organizations can enhance resilience and maintain a proactive security posture in the face of dynamic threat environments.

IV. Key Challenges in Vulnerability Management

Despite its central role in securing computer systems, vulnerability management faces persistent and multifaceted challenges that limit its effectiveness. These challenges arise from technical, organizational, and environmental factors and are further amplified by the scale and complexity of modern computing infrastructures.

One major challenge is the rapid growth in the number of disclosed vulnerabilities. Public vulnerability repositories and security advisories continuously report new weaknesses, making it difficult for organizations to maintain an up-to-date security posture. According to Foreman (2019), security teams are often overwhelmed by the volume of vulnerabilities, many of which compete for limited remediation resources. This situation leads to delayed patching and increased exposure to attacks.

Another critical issue is accurate vulnerability identification and assessment. Vulnerability scanning tools may generate false positives or fail to detect context-specific weaknesses, reducing confidence in assessment results. Pfleeger and Pfleeger (2012) emphasize that vulnerabilities cannot be evaluated in isolation; their impact depends on system configuration, threat likelihood, and existing countermeasures. Inaccurate assessments may result in misallocation of security efforts.

Prioritization of vulnerabilities also remains a significant challenge. While standardized scoring systems such as the Common Vulnerability Scoring System (CVSS) provide a baseline for severity ranking, they often fail to capture organizational context, asset criticality, or real-world exploitability. Walkowski, Oko, and Sujecki (2021) note that reliance on generic scores can cause organizations to prioritize vulnerabilities that pose minimal actual risk while neglecting more critical ones.

Operational constraints further complicate vulnerability management, particularly in environments requiring high availability. Applying patches may introduce system instability, compatibility issues, or downtime. Shahriar and Zulkernine (2012) highlight that fear of disrupting production systems frequently leads organizations to postpone remediation, thereby prolonging vulnerability exposure.

The heterogeneity of modern computing environments presents additional difficulties. Organizations increasingly rely on cloud services, Internet of Things (IoT) devices, and industrial control systems, each with distinct vulnerability profiles and management requirements. Research by Grobauer, Walloschek, and Stocker (2010) and Anand et al. (2020) demonstrates that traditional vulnerability management approaches are often insufficient for cloud-based and IoT systems due to shared responsibility models, limited visibility, and device constraints.

Finally, human and organizational factors play a substantial role. Lack of skilled personnel, insufficient security awareness, and weak governance structures can undermine vulnerability management initiatives. Kotha (2015) and Lai and Hsia (2007) argue that effective vulnerability management requires not only technical solutions but also strong policies, coordination across teams, and continuous training.

Table 2: Major Challenges in Vulnerability Management

Challenge	Description	Implications	Key Sources
Volume of vulnerabilities	Continuous disclosure of new vulnerabilities	Remediation delays and increased attack surface	Foreman (2019); Kotha (2015)
Inaccurate assessment	False positives and context-insensitive scanning	Misprioritization of security efforts	Pfleeger & Pfleeger (2012)
Prioritization limitations	Overreliance on generic severity scores	Ineffective risk management	Walkowski et al. (2021)
Patch deployment risks	Downtime and system instability concerns	Deferred remediation	Shahriar & Zulkernine (2012)
Diverse environments	Cloud, IoT, and SCADA-specific vulnerabilities	Limited visibility and control	Grobauer et al. (2010); Anand et al. (2020); Ten et al. (2008)

Human and organizational factors	Skills gaps and weak governance	Reduced program effectiveness	Kotha (2015); Lai & Hsia (2007)

These challenges demonstrate that vulnerability management is not a purely technical task but a continuous, risk-driven process requiring adaptive strategies, organizational commitment, and contextual decision-making to remain effective in dynamic threat environments.

V. Approaches and Tools

Effective vulnerability management relies on a combination of systematic approaches and supporting tools designed to identify, assess, prioritize, and remediate security weaknesses across computer systems. Scholars emphasize that no single method is sufficient; instead, organizations must adopt layered and adaptive strategies aligned with their system architecture and risk profile (Kotha, 2015; Foreman, 2019).

One widely adopted approach is automated vulnerability scanning, which uses specialized tools to continuously inspect systems, networks, and applications for known vulnerabilities. These tools compare system configurations and software versions against vulnerability databases, enabling organizations to detect weaknesses efficiently and at scale (Foreman, 2019). However, automated scanning is often complemented by manual assessment and penetration testing, which provide deeper insights into complex or context-specific vulnerabilities that automated tools may overlook (Pfleeger & Pfleeger, 2012).

Another key approach is risk-based vulnerability prioritization, which focuses on assessing vulnerabilities according to their potential impact and likelihood of exploitation rather than treating all vulnerabilities equally. The Common Vulnerability Scoring System (CVSS) is frequently used to standardize severity assessment and support decision-making regarding remediation priorities (Walkowski et al., 2021). This approach helps organizations allocate limited resources more effectively while reducing overall exposure to critical threats.

Patch and configuration management tools play a central role in remediation efforts. These tools automate the deployment of security patches and enforce secure configurations across systems, reducing the window of exposure between vulnerability discovery and mitigation (Shahriar & Zulkernine, 2012; Ye et al., 2021). In emerging environments such as cloud computing, IoT, and industrial control systems, specialized tools and frameworks are required to address unique vulnerability characteristics related to scalability, heterogeneity, and availability constraints (Grobauer et al., 2010; Anand et al., 2020; Ten et al., 2008).

The table below summarizes major vulnerability management approaches and commonly associated tools or techniques.

Table 3: Major Vulnerability Management Approaches and Tools

Approach	Description	Typical Tools / Techniques	Key References
Automated Vulnerability Scanning	Continuous identification of known vulnerabilities in systems and networks	Network scanners, host-based scanners, vulnerability databases	Foreman (2019); Lai & Hsia (2007)
Manual Assessment & Penetration Testing	In-depth analysis of system weaknesses through simulated attacks	Penetration testing frameworks, expert analysis	Pfleeger & Pfleeger (2012)
Risk-Based Prioritization	Ranking vulnerabilities based on severity, impact, and exploitability	CVSS scoring, risk assessment models	Walkowski et al. (2021); Kotha (2015)
Patch & Configuration Management	Automated remediation and secure configuration enforcement	Patch management systems, configuration tools	Shahriar & Zulkernine (2012); Ye et al. (2021)
Domain-Specific Assessment	Tailored vulnerability analysis for specialized systems (e.g., cloud, IoT, SCADA)	Cloud security tools, IoT assessment frameworks	Grobauer et al. (2010); Anand et al. (2020); Ten et al. (2008)

Overall, the integration of automated tools with structured assessment approaches and contextual risk analysis is essential for effective vulnerability management. Prior studies consistently highlight that combining technical tools with strategic processes improves resilience and enables organizations to respond more effectively to evolving security threats (Kotha, 2015; Foreman, 2019).

VI. Best Practices

Effective vulnerability management requires the adoption of structured, continuous, and risk-informed practices that align technical controls with organizational processes. One widely recognized best practice is the establishment of a formal vulnerability management lifecycle that integrates identification, assessment, remediation, and verification activities. Such a lifecycle ensures consistency and accountability while enabling organizations to respond systematically to newly discovered vulnerabilities (Kotha, 2015; Foreman, 2019).

Continuous and automated vulnerability scanning is another critical practice. Given the scale and complexity of modern computer systems, manual approaches are insufficient for timely detection. Automated tools enable frequent assessments across networks, applications, cloud platforms, and specialized systems, thereby improving visibility and reducing exposure windows (Lai & Hsia, 2007; Grobauer et al., 2010). However, scanning results should be carefully validated to minimize false positives and support informed decision-making.

Risk-based prioritization is essential to managing the large volume of identified vulnerabilities. Best practices emphasize evaluating vulnerabilities based on severity, exploitability, asset criticality, and potential business impact rather than treating all findings equally. The use of standardized scoring frameworks, such as vulnerability severity metrics, supports consistent prioritization and efficient allocation of remediation resources (Walkowski et al., 2021; Pfleeger & Pfleeger, 2012).

Timely patch management and remediation planning also represent core best practices. Organizations should implement controlled patch deployment processes that balance security urgency with operational stability. Where immediate patching is not feasible, compensating controls such as configuration hardening, access restrictions, or network segmentation should be applied to reduce risk (Shahriar & Zulkernine, 2012; Ye et al., 2021).

Integration of vulnerability management with broader security and development practices further strengthens effectiveness. Embedding vulnerability assessment into system design, software development, and operational workflows supports early detection and prevention, particularly in environments such as IoT, industrial control systems, and cloud infrastructures (Anand et al., 2020; Ten et al., 2008). This integration promotes a proactive security posture rather than reactive remediation.

Finally, organizational governance and human factors play a decisive role in successful vulnerability management. Clear policies, defined roles and responsibilities, regular security training, and management support are necessary to sustain continuous improvement. Aligning technical practices with organizational awareness and compliance requirements enhances resilience and ensures that vulnerability management remains an ongoing, strategic security function (Foreman, 2019; Kotha, 2015).

VII. Conclusion

Vulnerability management is a classic support block of computer system protection, especially in a time marked by growing system intricacy, boundaryless network connections and threat vectors which are changing fast. This paper has revealed that vulnerability management is not just about the simple identification of vulnerabilities in a system but involves the process of system weaknesses systematically evaluated, prioritized, remediated, and monitored. The approach to systematic and proactive vulnerability management strategies, as highlighted by Kotha (2015) and Foreman (2019), can enable organizations to mitigate surfaces of attacks and improve the general security posture.

The continued existence of the problems like high rate of weaknesses, scarce resources available to remediate them, and constraints of operations highlights why risk-based and automated strategies are required. Common Vulnerability Scoring System models are based on standardized measurements and can offer a more stable prioritization and decision-making platform (Walkowski, Oko, and Sujecki, 2021). Moreover, the threat vulnerability countermeasure model also emphasizes the need to align the vulnerability management efforts to larger security goals (Pfleeger and Pfleeger, 2012).

New computing models, such as cloud computing, IoT systems, and industrial control systems, create more levels of complexity and require dynamic and situational vulnerability management practices (Grobauer, Walloschek, and Stocker, 2010; Anand et al., 2020; Ten, Liu, and Manimaran, 2008). The development of automated vulnerability testing and patch testing is a promising avenue towards making remediation more scaled and timely (Ye, Martinez, and Monperrus, 2021) but human oversight and organizational governance cannot be withheld (Shahriar and Zulkernine, 2012).

To sum up, vulnerability management entails an integrated approach, continuous, and risk-oriented solution that considers both technical solutions and standardized assessment models as well as organizational commitment. It is necessary to enhance these factors to improve the resilience of the system, curb cyber risks, and maintain secure computing environments in the light of emerging threats (Lai and Hsia, 2007).

VIII. References

1. Kotha, N. R. (2015). Vulnerability Management: Strategies, Challenges, and Future Directions. NeuroQuantology, 13(2), 269-275.
2. Foreman, P. (2019). Vulnerability management. Auerbach Publications.
3. Pfleeger, C. P., & Pfleeger, S. L. (2012). Analyzing computer security: A threat/vulnerability/countermeasure approach. Prentice Hall Professional.
4. Walkowski, M., Oko, J., & Sujecki, S. (2021). Vulnerability management models using a common vulnerability scoring system. Applied Sciences, 11(18), 8735.
5. Shahriar, H., & Zulkernine, M. (2012). Mitigating program security vulnerabilities: Approaches and challenges. ACM Computing Surveys (CSUR), 44(3), 1-46.
6. Anand, P., Singh, Y., Selwal, A., Alazab, M., Tanwar, S., & Kumar, N. (2020). IoT vulnerability assessment for sustainable computing: Threats, current solutions, and open challenges. IEEE access, 8, 168825-168853.
7. Kumar, S. (2007). Patterns in the daily diary of the 41st president, George Bush (Doctoral dissertation, Texas A&M University).
8. Uppuluri, V. (2019). The Role of Natural Language Processing (NLP) in Business Intelligence (BI) for Clinical Decision Support. ISCSITR-INTERNATIONAL JOURNAL OF BUSINESS INTELLIGENCE (ISCSITR-IJBI), 1(2), 1-21.
9. Abraham, U. I. (2020). Deforestation, Air Quality Degradation and Increased Cardiopulmonary Diseases. SRMS JOURNAL OF MEDICAL SCIENCE, 5(02).
10. Taorui Guan, "Evidence-Based Patent Damages," 28 Journal of Intellectual Property Law (2020), 1-61.
11. Adepoju, S. (2021). Hybrid Retrieval Architectures: Integrating Vector Search into Production Systems.
12. Akinyemi, A. (2021). Cybersecurity Risks and Threats in the Era of Pandemic-Induced Digital Transformation. International Journal of Technology, Management and Humanities, 7(04), 51-62.
13. Azmi, S. K., Vethachalam, S., & Karamchand, G. (2022). The Scalability Bottleneck in Legacy Public Financial Management Systems: A Case for Hybrid Cloud Data Lakes in Emerging Economies.
14. Guan, T. (2020). Evidence-Based Patent Damages. J. Intell. Prop. L., 28, 1.
15. Taiwo, S. O., & Amoah-Adjei, C. K. (2022). Financial risk optimization in consumer goods using Monte Carlo and machine learning simulations.
16. Akinyemi, A. (2022). Securing Critical Infrastructure Against Cyber Attacks. SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology, 14(04), 201-209.
17. Uppuluri, V. (2020). Integrating behavioral analytics with clinical trial data to inform vaccination strategies in the US retail sector. J Artif Intell Mach Learn & Data Sci, 1(1), 3024-3030.
18. Akinyemi, A. (2022). Zero Trust Security Architecture: Principles and Early Adoption. International Journal of Technology, Management and Humanities, 8(02), 11-22.
19. Lai, Y. P., & Hsia, P. L. (2007). Using the vulnerability information of computer systems to improve the network security. Computer Communications, 30(9), 2032-2047.
20. Ten, C. W., Liu, C. C., & Manimaran, G. (2008). Vulnerability assessment of cybersecurity for SCADA systems. IEEE Transactions on Power Systems, 23(4), 1836-1846.
21. Grobauer, B., Walloschek, T., & Stocker, E. (2010). Understanding cloud computing vulnerabilities. IEEE Security & privacy, 9(2), 50-57.
22. Ye, H., Martinez, M., & Monperrus, M. (2021). Automated patch assessment for program repair at scale. Empirical Software Engineering, 26(2), 20.