International Journal of Safety and Security Engineering
Vol. 15, No. 2, February, 2025, pp. 267-273
Journal homepage: http://iieta.org/journals/ijsse

Design and Implementation of Distributed Web Application Vulnerability Assessment Tools for Securing Complex Microservices Environment

Muhammad Izzat, Ferry Astika Saputra*, Iwan Syarif

Department of Informatics and Computer Engineering, Politeknik Elektronika Negeri Surabaya, Surabaya 60111, Indonesia

Corresponding Author Email: ferryas@pens.ac.id

Copyright: ©2025 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

https://doi.org/10.18280/ijsse.150207

Received: 22 November 2024
Revised: 24 January 2025
Accepted: 12 February 2025
Available online: 28 February 2025

Keywords:
distributed vulnerability management, nuclei scanner, OWASP security framework, real-time vulnerability detection, web application security testing

ABSTRACT

Modern web applications with complex distributed architectures present significant challenges in vulnerability assessment that traditional approaches fail to address effectively. This research introduces the Distributed Vulnerability Management System (DVMS), implementing a multi-agent architecture to enhance vulnerability detection while eliminating single points of failure. The methodology employs the Nuclei vulnerability scanner across five Open Web Application Security Project (OWASP) security domains, expanding beyond conventional vulnerabilities to include Security Misconfiguration, Vulnerable Components, and Sensitive Data Exposure. Experimental results demonstrate detection accuracies of 80% for Injection, 85.71% for XSS, 80% for Security Misconfiguration, 50% for Vulnerable Components, and 90.91% for Sensitive Data Exposure. The distributed architecture enables parallel processing and optimizes security resource allocation across network infrastructures. While showing promising results in comprehensive security coverage, the system identifies areas for future enhancement in detection accuracy and vulnerability scope expansion. This research contributes a scalable, distributed approach to vulnerability management particularly suited for modern web applications, providing organizations with enhanced security assessment capabilities in complex technological environments.

1. INTRODUCTION

In the current age of rapid digital transformation, web-based applications are vital digital assets for multiple industries [1]. With technological advancements, the need for dependable and secure systems keeps increasing [2]. However, developing complex web-based architectures presents significant challenges, particularly in securing components against vulnerabilities. Modern technological ecosystems such as microservices, IoT, serverless computing, and cloud solutions require comprehensive monitoring strategies to safeguard applications and infrastructure [3]. Modern ecosystems are offered as independent services that interconnect through various architectural environments, which naturally broadens the attack surface and raises potential security risks [4]. Conventional systems focus on local vulnerability management within the local network, while modern technological architectures urge the development of a comprehensive distributed monitoring system for web-based applications to mitigate security challenges across complex, interconnected environments [5].

Current vulnerability monitoring systems struggle with capturing security flaws across unique security configurations and diverse performance requirements because centralize behavioral approach [6]. Complex environments have a potential vulnerability throughout the systems such as Cross-Site Scripting (XSS), SQL Injection [7], insecure data exchange between microservices, misconfigurations, and unverified external components [8]. Another limitation of centralizing the vulnerability monitoring system is about the scalability issues when the number of nodes increasing that leads to bottlenecks of the transaction volumes. This problem is considered as a single point of failure if the central server achieves a downtime or exploitation that leads the entire monitoring system to become ineffective and vulnerable [9].

In response to the challenges faced by vulnerability management systems, this research proposes a DVMS designed to enhance the penetration coverage of vulnerability tools and eliminate the single point of failure issue by utilizing multiple agents. Furthermore, it aims to expand the scope of vulnerability detection compared to existing tools and examines their accuracy. The specific objectives of this research include identifying gaps in the capabilities of current vulnerability tools in complex environments and evaluating the effectiveness of vulnerability management systems in providing practical assessments for mitigating security flaws within the system.

2. DEFINITION AND CHALLENGES

2.1 Complex environment

Complex applications are structured unit with reusable components and independent services that operate the

functionalities individually [10]. With this result, the individual model system provides a highly interdependent for each component. Another opportunity of implementing web application that already apply reusable component and independent services are the ability to implement a heterogeneous technology stack on the web application instead using a relatively fixed stack for every component. The advantages of heterogeneous technology stacks are easiness for supply of library and a huge possibility for innovation by different developer with their own known technologies. In addition of these opportunities in the complex environment, deploying microservices in the complex environment does not need a control of entire system since microservices are deployed independently.

As the web applications become more dynamic, the infrastructure was following up with the complexity of the system itself. This means infrastructure security models such as perimeter-based security that only focus on their local boundaries network are no longer effective [11]. It requires a shift of architecture model that had an ability of continuous security monitoring in their model, which able by adopting the Zero-Trust Architecture (ZTA) model [12]. In the ZTA architecture model, each component in the systems is considered as untrustworthy until verified. To support these challenges, ZTA require a security orchestration that manage the modern complex system including continuous monitoring system in real-time and vulnerability assessment to analyze the weakness [13].

2.2 Vulnerability assessment

Vulnerability assessment (VA) is a process to identify and classify weaknesses in a system or infrastructure [14]. The objective of performing VA in the system is to provide insight of the security posture and measure appropriate mitigation or elimination of the risks [15]. Performing VA in your system is applied by scanning various assets, such as servers, web applications, database, and network components for possible vulnerabilities. This performance is using both automated and manual techniques to evaluate the system [16]. The automated techniques are providing efficiency by rapidly detecting common vulnerabilities across the systems [17]. For the manual techniques, it offers a deeper and more detailed analyses for more complex issue rather than automated findings. Therefore, combining these two methods performs a comprehensive security evaluation throughout the complex system.

Table 1. Current vulnerability detection accuracy results

Open-Source Tools	Injection	XSS	Security Misconfiguration	Vulnerable Components	Sensitive Data Exposure
W3af	80%	-	-	-	-
OWASP ZAP	73%	72%	-	-	-
Wapiti	97%	74%	-	-	-
Arachni	98%	72%	-	-	-
Vega	67%	60%	-	-	-

The previous research [18] provides a comparative results of various security vulnerability scanning tools systematically compiled from various scientific article based on its evaluation result on the OWASP framework, that ranks the 10 most critical vulnerabilities. The research results mention SQL Injection and XSS vulnerabilities often reported in the articles as shown in Table 1.

The various scientific articles discussed show that five open-source-based Web Vulnerability Scanners (WVS) are majorly focused on only 2 types of vulnerabilities, Injection and XSS. Therefore, in the web application with complex system, it necessary to evaluate the capabilities of WVS to be able to expand the scope of scanning into other types of vulnerability. To solve this, this study aims to reach other scopes such as Security Misconfiguration, Vulnerable Components, and Sensitive Data Exposure.

2.3 OWASP framework

As system become complex, ensuring a standard security posture across applications becomes a critical challenge. For web-based applications, various framework establishes standards for assessing security weaknesses that serve as essential references for organization aiming to enhance their vulnerability assessment practices [19]. OWASP is an industry standard that address this challenge by providing a widely accepted list of critical vulnerabilities monitor in the OWASP Top 10 [20]. These standards guide developers and security practitioners in identifying and mitigating security risks, helping maintain robust security even as application grow in complexity. OWASP framework is designed to accommodate diverse configuration, making it suitable for complex environments that require consistent security measures.

2.4 Multi-agent system in distributed architecture

Multi-agent system is a system design approach that separate services in to multiple nodes or agent which able to work independently [21]. In the Distributed Architecture, this system generates a more efficient and scalable process in the context of VA. These advantages leverage the problem in centralize monitoring system where a single point of failure vulnerability exists by distributing the responsible for VA into multiple agents across the network. The challenge in implementing his multi-agent system is to ensure each scenario and interaction between the agent and the central system to be performed as scheduled.

2.5 F-Measure method

The F-Measure method is a metric to assess the performance of the classification system, particularly in contexts where there is an imbalance between positive and negative outcomes [22]. In this context, the classification of the vulnerability that found during VA, had an imbalance positive and negative cases as the result. The F-Measure calculate the precision and recall from The VA result, and accumulate it to become the overall accuracy using F1-Score. The precision identifies True Positive (TP) vulnerabilities compared with all detections, including the False Positive (FP), as shown in Eq. (1).

$$\text{Precision}=\frac{\text{TP}}{\text{TP+FP}} \quad (1)$$

The recall is a proportion of actual vulnerabilities that are correctly identified by the VA tools, including the False Negative (FN) as shown in Eq. (2).

$$\text{Recall}=\frac{\text{TP}}{TP+FN} \quad (2)$$

In this context, there are two goals that should be achieved by avoiding FP (irrelevant alerts) and minimizing FN (missed vulnerabilities). With the F1-Score, these objectives are measured, using the precision and recall rather each of the scores are high or low. The formulation of F-Measure to calculate the F1-Score is shown in Eq. (3).

$$\text{F1-Score}=2\times\frac{Precision\times Recall}{Precision+Recall} \quad (3)$$

F-Measure provides evaluation of the detection in each scanning node using different tools. It helps assess the consistency across multiple nodes by measuring how well the system balances the precision and recall.

3. METHODOLOGY

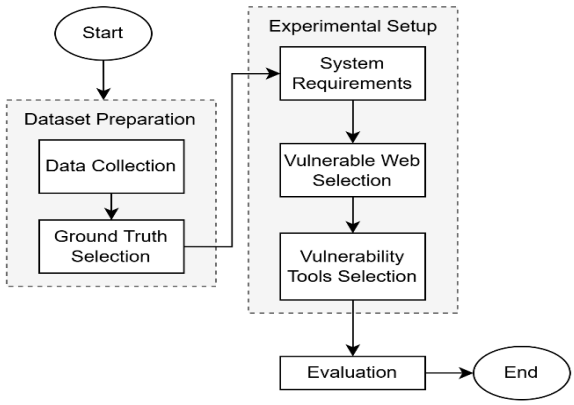

Figure 1. Research methodology workflow

Based on Figure 1, the research methodology starts with data preparation, which entails collecting data and ensuring it is suitable for reliable ground truth datasets that will serve as benchmarks for evaluation. The experimental setup section outlines the system requirements, selection of standard web applications vulnerable to testing and the appropriate vulnerability assessment tools needed for the experiments. The evaluation phase then examines the data from the ground truth implemented through the experimental setup to measure the accuracy and implementation of the proposed methodology. This workflow guarantees a systematic approach to validating the research outcomes and fulfilling the study's objectives.

3.1 Data collection

Data Collection phase is important for establishing robust dataset referencing design of web applications to simulate real-world security challenges in alignment with OWASP framework. First step of data collection is choosing data parameters that focuses on comprehensive vulnerability identification from this OWASP official web vulnerability identification [23]. These parameters are then categorized based on 5 primary security domains on this research:

(1) Injection
(2) XSS
(3) Security Misconfiguration
(4) Vulnerable Components
(5) Sensitive Data Exposure

These 5 primary security domains are chosen from OWASP Top 10 that are able to performed in automatic tools testing environments. This research identifies 21 standardized parameters that can be implemented in automated vulnerability tools, similar to other open-source tools listed in Table 1.

3.2 Ground truth selection

Based on Table 2, the scope selection of vulnerability from the web application was detailed into a baseline measurement for deciding the positive and negative cases in the experiment object. Ground truth is a sort of datasets that verify which vulnerabilities are present in the reliable source from OWASP framework. This research also verifies that the VA tools available to provide script or other penetration method for performing every parameter in the vulnerable site.

Table 2. Ground truth datasets

Injection	XSS	Security Misconfiguration	Vulnerable Components	Sensitive Data Exposure
SQL Injection	Stored XSS	Error Handling	Vulnerable Library	Access Log
Command Execution	Reflected XSS	HTTP Headers Missing	Local File Read	Confidential Document
Blind SQL Injection	CSP Bypass	Deprecated Interface	Supply Chain Attack	Email Leak
	Client Side XSS	Directory Browsing	Unsigned JWT	Leaked Access Logs
				Exposed Metrics
				Leaked Unsafe Product

3.3 System requirements

Based on Table 3, the system requirements specification ensures a consistent configuration setup across all nodes in the test environment. The vulnerable web application host is within a virtual machine, while the VA Agents and Central Dashboard deployment are on the host machine. This setup maintain consistency across the test environment, ensuring a fair comparison of our results.

Table 3. System requirements

System Requirements	
Operating System	Ubuntu 22.04 LTS
CPU	Intel i7-10710U 1.10GHz
Memory	4 GB
Machine Type	Virtual Machine
Network Speed	1 Gbps

3.4 Vulnerable web selection

The selection of vulnerable web applications is necessary with accurate research sites which do not produce false information during experiments. The research methodology strategically selects web applications with the key criteria such as, Demonstrable based on OWASP Top 10 Vulnerabilities and Compatibility with automated vulnerability assessment tools. The deployment of vulnerable web is within an interconnected network infrastructure that simulate distribution of modern technological services.

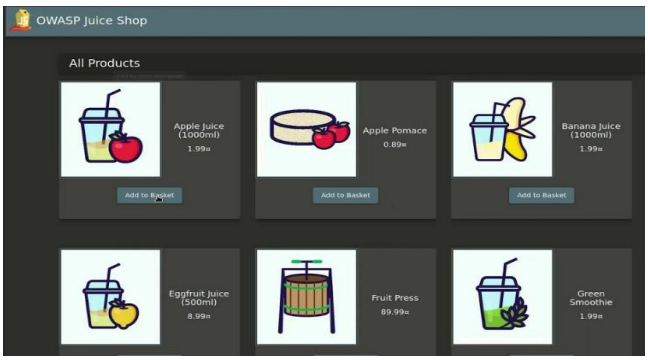

Figure 2. OWASP juice shop interface

As shown in Figure 2, OWASP juice shop is a web-based application with security vulnerabilities that is used for security training and awareness.

3.5 Vulnerability tools selection

In this research, the selection of Nuclei vulnerability scanner tools is crucial, that combines speed, flexibility, and comprehensive vulnerability coverage [24]. Nuclei vulnerability scanner is a robust scanner with an adaptable template to perform security checks. The template development is based on YAML Languages that scan hosts for recent attacks, vulnerabilities, CVEs, endpoints, and misconfigurations. Using this scanner in this research would help detect scenario attack vulnerabilities such as Injections, XSS, Misconfiguration, and Sensitive Data. Another advantage of Nuclei is the ability to process thousands of hosts in a short amount of time. These advantages would help increase the performance of assessing multiple hosts in real-time.

3.6 Evaluation

To assess the proposed system's effectiveness, the researcher performs a thorough analysis of experimental results. The evaluation emphasizes measuring the system's accuracy and performance in vulnerability identification, concurrently comparing it with current vulnerability assessment tools. The F-Measure method, a well-regarded metric, is employed to assess the system's accuracy by evaluating the balance between precision and recall. This methodology guarantees reliable vulnerability detection across various scenarios, offering a strong indication of the system's effectiveness in tackling real-world security issues.

4. IMPLEMENTATION AND EXPERIMENTATION

4.1 System detection accuracy

The experiment was conducted by scanning using Nuclei vulnerability scanner with 5 iterations for each agent. This iteration was performed to ensure the results were stable within the subjects. From the results, the found vulnerabilities are compared with the ground truth that already defined, and calculate each vulnerability results using the F-Measure method as shown in Table 2.

From the results in Table 4, indicates that the Nuclei vulnerability scanner demonstrates high detection accuracy for Injection with 80% accuracy, XSS with 85.71% accuracy, and Sensitive Data Exposure with 90.91% accuracy. While showing moderate accuracy for Security Misconfiguration and Vulnerable Components with 50% accuracy. Comparing with other tools [18], Nuclei outperforms by covering five vulnerability types instead of only Injection and XSS. The broader scope and balanced precision and recall performance highlight its effectiveness in diverse scenarios. For overall detection accuracy results are shown in Table 5.

Comparing with overall results of previous research, the Nuclei Scanner demonstrates broader vulnerability coverage and higher accuracy in several scopes. While its Injection accuracy only achieve 80% which lower than Wapiti with 97% accuracy and Arachni with 98% accuracy, Nuclei matches Vega in XSS detection with 85% accuracy and outperforms other tools in overall scopes. Additionally, Nuclei extends the benchmark by effectively detecting vulnerabilities in more complex categories such as Security Misconfiguration with 80% accuracy, Sensitive Data Exposure with 90.9% accuracy, and Vulnerable Components with 50% accuracy.

Table 4. Nuclei vulnerability scanner accuracy result

Calculation	Injection	XSS	Security Misconfiguration	Vulnerable Components	Sensitive Data Exposure
Precision	100%	100%	66.67%	50%	100%
Recall	66.67%	75%	100%	50%	83.33%
F1-Score	80%	85.71%	80%	50%	90.91%

Table 5. Overall vulnerability detection accuracy results

Open-Source Tools	Injection	XSS	Security Misconfiguration	Vulnerable Components	Sensitive Data Exposure
W3af	80%	-	-	-	-
OWASP ZAP	73%	72%	-	-	-
Wapiti	97%	74%	-	-	-
Arachni	98%	72%	-	-	-
Vega	67%	85%	-	-	-
Nuclei	80%	85%	80%	50%	90.9%

Nuclei outperforms other vulnerability scanning tools due to its unique combination of speed, customizability, and community-driven template system. Its high-performance engine enables rapid scanning across various protocols, including TCP, SSH, DNS, HTTP, and SSL, facilitating efficient large-scale assessments. The use of simple YAML-based templates allows users to define custom detection scenarios, enhancing flexibility and enabling tailored scans to meet specific requirements. Additionally, Nuclei's active open-source community contributes to a continually updated repository of over 8,000 templates, ensuring timely detection of emerging vulnerabilities. This collaborative approach, combined with its modular architecture, allows Nuclei to adapt to a wide range of needs, making it a preferred choice for comprehensive and up-to-date vulnerability assessments.

4.2 Distributed architecture implementation

The DVMS architecture directly addresses the single point of failure limitation inherent in centralized vulnerability management systems through its distributed, multi-agent design. In the implementation of Distributed Architecture, it maintains the current process to be stable while the multi-agent systems are scaled horizontally. This solution ensures the system relies on continuous monitoring while still monitoring the central dashboard. As illustrated in the Figure 3, through this approach, DVMS topology optimizes detection accuracy in complex environments.

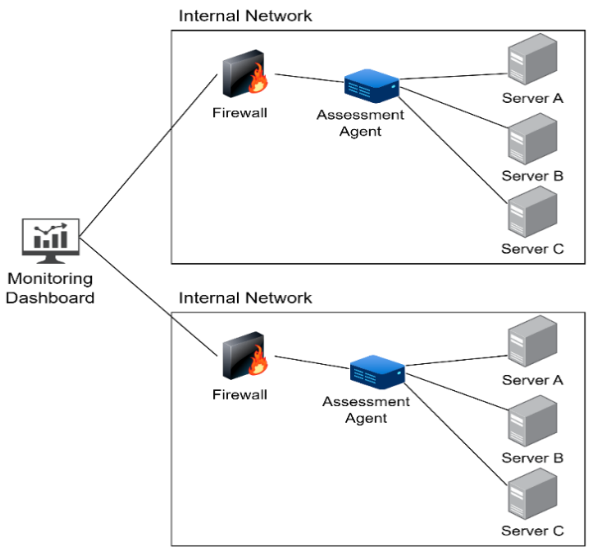

Figure 3. The network topology of DVMS

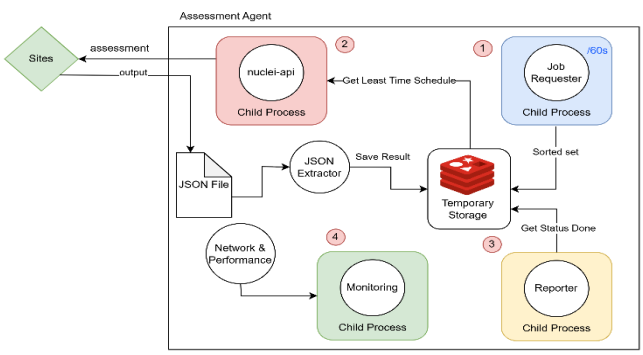

Figure 4. Architecture design of DVMS assessment agent

In the distributed architecture, the agent is scaled horizontally without affecting other components in the architecture [25]. With the ability to specifically scan a part of the system, it enhances the performance of accuracy in each system with their parallel processing. In Figure 4, the DVMS agent architecture has four main processes that work in parallel, so the process is performed simultaneously without interrupting any other child process.

Job Requester

This process retrieves the schedule for the task from the central management system. If any asset is scheduled for imminent scanning, the job requester accepts the task and store it temporary in the agent Redis database.

Vulnerability Assessment

This process uses the Nuclei VA tools to perform the vulnerability scan. It processes only the tasks stored in temporary storage, executing scans based on the predefined schedule.

Reporter

Once the vulnerability scan is complete, this process reports the assessment results to the central system for aggregation and analysis.

Monitoring

This independent process monitors the computational performance of the entire agent, ensuring optimal resource utilization and preventing potential bottlenecks.

After this process, each report and process monitoring are sent to the central management system for data aggregation and managing the result into the central dashboard that enables the administrator to respond to threats effectively. The central management dashboard is able to assign the VA tasks to the suitable agent using a task scheduling algorithm. This algorithm ensures every agent receives tasks based on its current workload and latency within the network. When the central management system identifies the asset needs assessment from the time defined by the users, it allocates the tasks in to the agents immediately.

5. CONCLUSIONS

The DVMS offers solutions in cybersecurity, addressing the fundamental limitations of local and centralized vulnerability management systems. This research's primary focus lies in the comprehensive approach to vulnerability detection. While conventional tools narrowly focus on Injection and XSS vulnerabilities, DVMS expands the security assessment scope across five critical domains. This approach represents a significant advancement in mitigating security risks in complex technological environments. By choosing Nuclei vulnerability scanner, the research demonstration achieves the potential for more comprehensive security evaluations, achieving detection accuracies from 50% to 90% across different vulnerability categories.

The distributed multi-agent's architecture provides a solution for single point of failure problem that inherent in centralized monitoring systems. This solution enables real-time, dynamic vulnerability assessment, and allowing security resource allocation more effectively across network infrastructures. The ability to perform parallel processing and maintain continuous monitoring represents a fundamental shift in how organizations approach cybersecurity in increasingly

complex technological ecosystems.

However, the research also acknowledges its limitations. Nuclei vulnerability tools require further refinement in accuracy detection for Injection and XSS. Other limitations in the research are the expansion capability of vulnerability detection scopes that need to cover more scopes in OWASP Top 10. DVMS has significant potential for future development with advance integration tools such as threat intelligence and other machine learning techniques to keep provide comprehensive security mitigation for organizations.

In conclusion, this research transcends traditional boundaries of vulnerability assessment. It challenges existing paradigms, offers innovative solutions, and provides a roadmap for addressing the complex security challenges of our increasingly interconnected digital world.

REFERENCES

[1] Sotnik, S., Shakurova, T., Lyashenko, V. (2023). Development features web-applications. International Journal of Academic and Applied Research (IJAAR), 7(1): 79-85. https://openarchive.nure.ua/handle/document/21600.

[2] Gong, Y., Yang, J., Shi, X. (2020). Towards a comprehensive understanding of digital transformation in government: Analysis of flexibility and enterprise architecture. Government Information Quarterly, 37(3): 101487. https://doi.org/10.1016/j.giq.2020.101487

[3] Shabani, I., Mëziu, E., Berisha, B., Biba, T. (2021). Design of modern distributed systems based on microservices architecture. International Journal of Advanced Computer Science and Applications, 12(2).

[4] Berardi, D., Giallorenzo, S., Mauro, J., Melis, A., Montesi, F., Prandini, M. (2022). Microservice security: A systematic literature review. PeerJ Computer Science, 8: e779. https://doi.org/10.7717/peerj-cs.779

[5] Farida, I., Setiawan, R., Maryatmi, A.S., Juwita, M.N. (2020). The implementation of e-government in the industrial revolution era 4.0 in Indonesia. International Journal of Progressive Sciences and Technologies, 22(2): 340-346. http://ijpsat.ijsht-journals.org.

[6] Waseem, M., Liang, P., Shahin, M., Di Salle, A., Márquez, G. (2021). Design, monitoring, and testing of microservices systems: The practitioners' perspective. Journal of Systems and Software, 182: 111061. https://doi.org/10.1016/j.jss.2021.111061

[7] Kollepalli, R.P.K., Reddy, M.J.S., Sai, B.L., Natarajan, A., Mathi, S., Ramalingam, V. (2024). An experimental study on detecting and mitigating vulnerabilities in web applications. International Journal of Safety and Security Engineering, 14(2): 523-532, https://doi.org/10.18280/ijsse.140219

[8] Hannousse, A., Yahiouche, S. (2021). Securing microservices and microservice architectures: A systematic mapping study. Computer Science Review, 41: 100415. https://doi.org/10.1016/j.cosrev.2021.100415

[9] Ahmad, S., Mir, A.H. (2021). Scalability, consistency, reliability and security in SDN controllers: A survey of diverse SDN controllers. Journal of Network and Systems Management, 29: 1-59. https://doi.org/10.1007/s10922-020-09575-4

[10] Baškarada, S., Nguyen, V., Koronios, A. (2020). Architecting microservices: Practical opportunities and challenges. Journal of Computer Information Systems, 60(5): 428-436. https://doi.org/10.1080/08874417.2018.1520056

[11] Kotari, M., Chiplunkar, N.N. (2020). Investigation of security issues in distributed system monitoring. In Handbook of Computer Networks and Cyber Security: Principles and Paradigms, Springer, Cham, 609-634. https://doi.org/10.1007/978-3-030-22277-2_24

[12] Syed, N.F., Shah, S.W., Shaghaghi, A., Anwar, A., Baig, Z., Doss, R. (2022). Zero-Trust Architecture (ZTA): A comprehensive survey. IEEE Access, 10: 57143-57179. https://doi.org/10.1109/ACCESS.2022.3174679

[13] Sarkar, S., Choudhary, G., Shandilya, S.K., Hussain, A., Kim, H. (2022). Security of zero trust networks in cloud computing: A comparative review. Sustainability, 14(18): 11213. https://doi.org/10.3390/su141811213

[14] Fatima, A., Khan, T.A., Abdellatif, T.M., Zulfiqar, S., Asif, M., Safi, W., Al Hamadi, H., Al-Kassem, A.H. (2023). Impact and research challenges of penetrating testing and vulnerability assessment on network threat. In 2023 International Conference on Business Analytics for Technology and Security (ICBATS), Dubai, United Arab Emirates, pp. 1-8. https://doi.org/10.1109/ICBATS57792.2023.10111168

[15] Aboelfotoh, S.F., Hikal, N.A. (2019). A review of cyber-security measuring and assessment methods for modern enterprises. JOIV: International Journal on Informatics Visualization, 3(2): 157-176. https://doi.org/10.30630/joiv.3.2.239

[16] Stefinko, Y., Piskozub, A., Banakh, R. (2016). Manual and automated penetration testing. Benefits and drawbacks. Modern tendency. In 2016 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET), Lviv, Ukraine, pp. 488-491. https://doi.org/10.1109/TCSET.2016.7452095

[17] Aslan, Ö., Aktuğ, S.S., Ozkan-Okay, M., Yilmaz, A.A., Akin, E. (2023). A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. Electronics, 12(6): 1333. https://doi.org/10.3390/electronics12061333

[18] Alazmi, S., De Leon, D.C. (2022). A systematic literature review on the characteristics and effectiveness of web application vulnerability scanners. IEEE Access, 10: 33200-33219. https://doi.org/10.1109/ACCESS.2022.3161522

[19] Shanley, A., Johnstone, M.N. (2015). Selection of penetration testing methodologies: A comparison and evaluation. In Australian Information Security Management Conference, AISM 2015, SRI Security Research Institute, Edith Cowan University, pp. 65-72. https://doi.org/10.4225/75/57b69c4ed938d

[20] Fredj, O.B., Cheikhrouhou, O., Krichen, M., Hamam, H., Derhab, A. (2021). An OWASP top ten driven survey on web application protection methods. In Risks and Security of Internet and Systems: 15th International Conference, CRiSIS 2020, Paris, France, pp. 235-252. https://doi.org/10.1007/978-3-030-68887-5_14

[21] Dehimi, N.E.H., Tolba, Z., Djabelkhir, N. (2024). Testing inclusive, exclusive, and parallel interactions in multi-agent system: A new model-based approach.

International Journal of Safety and Security Engineering, 14(4): 1125-1138. https://doi.org/10.18280/ijsse.140411
[22] Christen, P., Hand, D.J., Kirielle, N. (2023). A review of the F-Measure: Its history, properties, criticism, and alternatives. ACM Computing Surveys, 56(3): 1-24. https://doi.org/10.1145/3606367
[23] Pwning OWASP Juice Shop. https://pwning.owasp-juice.shop/companion-guide/, accessed on Aug. 24, 2024.
[24] Solanki, H.V. (2023). Limiting attack surface for infrastructure applications using custom YAML templates in nuclei automation, Doctoral Dissertation, Dublin, National College of Ireland.
[25] Pandiya, D.K. (2021). Scalability patterns for microservices architecture. Educational Administration: Theory and Practice, 27(3): 1178-1183. https://doi.org/10.53555/kuey.v27i3.6897