Penetration Testing and Vulnerability Analysis of SINTA Platform to Strengthen Privacy and Data Protection

Supangat Supangat [1,*] [ID], Anis Rahmawati Amna [2,] [ID], and Mochamad Yovi Fatchur Rochman [3]

[1,2] Department of Information Systems and Technology, Universitas 17 Agustus 1945 Surabaya, Indonesia
[3] Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Malaysia
[*] Corresponding author: supangat@untag-sby.ac.id

To cite this article: Supangat, S., Amna, A. R., & Rochman, M. Y. F. (2025). Penetration Testing and Vulnerability Analysis of SINTA Platform to Strengthen Privacy and Data Protection. *Journal of Information Technology and Cyber Security, 3*(2), 79-83. https://doi.org/10.30996/jitcs.12216

Abstract

The increasing reliance on digital platforms for academic and governmental purposes necessitates robust cybersecurity measures. Consequently, identifying vulnerability is critical to ensuring data security and providing actionable recommendations for cybersecurity officers. Platforms like Sinta (Science and Technology Index), which focus on collecting peer-reviewed papers and maintaining researcher's research records, represents significant governmental contributions in academia. Cybersecurity awareness is demonstrated through events organized to evaluate the vulnerability of the platform, enabling researchers to access its security and report potential issues. This study addresses these concerns by conducting system penetration testing using the OWASP and Burp Suite Framework, focusing on identifying five critical vulnerabilities. The evaluation examines issues, such as sensitive data exposure in API responses, error log disclosures, email enumeration, and improper access to system files. The results reveal that the platform suffers from multiple levels of security vulnerabilities, prompting recommendations for authorities to take actions to mitigate potential risks effectively.

Keywords: cybersecurity, pentesting, Science and Technology Index, Sinta.

1. Introduction

The increasing prevalence of cyber threads forces software developer to prioritize cybersecurity to protect privacy and ensure data security. As the organization managing SINTA, a platform for collecting, indexing, and disseminating researcher's research and activities, the Indonesian Ministry of Education, Culture, Research, and Technology is responsible for ensuring robust data security (Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia, 2020). SINTA stores sensitive information, including personal details, national identification research number (NIDN), and research projects. Therefore, safeguarding its data and identifying potential vulnerabilities are essential to maintaining its security and protecting user information (Cahyanto, 2023; Alhamed & Rahman, 2023)[3].

Identifying vulnerabilities is a critical step in ensuring data security. To address this, the Ministry of Education, Culture, Research, and Technology organizes an event called Sinta Talk to invite researchers to evaluate the system, identify vulnerabilities, and provide actionable recommendations. In response, this study evaluates sensitive data exposure, analyzes risk levels, and assesses potential impacts. The goal of the study is to propose mitigation actions to strengthen SINTA platform and ensure sustainable privacy and data security. The findings may also benefit other platforms with similar characteristics and security requirements by providing a framework for enhancing cybersecurity.

Cybersecurity domain receives an increasing attention from both academia and practitioners, given the risk of data breaches, unauthorized access, and system vulnerabilities. Several studies have extensively explored issues such as inadequate system configurations, weak authentication protocols, and the exposure of sensitive data through APIs (Folorunso, Wada, Samuel, & Mohammed, 2024). For instance,
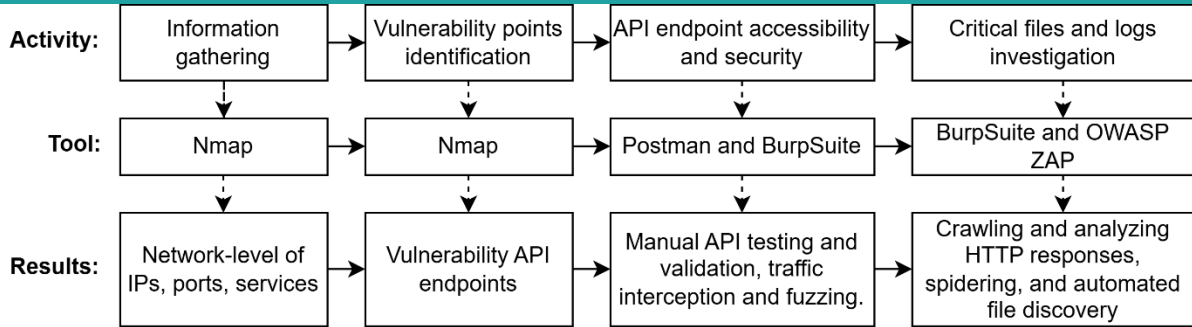
Activity:	Information gathering	→	Vulnerability points identification	→	API endpoint accessibility and security	→	Critical files and logs investigation
Tool:	Nmap		Nmap	→	Postman and BurpSuite		BurpSuite and OWASP ZAP
Results:	Network-level of IPs, ports, services	→	Vulnerability API endpoints	→	Manual API testing and validation, traffic interception and fuzzing.	→	Crawling and analyzing HTTP responses, spidering, and automated file discovery

Fig. 1. Research methodology.

Fig. 2. Illustration of NIST's CVSS Calculator (https://www.first.org/cvss/calculator/3.0).

Yazıcıoğlu (2024) conducted a comparative study of manual, customized, and automated testing approaches. The findings revealed that the customized approach provided more actionable insights due to its alignment with real-world conditions. Additionally, Kyriazoglou (2024) highlighted that security policy frameworks, such as GDPR, can significantly mitigate security breaches, particularly in areas related to data privacy and user rights, when effectively implemented.

Given the situation, our study aims to examine vulnerabilities on API access control and insecure file configuration as these are the most frequent issues occur (OWASP, 2024). To do so, penetration testing is conducted using Burp Suite, OWASP ZAP, Nmap, and Postman (Santos & Acosta, 2023; Seara & Serrão, 2024; Shahid, et al., 2022; Albalawi, et al., 2023). These packages are selected solely due to practicality and familiarity.

2. Methods

The study applies penetration testing during the BlackBox Testing phase, allowing the researchers to evaluate the system without access to its source code or internal configurations (Althunayyan, Saxena, Li, & Gope, 2022). This approach simulates real-world scenarios where cyber attacts exploit system vulnerabilities. Burp Suite (Albalawi, et al., 2023), OWASP ZAP (Shahid, et al., 2022), Nmap (Seara & Serrão, 2024), and Postman (Santos & Acosta, 2023) were employed for various tasks.

First, reconnaissance and scanning were conducted to gather system information, including IP addresses, open ports, and active services. Nmap was used to identify vulnerability points that could be exploited by external users. Next, Postman and BurpSuite were used to the accessibility and security of API endpoints, specifically verifying if sensitive data (e.g., NIDN or other personal information) could be accessed without proper authorization. Finally, the investigation checked whether critical files and logs (e.g., `.gitignore`, installed.json) were publicly accessible, as these files could expose sensitive configurations and pave the way for further attack (Fig. 1).

Table 1
Vulnerability analysis and recommended mitigation.

Risk Level	Vulnerability	CVSS	Potential Impact	Mitigation Recommendation
High	Sensitive Data Exposure in API Responses	7.5	Data breach, privacy violation	Enforce strict authentication on all API endpoints and mask sensitive data in API responses.
Moderate	Error Log Webmail Roundcube File Disclosure	5.3	Email exposure, potential phishing attacks	Restrict access to log files to internal servers, remove unnecessary files, and store logs in secure directories.
Moderate	Email Enumeration Vulnerability on Password Reset Page	5.3	Increased risk of brute force and phishing attacks	Implement CAPTCHA and limit repeated attempts from the same IP address on the password reset page.
Low	`.gitignore` File Detected	3.9	Internal configuration exposure	Remove the `.gitignore` file from the public directory and ensure secure configuration.
Low	`composer.lock` or `installed.json` Publicly Accessible	3.6	Third-party library exposure and system version disclosure	Restrict public access to these files and enforce proper access controls to protect sensitive information.

The results were then analyzed using CVSS (Common Vulnerability Scoring System) to determine severity of the impacts and mitigation priority based on OWASP standard and data protection regulation (OWASP, 2024; Sánchez-García, Mejía, & Gilabert, 2023). The score is calculated using NIST's CVSS calculator by entering base score, temporal score, and environmental score metrices (https://www.first.org/cvss/calculator/3.0). Each metric reflects different characteristics of a vulnerability, from intrinsic (i.e., base metric), change over time (i.e., temporal metric), and specific environment (i.e., environmental metric). The online calculator enables users to select options according to the situation occurs, offering simplicity to assess incidents and provide mitigation recommendations.

3. Results and Discussion

This section presents security testing results for the SINTA platform. The findings reveal that the system has several vulnerabilities, each associated with varying levels of potential risk. Table 1 summarizes the penetration testing results, including the type of vulnerability, CVSS score, potential impact, and recommended mitigation measures. The CVSS score were used to assess the severity of each vulnerability and prioritize mitigation efforts based on the associated risks.

The findings reveal that high and intermediate risks of vulnerability are present in Sensitive Data Exposure in API responses and insufficient control over API access. These issues are potential breach for unauthorized access to personal information, which could threat user privacy and security (OWASP, 2017). CVSS Score of 7.5 shows that the vulnerability is significant and the mitigation recommendation provided should be addressed immediately as it could lead to severe impact on the system.

A moderate risk of vulnerability arises from Error Log Disclosure and Email Enumeration, with a CVSS score of 5.3. While these vulnerabilities could trigger exploits, particularly phishing, the threat typically requires specific conditions or some level of user interaction (OWASP, 2017; OWASP, 2021; Dobon, 2023). Therefore, the mitigation actions should focus on strengthening user security by moving log files to secure directories, implementing two-factors authentication (e.g., CAPTCHA, Windows authentication), and limiting access attempts to specific pages when users enter incorrect password. Low-risk vulnerabilities, indicated by CVSS scores of 3.6 and 3.9, emerge in Gitignore File Detection and Composer `installed.json` Exposure, suggesting that the system lacks proper file configuration security, which could potentially lead to further exploitation. (Acunetix, 2023; Tenable, 2019). While vulnerabilities in this level are typically minor and have little to no impact on the system confidentiality, it is recommended to remove configuration files from public directory and limit public access into such directories.

However, this study has several limitations, including the use of BlackBox Testing. While this approach may reflect real-world scenarios, the investigation could not thoroughly identify which lines of code or internal configurations may causing issues. Additionally, as the study was conducted over a limited time span, it may not have captured all potential threats comprehensively. Given these limitations, it is recommended to also conduct WhiteBox Testing and GreyBox Testing to provide broader access and enable a more secure evaluation from both the internal and external perspectives of the platform (Zardari,

et al., 2022; Stef & Polgar, 2024).

4. Conclusions

This study assesses vulnerabilities on the SINTA platform across various types of risks. Our main findings sugges that the system should strengthen their protection on user privacy data, API access control, and log files configurations. These findings are not only beneficial for the organizers of the SINTA platform but also for those who manage similar platforms to prevent data exploitation.

Although there is no direct request from the Indonesian Ministry of Education, Culture, Research, and Technology, the research call Sinta Talk has been regarded as a formal invitation for researchers to analyze the system vulnerability. However, to the best of our knowledge, no previous publications have conducted a similar analysis on this specific system. Therefore, we cannot compare our results with an identical study. Instead, we refer to regular vulnerability analyses conducted on similar systems.

Further study recommends to implement BlackBox Testing together with WhiteBox Testing or GreyBox Testing. The combination is useful to gain better understanding on internal security of the system and potential risks. Creating a guideline moduls providing a list of potential risks and mitigations will be useful for administrators to detect and maintain potential data breaching in order to secure the platform.

The practical implication of this study could enhance data security and reduce the risk of information breach which could harm both individual and organizations. By stricting API access control and protecting configuration files and logs, it is hoped that the data can be protected better.

5. CRediT Authorship Contribution Statement

Supangat: Conceptualization, Methodology, Data curation, Formal analysis, Investigation, Validation, Visualization, and Writing – original draft. **Anis Rahmawati Amna**: Writing – review & editing. **Mochamad Yovi Fatchur Rochman:** Conceptualization, Formal analysis, and Writing – review & editing.

6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

7. Acknowledgments

We thank Edho Ferdian Dwi Cahyo who helped us conduct a pentest.

8. Data Availability

Data will be made available on request.

9. References

Acunetix. (2023, 09 28). *Composer installed.json publicly accessible*. Retrieved from Acunetix: https://www.acunetix.com/vulnerabilities/web/composer-installed-json-publicly-accessible/

Albalawi, N., Alamrani, N., Aloufi, R., Albalawi, M., Aljaedi, A., & Alharbi, A. R. (2023). The Reality of Internet Infrastructure and Services Defacement: A Second Look at Characterizing Web-Based Vulnerabilities. *Electronics, 12*(12), 2664. doi:https://doi.org/10.3390/electronics12122664

Alhamed, M., & Rahman, M. M. (2023). A Systematic Literature Review on Penetration Testing in Networks: Future Research Directions. *Applied Sciences, 13*(12), 6986. doi:https://doi.org/10.3390/app13126986

Althunayyan, M., Saxena, N., Li, S., & Gope, P. (2022). Evaluation of Black-Box Web Application Security Scanners in Detecting Injection Vulnerabilities. *Electronics, 11*(13), 2049. doi:https://doi.org/10.3390/electronics11132049

Cahyanto, I. (2023). Privacy Challenges in Using Wearable Technology in Education Literature Review. *Formosa Journal of Applied Sciences (FJAS), 2*(6), 909-928. doi:https://doi.org/10.55927/fjas.v2i6.4272

Dobon, D. (2023, 08). *Email enumeration vulnerability on "Password Reset" dialogue*. Retrieved from Discourse: https://meta.discourse.org/t/email-enumeration-vulnerability-on-password-reset-dialogue/273449

Folorunso, A., Wada, I., Samuel, B., & Mohammed, V. (2024). Security compliance and its implication for cybersecurity. *World Journal of Advanced Research and Reviews, 24*(1), 2105–2121. doi:https://doi.

org/10.30574/wjarr.2024.24.1.3170

Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia. (2020). *Sinta (Science and Technology Index)*. Retrieved 02 10, 2025, from Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia: https://sinta.kemdikbud.go.id/

Kyriazoglou, J. (2024). Summarizing ISO 27K and Major Privacy Regulations. In J. Kyriazoglou, *Information Security Incident and Data Breach Management: A Step-by-Step Approach* (pp. 15–26). Berkeley, CA: Apress. doi:https://doi.org/10.1007/979-8-8688-0870-8_2

OWASP. (2017). *OWASP Top Ten 2017: A3:2017-Sensitive Data Exposure*. Retrieved from OWASP: https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure

OWASP. (2021). *A07:2021 – Identification and Authentication Failures*. Retrieved from OWASP: https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/

OWASP. (2024). *OWASP Top Ten*. Retrieved 02 10, 2025, from OWASP: https://owasp.org/www-project-top-ten/

Sánchez-García, I. D., Mejía, J., & Gilabert, T. S. (2023). Cybersecurity Risk Assessment: A Systematic Mapping Review, Proposal, and Validation. *Applied Sciences, 13*(1), 395. doi:https://doi.org/10.3390/app13010395

Santos, F., & Acosta, N. (2023). An Approach Based on Web Scraping and Denoising Encoders to Curate Food Security Datasets. *Agriculture, 13*(5), 1015. doi:https://doi.org/10.3390/agriculture13051015

Seara, J. P., & Serrão, C. (2024). Automation of System Security Vulnerabilities Detection Using Open-Source Software. *Electronics, 13*(5), 873. doi:https://doi.org/10.3390/electronics13050873

Shahid, J., Hameed, M. K., Javed, I. T., Qureshi, K. N., Ali, M., & Crespi, N. (2022). A Comparative Study of Web Application Security Parameters: Current Trends and Future Directions. *Applied Sciences, 12*(8). doi:https://doi.org/10.3390/app12084077

Stef, M. P., & Polgar, Z. A. (2024). Software Platform for the Comprehensive Testing of Transmission Protocols Developed in GNU Radio. *Information, 15*(1), 62. doi:https://doi.org/10.3390/info15010062

Tenable. (2019, 05 16). *Gitignore File Detected*. Retrieved from tenable: https://www.tenable.com/plugins/was/98595

Yazıcıoğlu, M. B. (2024). ISO 27001, KVKK, and GDPR: A Comparison of Information Security and Data Protection Standards. *Journal of Engineering and Technology, 5*(1), 11-21. Retrieved from https://dergipark.org.tr/en/pub/jetech/issue/85597/1488191

Zardari, S., Alam, S., Salem, H. A., Reshan, M. S., Shaikh, A., Malik, A. F., . . . Mouratidis, H. (2022). A Comprehensive Bibliometric Assessment on Software Testing (2016–2021). *Electronics, 11*(13), 1984. doi:https://doi.org/10.3390/electronics11131984