maka yang seharusnya negatif maka tetap positif dikarenakan payload pertama sudah mentrigger untuk fungsi alert yang disisipkan ke dalam payload pertama.

5. Kesimpulan

Static code analysis merupakah salah satu teknik yang dapat digunakan untuk menemukan adanya kemungkinan celah keamanan XSS. Sistem fuzzer digunakan sebagai pengujian otomatis dari hasil informasi pengiriman data yang didapatkan. Berdasarkan pengujian yang dilakukan pada dua security level low dan medium DVWA, untuk stored XSS selalu mendapatkan akurasi yang paling rendah dibanding dua jenis XSS lainnya. Stored XSS mendapatkan akurasi paling rendah dikarenakan untuk alur pengujian stored XSS lebih panjang dibanding yang lain dan pengujian untuk stored xss diperlukan lebih banyak interaksi dari pengguna. Total vulnerability yang ditemukan pada security level medium mendapatkan hasil lebih rendah dibanding pengujian pada security level low karena pada security level medium sudah diterapkan sanitasi pada form input. Untuk saran penelitian selanjutnya, penulis menyarankan untuk menggunakan *static code analysis* untuk deteksi serangan lainnya seperti *SQL injection*, *Command Injection*, dan *ORM Injection*.

Daftar Pustaka

[1] Stackoverflow, “Stack Overflow Developer Survey 2020,” 2020. [Online] Available: https://insights.stackoverflow.com/survey/2020#technology-programming-scripting-and-markup-languages-all-respondents [access: 15-Nov-2020].

[2] Owasp, “Cross Site Scripting (XSS) Software Attack,” 2020. [Online] Available: https://owasp.org/www-community/attacks/xss/ [access: 15-Nov-2020].

[3] Portswigger, “Critical stored XSS vulnerability in Instagram’s Spark AR Studio nets 14-year-old researcher $25,000”, 2020. [Online] Available: https://portswigger.net/daily-swig/critical-stored-xss-vulnerability-in-instagrams-spark-ar-studio-nets-14-year-old-researcher-25-000 [access: 15-Nov-2020].

[4] Watchcom, “Watchcom uncovers Cisco Jabber vulnerabilities,” 2020. [Online] Available: https://watchcom.no/nyheter/nyhetsarkiv/uncovers-cisco-jabber-vulnerabilities/ [access: 15-Nov-2020].

[5] A. Takanen, J. DeMott, and C. Miller, Fuzzing for Software Security Testing and Quality Assurance. Artech House, Inc., Norwood, USA, 2008.

[6] PacketLabs, “How does Cross-site Scripting (XSS) impact customers?,” 2020. [Online] Available: https://www.packetlabs.net/cross-site-scripting-xss/ [access: 22-Nov-2020].

[7] Owasp, “Types of XSS,” 2020. [Online] Available: https://owasp.org/www-community/Types_of_Cross-Site_Scripting [access: 22-Nov-2020].

[8] Owasp, “DOM Based XSS,” 2020. [Online] Available: https://owasp.org/www-community/attacks/DOM_Based_XSS [access: 22-Nov-2020].

[9] TechTarget, “What is payload (computing)?,” 2020. [Online] Available: https://searchsecurity.techtarget.com/definition/payload [access: 25-Nov-2020].

[10] Owasp, “XSS Filter Evasion Cheat Sheet,” 2020. [Online] Available: https://owasp.org/www-community/xss-filter-evasion-cheatsheet [access: 25-Nov-2020].

[11] Choi, H., Hong, S., Cho, S., & Kim, Y. G. (2018). HXD: Hybrid XSS detection by using a headless browser. *Proceedings of the 2017 4th International Conference on Computer Applications and Information Processing Technology, CAIPT 2017*, *2018-January*, 1–4. https://doi.org/10.1109/CAIPT.2017.8320672.

[12] S. Gupta and B. B. Gupta, “XSS-secure as a service for the platforms of online social network-based multimedia web applications in cloud,” Multimedia Tools and Applications, vol. 77, no. 4, pp. 4829–4861, Jul. 2016, doi: 10.1007/s11042-016-3735-1.

[13] W. Ben Jaballah and N. Kheir, “A Grey-Box Approach for Detecting Malicious User Interactions in Web Applications,” Proceedings of the 8th ACM CCS International Workshop on Managing Insider Security Threats, Oct. 2016, doi: 10.1145/2995959.2995966.

[14] Li, L., & Wei, L. (2019). Automatic XSS detection and automatic anti-anti-virus payload generation. *Proceedings - 2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, CyberC 2019*, 71–76. https://doi.org/10.1109/CyberC.2019.00021.

[15] M. Liu, B. Zhang, W. Chen, and X. Zhang, “A Survey of Exploitation and Detection Methods of XSS Vulnerabilities,” IEEE Access, vol. 7, pp. 182004–182016, 2019, doi: 10.1109/access.2019.2960449.

[16] M. Khder, “Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Application,” International Journal of Advances in Soft Computing and its Applications, vol. 13, no. 3, pp. 145–168, Nov. 2021, doi: 10.15849/ijasca.211128.11.

[17] Ari Takanen; Jared Demott; Charles Miller; Atte Kettunen, Fuzzing for Software Security Testing and Quality Assurance, Second Edition , Artech, 2018.

[18] Repository, “Payload All The Things” 2022. https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/XSS%20Injection [access: 1 Juni 2022].
[19] Repository, “Cross Site Scripting (XSS) Vulnerability Payload List” 2022. https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/XSS%20Injection [access: 1 Juni 2022].
[20] Owasp, “Cross Site Scripting Prevention Cheat Sheet,” 2022. [Online] Available: https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html#html-sanitization [access: 28-Sep-2022]

Lampiran

Link artefak dapat di akses melalui : https://drive.google.com/drive/folders/1W5muezQLocx7hOZBZKlDEq121NGx-J7d?usp=sharing