Vulnerability Audit and Assessment - Results and Executive Summary

Introduction

This report presents results returned from a vulnerability assessment of http://demo.testfire.net/, a testing site for an online bank. It analyses the findings of a vulnerability scan and security weaknesses that deviate from baseline standards based on recommendations from OWASP and NIST. The websites ability to meet various compliance standards will be evaluated. Conclusions will be made about the overall level of security and recommendations for improvements will be given.

Methodology

Kali Linux was chosen as the operating system and was run through a virtual machine.

Reconnaissance was performed with theHarvester, WHOIS and Dig.

Nmap was used to find open ports for scanning and to view SSL/TLS versions and encryption cypher strength. To confirm the results an online testing lab was used (Qualys, 2025).

Automated vulnerability scanning was conducted with Zap v2.16.0. The initial scan was passive, utilising both the traditional spider and modern AJAX spider functionality. A general active scan was performed, before a higher strength active scan focusing on injection attacks was conducted. Some manual testing was done to confirm returned results.

One issue encountered was Zap repeatedly crashing during active scanning, with no reason given. This was resolved by increasing the memory capacity of the Kali Linux virtual machine. The problem highlighted the intensive performance demands of the scanner. Testing was conducted outside of regular business hours as a precaution.

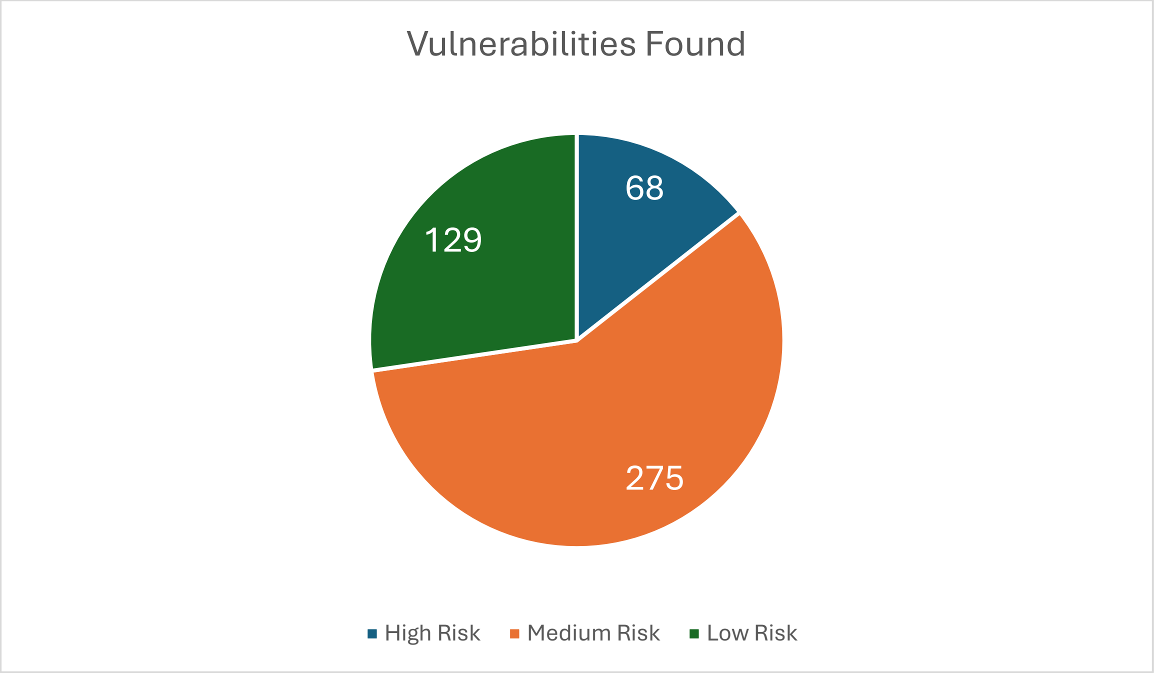
Summary of Data

The initial tests conducted with theHarvester, WHOIS and Dig revealed basic information about the website. WHOIS revealed that the address is hosted by Rackspace Technology Inc, whilst Dig found the IP address and SOA records showing authoritative servers. theHarvester returned no results. This reconnaissance section of testing was expected to return few useful results due to the fact the website is for demonstration purposes and does not belong to a working business.

Scanning with Nmap returned 3 open TCP ports: 80, 443, and 8080. Port 80 is used for http traffic and port 443 is used for https traffic (Jernigan and Meyers, 2022). Port 8080 is used as an alternative to port 80 (Walker, 2024). This information proved useful, as port 8080 was used for Zap vulnerability scanning.

Scanning for SSL/TLS versions and encryption cypher strength evidenced use of TLS versions 1.0, 1.1 and 1.2, alongside Diffie-Hellman key lengths of 1024. OWASP recommends using TLS 1.3, with 1.2 to be supported, and a best practice key length of 2048 bits (OWASP, 2025). NIST recommends a minimum key length of 2048 bits (Barker and Roginsky, 2024). Cryptographic failures were ranked 2nd in OWASP's Top 10 Web Application Security Risks, Vulnerable or Outdated components ranked 6th (OWASP, 2021). This Top 10 informed the baseline for evaluating the target.

Zap vulnerability scanning found 472 vulnerability instances. Fig 1 shows the variation of risk level, according to Zap. The table is a more detailed breakdown of the vulnerability instances.



(Fig 1)

VULNERABILITY	SEVERITY	INSTANCES
Cross Site Scripting **(**DOM Based)	High	63
Cross Site Scripting (Reflected)	High	3
SQL Injection	High	2
Absence of Anti-CSRF Tokens	Medium	4
Cross-Domain Misconfiguration	Medium	1
Missing Anti-clickjacking Header	Medium	79
Content Security Policy Header not set.	Medium	191
Cookie No HttpOnly Flag	Low	2
Cookie without SameSite Attribute	Low	4

Cross Domain JavaScript Source File Inclusion	Low	1
Server Leaks Version Information	Low	1
Timestamp Disclosure - Unix	Low	2
X Content Type Options Header Missing	Low	119

Cross site scripting (XSS) occurs when malicious code is inserted into a webpage. When a user loads the webpage, the code executes, and a variety of attacks can be performed. DOM Based and Reflected XSS refers to the delivery method of the malicious code; client and server respectively (Mitre, 2024).

SQL injection (SQLi) vulnerabilities allow attackers to use database queries to manipulate website and database interaction (Shema and Alcover, 2012). This can lead to multiple attacks; one possible injection could lead to the deletion of a database table, causing information to be unavailable to the owner.

Both XSS and SQLi are classified as Injection, the 3rd ranked risk (OWASP, 2021).

The three vulnerabilities with the most instances all concern headers. Headers can be used as part of a defensive strategy to mitigate certain attacks. For example, Content Security Policy Headers can help guard against XSS scripts (OWASP, 2025). Anti-CSRF tokens provide a similar function, guarding against Cross Site Request Forgery attacks, an attack that hijacks the user’s browser to fulfil attacker requests (Shema and Alcover, 2012). All these attacks, alongside the two cookie vulnerabilities, are part of Insecure Design, the 5th ranked risk (OWASP, 2021).

Use of the target site uncovered Identification and Authentication Failures; the 7th ranked risk (OWASP, 2021). One problem encountered was use of single factor authentication for login, which can lead to authentication attacks. NIST recommends multifactor authentication (Grassi, Garcia and Fenton, 2020).

As part of the testing process the administration login details of username admin, password admin were provided. This presents two problems; default credentials present a security risk in themselves, and passwords should be subject to minimal standards of complexity (OWASP, 2021).

GDPR Compliance

The vulnerabilities found during the investigation place significant doubt on the targets sites ability to meet GDPR compliance.

GDPR calls for appropriate use of security technology in both data in transit and for password policy. Appropriate use can be considered as the recommendations of a national body. According to the NIST guidelines mentioned earlier, the target site would not meet the criteria.

Another area that needs addressing is the sites' lack of a cookie policy. GDPR regulations require opt-in consent for cookies, and for the website to be usable if cookies are refused (European Union, 2016).

PCI DSS and PSR Compliance

These same vulnerabilities also cause compliance issues for specific banking and payment standards. PCI DSS requires multi-factor authentication to be in place by the 31st of March 2025 (Mennes, 2024). PSR compliance is built around following PSD2 regulations, primarily Strong Customer Authentication (SCA). SCA requires multi-factor authentication (Delitz, 2025).

Conclusions and Recommendations

The target contains numerous vulnerabilities that put user security at risk and do not meet required standards for compliance with multiple regulatory bodies. Most vulnerabilities concern insecure design, but the more damaging relate to encryption failures, weakness to injection attacks and authentication failures.

It is recommended to implement the following changes, ranked in order of business importance:

1. Improve authentication and identification processes: add multi-factor authentication, create a password policy that includes complexity rules and make sure default passwords are never used. This addresses the principal factor in failures to meet compliance requirements, whilst also fixing a security issue that affects confidentiality, integrity and availability of data.
2. Improve encryption standards: Upgrade TLS to a minimum of version 1.2, with accommodations made for a further move to version 1.3 at a future date. Upgrade Diffie-Hellman keys to a minimum of 2048 bits. Encryption failures also had an impact on compliance failures, whilst also ranking 2nd in the OWASP top ten website vulnerabilities. These mitigations will protect the integrity of data.
3. Set a cookie policy: introduce opt-in process to bring site in line with GDPR regulations.
4. Set Content Security Policy and X Content Type Options Headers: This helps to resolve most vulnerabilities found in the Zap scan, by mitigating XSS attacks. These measures should be complemented by others, for defence-in-depth a combination of techniques is best (Portswigger, 2025).
5. Set parameterised queries for input to guard against SQL injection attacks (Shema & Alcover, 2012).
6. Resolve insecure design issues: add anti-CSRF tokens, add anti-clickjacking headers and cookies with SameSite only attributes. The most secure websites are secure by design, an approach that should be followed.

Word Count: 1274

References

Barker, E., and Roginsky, A. (2024) Transitioning the Use of Cryptographic Algorithms and Key Lengths. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-131Ar3 ipd. Available at https://doi.org/10.6028/NIST.SP.800-131Ar3.ipd.

Checkmarx (2025) *Zap Documentation*. Available at: https://www.zaproxy.org/docs/ (Accessed 7 March 2025).

Delitz, V. (2025) *PSD2 Passkeys: Phishing-Resistant PSD2-Compliant MFA,* Corbado. Available at https://www.corbado.com/blog/psd2-passkeys (Accessed 8 March 2025)

European Union. (2016) *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation)*. Official Journal of the European Union, L119, 1 – 88. Available at: https://eur-lex.europa.eu/eli/reg/2016/679/oj (Accessed 8 March 2025).

HCL Software (2025) *AltoroJ*, GitHub. Available at https://github.com/HCL-TECH-SOFTWARE/AltoroJ (Accessed 16 February 2025).

HCL Technologies Ltd. (2025) *Altoro Mutual*. Available at http://demo.testfire.net/ (Accessed 9 March 2025).

Hertzog et al. (2021) *Kali Linux Revealed*. United States of America: Offsec Press.

Jernigan, S., and Meyers, M. (2022) *Mike Meyers' CompTIA Network+ Guide to Managing and Troubleshooting Networks, Sixth Edition (Exam N10-008)*. 6th edn. McGraw-Hill.

Lyon, Gordon. (2009) *Nmap Network Scanning: Official Nmap Project Guide to Network Discovery and Security Scanning*. Sunnyvale, CA, USA: Insecure.

Martorella, C. (2025) *theHarvester*, GitHub. Available at https://github.com/laramies/theHarvester (Accessed 16 February 2025).

Mennes, F. (2024) *PCI DSS 4.0: New multi-factor authentication requirements,* OneSpan. Available at https://www.onespan.com/blog/new-mfa-requirements-in-PCI-DSS-4.0 (Accessed 9 March 2025).

Mitre (2024) *CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')*. Available at https://cwe.mitre.org/data/definitions/79.html (Accessed 9 March 2025)

OWASP Foundation (2025) *Transport Layer Security Cheat Sheet*. Available at https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Security_Cheat_Sheet.html (Accessed 8 March 2025).

OWASP Foundation. (2021) *OWASP Top Ten*. Available at https://owasp.org/www-project-top-ten/ (Accessed 8 March 2025).

Portswigger. (2025) *Cross-site Scripting*. Available at https://portswigger.net/web-security/cross-site-scripting (Accessed 8 March 2025).

QUALYS (2025) *SSL Report: demo.testfire.net*. Available at https://www.ssllabs.com/ssltest/analyze.html?d=demo.testfire.net (Accessed 9 March 2025).

Shema, M. & Alcover, J. B. (2012) *Hacking web apps: detecting and preventing web application security problems*. 1st edition. Waltham, Mass: Syngress.

Walker, K. (2024) *Localhost 8080: What is it and how to access it?* Available at https://codeop.tech/blog/accessing-localhost-8080/ (Accessed 9 March 2025).