**Vulnerability Audit and Assessment – Results and Executive Summary**

**Word count: 1,281** (excluding references), below the maximum allowed of 1,320 words (= 1,200 + 120 (10%) words)

Table of Contents

[1. Introduction 2](#_Toc109681954)

[2. Threat Modelling Methodology 2](#_Toc109681955)

[3. Penetration Testing Methodology 4](#_Toc109681956)

[4. Summary of findings 4](#_Toc109681957)

[4.1. Website’s main information 4](#_Toc109681958)

[4.2. Exploratory analysis of security vulnerabilities 4](#_Toc109681959)

[4.3. Systematic vulnerability analysis, risk assessment, and mitigation strategies 6](#_Toc109681960)

[5. Evaluation of website against two security standards: GDPR and PCI DSS 11](#_Toc109681961)

[6. Conclusions 13](#_Toc109681962)

[7. Prioritised recommendations 13](#_Toc109681963)

# Introduction

This document provides an executive summary of the results from a vulnerability assessment carried out on the website <https://ehr-online.co.uk/>, which **detected** security **vulnerabilities** of various degrees of severity further to performing penetration testing (Sheikh, 2021) (**Table 1**). It also outlines the **methodology** leveraged, a summary of the **data**, a comparison against **security standards** **and** applicable **directives**, such as the European Union (EU) “General Data Protection Regulation” (GDPR) (Hussain *et al*., 2020). This document ultimately provides a conclusion and **recommendations** to **improve** the website’s **security** based on the business impact of the vulnerabilities identified (Sheehan *et al*., 2021).

**Table 1**. The characteristics of the penetration testing conducted.

|  |  |
| --- | --- |
| **Organisation’s name** | LibreHealth |
| **Web application’s name** | LibreHealth EHR |
| **URL** | https://ehr-online.co.uk/ |
| **Type of website** | Healthcare-related platform to manage electronic health records (EHR) |
| **Security standards and directives** | ISO/IEC 27001:2005 (ISO, 2013), GDPR, MDR |
| **Threat modelling frameworks** | AWS Well-Architected Framework (Pillar, 2018), STRIDE, DREAD |
| **Penetration testing strategy** | Remote, black-box |
| **Penetration testing frameworks** | PTES, OWASP |

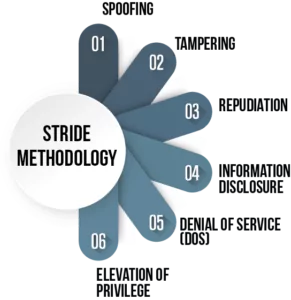
# Threat Modelling Methodology

Amazon Web Services (**AWS**) **Well-Architected Framework** (**Fig. 1**) (Pillar, 2018) and the **STRIDE model** (**Fig. 2**) were leveraged to detect security vulnerabilities (Khan, 2017), which were then ranked based on their risk via the **DREAD model** (**Fig. 3**) (Gómez-Hernández *et al.*, 2021).

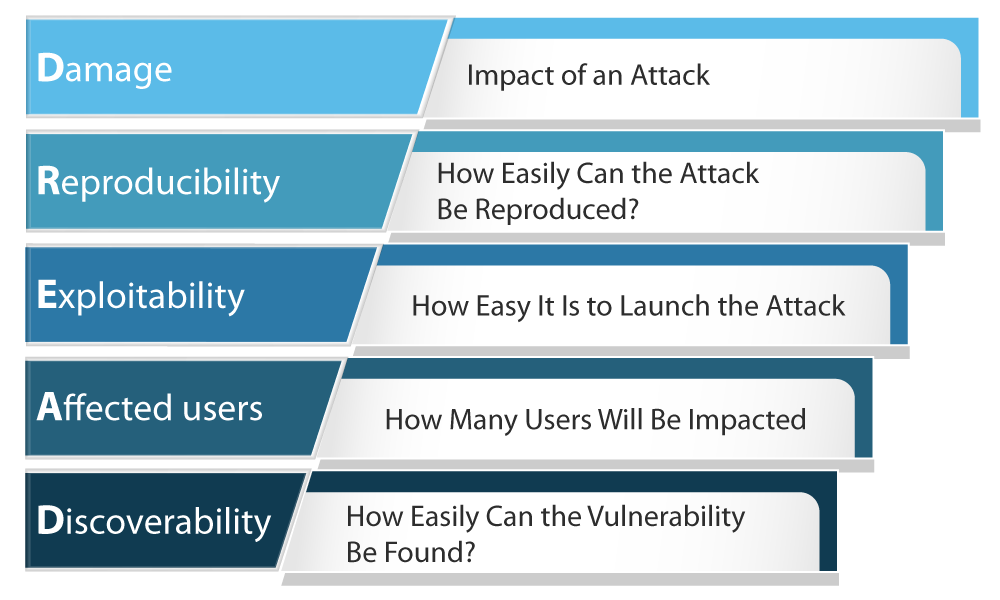
Diagram

Description automatically generated

**Figure 1**. Security-related component of the AWS Well-Architected Framework (adapted from van Staden, 2021).



**Figure 2**. STRIDE model (EC-Council, 2022).



**Figure 3**. DREAD model (EC-Council, 2022).

# Penetration Testing Methodology

The “Penetration Testing Execution Standard” (**PTES**, 2017) and **OWASP** (2021) frameworks were leveraged to guide **black-box remote** penetration testing (Chapple *et al.*, 2018) via **manual and automated** tests to provide more holistic recommendations (Mahmood *et al.*, 2022).

# Summary of findings

## 4.1. Website’s main information

The selected website (ehr-online.co.uk/) was found to use Apache as its web server, and it does not run any CMS. The website is hosted in Amsterdam, the Netherlands. Via WHOIS, the main nameservers for this website were found to be: ns1.a2hosting.com, ns2.a2hosting.com, ns3.a2hosting.com, and ns4.a2hosting.com. Via the DNS Checker, the MX record for this website was found to be: mail.ehr-online.co.uk.

## 4.2. Exploratory analysis of security vulnerabilities

Via the tools available on Kali Linux, the main security vulnerabilities of the website of interest were identified. The website was found **no**t to have a **firewall**.

Out of 13 TCP open ports identified via the NMAP scanner as per **Fig. 4**, the following 4 ports are expected to be open: port 53 for DNS; port 443 for HTTPS (encrypted) traffic; port 993 as a secure port for IMAP, which works via TLS/SSL encryption; port 995 as the encrypted port for POP3, which also works over TLS/SSL. The **other** **9 TCP ports should not be open**, as they may be dangerous when the service listening on such ports is not configured properly, without the required patches, vulnerable to exploits, or has poor network security.

Via the NMAP scanner, one **UDP open** **port** was also identified as per **Fig. 5**, which is a frequent and visible vulnerability in the DNS Bypass Firewall Rules.

**Figure 4**. 13 open TCP ports identified.

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

**Figure 5**. One open UDP port identified.

Graphical user interface, application

Description automatically generated

## 4.3. Systematic vulnerability analysis, risk assessment, and mitigation strategies

The Open Web Application Security Project’s (OWASP) list of vulnerability scanning tools (OWASP, 2022) was reviewed; as a result, the website ‘HostedScan’ was identified and leveraged to conduct a vulnerability assessment on the chosen website.

The OWASP Zed Attack Proxy (ZAP) web penetration testing was performed via the ‘HostedScan’ tool. The OWASP ZAP (Makino & Klyuev, 2015) generates a proxy between the website and the client, trying the website’s features and recording the actions performed. Thereafter, it attacks the website via known methods (Sudhodanan *et al*., 2016).

By leveraging the DREAD risk assessment model, two medium-, four low-, and one informational-level alerts were identified as per **Table 2**. The names and the security risk of the vulnerabilities identified are shown in **Table 3**.

**Table 2**. The number and risk level of the alerts identified.

Table

Description automatically generated with low confidence

**Table 3**. The names and risks of the vulnerabilities identified.

Table

Description automatically generated

Further information on such vulnerabilities is provided in **Tables 4-10**, including evidence of the requests and observations performed during penetration testing and suggested solutions to mitigate the issues identified.

The following security vulnerabilities identified on the selected website were found to be part of OWASP’s (2021) top 10 security vulnerabilities:

* + A04 (the top 4th) – Insecure Design.
    - One vulnerability of medium risk was found as per **Table 4**.
    - Two vulnerabilities of low risk were identified as per **Tables 9 and 10**.
  + A05 (the top 5th) – Security Misconfiguration.
    - One vulnerability of low risk was found as per **Tables 6 and 7**.
  + A06 (the top 6th) – Vulnerable and Outdated Components.
    - One vulnerability of medium risk was identified as per **Table 5**.
  + A10 (the top 10th) – Server-Side Request Forgery (SSRF).
    - One vulnerability of low risk was found as per **Table 8**.

**Table 4**. The vulnerability regarding the ‘Absence of Anti-CSRF Tokens’ and suggested solutions to mitigate it.

Table

Description automatically generated

**Table 5**. The outdated ‘Vulnerable JS Library’ and suggested solution to mitigate it.

Graphical user interface, text, application, email

Description automatically generated

**Table 6**. The vulnerability regarding the ‘Cookie No HttpOnly Flag’ and the suggested solution to mitigate it.

Graphical user interface, text, application, email

Description automatically generated

**Table 7**. The vulnerability regarding the ‘Cookie Without Secure Flag’ and the suggested solution to mitigate it.

Graphical user interface, text, application, email

Description automatically generated

**Table 8**. The vulnerability regarding the ‘Cookie Without SameSite Attribute’ and the suggested solution to mitigate it.

Graphical user interface, text, application, email

Description automatically generated

**Table 9**. The vulnerability regarding the ‘Server Leaks Information via “X-Powered-By” HTTP Response Header Field(s)’ and the suggested solution to mitigate it.

Graphical user interface, application, email

Description automatically generated

**Table 10**. The vulnerability regarding the ‘Re-examine Cache-control Directives’ and the suggested solution to mitigate it.

Graphical user interface, text, application, email

Description automatically generated

# Evaluation of website against two security standards: GDPR and PCI DSS

Via the ImmuniWeb’s GDPR compliance test (Hussain *et al*., 2020), three main vulnerabilities were found that do not comply with GDPR (Hussain *et al*., 2020) as per **Fig. 6**: 1) the lack of a privacy policy, thus infringing Article 13 of GDPR; 2) the website’s components being outdated and containing publicly known security vulnerabilities, as described in detail in **Table 5**, thusbreaching Articles 5.1.f, 24.1, and 32 of GDPR; 3) cookies with sensitive information being sent without a secure flag, as outlined in detail in **Table 7**, thusfurther infringing Articles 5.1.f, 24.1, and 32 of GDPR.

**Figure 6**. Results from the GDPR compliance test.

Graphical user interface, text, application

Description automatically generated

Moreover, as per the PCI DSS security standard, the PCI DSS compliance test was carried out and three requirements were found not have been met (**Fig. 7**), i.e., requirements no. 6.2, 6.5, and 6.6, respectively indicating that outdated components have been used (**Table 5**), which also have publicly known vulnerabilities (**Table 5**), and that no WAF was detected on the selected website, thus being vulnerable against common cyber-attacks.

**Figure 7**. Results from the PCI DSS compliance test.

Graphical user interface, text, application

Description automatically generated

# Conclusions

Further to adequate threat modelling via three frameworks to identify and rank security risks based on their severity (AWS Well-Architected Framework (Pillar, 2018), STRIDE (Khan, 2017), and DREAD (Gómez-Hernández *et al.*, 2021)) as per section 2 and via the industry-grade penetration testing methodologies (PTES (2017) and OWASP (2021, 2022)) in section 3, the main findings were presented in section 4, including the main security vulnerabilities identified, their risks and strategies to mitigate them. Section 5 evaluated the website’s security against two applicable standards (GDPR and PCI DSS) (Hussain *et al*., 2020), which highlighted the importance of the prioritised recommendations in section 7.

# Prioritised recommendations

It is recommended to provide an expedited resolution of the:

1. three misconfigurations and security weaknesses in **Fig. 6**, to ensure compliance with GDPR by resolving considering the importance of such a directive from a business standpoint (Hussain *et al*., 2020),
2. medium-risk security vulnerabilities in **Tables 4 and 5**.

**References**

Chapple, M., Stewart, J. M., & Gibson, D. (2018) *(ISC) 2 CISSP Certified Information Systems Security Professional Official Study Guide*. John Wiley & Sons.

EC-Council (2022) Threat modeling. Retrieved from: <https://www.eccouncil.org/threat-modeling/>. Last accessed on July 21st, 2022.

Gómez-Hernández, J. A., Camacho, J., Holgado-Terriza, J. A., García-Teodoro, P., & Maciá-Fernández, G. (2021) ARANAC: A Bring-Your-Own-Permissions Network Access Control Methodology for Android Devices. *IEEE Access*, 9: 101321-101334.

Hussain, F., Hussain, R., Noye, B., & Sharieh, S. (2020) Enterprise API security and GDPR compliance: Design and implementation perspective. *IT Professional*, 22(5): 81-89.

Khan, S. A. (2017). A stride model based threat modelling using unified and-or fuzzy operator for computer network security. *International Journal of Computing and Network Technology*, 5(01): 13-20.

ISO (2013) ISO/IEC 27001:2013. Information technology - Security techniques - Information security management systems - Requirements. Retrieved from: <https://www.iso.org/standard/54534.html>. Last accessed on June 28th, 2022.

Mahmood, S., Nguyen, H. N., & Shaikh, S. A. (2022) Systematic threat assessment and security testing of automotive over-the-air (OTA) updates. *Vehicular Communications*, 35: 100468.

Makino, Y., & Klyuev, V. (2015) Evaluation of web vulnerability scanners. In *2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)* (Vol. 1, pp. 399-402). IEEE.

OWASP (2021) OWASP Top 10. Retrieved from <https://owasp.org/www-project-top-ten/>. Last accessed on July 15th, 2022.

OWASP (2022) Vulnerability Scanning Tools. Retrieved from <https://owasp.org/www-community/Vulnerability_Scanning_Tools>. Last accessed on July 19th, 2022.

Pillar, R. (2018) AWS well-architected Framework. *Amazon WebServices*, 45.

PTES (2017) PTES Technical Guidelines. Retrieved from: <http://www.pentest-standard.org/index.php/PTES_Technical_Guidelines>. Last accessed on July 20th, 2022.

Sheehan, B., Murphy, F., Kia, A. N., & Kiely, R. (2021). A quantitative bow-tie cyber risk classification and assessment framework. *Journal of Risk Research* 24(12): 1619-1638.

Sheikh, A. (2021) Penetration Testing. In *Certified Ethical Hacker (CEH) Preparation Guide* (pp. 189-196). Apress, Berkeley, CA.

Sudhodanan, A., Armando, A., Carbone, R., & Compagna, L. (2016) Attack Patterns for Black-Box Security Testing of Multi-Party Web Applications. In *NDSS*.

van Staden, M. (2021) AWS Well-Architected Framework. Retrieved from: <https://malcolm.cloud/2021/08/aws-well-architected-framework-summary/waf-summary-min.png>. Last accessed on July 20th, 2022.