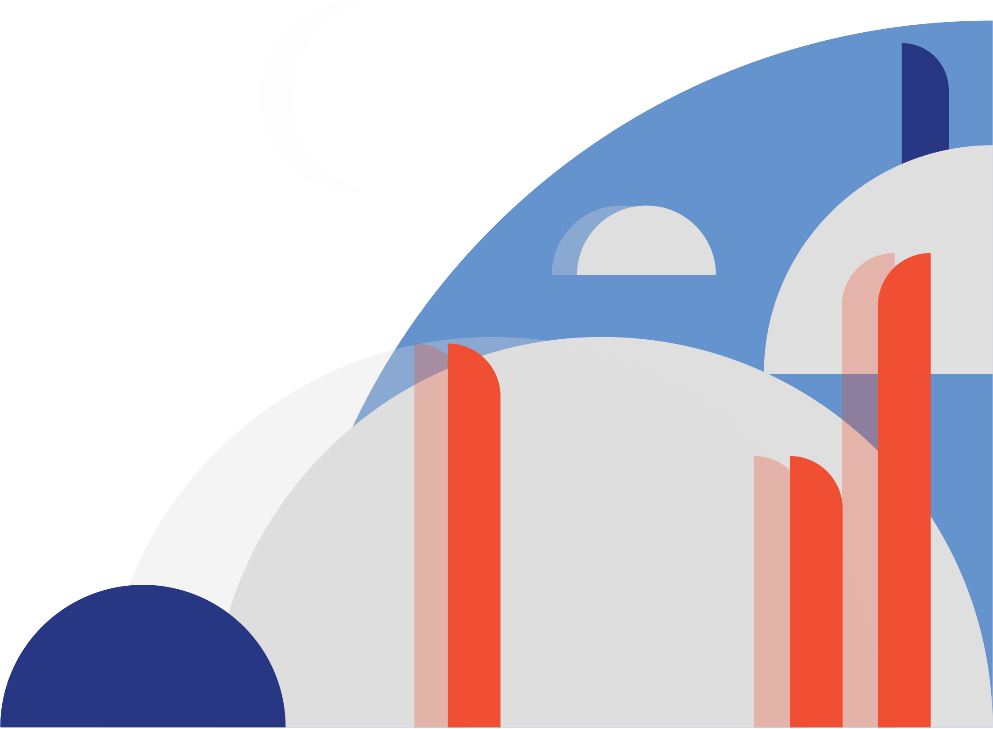
**Vulnerability Assessment Report**

**{{replace\_company}}**

**網站弱點掃描報告**

{{replace\_date\_2}}

1. 摘要

1.1. 檢測依據

**弱點掃瞄方法：**

依循 NIST 800-115 Technical Guide to Information Security Testing and Assessment 。

**弱點掃描內容：**

依據NVD及CVE網站相關弱點，結合National Center for Cyber Security Technology (NCCST)，評估掃描標的是否存在已知的安全弱點，並比照Open Web Application Security Project (OWASP) 的標準將問題進行分類，最後提出修補建議與掃描結果報告。

**弱點掃描工具列表：**

VAS (Vulnerability Assessment as a Service) 服務除了自主研發的Vulnerability Scanner外還整合 了知 名掃描工具，從多種角度驗證弱點，藉此提升掃描的準確性並提供更完整的測試。

所有工具皆已更新至{{replace\_date\_1}}可升級之最新版本，以下為版本資訊與工具簡介。

1. Vulnerability Scanner:

版本: 1.0

簡介:結合OWASP TOP 10、CWE及CVE合規標準及最新資安情資，更全面性、深入 地完成自動化且持續性的弱點評估。

2. OWASP ZAP:

版本: 2.11.1

簡介: ZAP是OWASP組織主導開發的項目，全稱是OWASP Zed attack proxy，是一款 web application集成滲透測試和弱點掃描的檢測工具。

3. Nessus:

版本:10.1.2

簡介: Nessus目前是業界最廣泛使用的漏洞與系統設定評估產品，能對各種遠端系統進行安全 掃瞄，支援超過59,000個漏洞和系統設定檢查。

1.2. 測試範圍

**檢測日期:** {{replace\_date\_2}}–{{replace\_date\_3}}

**受測對象:** 1 網域 (僅1目標檢測到網站功能)

|  |  |
| --- | --- |
| **名稱** | **目標** |

1.3. 結果彙總

{{image\_score}}{{image\_dis}}

表 2：風險資訊彙總

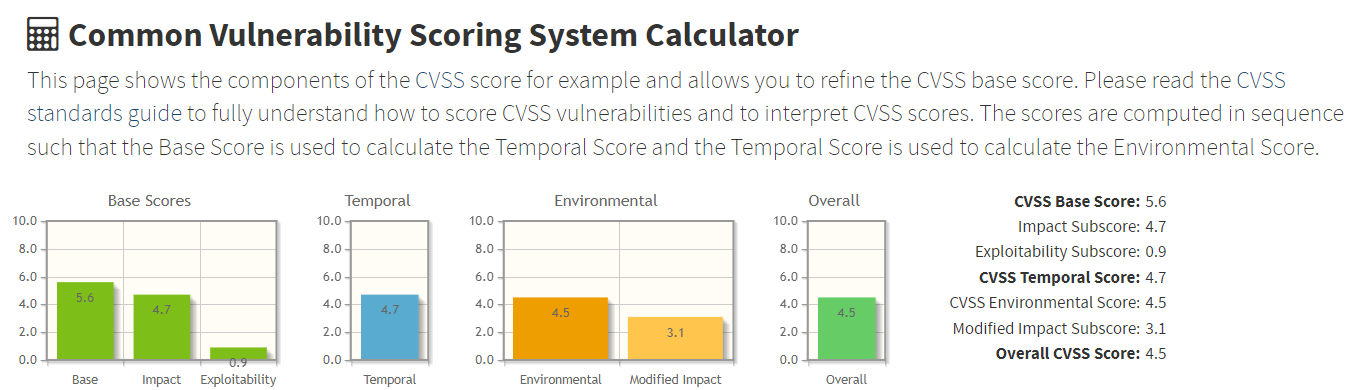
|  |  |  |
| --- | --- | --- |
| **風險編號** | **風險名稱** | **風險等級** |

2. 測試結果

2.1. 風險評估方式

風險評估方式是以CVSSv3為主要參考依據，CVSS (Common Vulnerability Scoring System)漏洞評 鑑系統是由美國國家基礎建設諮詢委員會 (NIAC) 委託製作，是一套公開的評鑑標準，經常被用 來評比企業資訊科技系統的安全性。

依照CVSS的評分分數將弱點分為四個等級，1~3為低風險，4~6為中風險，7~8為高風險，9~10 為最高風險，並對各種等級的問題提供建議採取的措施。



|  |
| --- |
| 表 3：風險計算指標 |
| |  |  |  | | --- | --- | --- | | **風險等級** | **分數** | **說明** | | 低 | 1~3 | 較不易被利用或攻擊的漏洞，不需立即採取補救措施。 | | 中 | 4~6 | 被揭示出重要的資訊，若與其他已發現的風險結合，可能會造成一定程度的影響，建議持續觀察並規劃改善。 | | 高 | 7~8 | 存在已知的弱點或漏洞，有較高的可能被利用並攻擊，建議立即採取修正措施。 | | 極高 | 9~10 | 存在已知的弱點或漏洞，高機率直接影響系統運作，建議立即採取修正措施。 | |

2.2. 風險分類方式

問題分類方式主要參考OWASP訂定的最新版OWASP Top 10，OWASP(Open Web Application Security Project) 是一個開放社群、非營利性組織，全球目前有82個分會，其主要目標是協助 解決網路軟體安全之標準、工具與技術文件，長期致力於協助政府或企業瞭解並改善應用程式的 安全性。

目前OWASP有30多個進行中的計畫，包括最知名的OWASP Top 10，當前之最新版本為 OWASP Top 10 2021，內容如下。

**Top 10 Web Application Security Risks**

[**A01:2021-權限控制失效Broken Access Control**](https://owasp.org/Top10/A01_2021-Broken_Access_Control/)

[**A02:2021-加密機制失效Cryptographic Failures**](https://owasp.org/Top10/A02_2021-Cryptographic_Failures/)

[**A03:2021-注入式攻擊Injection**](https://owasp.org/Top10/A03_2021-Injection/)

[**A04:2021-不安全設計Insecure Design**](https://owasp.org/Top10/A04_2021-Insecure_Design/)

[**A05:2021-安全設定缺陷Security Misconfiguration**](https://owasp.org/Top10/A05_2021-Security_Misconfiguration/)

[**A06:2021-危險或過舊的元件Vulnerable and Outdated Components**](https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/)

[**A07:2021-認證及驗證機制失效Identification and Authentication Failures**](https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/)

[**A08:2021-軟體及資料完整性失效Software and Data Integrity Failures**](https://owasp.org/Top10/A08_2021-Software_and_Data_Integrity_Failures/)

[**A09:2021-資安記錄及監控失效Security Logging and Monitoring Failures**](https://owasp.org/Top10/A09_2021-Security_Logging_and_Monitoring_Failures/)

[**A10:2021-伺服端請求偽造Server-Side Request Forgery**](https://owasp.org/Top10/A10_2021-Server-Side_Request_Forgery_%28SSRF%29/)

2.3. 風險說明

# 3. 結論

{{image\_9}}

依照掃描結果可將問題分為2大類，再針對各類別進行更詳細的說明與建議。

**1. 不安全的設計：**

* 建立與使用安全開發生命週期並且協同應用程式安全的專業人士來評估與設計安全與隱私相關的 控制措施。
* 建立與使用安全設計模式的函式庫或是已完成可使用的元件。
* 使用威脅建模在關鍵的認證、存取控制、商業邏輯與關鍵缺陷上。
* 撰寫單元測試與整合測試來驗證所有的關鍵流程對威脅建模都有抵抗。

**2. 未進行最佳化安全設定：**

* 一個可重複的安全強化流程，必需可達到快速且簡單的佈署，而且能在分隔且封鎖的環境下執行。開發，品質管理，以及實際營運的環境，都須有一致相同的設定，並且使用不同的認證資訊。這種步驟需要盡可能的自動化，降低需要建立安全環境時，所需要的投入。
* 一個最精簡的平台，上面不會搭配任何不需要的功能，套件，檔案，以及範本。移除或不安裝任何不須使用的功能或框架。

4. 參考資料

[1] CVE(https://cve.mitre.org/)

[2] NIST(https://www.nist.gov/)

[3] NVD(https://nvd.nist.gov/)

[4] OWASP(https://owasp.org/)