網站安全重點防禦
網站常見漏洞與案例分享

2025/08/05

目錄

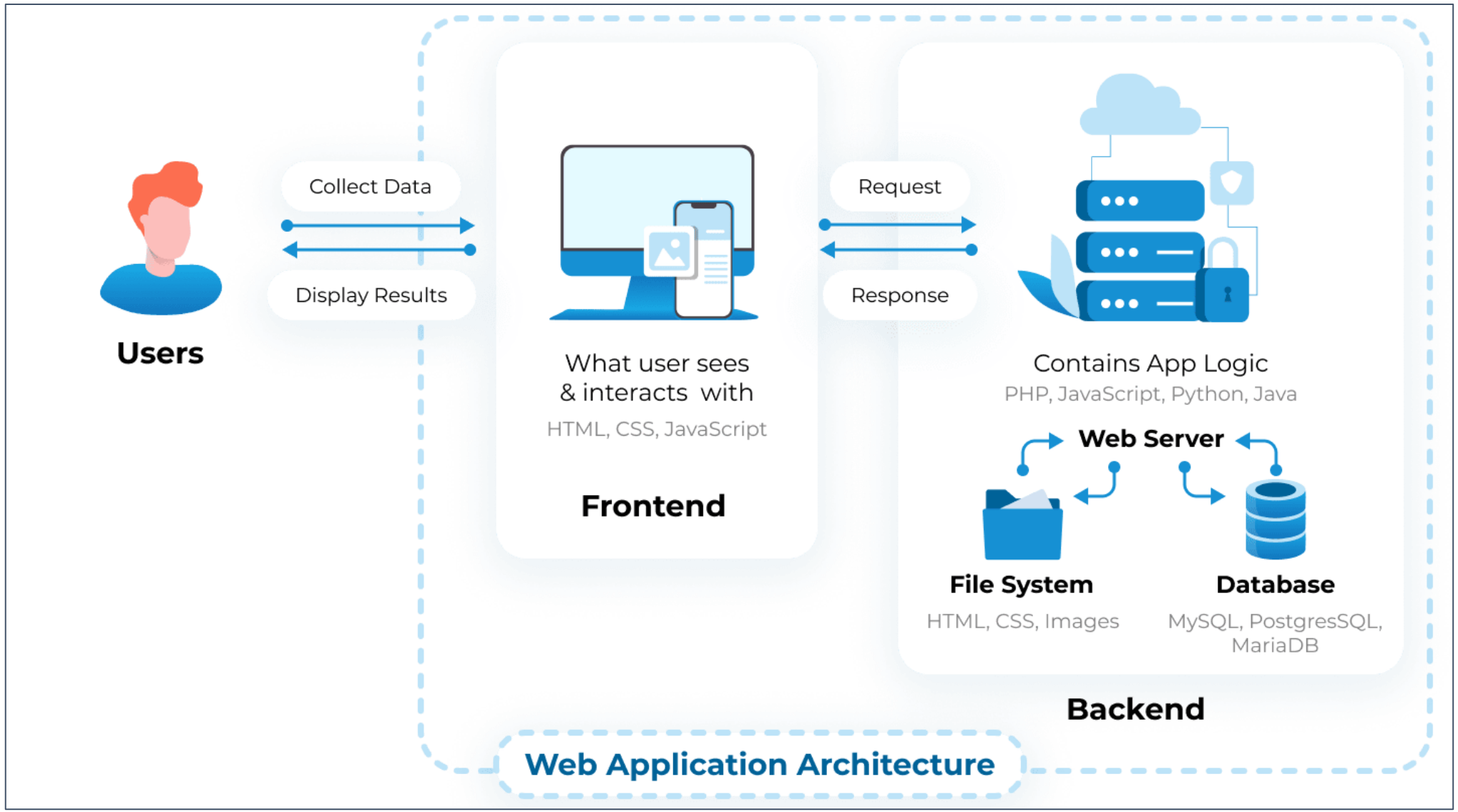
前言

前言

- 網站通常是攻擊者的首選目標
  - 對外長時間開放
  - 結構複雜時，維護困難
  - 從網站突破，橫向移動
- 生成式AI使攻擊門檻降低
  - 自動生成攻擊程式碼
  - 技術門檻降低
  - Cybercrime as a Servant(網路犯罪即代理)

網站架構與基礎簡介

網站架構-前端與後端



https://litslink.com/blog/web-application-architecture

網站架構-前端與後端

網站組成	工具/語言	功能	常見風險
使用者端	瀏覽器請求(Request)		
前端(Frontend)	• HTML/CSS/JavaScript • Framework (Vue/React/Angular) • 與後端API溝通 (AJAX/Fetch/Axios)	• UI呈現 • 互動處理(表單、動畫) • 請求後端API	• XSS • 未驗證的輸入 • 暴露API結構 • 前端框架漏洞 (Jquery、Vue)
後端(Backend)	• Node.js/PHP/Python/JAVA • Database、File Storage	• 邏輯處理 • 存取資料庫 • 驗證與授權	• SQL Injection • Broken Authentication (不安全的身份驗證) • 檔案上傳漏洞 • 未授權的API存取

HTTP Request與Response

- HTTP Request(請求)
  - HTTP Method(方法)
  - HTTP Header(標頭)
  - Path(資源路徑)
  - CRLF(換行符號)
  - Body(內容)
- HTTP Response(回應)
  - HTTP Status Code(狀態碼)
  - HTTP Header(標頭)
  - CRLF(換行符號)
  - Body(內容)

HTTP Method(方法)

- 指定這個資源(URL)進行期望的操作

常見HTTP Method	說明	風險
GET	請求取得(顯示)資源	資訊洩漏(URL帶參數)
POST	傳送(提交)資料 Body攜帶傳輸的資料	XSS、CSRF、SQL Injection
PUT	更新、上傳或取代資源	權限未控管恐遭竄改
DELETE	刪除指定的資源	未驗證重要資源即遭刪除
HEAD	與GET相似 僅取得HTTP狀態碼與標頭	探測資源是否存在
CONNECT	建立與目標資源的通訊隧道 （例如 HTTPS）	可能被用作跳板進行攻擊 （Proxy濫用）
OPTIONS	查詢伺服器支援的方法與設定	暴露伺服器支援的功能
TRACE	回傳收到的請求	可能導致 XST （跨站追蹤攻擊）
PATCH	修改資源的局部內容	權限未控管恐遭竄改

HTTP Status Code(回應狀態碼)

- 主要分成五類：
  - 1XX資訊回應
  - 2XX成功回應
  - 3XX重新導向
  - 4XX用戶端錯誤
  - 5XX伺服器錯誤

常見HTTP Status Code	類型	說明
200 OK	成功	請求成功回應
301	重新導向	永久移動 資源已被永久搬移到新位置
302	重新導向	臨時移動 資源暫時搬移到其他位置
304	重新導向	資源未修改 可使用快取版本
400	用戶錯誤	格式錯誤、不合法參數
401	未授權	沒有登入或Token失效
403	禁止存取	權限不足
404	找不到資源	URL錯誤
500	伺服器錯誤	程式錯誤、資料庫連接錯誤
502	伺服器錯誤	伺服器從上游伺服器收到無效的回應
503	伺服器錯誤	服務暫時無法使用 可能是過載或維護中

HTTP Request(請求)

- Request Line：請求方法+請求資源(URI)+HTTP版本

Request

Pretty Raw Hex

```
1 GET / HTTP/1.1
2 Host: www.klzea.gov.tw
3 Sec-Ch-Ua: "Chromium";v="123", "Not:A-Brand";v="8"
4 Sec-Ch-Ua-Mobile: ?0
5 Sec-Ch-Ua-Platform: "Windows"
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/123.0.6312.122 Safari/537.36
8 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application
  /signed-exchange;v=b3;q=0.7
9 Sec-Fetch-Site: none
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-User: ?1
12 Sec-Fetch-Dest: document
13 Accept-Encoding: gzip, deflate, br
14 Accept-Language: zh-TW,zh;q=0.9,en-US;q=0.8,en;q=0.7
15 Priority: u=0, i
16 Connection: close
17
18
```

HTTP

- Stat

Response

Pretty Raw Hex Render

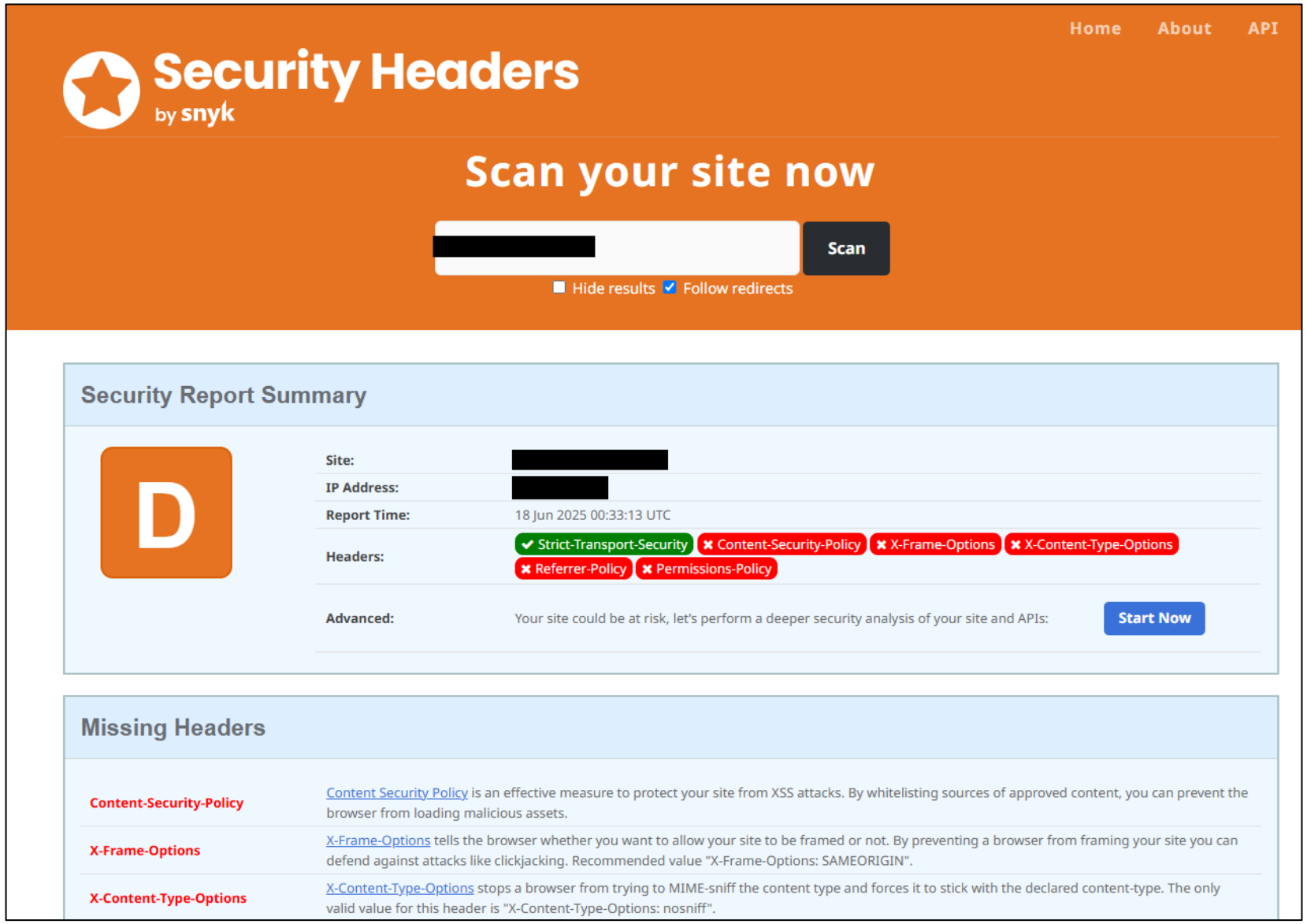
```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html; charset=utf-8
X-XSS-Protection: 1; mode:block
X-Content-Type-Options: nosniff
Feature-Policy: microphone 'none'
Permissions-Policy: camera=()
Referrer-Policy: origin
Strict-Transport-Security: max-age=480;includeSubDomains
Content-Security-Policy: default-src 'none';      script-src 'self' 'unsafe-eval' 'unsafe-inline'
https://cse.google.com/cse.js https://cse.google.com/cse/element/ https://cse.google.com/adsense/search/async-ads.js
https://www.googletagmanager.com/gtm.js https://www.googletagmanager.com/gtag/js
https://www.google-analytics.com/analytics.js https://www.google.com/cse/static/element/
https://partner.googleadservices.com/gampad/cookie.js *.facebook.net;      img-src 'self' data: *.youtube.com
*.ytimg.com https://www.googletagmanager.com/a https://www.google.com.tw/ads/ https://www.google-analytics.com
https://ssl.gstatic.com/ https://www.google.com/cse/static/ https://clients1.google.com
https://encrypted-tbn2.gstatic.com;      frame-src 'self' 'unsafe-eval' 'unsafe-inline' *.youtube.com *.facebook.com
*.google.com https://nchdb.boch.gov.tw;      style-src 'self' 'unsafe-eval' 'unsafe-inline' https://cdn.jsdelivr.net
https://www.google.com/cse/static/;      font-src 'self' 'unsafe-eval' 'unsafe-inline';      frame-ancestors 'self';
     connect-src 'self' https://analytics.google.com/g/collect https://csp.withgoogle.com;      media-src 'self'
X-Frame-Options: SAMEORIGIN
Date: Mon, 02 Jun 2025 06:50:09 GMT
Connection: close
Content-Length: 12541

<!DOCTYPE html>
<html lang="zh-Hant">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1, maximum-scale=1" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <title>
      □□□□□□□□□□□
    </title>
    <meta name="description" content="□□□□□□□□□□□" />
    <meta name="author" content="□□□□□□□□□□□" />

    <!--favicon-->
    <link rel="shortcut icon" href="https://www.k12ea.gov.tw/favicon.ico" />
    <!--main style css-->
    <link rel="stylesheet" href="/Content/geometry/css/style.css?v=20241022" />


    <link rel="stylesheet" href="/Content/geometry/css/style2.css?v=20241022" />
    <!--jquery-->

    <script src="/Scripts/jquery-3.6.0.min.js">
    </script>
```

增加安全性的 HTTP Headers

標頭名稱	參數/值	防禦風險
Set-Cookie	HttpOnly	防止 JavaScript 存取 Cookie 避免 XSS（跨站腳本攻擊）
	Secure	僅允許在 HTTPS 傳輸 Cookie 防止中間人攻擊（MITM）
Content-Security-Policy (CSP)	default-src script-src img-src font-src frame-src	防止 JavaScript 存取 Cookie 避免 XSS（跨站腳本攻擊）
X-Frame-Options	DENY SAMEORIGIN ALLOW-FROM	Clickjacking
X-Content-Type-Options	style script	防止 Content-Type 被竄改
Strict-Transport-Security (HSTS)	Max-age=……	強制使用HTTPS 防止中間人攻擊（MITM）
Referrer-Policy	no-referrer	增加隱私
	origin	

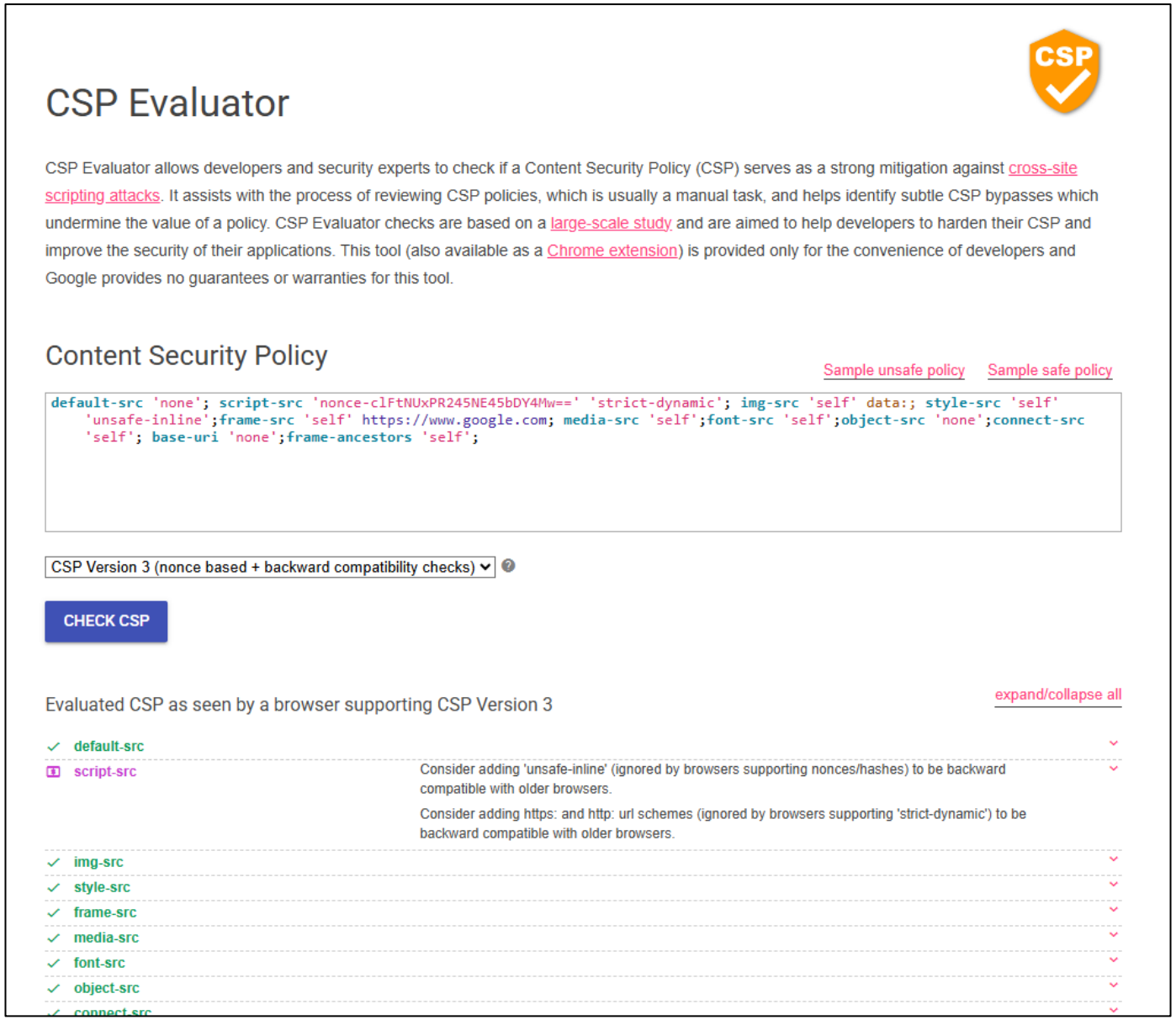
https://yu-jack.github.io/2017/10/20/secure-header/

Security Headers



https://securityheaders.com/

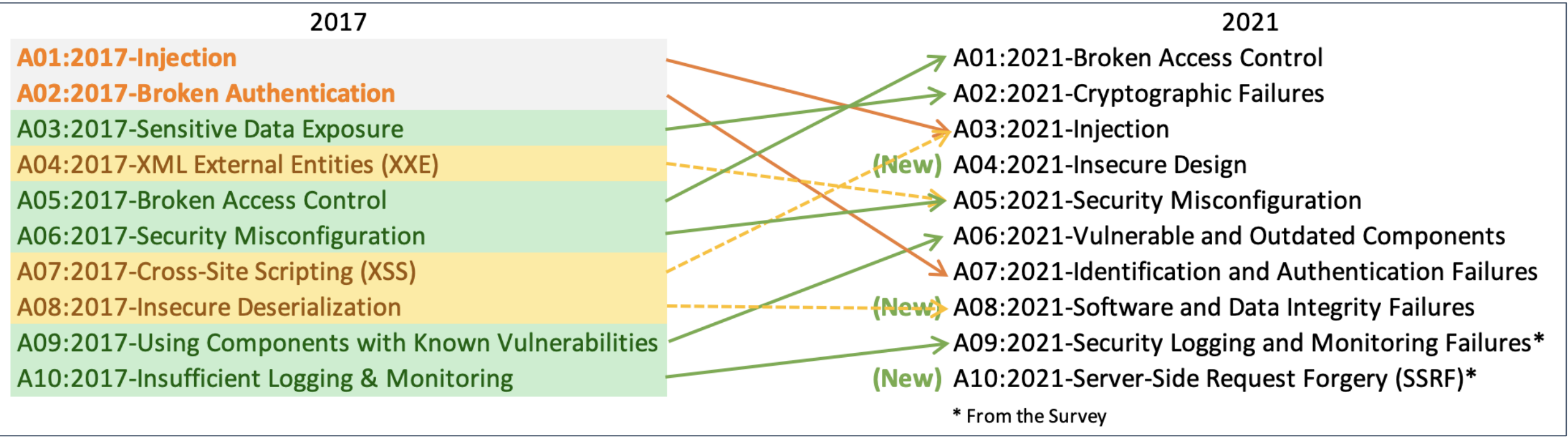
Google CSP



https://csp-evaluator.withgoogle.com/

網站常見漏洞
OWASP TOP 10

OWASP TOP 10



https://owasp.org/Top10/zh_TW/

A01:2021-Broken Access Control
無效的存取控管

A01:2021-Broken Access Control
無效的存取控管

- 未正確限制用戶對資源的存取權限，導致未經授權的使用者可以執行特權操作或存取非公開資料

直接引用資料庫物件的IDOR漏洞

以下URL可直接存取客戶的帳戶頁面

`https://example.com/customer_account?customer_ID=`**122333**

若無存取控制，攻擊者可直接修改值(customer_ID)，繞過存取控制查看其他客戶帳戶頁面，此為水平越權範例，可能導致資訊外洩，以及資訊遭竄改。

一般使用者可執行管理員功能

- 水平越權（Horizontal Privilege Escalation）

  使用者存取其他使用者的資料

- 未驗證資源刪除/修改操作API

  缺乏權限驗證，導致非法操作

A01:2021-Broken Access Control
建議防護方法

- 除可公開資源，其他資源一律禁止任何存取行為
- 停用伺服器的目錄清單功能(Directory Listing)
- 確認每個功能(及資源)的各類型權限皆分配合宜
- 設定適當的CORS存取來源與方法
- 正確使用OAuth、SAML、JWT、OpenID Connect機制

A02:2021-Cryptographic Failures
加密機制失效

A02:2021-Cryptographic Failures
加密機制失效

- 資料在傳輸過程中，未正確使用加密方式或使用較弱的演算法，導致敏感資訊可能被竊取、竄改或濫用。
- 漏洞類型：
  - 傳輸過程中使用明文(HTTP)
  - 使用較弱的加密演算法(MD5、SHA-1、DES、RC4)
  - 敏感資料未加密儲存
  - 金鑰寫在程式碼內

A02:2021-Cryptographic Failures
建議防護方法

- 選用較強的加密演算法
  - AES-256（對稱加密）RSA-2048
  - ECC（非對稱加密）
  - SHA-256 以上雜湊演算法
- 傳輸敏感資料使用TLS 1.2(1.3)/HTTPS
- 加強密碼與帳號的管控
  - 加鹽雜湊
  - 限制嘗試次數
- 安全的憑證
  - 定期更新憑證
- 設定安全標頭(CSP、HSTS)

Google Chrome瀏覽器 x 中華電信簽發憑證

8 月起全面生效，Google Chrome 將移除中華電信憑證預設信任

作者 邱 德芯 | 發布日期 2025 年 06 月 02 日 18:31 | 分類 Google , 科技生活 , 資訊安全 分享 分享 Follow 分享 查看翻譯

Google 近日宣布，自 2025 年 8 月起，將在 Chrome 瀏覽器中取消對中華電信與匈牙利憑證機構 Netlock 所簽發憑證的預設信任，此變更將適用於 Windows、macOS、ChromeOS、Android 與 Linux 平台的 Chrome 139 版本以上。Google 強調，這項措施是為了維護使用者安全與憑證體系的整體信任基礎。

https://infosecu.technews.tw/2025/06/02/google-chrome-cht-certification-authority/?fbclid=IwY2xjawKqhnhleHRuA2FlbQIxMQABHgdVvBPmLp06tEKu

A03:2021-Injection
注入攻撃

A03:2021-Injection
注入攻擊

- 應用程式未對使用者提供的數據經過安全性的檢查、驗證及過濾。
- 常見漏洞類型：
  - SQL Injection
  - Cross-Site Scripting
  - Command Injection
  - NoSQL Injection

A03:2021-Injection
SQL Injection

- 將 SQL 指令注入資料查詢語句，操控資料庫行為
  - Authentication Bypass：繞過登入驗證
  - Union-based Injection：以union串聯查詢資料
  - Error-based Injection：從錯誤訊息中取得資訊
  - Blind Injection：由於無錯誤訊息回傳，透過多次測試來取得資訊
    (系統反應的時間或邏輯差異)

https://tech-blog.cymetrics.io/posts/nick/sqli/

A03:2021-Injection
Cross Site Scripting(XSS)

- 攻擊者將惡意腳本注入到網頁中，在其他使用者瀏覽時執行，目的是竊取cookie、session、進行釣魚或偽造使用者操作。
- 常見類型：
  - 反射型 (Reflected XSS)

    使用者點擊惡意連結，立即回應執行
  - 儲存型 (Stored XSS)

    被害者瀏覽網頁時自動觸發，內容儲存在伺服器(如:留言板、聊天室)
  - DOM-Based XSS

    惡意腳本存在於前端 JavaScript 操作 DOM 時

https://blog.huli.tw/2021/10/25/learn-frontend-from-security-pov/

A03:2021-Injection
Command Injection

- 攻擊者將惡意系統命令注入應用程式，並在伺服器上執行多數發生在使用者輸入拼接成系統指令(如:shell、PowerShell、cmd)，藉此獲取機敏資訊或執行未經授權的任意指令。
- 風險：
  - 執行任意系統指令
  - 資料遭竄改、外洩或刪除
  - 伺服器被植入惡意程式

https://www.cc.ntu.edu.tw/chinese/epaper/0039/20161220_3905.html

A03:2021-Injection
建議防護方法

- 使用參數化查詢(Parameterized Queries)
- 使用ORM(Object-Relational Mapping，物件關聯對映)工具避免動態拼接 SQL 字串
- 對輸入的資料進行嚴格驗證與清洗
- 採用白名單機制(Whitelisting)檢查資料類型、格式、長度
- 使用安全的 API 或框架避免直接使用 Shell、系統命令函式（如： exec(), system()）
- 針對不同語言或上下文(HTML, SQL, JSON, JavaScript)採用適當轉義 (Escaping)
- 限制帳號權限並針對使用者授與最低權限(Least Privilege Principle)
- 啟用 Web Application Firewall (WAF)偵測並阻擋常見的注入攻擊模式

A04:2021-Insecure Design
不安全設計

A04:2021-Insecure Design
不安全設計

- 系統設計階段未考慮資安風險與保護措施，屬系統設計與架構中的風險。
- 屬於這類型的漏洞範例：
  - 無強制驗證使用者身分或授權機制
  - 功能濫用
  - 敏感資料傳輸未加密
  - 不合理的前端邏輯
  - 缺乏防禦暴力破解或嘗試次數限制

A04:2021-Insecure Design
建議防護方法

- 最小權限原則（Principle of Least Privilege）
- 每個帳號或服務僅擁有完成工作所需的最低權限。
- 限制功能性 API、管理介面等的存取。
- 拒絕預設允許（Deny by Default）系統預設應拒絕所有請求，僅允許特定信任來源或操作。
- 防失敗機制（Fail Secure）系統在錯誤、逾時或異常時，應保有安全狀態。

A05:2021-Security Misconfiguration
不安全的組態設定

A05:2021-Security Misconfiguration
不安全的組態設定

- 伺服器、網路服務、平台、應用程式、元件、框架、虛擬機或容器等在部署時沒有實施正確的設定，或使用了預設設定、錯誤的設定、過度授權、顯示過多錯誤訊息等情況，皆屬於「不安全的組態設定」。
- 常見的情境：
  - 使用預設帳號密碼
  - 錯誤訊息詳細顯示伺服器資訊
  - 未移除不必要的功能 / 模組(如:測試頁、debug工具)
  - 檔案或目錄權限過大

A05:2021-Security Misconfiguration
建議防護方法

- 強化預設組態與安裝安全性
  - 移除預設帳號與密碼
  - 停用未使用的功能、服務或模組
  - 移除phpinfo()、除錯介面
- 自動化與標準化配置管理
- 權限與存取控制
  - 依權責與功能實行最小化權限
  - 僅允許可信任的IP存取管理介面
  - 多重因素驗證(MFA)
- 監控與漏洞管理
- 安全導向的開發與部署流程

A06:2021-Vulnerable and Outdated Components
危險或過舊的元件

A06:2021-Vulnerable and Outdated Components
危險或過舊的元件

- 系統中使用了包含已知漏洞或已停止更新或維護的：
  - 函式庫 / 套件(Library/Package)
  - 框架(Framework)
  - 作業系統
  - 平台依賴服務(如 :Web Server、資料庫等)
- 這些元件可能被攻擊者利用，導致資料洩漏、遠端執行、權限提升等風險。

A06:2021-Vulnerable and Outdated Components
建議防護方法

- 維護資產清單
  - 列出已使用的第三方元件(包含版本、來源及授權等資訊)
- 定期執行弱點掃描
- 使用受信任的來源安裝元件
  - 優先使用官方套件庫
- 定期更新與修補弱點
- 移除未使用的元件

A07:2021-Identification and Authentication Failures
認證及驗證機制失效

A07:2021-Identification and Authentication Failures
認證及驗證機制失效

- 系統身分驗證相關功能包含缺陷，未能妥善實施或保護「身份驗證」與「使用者會話管理」的機制，導致攻擊者能夠繞過相關驗證機制或者利用開發設計上的缺陷，破解取得密碼或Session Token，暫時性或永久性的冒充其他用戶身分並存取相關資源。
- 相關漏洞：
  - 允許暴力密碼破解攻擊
  - 使用預設帳號密碼、弱密碼、暴露在公開網路上的密碼
  - 使用明文、較弱的加密方式儲存密碼
  - 缺少或無效的多重身分驗證
  - 在URL中公開Session

A07:2021-Identification and Authentication Failures
建議防護方法

- 強化密碼政策
  - 密碼長度、複雜度可參考密碼規範NIST-800-63
- 防止密碼暴力破解，實施帳戶鎖定，限制或延遲失敗的登錄嘗試
- 使用安全的方式儲存密碼
- 實施多重因素驗證機制(multi-factor authentication, MFA)
- 安全的Session管理
- 使用成熟的認證框架
- 加強稽核與監控
- 定期執行檢測與滲透測試

A08:2021-Software and Data Integrity Failures
軟體及資料完整性失效

A08:2021-Software and Data Integrity Failures
軟體及資料完整性失效

- 當系統的程式碼或基礎架構未能保護重要軟體或資料的完整性，使攻擊者有機會植入惡意程式、篡改資料或操控更新流程。
- 相關漏洞：
  - 應用程式的外掛或套件來自於不受信任的來源
  - 供應鏈攻擊
  - 不安全的持續性整合/部署(CI/CD)流程

A08:2021-Software and Data Integrity Failures
軟體

下載超過2千萬次的Docker映像檔被爆含有挖礦軟體

這次Palo Alto Networks的研究人員，只利用電子錢包位址關聯分析，就找出大量活躍的惡意映像檔，代表實際災情可能更為嚴重

文/ 林妍溱 | 2021-03-31 發表

讚 680 分享

情境示意圖，Photo by Ayana Fragoso on pixy

https://www.ithome.com.tw/news/143546

A08:2021-Software and Data Integrity Failures
建議防護方法

- 使用受信任的套件與映像來源
- 強化 CI/CD 安全性
  - 加強修改 / 部屬程式碼權限控管
- 原始碼版本控制加簽驗證
- 執行軟體組成(成分)分析(SCA)掃描
- 建立供應鏈風險管理政策
- 實施 DevSecOps

A09:2021-Security Logging and Monitoring Failures
資安記錄及監控失效

A09:2021-Security Logging and Monitoring Failures
資安記錄及監控失效

- 未記錄或監控系統相關事件/日誌/紀錄，導致攻擊發生時無法偵測、即時反應、調查及應變，甚至攻擊者進一步持續探索並攻擊其他更多系統。
- 常見情況與風險：
  - 未紀錄系統登入失敗、異常行為與存取敏感資料
  - 未設立異常警示或即時阻斷異常行為
  - 無法調查或還原事件發生過程
  - 相關日誌只儲存在本地(本機)，已遭攻擊者刪除或竄改

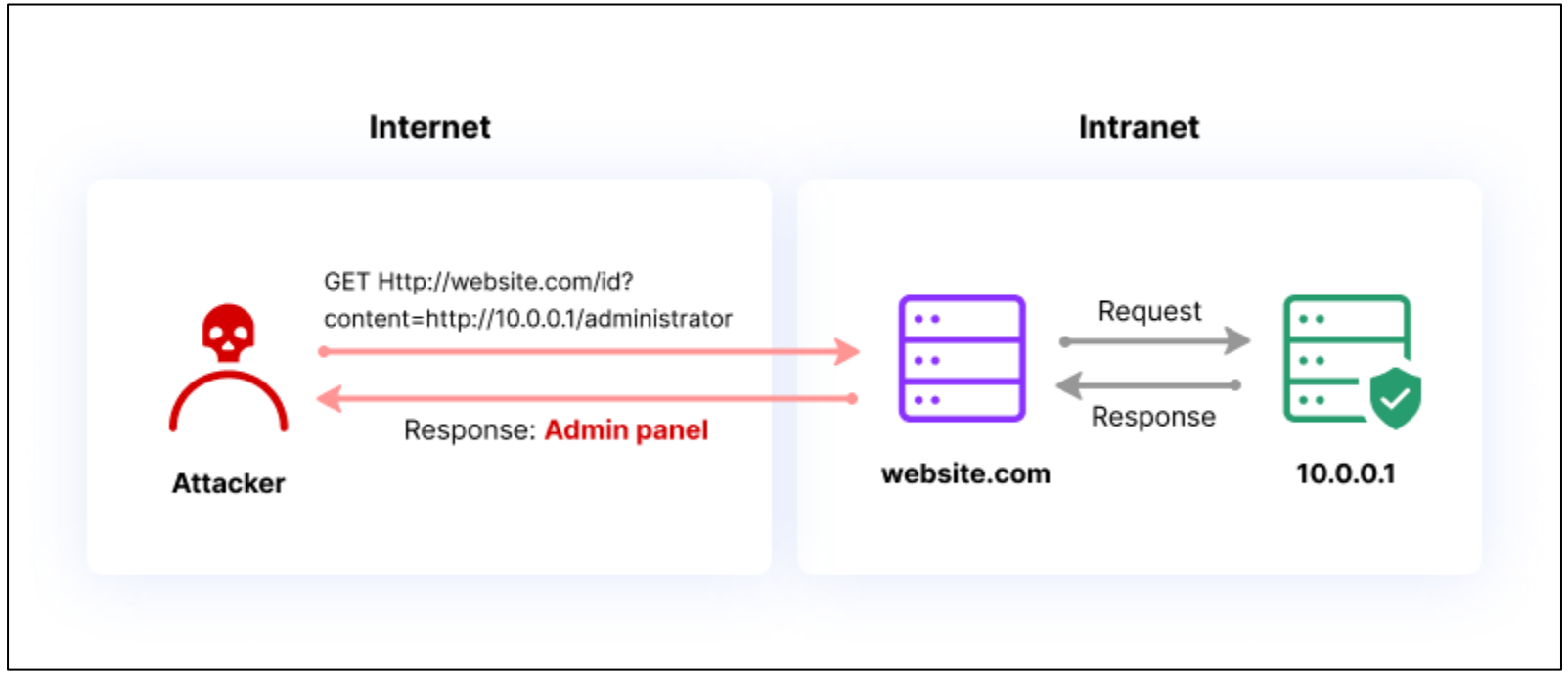
A09:2021-Security Logging and Monitoring Failures
建議防護方法

- 啟用完整的安全記錄(Security Logging)
  - 確保所有登入、存取控制失敗、輸入驗證失敗能夠被記錄到日誌中
- 實施異常行為偵測（Behavior Analytics）
- 事件警示與通知
- 定期稽核與日誌審查
- 建立事件回應流程
  - 安全事件應變(IR)
  - 緊急應變處理(ERS)
  - 復原計畫(DR)

A10:2021-Server-Side Request Forgery(SSRF)
伺服端請求偽造

A10:2021-Server-Side Request Forgery(SSRF)
伺服端請求偽造

- 攻擊者誘使伺服器向非預期的位置發送惡意請求，存取內部資源或外部系統，可能導致資訊外洩或橫向攻擊。



https://www.imperva.com/learn/application-security/server-side-request-forgery-ssrf/

A10:2021-Server-Side Request Forgery(SSRF)
建議防護方法

- 封鎖內部網路請求
  - 設定防火牆(如:iptables)拒絕內部 IP 的 outbound 請求
- 白名單 URL 檢查
  - 僅允許信任的URL或網域，並對輸入參數進行嚴格驗證。
  - 不允許用戶輸入完整URL或IP直接作為請求目標。
- 停用不必要的 URL 轉發功能
- 監控與紀錄
  - 記錄外部與內部的HTTP請求log，進行異常行為偵測與追蹤。

重大漏洞與案例

CVE
Common Vulnerabilities and Exposures

- 由美國 MITRE (非營利機構)主導與維護
- 漏洞編號組成CVE+2025+12345
- Known Exploited Vulnerabilities Catalog
  - 已分配CVE編號
  - 漏洞已有被廣泛利用的跡象
  - 漏洞有明確的修補措施，或已提供更新

平台/工具	嚴重程度
NVD	• CVSS 評分 • 漏洞細節 • 修補建議
MITRE CVE	漏洞基本資訊
Exploit-DB GitHub	漏洞PoC
Nessus OpenVas	漏洞檢測工具

CVSS
Common Vulnerability Scoring System

- CVSS v3.x分數組成：
- **基本評分(Base Score)**
  - **攻擊向量（Attack Vector, AV）**
  - **攻擊複雜度（Attack Complexity, AC）**
  - **所需權限（Privileges Required, PR）**
  - **使用者互動（User Interaction, UI）**
  - **影響程度（機密性C、完整性I、可用性A）**
- 時間評分(Temporal Score)
  - 漏洞修補狀態、攻擊代碼成熟度等
- 環境評分(Environmental Score)
  - 對特定組織環境的影響調整（是否部署了緩解措施）

分數範圍	嚴重程度
0.0	無風險
0.1-3.9	低(Low)
4.0-6.9	中(Medium)
7.0-8.9	高(Hight)
9.0-10.0	嚴重(Critical)

https://nvd.nist.gov/
https://cve.mitre.org/
https://www.first.org/cvss/calculator/3-1

CVE-2024-4577 x CVSS v3.x

CVE-2024-4577 Detail

Description

In PHP versions 8.1.* before 8.1.29, 8.2.* before 8.2.20, 8.3.* before 8.3.8, when using Apache and PHP-CGI on Windows, if the system is set up to use certain code pages, Windows may use "Best-Fit" behavior to replace characters in command line given to Win32 API functions. PHP CGI module may misinterpret those characters as PHP options, which may allow a malicious user to pass options to PHP binary being run, and thus reveal the source code of scripts, run arbitrary PHP code on the server, etc.

Metrics

CVSS Version 4.0 | CVSS Version 3.x | CVSS Version 2.0

NVD enrichment efforts reference publicly available information to associate vector strings. CVSS information contributed by other sources is also displayed.

CVSS 3.x Severity and Vector Strings:

CNA: PHP Group **Base Score:** 9.8 CRITICAL **Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

CVSS v3.1 Severity and Metrics:
Base Score: 9.8 CRITICAL
Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
Impact Score: 5.9
Exploitability Score: 3.9

Attack Vector (AV): Network
Attack Complexity (AC): Low
Privileges Required (PR): None
User Interaction (UI): None
Scope (S): Unchanged
Confidentiality (C): High
Integrity (I): High
Availability (A): High

References to Advisories, Solutions, and Too

By selecting these links, you will be leaving NIST webspace. We have provided these have information that would be of interest to you. No inferences should be drawn on acc from this page. There may be other web sites that are more appropriate for your purpose. NIS xpressed, or concur with the facts presented on these sites. Further, NIST does not endorse an ned on these sites. Please address comments about this page to nvd@nist.gov.

Hyperlink
http://www.openwall.com/lists/oss-security/2024/06/07/1

QUICK INFO

CVE Dictionary Entry:
CVE-2024-4577
NVD Published Date:
06/09/2024
NVD Last Modified:
03/28/2025
Source:
PHP Group

https://nvd.nist.gov/vuln/detail/cve-2024-4577

KEV

微軟 | SHAREPOINT

CVE-2025-49706

Microsoft SharePoint 驗證不當漏洞： *Microsoft SharePoint 包含一個驗證不當漏洞，允許授權攻擊者透過網路進行欺騙。成功利用該漏洞，攻擊者可以查看敏感資訊並對已披露的資訊進行一些更改。此漏洞可能與 CVE-2025-49704 關聯。 CVE-2025-53771 的更新比 CVE-2025-49706 的更新提供了更強大的保護措施。*

相關 CWE: CWE-287

已知用於勒索軟體活動？**未知**

行動： CISA 建議中斷已達到生命週期終止 (EOL) 或服務終止 (EOS) 的面向公眾的 SharePoint Server 版本。例如，SharePoint Server 2013 及更早版本已達到生命週期終止，如果仍在使用，應立即停止使用。對於受支援的版本，請根據 CISA 和供應商的說明採取緩解措施。請遵守適用於雲端服務的 BOD 22-01 指南，或如果沒有可用的緩解措施，請停止使用該產品。

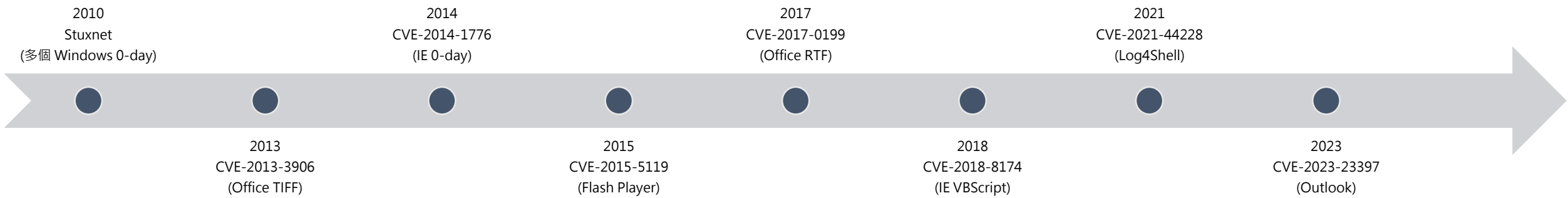
- **新增日期：** 2025-07-22
- **到期日：** 2025-07-23

附加說明 +

https://www.cisa.gov/known-exploited-vulnerabilities-catalog

零時差漏洞利用
0 day(zero-day)零日漏洞

- CVE推出，但供應商還未釋出更新(修補)程式
  - 駭客就目前釋出資訊研究出利用方法
- 更新剛推出，但使用者還來不及更新
- 黑帽駭客找到漏洞但不去通報
  - 自己利用來攻擊
  - 釋出於暗網交易平台(高價售出)



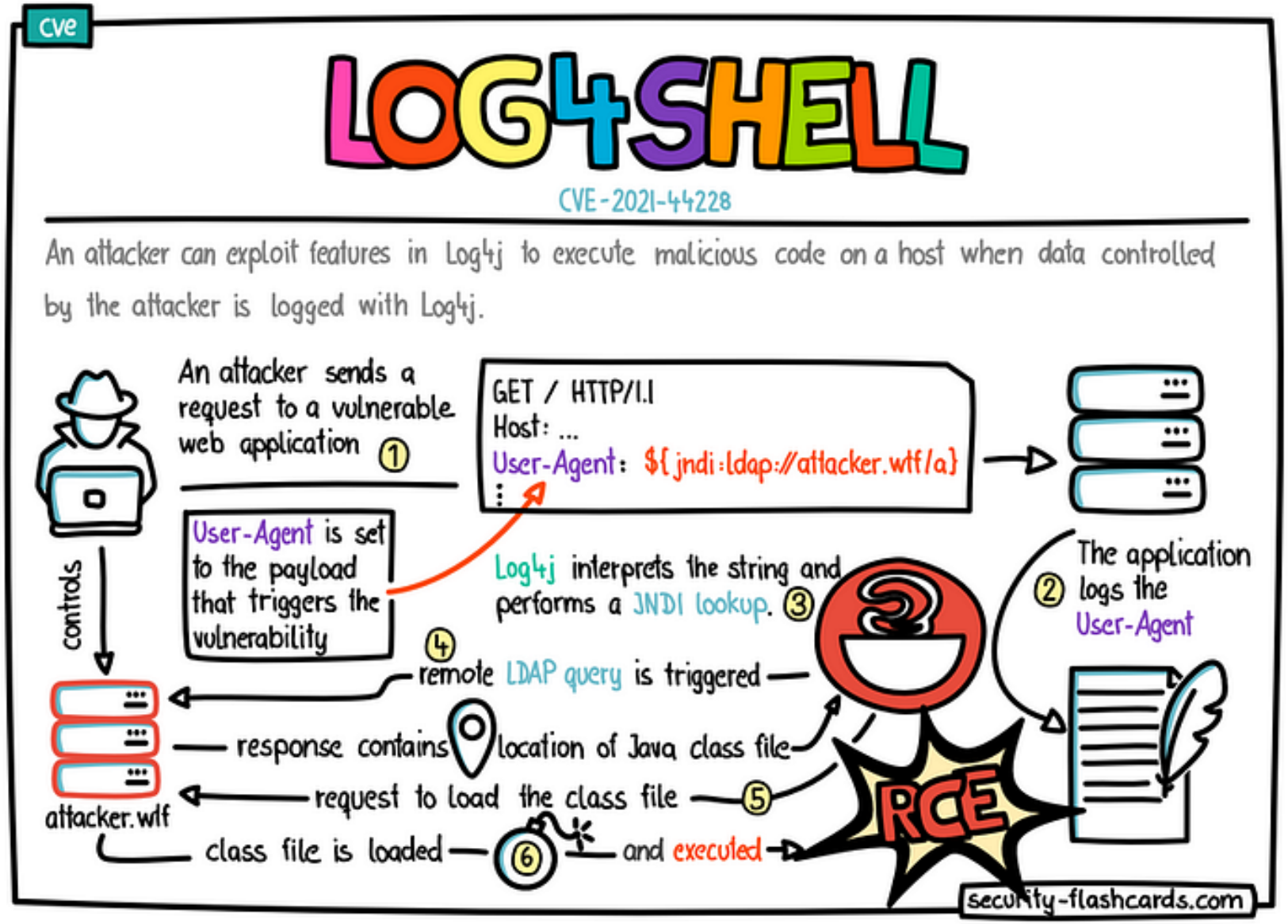
CVE-2021-44228
Apache Log4j2零日漏洞

CVE-2021-44228
Apache Log4j2零日漏洞

- Log4Shell x Apache Log4j2 RCE漏洞
- CVSS 3.0評分10分(最嚴重10分)
- 受影響範圍：Java開源日誌資料庫Apache Log4j 2.0-beta9至2.14.1 JNDI Lookups 插件
- 受影響企業：Microsoft、Apple iCloud、Cisco、NetApp、Steam、Minecraft
- 解決方法：
  - 1. 安裝Apache推出的Log4j 2.15版
  - 2. 使用較新版本的Java SDK（JDK），限制JNDI漏洞利用

https://medium.com/cycraft/cve-2021-44228-ffad06b53002
https://www.cht.com.tw/home/campaign/IntranetService/doc/002.CVE_2021_44228.pdf

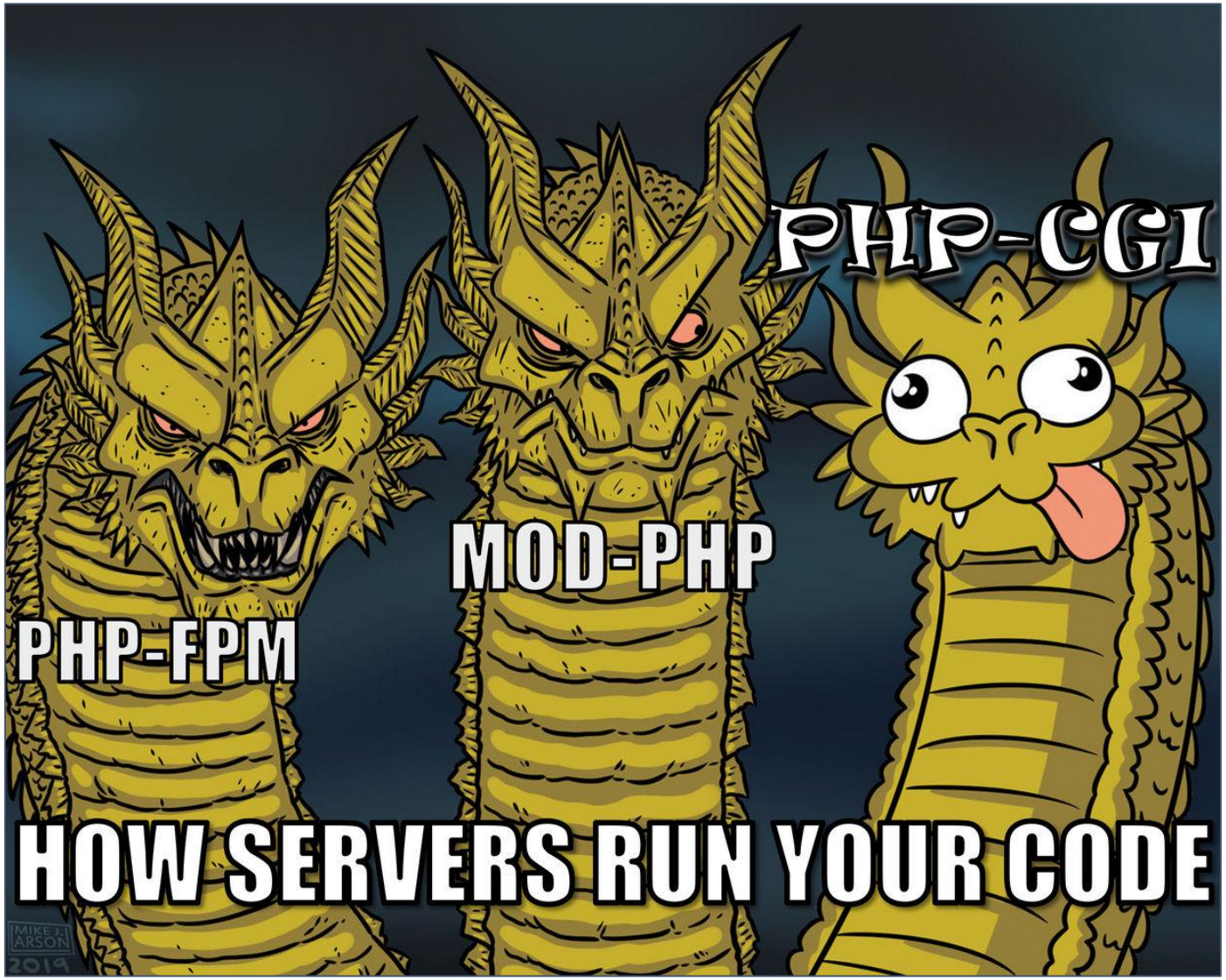
CVE-2021-44228



https://medium.com/@aka.0x4C3DD/log4j-vulnerability-cve-2021-44228-log4shell-a-zero-day-remote-code-execution-exploit-9440cdb1e771

CVE-2024-4577
Windows x PHP CGI

CVE-2024-4577



https://devco.re/blog/2024/06/06/security-alert-cve-2024-4577-php-cgi-argument-injection-vulnerability/

CVE-2024-4577
PHP x XAMPP

2. 對 XAMPP for Windows 使用者

在撰寫本文的當下，XAMPP 尚未針對此漏洞釋出相對應的更新安裝檔，如確認自身的 XAMPP 並無使用到 PHP CGI 之功能，可透過修改下列 Apache Httpd 設定檔以避免暴露在弱點中:

> C:/xampp/apache/conf/extra/httpd-xampp.conf

找到相對應的設定行數:

```
ScriptAlias /php-cgi/ "C:/xampp/php/"
```

並將其註解:

```
# ScriptAlias /php-cgi/ "C:/xampp/php/"
```

https://devco.re/blog/2024/06/06/security-alert-cve-2024-4577-php-cgi-argument-injection-vulnerability/

驗證工具與方法

資訊蒐集與漏洞測試工具

- 資訊蒐集
  - Google Hacking
  - Wapplyzer
  - Shodan(Censys/FOFA/ZoomEye)
  - WPScan(WordPress)
  - 瀏覽器
- 漏洞測試
  - nmap
  - Burp Suite
  - Sqlmap
  - Exploit DB
  - 瀏覽器+curl(Bash)+CurlConverter
  - Metasploit

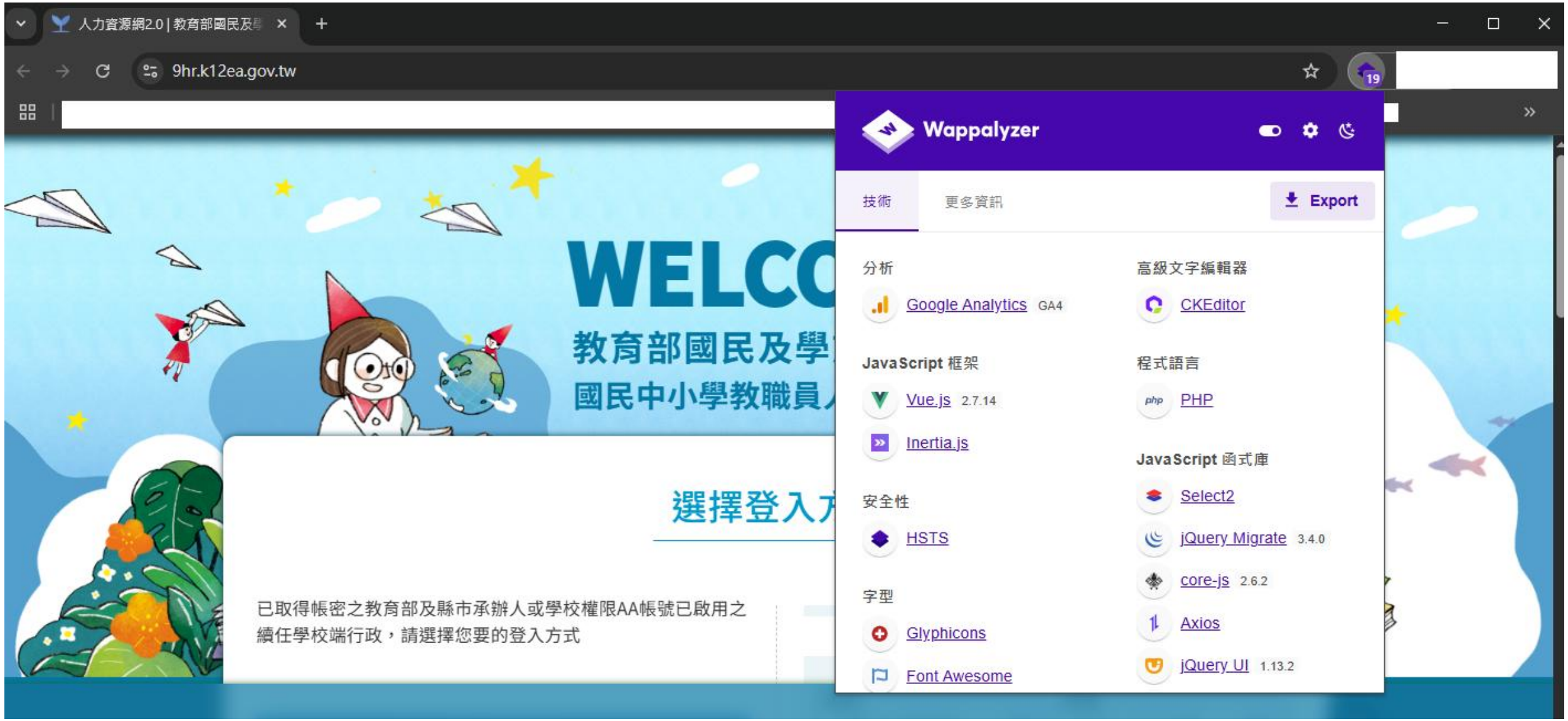
資訊蒐集

Google Hacking

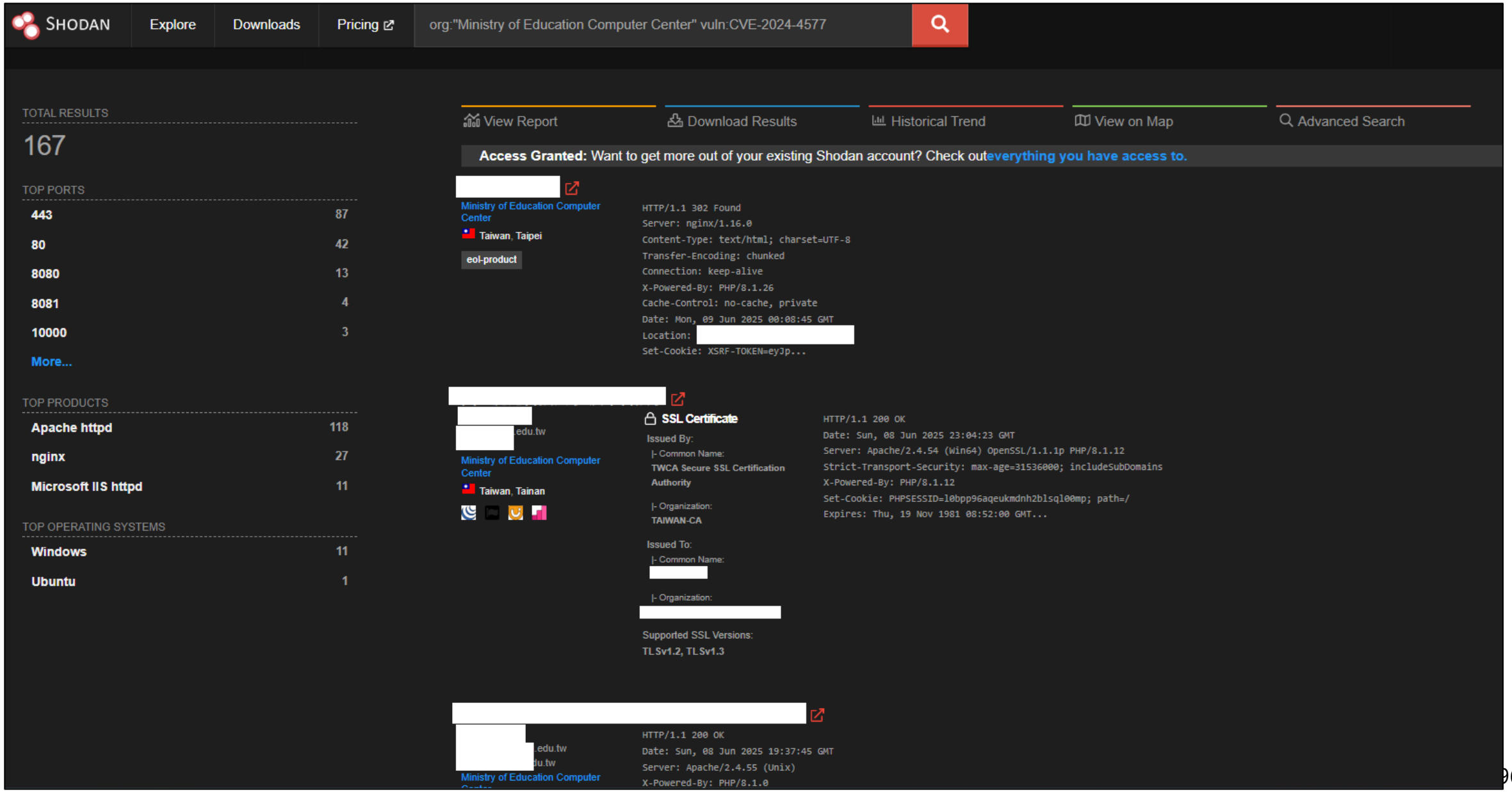
Operator	Purpose	Mixes with Other Operators?	Can be used Alone?	Web	Images	Groups	News
intitle	Search page Title	yes	yes	yes	yes	yes	yes
allintitle[4]	Search page title	no	yes	yes	yes	yes	yes
inurl	Search URL	yes	yes	yes	yes	not really	like intitle
allinurl	Search URL	no	yes	yes	yes	yes	like intitle
filetype	specific files	yes	no	yes	yes	no	not really
intext	Search text of page only	yes	yes	yes	yes	yes	yes
allintext	Search text of page only	not really	yes	yes	yes	yes	yes
site	Search specific site	yes	yes	yes	yes	no	not really
link	Search for links to pages	no	yes	yes	no	no	not really
inanchor	Search link anchor text	yes	yes	yes	yes	not really	yes
numrange	Locate number	yes	yes	yes	no	no	not really
daterange	Search in date range	yes	no	yes	not really	not really	not really
author	Group author search	yes	yes	no	no	yes	not really
group	Group name search	not really	yes	no	no	yes	not really
insubject	Group subject search	yes	yes	like intitle	like intitle	yes	like intitle
msgid	Group msgid search	no	yes	not really	not really	yes	not really

https://yu-jack.github.io/2017/10/17/google-hacking/
https://www.exploit-db.com/google-hacking-database

Wapplyzer


人力資源網2.0 | 教育部國民及
9hr.k12ea.gov.tw
19
WELCO
教育部國民及學
國民中小學教職員
選擇登入
已取得帳密之教育部及縣市承辦人或學校權限AA帳號已啟用之
續任學校端行政，請選擇您要的登入方式
Wappalyzer
技術
更多資訊
Export
分析
Google Analytics GA4
JavaScript 框架
Vue.js 2.7.14
Inertia.js
安全性
HSTS
字型
Glyphicons
Font Awesome
高級文字編輯器
CKEditor
程式語言
PHP
JavaScript 函式庫
Select2
jQuery Migrate 3.4.0
core-js 2.6.2
Axios
jQuery UI 1.13.2

Shodan



https://www.shodan.io/search/examples

漏洞測試

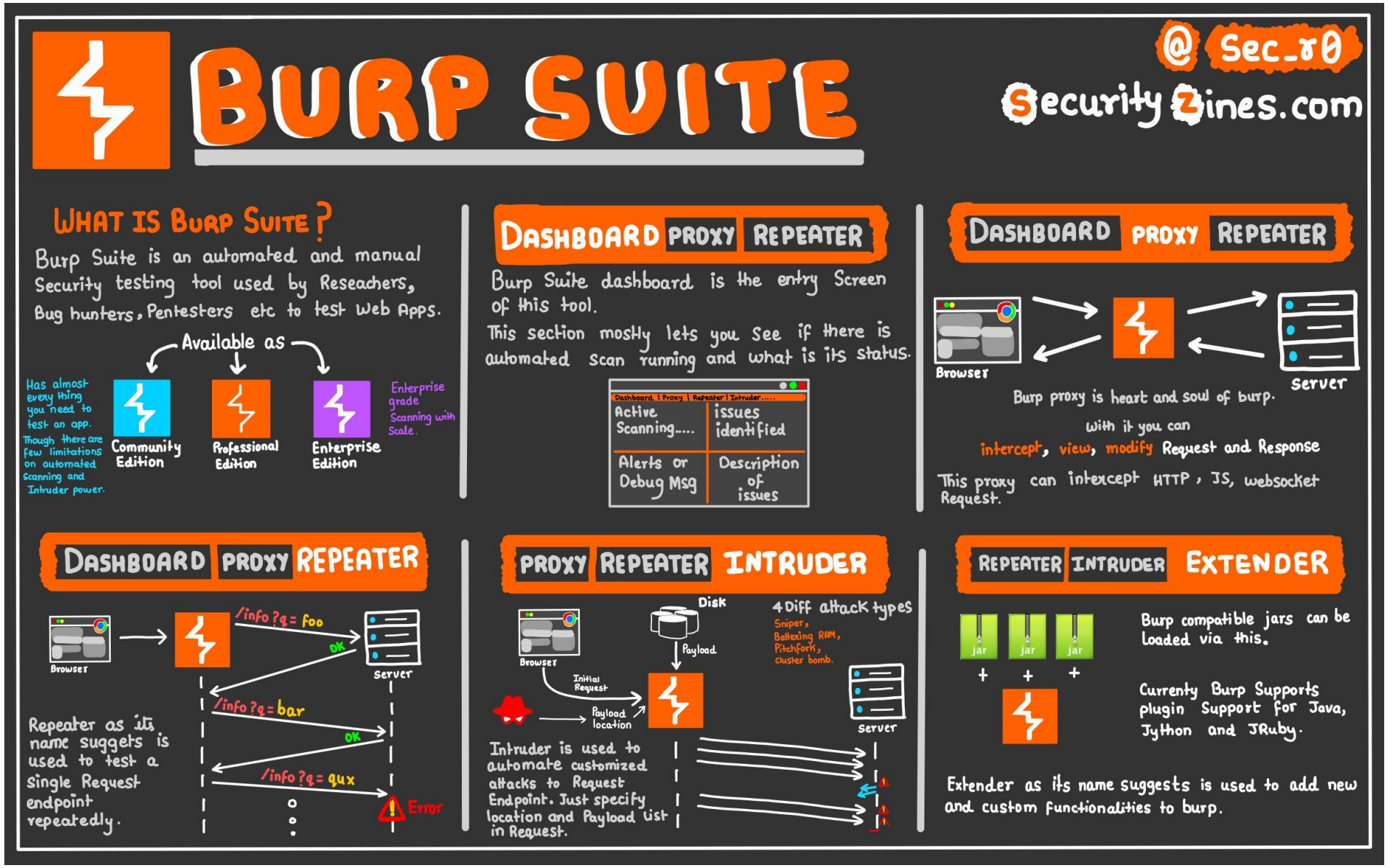
nmap

- Nmap 為開放原始碼的網路掃描與探測工具，提供掃描整個子網域或主機開放的連接埠、識別主機的存活、檢測作業系統版本與服務，同時也提供自動化腳本的功能(Nmap Scripting Engine)。

指令	用途
nmap 192.168.1.1	掃描單一IP(主機)
nmap -sP 192.168.1.0/24	以Ping方式掃描整個網段
nmap -sS -p 22,80 192.168.1.1	SYN方式掃描單一IP(主機)特定埠
nmap -A 192.168.1.1	進階掃描單一IP(主機)，開放的連接埠、對應服務與版本、主機作業系統等
nmap -p 21 -Pn --script=ftp-anon -iL ftp_ip.txt -oN scan_results.txt	FTP匿名登入腳本測試，指定21 Port，掃描範圍為指定檔案ftp_ip.txt，輸出scan_results.txt

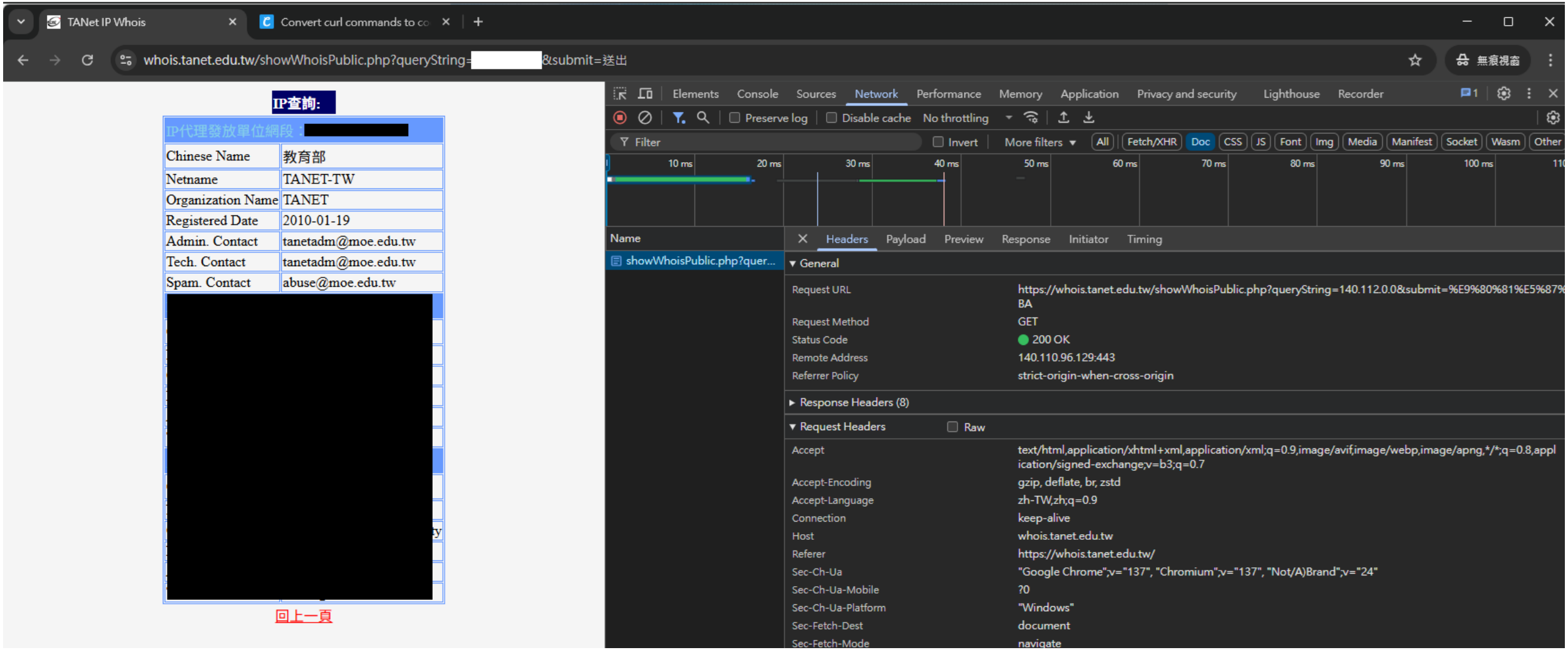
https://nmap.org/

Burp Suite



https://securityzines.com/pages/flyers/burp

瀏覽器+curl(Bash)+CurlConverter



https://curlconverter.com/

sqlmap

- 檢測到包含SQL Injection漏洞的注入點後，利用自動化工具進行攻擊。
- 指令參考：
  - sqlmap -u [url] --dbs
    列出資料庫
  - sqlmap -u [url] -D [database] --tables
    列出 tables(資料表)
  - sqlmap -u [url] -D [database] -T [tables] --columns
    列出 columns(欄位)
  - sqlmap -u [url] -D [database] -T [tables] -C [columns] --dump
    dump資料

結語

防禦對策

偵查目標 Reconnaissance ・駭客辨識及選擇攻擊目標

武裝工具 Weaponization ・駭客組織自動化攻擊工具

傳遞攻擊 Delivery ・透過載具進入攻擊目標環境

弱點攻擊 Exploitation ・弱點攻擊觸發入侵者程式

控制環境 Control ・遭安裝遠端存取之後門或木馬

攻擊執行 Execute ・在環境中布建內網、竊取資料

維持潛伏 Maintain ・潛伏於環境中清除軌跡

追蹤補強 Tracing ・駭客離場後，守方趁勢補強

前期▼預防

- 審視整體組織的資安管理政策
- 建立安全備份機制，確保攻擊事件發生後能有效還原
- 「資料」分類與權限控管、「身分」二階段驗證及透過AD追蹤行為分析、「端點」防毒和安全列管
- 在環境中部署進階威脅防禦工具，讓AI學習、優化識別
- 使用整合型威脅與漏洞管理工具，即時探索並評估各端點上與軟體漏洞，進行集中化動態調整

中期▼處理

- 從網路、系統等層面排查弱點，找出系統漏洞
- 資安工具即時告警並主動攔截攻擊，防止災難擴大
- 由資安顧問進行事件分析與處理(Incident Response)
- 攻擊的當下可以透過整合型威脅與漏洞管理工具，來追蹤發動的源頭與路徑，甚至可以進行即時的阻斷跟告警，讓公司不會亂了分寸造成災情擴大

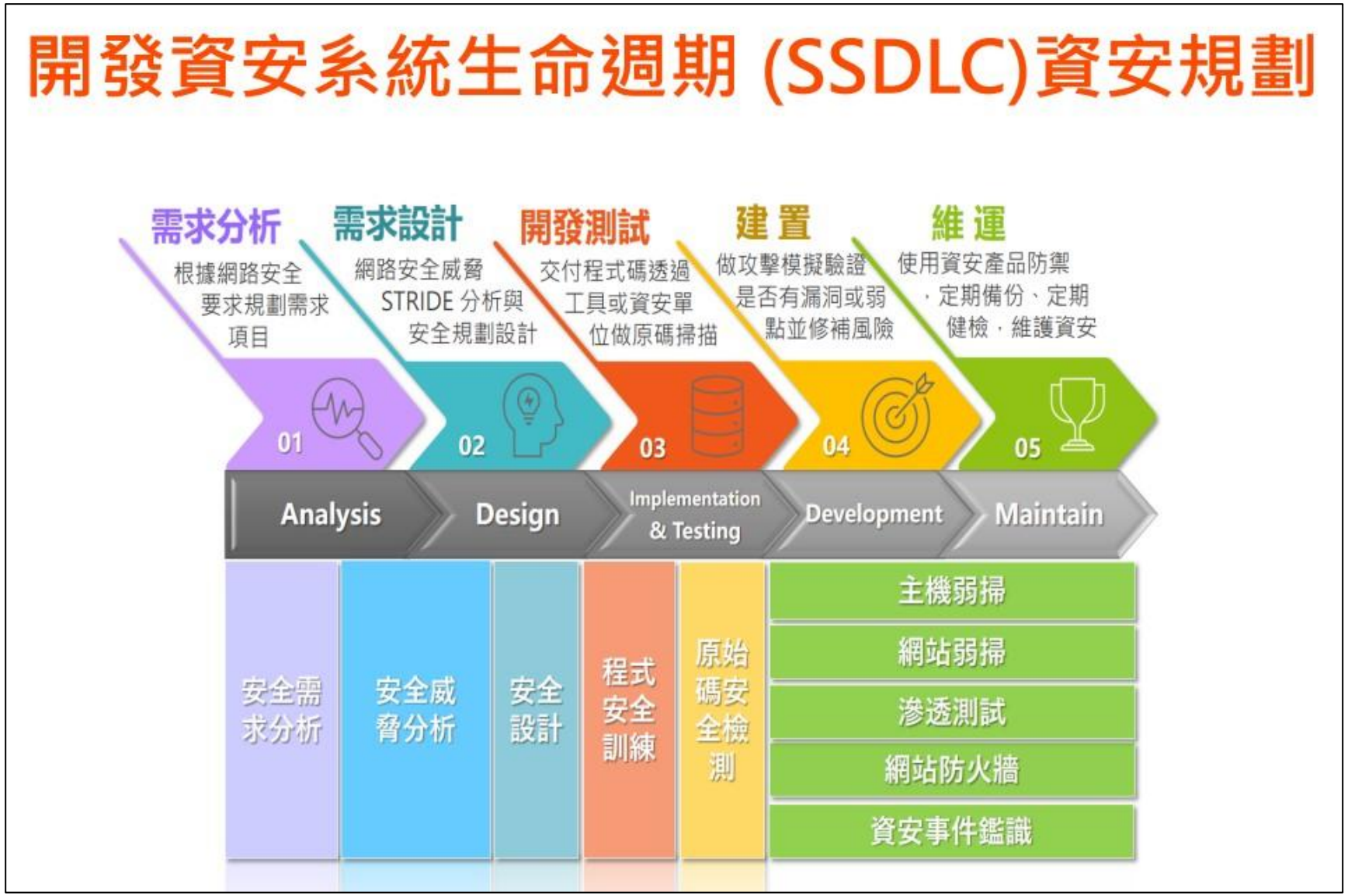
後期▼補強

- 透過整合型威脅與漏洞管理工具，彙整攻擊路徑紀錄，並結合SIEM所收集的Log進行比對分析，有效地針對弱點進行補強及處理，避免相同的弱點一再被針對

從長期管理策略不斷回饋優化

https://www.freedom.net.tw/ict-insight/security/cyber-kill-chain.html

防禦對策
開發資安系統生命週期(SSDLC)



https://ithelp.ithome.com.tw/questions/10214833

感謝聆聽!