果核數位
Digicentre

OWASP
Open Web Application
Security Project

以資安治理角度理解 OWASP Top 10 2021

目錄

01 OWASP Top 10 2021
年度項目

02 OWASP Top 10 2021
項目概略介紹、預防及修補

03 結論

01 OWASP Top 10 2021 年度項目

OWASP
TOP10

A01:2021 Broken Access Control
A02:2021 Cryptographic Failures
A03:2021 Injection
A04:2021 Insecure Design
A05:2021 Security Misconfiguration
A06:2021 Vulnerable and Outdated Components
A07:2021 Identification and Authentication Failures
A08:2021 Software and Data Integrity Failures
A09:2021 Security Logging and Monitoring Failures
A10:2021 Server-Side Request Forgery (SSRF)

OWASP TOP10 2017

- A01:2017 Injection
- A02:2017 Broken Authentication
- A03:2017 Sensitive Data Exposure
- A04:2017 XML External Entities (XXE)
- A05:2017 Broken Access Control
- A06:2017 Security Misconfiguration
- A07:2017 Cross-Site Scripting (XSS)
- A08:2017 Insecure Deserialization
- A09:2017 Using Components with Known Vulnerabilities
- A10:2017 Insufficient Logging & Monitoring

OWASP TOP10 2021

- A01:2021 Broken Access Control
- A02:2021 Cryptographic Failures
- A03:2021 Injection
- A04:2021 Insecure Design (NEW!)
- A05:2021 Security Misconfiguration
- A06:2021 Vulnerable and Outdated Components
- A07:2021 Identification and Authentication Failures
- A08:2021 Software and Data Integrity Failures (NEW!)
- A09:2021 Security Logging and Monitoring Failures *
- A10:2021 Server-Side Request Forgery (SSRF) * (NEW!)

* 調查結果而來

詳細解說請參考：https://www.youtube.com/watch?v=3h9HzTYxUX4

02 OWASP Top 10 2021 細項介紹

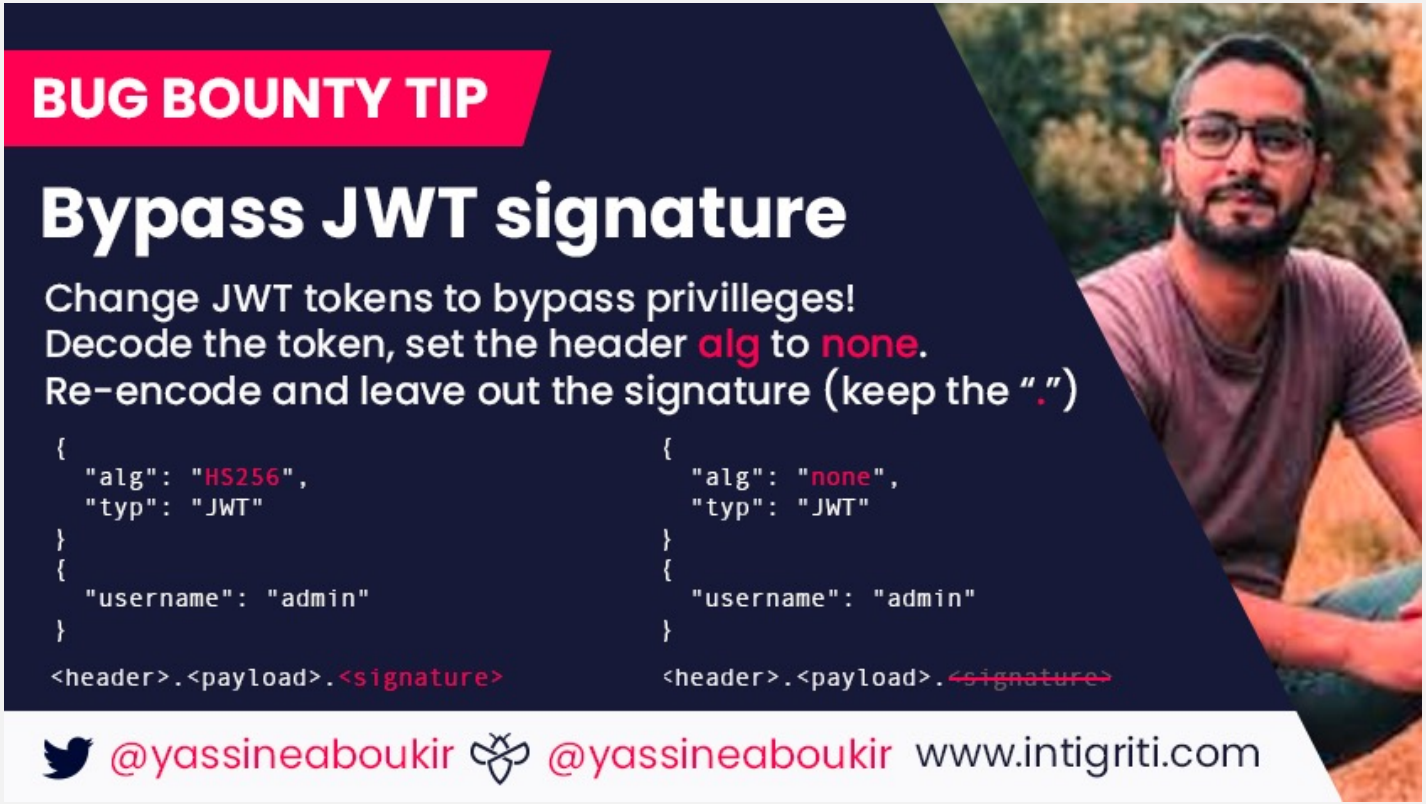
A01 Broken Access Control

A01:2021 Broken Access Control 權限控制失效

存取控制措施讓使用者不能執行在預期權限之外的行為
控制措施失效會導致未經授權的**資訊洩露、修改、毀損，或執行超出原權限的業務功能**。常見的存取控制弱點包括：

- 權限提升：水平提權、垂直提權
- IDOR 流水號問題
- 存取權限問題：如 Cookie、JWT Token
- CORS HTTP Header 問題
- 資訊洩漏問題

A01:2021 Broken Access Control 權限控制失效



A01:2021 Broken Access Control 如何修補及預防

存取控制僅在伺服器端有效

- 除公開的資源外，預設為拒絕存取
- 建立存取控制機制，並重複套用
- 停用 Web 目錄列表，並確保中繼資料（例如：.git) 和備份檔案不在根目錄中
- 記錄存取控制失效，並設定告警（例如：重覆登入失敗）
- 對 API 存取進行流量限制，降低自動攻擊帶來的損害
- 具身份識別之 Cookie，於登出、超時後被註銷
- JWT 使用最短存取時間，若長時間存取 JWT 建議遵循 OAuth 標準使其失效

開發工程師和 QA 測試工程師應納入與功能有關之存取控制的單元和整合測試

A02 Cryptographic Failures

A02:2021 Cryptographic Failures 加密機制失效

首先確定靜態資料及資料傳輸的防護需求
舉例來說，密碼、信用卡卡號、健康紀錄、個資、以及需要額外保護的營業祕密...等等主要被隱私法所保護的資料，以歐盟 GDPR 或 PCIDSS 等等金融業相關的資料保護法或標準。對於這些資料需考量：

- 傳輸加密與設定問題
- 憑證管理與驗證
- 金鑰管理問題
- 使用的加密演算法選擇、是否有 EOL 的演算法使用

A02:2021 Cryptographic Failures 如何修補及預防

至少執行以下措施，並參考相關資料：

- 以處理、儲存、傳輸行為模式分類，執行對應的控制措施
- 非必要不儲存敏感性資料，捨棄或使用 PCIDSS tokenization 或截斷(truncation)
- 將所有靜態的敏感性資料加密
- 針對包含敏感資料的回應停用快取
- 使用最新版且標準的強演算法、協定及金鑰並使用適當的金鑰管理
- 使用安全的加密傳輸協定如 TLS 1.2, TLS 1.3
- 使用具迭代方法之雜湊演算法如 Argon2, scrypt, bcrypt 或 PBKDF2 搭配加鹽來儲存密碼

A03 Injection

A03:2021 Injection 注入式攻擊

應用程式的直譯器在以下情況容易遭受攻擊：

- 程式未驗證、過濾、轉化或未參數化使用者提供的資料
- 在物件關係對映 (ORM) 的搜尋參數中，使用惡意語法來擷取額外或敏感資料
- 在動態查詢、指令或儲存行為中，直接使用或連結了惡意資料

將字串與使用者輸入做串接後，將之執行。

A03:2021 Injection 如何修補及預防

需要將命令與查詢資料分開

- 使用參數化或整合到物件關係對映 (ORMs) 中
- 使用正規表示式或白名單
- 動態查詢因應不同資料庫及業務需求，跳脫特殊符號進行過濾
- 使用如 LIMIT 等控制輸出量的語法

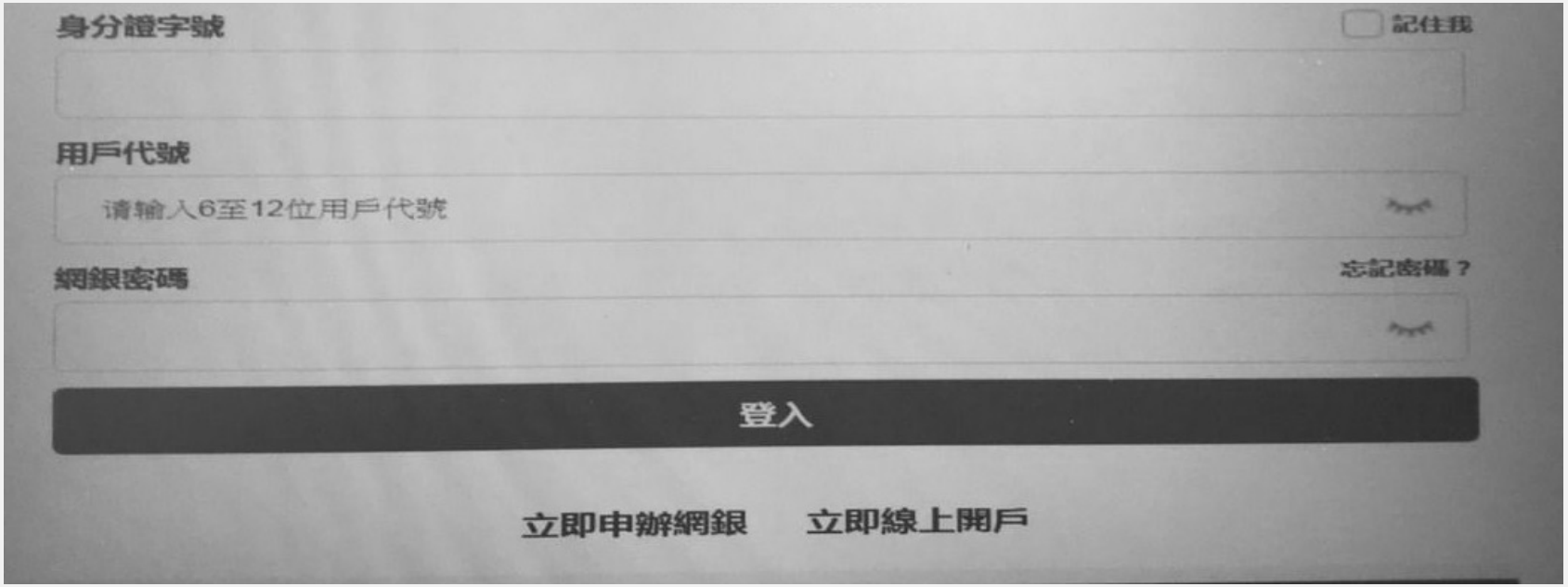
A04 Insecure Design

A04:2021 Insecure Design 不安全設計

不安全設計是指沒有控制措施或程式執行的流程有安全問題：
如：應該加密敏感資料但是沒有對應的實作方法。
如：實作 OTP 流程，其 OTP 並非隨機數且不具時效性。

安全設計用來評估威脅並預防可能發生的攻擊，達成這樣的結果需要安全的開發生命週期。

A04:2021 Insecure Design 不安全設計



A04:2021 Insecure Design 如何修補及預防

- 建立與使用軟體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
- 使用安全設計模式的函式庫或是已完成可使用的元件 (請參照 A06:2021-危險或過舊的元件)
- 使用威脅建模在認證、存取控制、商業邏輯上
- 撰寫單元測試與整合測試來驗證所有的重要流程對威脅建模都有抗衡

A05 Security Misconfiguration

A05:2021 Security Misconfiguration 安全設定缺陷

- 不必要的功能啟用或是安裝 (例：埠號、服務、頁面、帳號、或是特權)
- 使用預設帳號與密碼
- 錯誤處理暴露出的堆疊追蹤
- 軟體或使用元件過舊 (請參照 A06:2021-危險或過舊的元件)
- 升級後未啟用安全設定
- 未設定安全參數，程式框架、各種函式庫、資料庫等
- 未傳送安全標頭

當沒有一致且可重複的安全設定流程時，系統將會面對高風險。

A05:2021 Security Misconfiguration安全設定缺陷

github.com/mpgn/Spring-Boot-Actuator-Exploit

Spring Boot Actuator (jolokia) XXE/RCE

Information and payload from the following article: https://www.veracode.com/blog/res
actuators

Edit 28/02/2020: another article to achieve RCE using H2 Database Remote Code Execu
Exposed Actuators and H2 Database Aliases in Spring Boot 2

Tested on Spring Boot Actuator < 2.0.0 and Jolokia 1.6.0.

If you have access to the following ressource `/actuator/jolokia` or `/jolokia` with Spr
following ressource: **reloadByURL**, this writeup can help you to exploit an XXE and ultim

Setup the environment:

```
git clone https://github.com/artsploit/actuator-testbed
cd actuator-testbed
mvn install
```

A05:2021 Security Misconfiguration 如何修補及預防

安全的部署流程，並實際實行，包含以下:

- 自動化的流程，可以部署及確認環境中各類的安全設定
- 開發及營運環境，都須有一致相同的設定，並且使用不同的認證資訊
- 伺服器的功能最小化、區域化
- 依據相關更新，來執行安全審視及更動 (請參照 A06:2021-危險或過舊的元件)
- 使用安全表頭

A06 Vulnerable and Outdated Components

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

以下至少包含所使用之作業系統、軟體、套件、函式庫、框架等：

- 使用官方已淘汰的版本 (或不清楚使用的版本)
- 沒有定期執行弱點掃描
- 未即時或安排風險修補，並測試修補後的相容性
- 未啟用安全設定 (請參照 A05:2021 - 安全設定缺陷)

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

A06:2021 Vulnerable and Outdated Components 如何修補及預防

應設置修補程式管理流程：

- 刪除未使用的相依套件、不必要的功能、組件、檔案及文件。
- 定期盤點所使用之套件及相依套件的版本，如版控工具、OWASP Dependency Check、retire.js 等。
- 使用軟體組合分析工具來自動化該流程、訂閱使用套件的資安通報。
- 僅透過官方提供的安全連結更新（請參照 A08:2021-軟體及資料完整性失效）。
- 監控未維護的程式庫和套件。

每個組織都必須於應用程式或開發專案(portfolio)的生命週期內制訂持續監控、鑒別分類(triaging)、更新或更改配置的計劃。

A07 Identification and Authentication Failures

A07:2021 Identification and Authentication Failures 認證與驗證機制失效

使用者的身分、認證、會話(session)等管理對於防止認證相關的攻擊至關重要，如果應用程式存在以下情況，則可能有認證的漏洞：

- 暴力或撞庫攻擊
- 可被繞過的忘記密碼流程 (請參照 A4: 2021 不安全設計)
- 密碼儲存使用明碼或較脆弱雜湊法 (請參照 A2: 2021- 加密機制失效)
- 不具有或是無效的多因子認證
- 於URL中洩漏會話(session) ID
- 登入後沒有重新派送會話(session) ID
- 會話(session) ID 沒有正確註銷或沒有時效性

A07:2021 Identification and Authentication Failures 如何修補及預防

- 實作多因子認證
- 不使用任何預設的授權資料
- 實作弱密碼的檢查及輪換政策 (NIST 800-63b第5.1.1節- Memorized Secrets)
- 使用相同的 response 避免使用者帳號被列舉
- 限制登入失敗次數或延遲
- 會話(session)ID不應出現在URL中，且在登出後、閒置、超時後被註銷

A08 Software and Data Integrity Failures

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

- 程式碼或基礎架構未能保護軟體及資料之完整性
- 不安全的反序列化
- 使用不受信任來源之套件、函式庫、模組等
- 不安全的持續性整合/部署(CI/CD)流程
- 使用缺乏充足完整性驗證的自動更新

反序列化攻擊及第三方套件信任問題是此項目中之重點

A08:2021 Software and Data Integrity Failures 如何修補及預防

- 針對資料傳輸實施完整性檢查或數位簽章來防止竄改、重送、反序列化攻擊
- 從受信任的來源取得函式庫及相依套件
- 使用軟體供應鏈安全工具確保元件沒有已知弱點 (請參照 A06:2021危險或過舊的元件)
- 適當地設定持續性整合/部署(CI/CD)流程的組態及存取控制確保程式碼在組建及部署流程中的完整性

A09 Security Logging and Monitoring Failures

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

缺乏記錄及監控時無法偵測資安事件發生：

- LOG紀錄不足、甚至沒紀錄、設計不良
- 未集中收容
- 集中收容未客製化設計規則
- 資安測試時未觸發告警

允許使用者或攻擊者讀取日誌或告警事件可能早成資訊洩漏 (請參照 A01:2021權限控制失效)

A09:2021 Security Logging and Monitoring Failures 如何修補及預防

開發者應依據應用程式所面臨風險實作下列部分或全部的控制項：

- 確保日誌完整紀錄，並留存充足時間
- 日誌格式符合一般日誌管理系統常用格式
- 應正確編碼以防止系統遭受攻擊 (請參照 A03:2021 – 注入攻擊)。
- 實作完整性控制以避免系統資料竄改或刪除
- 有效地監控及告警機制
- 建立或導入事件應變及復原計畫，如 NIST 800-61r2 或更新版本

現有多種商業化及開放原始碼應用程式保護架構，如 OWASP ModSecurity Core Rule Set 及開放原始碼日誌關聯軟體可客製化儀表板及告警，如 ELK stack。

A10 Server-Side Request Forgery (SSRF)

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

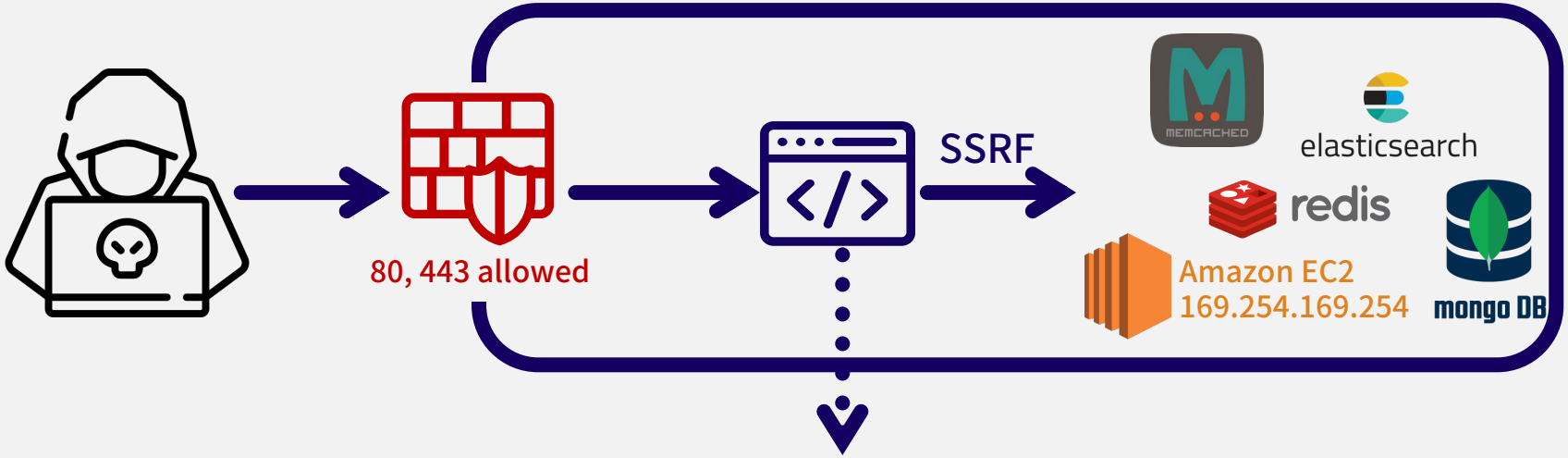
當網站伺服器取得遠端資源，卻未驗證由使用者提供的網址，此時就會發生偽造伺服端請求。

即便有防火牆、VPN 或其他網路 ACL 保護的情況下，仍得以達成攻擊。

雲端服務和其結構的複雜性，偽造伺服端請求的嚴重性將會愈來愈嚴峻。

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



https://www.digicentre.com.tw

A10:2021 Server-Side Request Forgery (SSRF) 如何修補及預防

從網路層著手：

- 切割成多個子網路以降低偽造伺服端請求之衝擊
- 於防火牆政策或於網路存取控制規則實施 "預設全拒絕 (deny by default)" ，以封鎖全部來自外部之網路流量

A10:2021 Server-Side Request Forgery (SSRF) 如何修補及預防

從應用層

- 過濾使用者端輸入之內容
- 使用白名單
- 不傳送原始回應給使用者端
- 停用 HTTP 重新導向

別透過拒絕清單或正規表示式來減緩偽造伺服端請求

03 結論

結論

- 客戶的錯誤期許：弱掃可以解決全部 OWASP Top10 問題
- 企業應盡量完整企業內部 SSDF 流程
  - 分階段導入code review, 弱點掃描, 滲透測試
  - DevSecOps：若有CI / CD 流程可導入自動化原碼檢測、第三方套件檢查
  - 行有餘力可以導入最前期的 Security Architecture 資安架構設計與相關威脅建模流程
- SOC 資安監控範圍可考慮延伸至 AP 應用程式層面
- 提供開發工程師更多安全程式碼相關的教育訓練
- 以 OWASP SAMM 評估企業內部軟體成熟度