資安攻防技術與實作

2025年4月15日

◦ 國立中央大學
許時準組長

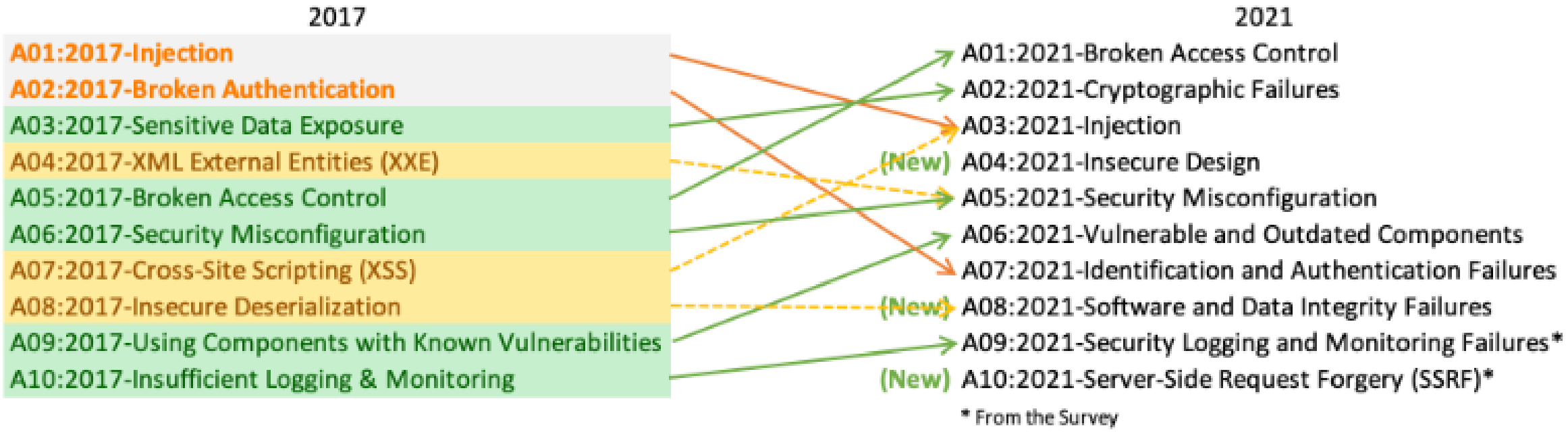
課程目標

- 了解常見的Web安全漏洞
- 實際演練資安攻防

大綱

- OWASP TOP 10
- 網站靶機安裝
- 資安攻防實作
- 結論

第一部分：
OWASP TOP 10



https://owasp.org/Top10/

1:

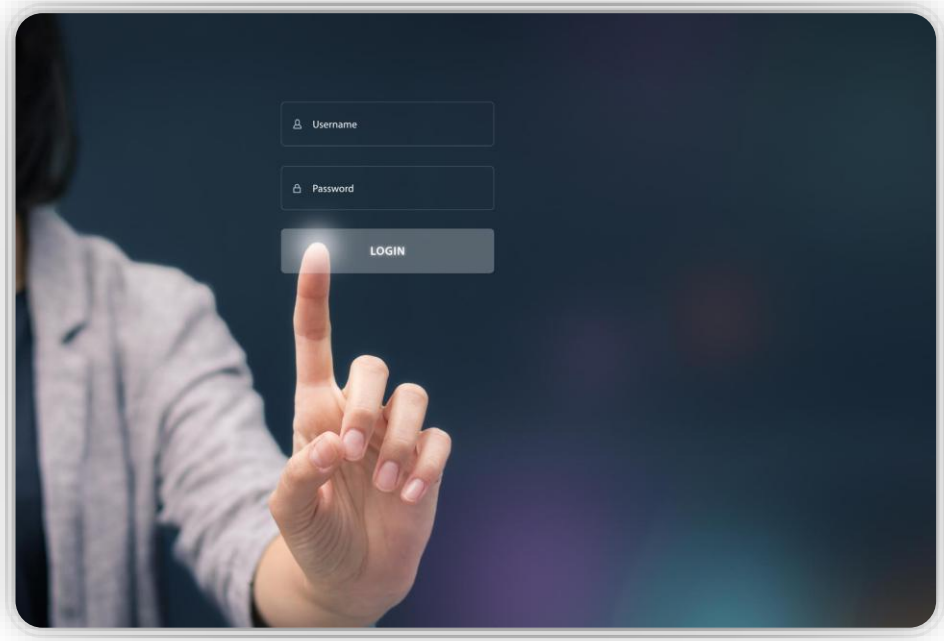
A01:2021 – 權限控制失效

修改URL、內部應用程式狀態或HTML頁面來繞過存取控制檢查。

容許主鍵被更改為其他用戶的記錄，允許查看或編輯其他人的帳戶。

特權提升。未登入即成為用戶，或以用戶身份登入即成為管理員。

如何預防權限控制失效



1. 存取控制僅在受信任的伺服器端檢查。
2. 除公開的資源外，預設為拒絕存取。
3. 停用Web伺服器目錄列表，並確保檔案中繼資料不在web根目錄中。
4. 記錄存取控制失效，並警示管理員。
5. 對API和控制器存取進行流量限制。

2:

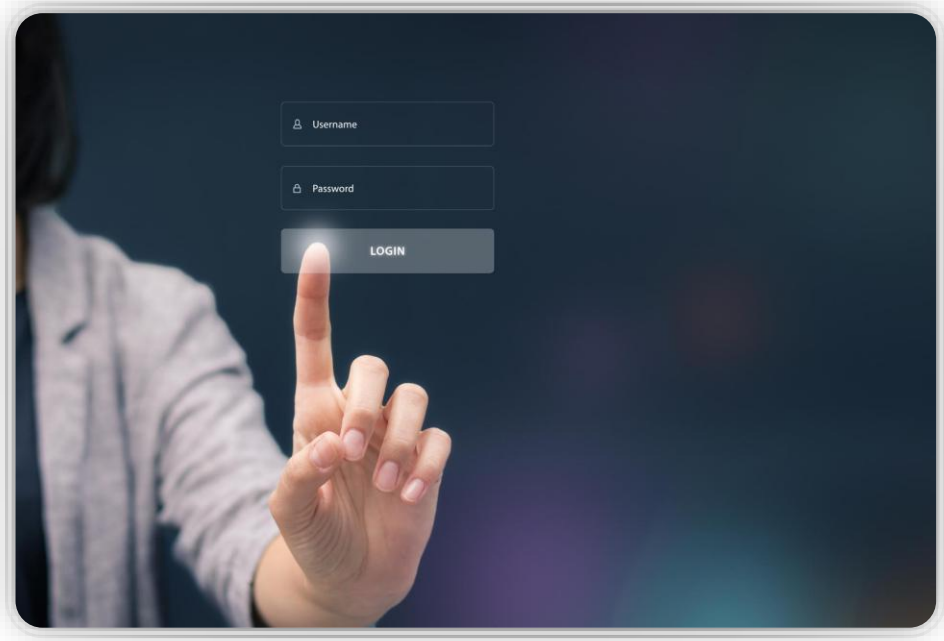
A02:2021 – 加密機制失效

資料是否以明碼傳輸? 像是HTTP, SMTP, FTP等協定。

是否有任何老舊或脆弱的加密演算法被預設使用或存在於較舊的程式碼?

是否有任何預設的加密金鑰被使用、脆弱的加密金鑰被重複使用?

如何預防加密機制失效



1. 對應用程式處理、儲存、傳輸的資料進行分類，辨識哪些為敏感性資料，依照分類執行對應的控制措施。

2. 確保將所有靜態的敏感性資料加密。

3. 確認使用最新版且標準的強演算法、協定及金鑰。

4. 使用安全的協定加密傳輸中的資料。

3:

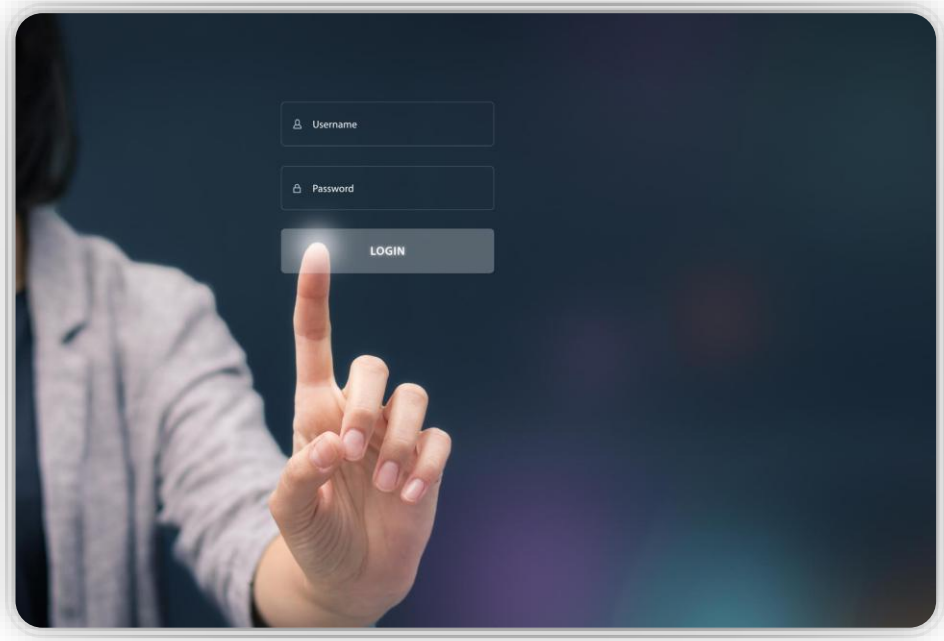
A03:2021 – 注入式攻擊

應用程式未驗證、過濾或清理使用者提供的資料。

在動態查詢、命令或儲存的程序，SQL、指令或儲存的程序中，直接使用或連結惡意資料。

https://owasp.org/Top10/

如何預防注入式攻擊



1. 需要將命令與查詢資料分開，以防止注入式攻擊。

2. 使用正面或白名單在伺服器端驗證輸入的資料。

3. 對於任何剩餘的動態查詢，在轉譯中使用特殊符號進行查詢將對查詢語法帶來不同的涵義。

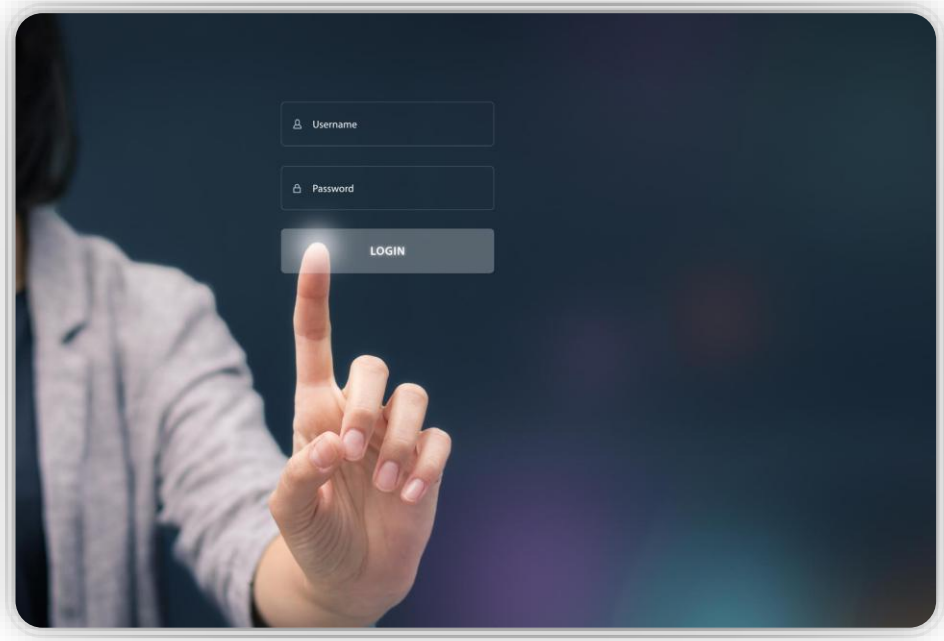
4:

A04:2021 – 不安全設計

- 不安全設計是一個廣泛的類別呈現許多不同的弱點，代表為"缺乏或無效的控制設計"。 缺乏不安全設計是指没有控制措施。

https://owasp.org/Top10/

如何預防不安全設計



1. 建立與使用安全開發生命週期，協同專業人士來評估與設計安全與隱私相關的控制措施。
2. 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。
3. 撰寫單元測試與整合測試來驗證所有的關鍵流程。

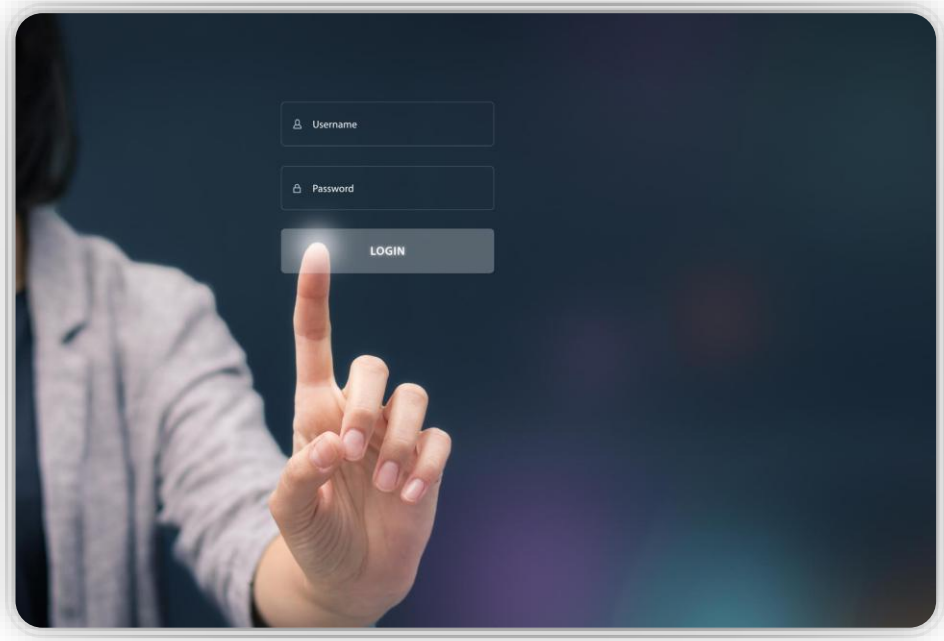
https://owasp.org/Top10/

5:

A05:2021 – 安全設定缺陷

- 只啟用必要的功能或是安裝（例如，不必要的port，服務，頁面，帳號，或是特權）。
- 預設帳號與密碼還可使用，並且未更改。
- 因錯誤處理而暴露出的堆疊追蹤，或是向使用者，暴露出過多的錯誤警告資訊
- 因為系統升級，導致最新的安全功能被關閉，或是造成不安全的設定

如何預防安全設定缺陷



1. 確保每個用戶、進程或服務僅獲得其所需的最小權限，以降低攻擊風險。
2. 僅啟用需要的服務和功能，關閉不必要的服務。
3. 使用強大的密碼政策，要求用戶使用長、複雜、並且定期更換的密碼。
4. 實施多因素身份驗證。

https://sites.google.com/gs.ncku.edu.tw/vulnerability-patching-guide/%E5%BC%B1%E9%BB%9E%E4%BF%AE%E8%A3%9C/owasp-top10-web-patch

6:

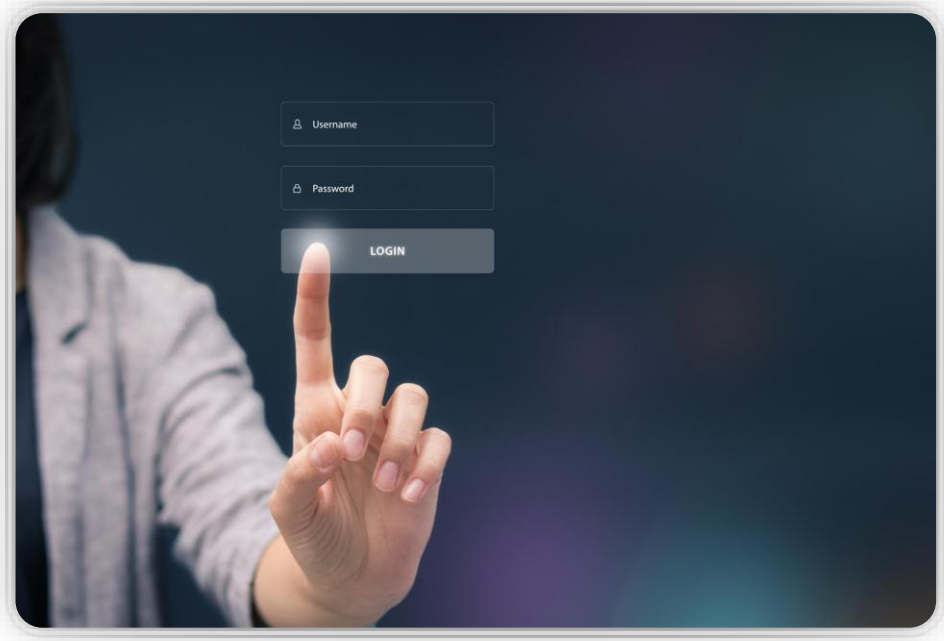
A06:2021 – 易受攻擊和已淘汰的組件

過時的組件的版本（用戶端和伺服器端）。這包括直接使用的組件以及嵌入的相依套件。

已不支援或已淘汰的作業系統、網頁/應用程式伺服器、資料庫、應用程式、 API 以及所有組件、執行環境和程式庫。

軟體開發人員未測試更新、升級或修補後程式庫的相容性。

如何預防易受攻擊和已淘汰的組件



1. 刪除未使用的相依套件、不必要的功能、組件、檔案及文件。

2. 持續使用版控工具來盤點客戶端和伺服器端組件（例如框架、程式庫）及相依組件的版本。

3. 僅透過官方提供的安全連結來取得組件。 優先選擇已簽署的更新包，以降低更新包被加入惡意組件的可能。

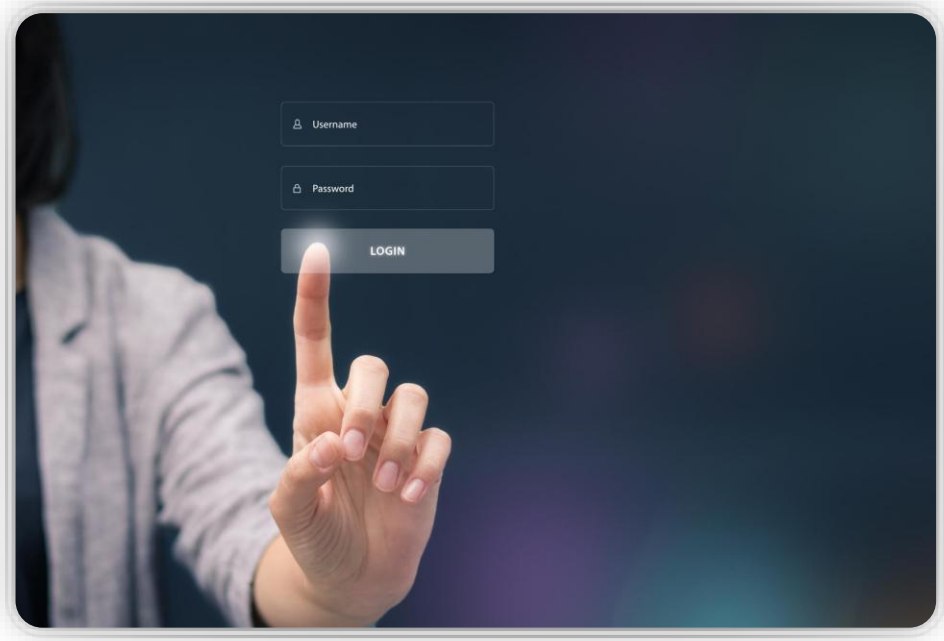
https://owasp.org/Top10/

7:

A07:2021 – 認證及驗證機制失效

- 允許像是攻擊者已經擁有有效用戶名稱和密碼列表的撞庫自動化攻擊。
- 允許預設、脆弱、常見的密碼，像是"Password"或"admin/admin"。
- 使用脆弱或無效的認證資訊回復或忘記密碼的流程。
- 將密碼使用明碼、加密或較脆弱雜湊法的方式儲存。

如何預防認證及驗證機制失效



1. 實作多因子認證來防止自動化撞庫攻擊、暴力破解、以及遭竊認證資訊被重複利用的攻擊。

2. 不要交付或部署任何預設的認證資訊，特別是管理者。

3. 實作脆弱密碼的檢查。

4. 限制或增加失敗登入嘗試的延遲。

https://owasp.org/Top10/

8:

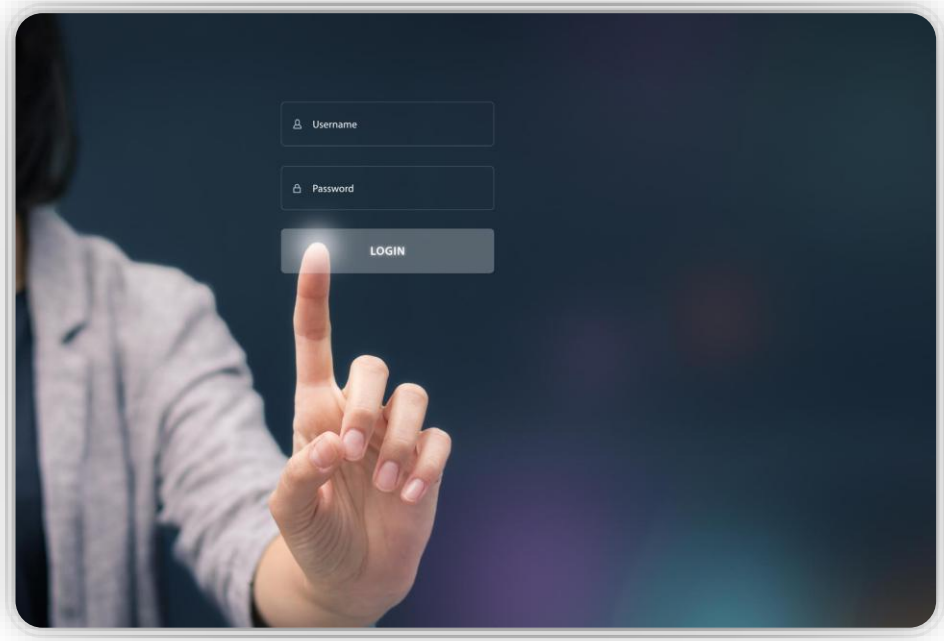
A08:2021 – 軟體及資料完整性失效

程式碼或基礎架構未能保護軟體及資料之完整性受到破壞。

應用程式依賴來自於不受信任來源，典藏庫及內容遞送網路之外掛，函式庫或模組。

自動更新功能在缺乏充足完整性驗證功能時就下載並安裝更新到處於安全狀態下的應用程式。

如何預防軟體及資料完整性失效



1. 利用完整性檢查或數位簽章來偵測竄改或重放攻擊。

2. 利用數位簽章或類似機制確保軟體或資料來自預期之提供者

3. 確保函式庫及從屬套件，例如npm或Maven，是從受信任的典藏庫取得。

4. 使用軟體供應鏈安全工具(例如OWASP Dependency Check 或 OWASP CycloneDX)確保元件沒有已知弱點。

https://owasp.org/Top10/

9:

A09:2021 – 資安記錄及監控失效

可稽核事件未記錄，如登入成功，登入失敗及高價值交易。

警告或錯誤發生時未產生，產生不充足或產生不明確日誌。

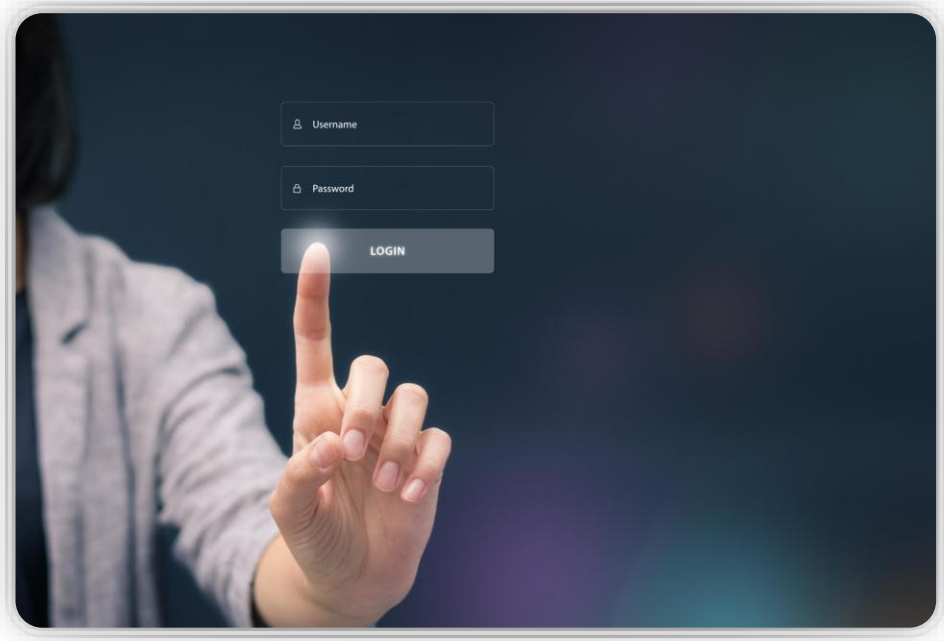
未監控應用程式或應用程式介面(API)日誌中的可疑活動。

日誌僅儲存於本地端。

滲透測試及DAST工具(如OWASP ZAP)掃描沒有觸發告警。

https://owasp.org/Top10/

如何預防資安記錄及監控失效



1. 確保記錄所有登入，存取控制及伺服器端輸入驗證之失敗，日誌應存留充足時間以利未來可能之鑑識分析要求。

2. 確保日誌格式符合一般日誌管理系統常用格式。

3. 確保日誌經正確編碼以防止遭受注入攻擊或日誌/監控系統遭受攻擊。

4. 建立或導入事件應變及復原計畫。

https://owasp.org/Top10/

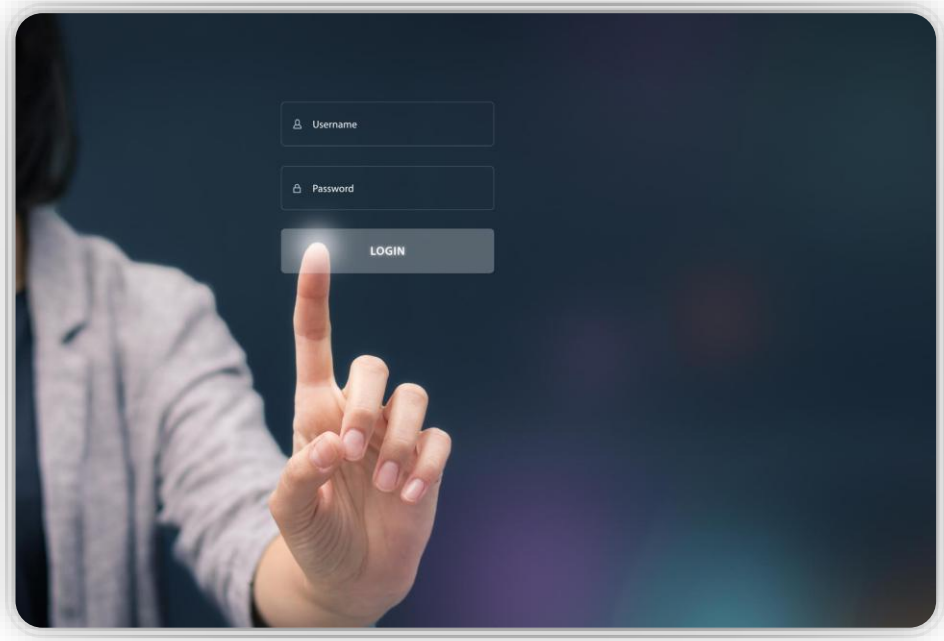
10:

A10:2021 – 伺服端請求偽造

1. 當網頁應用程式正在取得遠端資源，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。

2. 即便有防火牆、VPN或其他網路ACL保護的情況下，攻擊者仍得以強迫網頁應用程式發送一個經過捏造的請求給一個非預期的目的端。

如何預防伺服端請求偽造



1. 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

2. 過濾並驗證來自於用戶端提供之全部輸入

3. 以正面表列方式列出URL、port、目的地清單

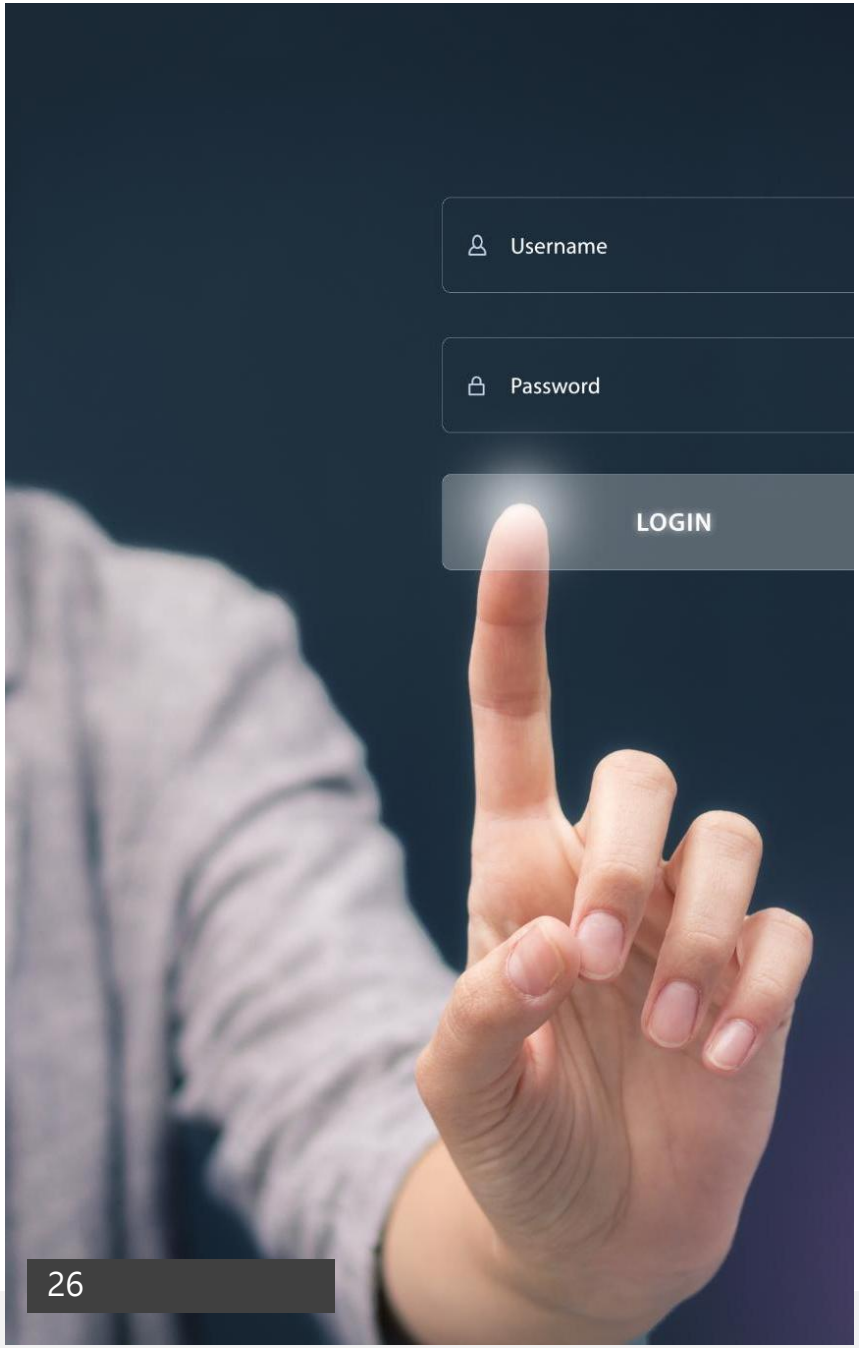
4. 停用HTTP重新導向

https://owasp.org/Top10/

第二部分：

網站靶機安裝

- 環境設置
- 安裝與啟動DVWA
- 確認環境運行正常

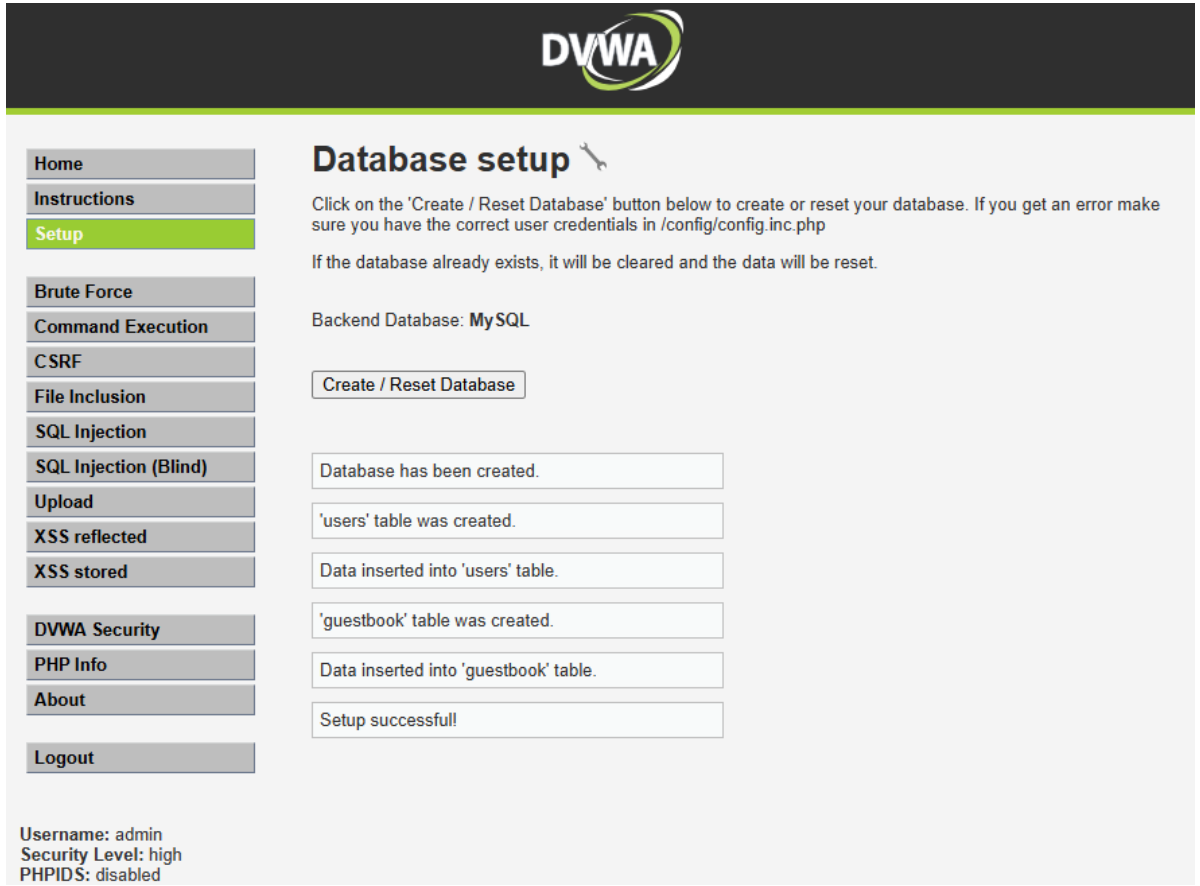


DVWA安裝

- Damn Vulnerable Web Application(DVWA) 可以讓使用者學習找出網站弱點的靶機系統。
- DVWA可以調整難易度，分為low、medium、high，難度越高能破解的難度也就越高。

重設資料庫

- 選擇”Setup” ➔ “Create / Reset Database”重設資料庫



第三部分：

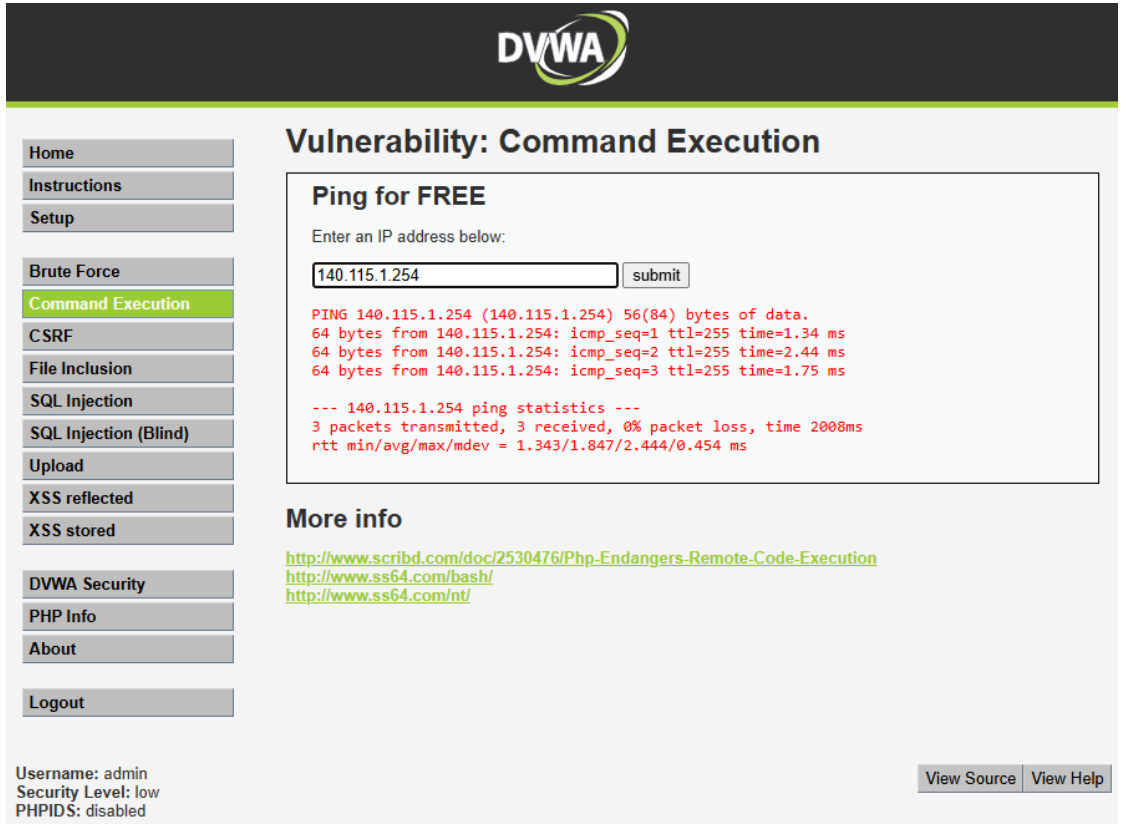
資安攻防實作

- 模擬攻擊
  - 嘗試通過各種方式找尋可能包含敏感資料的URL
  - http://127.0.0.1:12345/

LAB 1:

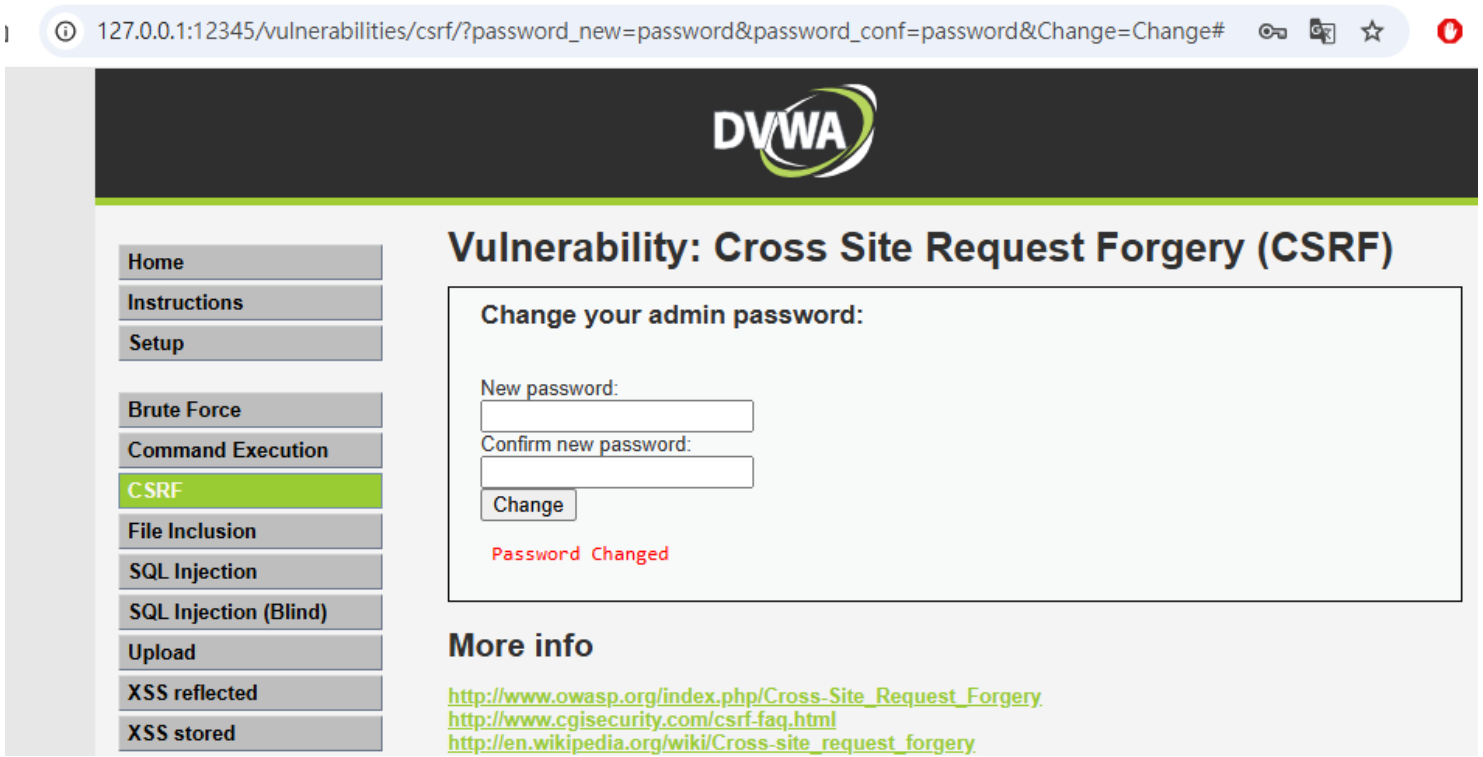
Command Execution

- 題目: 請嘗試取得作業系統使用者帳號密碼檔?
- Hint: 正常輸入IP : 140.115.1.254，
  系統執行指令: ping 140.115.1.254



LAB 2:
Cross Site Request Forgery (CSRF)
跨站請求偽造

- 題目: 請試著讓使用者在不知情下被修改密碼?
- Hint : 在當前已登入的Web應用程式上執行非本意的操作的攻擊方法
- 觀察修改密碼的URL
  http://127.0.0.1:12345/vulnerabilities/csrf/?password_new=password&password_conf=password&Change=Change#



LAB 3:

File Inclusion

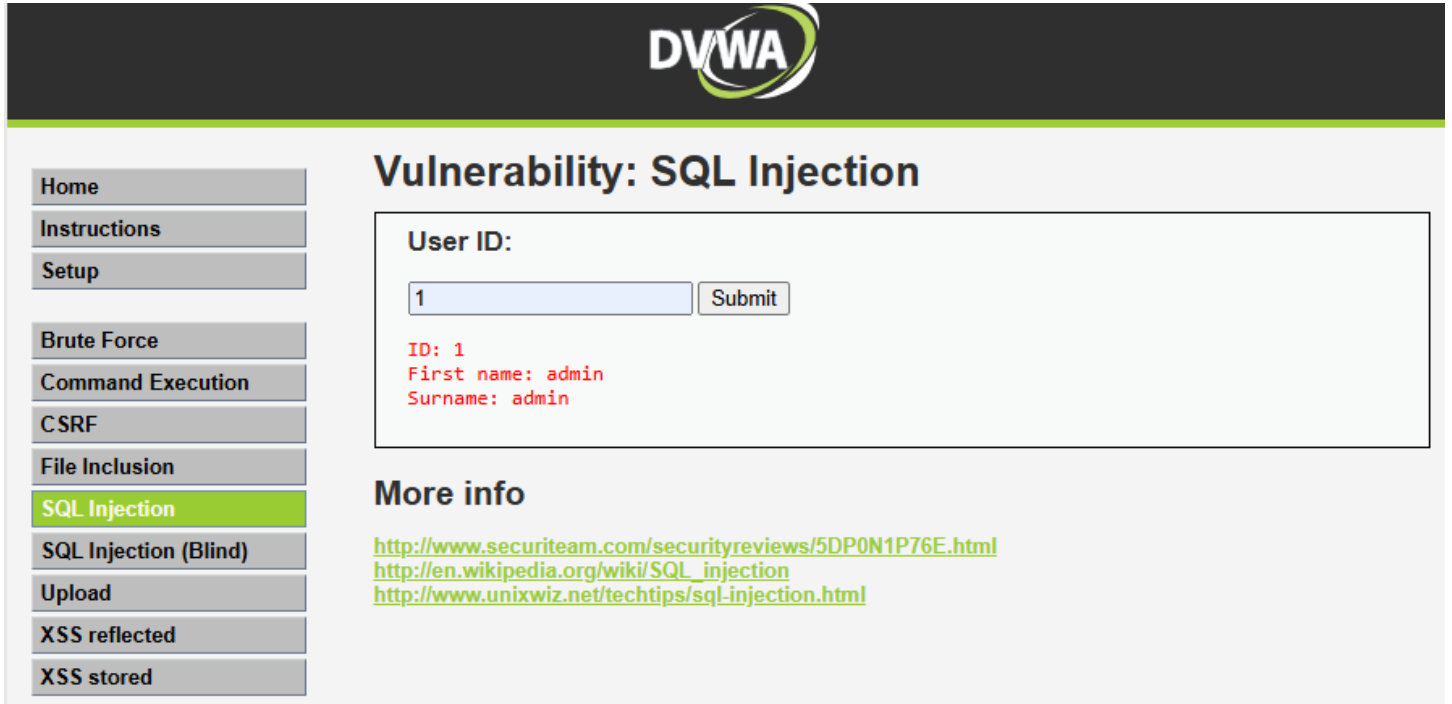
- 題目: 後端程式語言使用 include 引入其他檔案的時候，沒有去驗證輸入的值，讓駭客繞過驗證，請試著找到使用者帳號密碼?
- Hint: 觀察網址列或網頁原始碼
  http://127.0.0.1:12345/vulnerabilities/fi/?page=include.php

LAB 4:

SQL Injection

- 題目: 輸入甚麼可以繞過登入驗證? 顯示所有使用者名稱?

- Hint: 在輸入框中輸入單引號 (') ，送出表單以確認系統是否有檢查SQL 語法?
- 假設網站的登入驗證的SQL
  strSQL = "SELECT * FROM users WHERE (id = '" + id + "') ;"



LAB 5:

SQL Injection

- 題目: 如何找到資料庫中所有使用者帳號密碼?
- Hint:

✓ 找到資料庫名稱

#

✓ 找到資料表名稱

#

LAB 6:

XSS (Reflected)

- 題目:跨網站指令碼（Cross-site scripting，XSS）攻擊漏洞，允許駭客輸入惡意程式碼
- https://www.ithome.com.tw/news/139205

LAB 7:

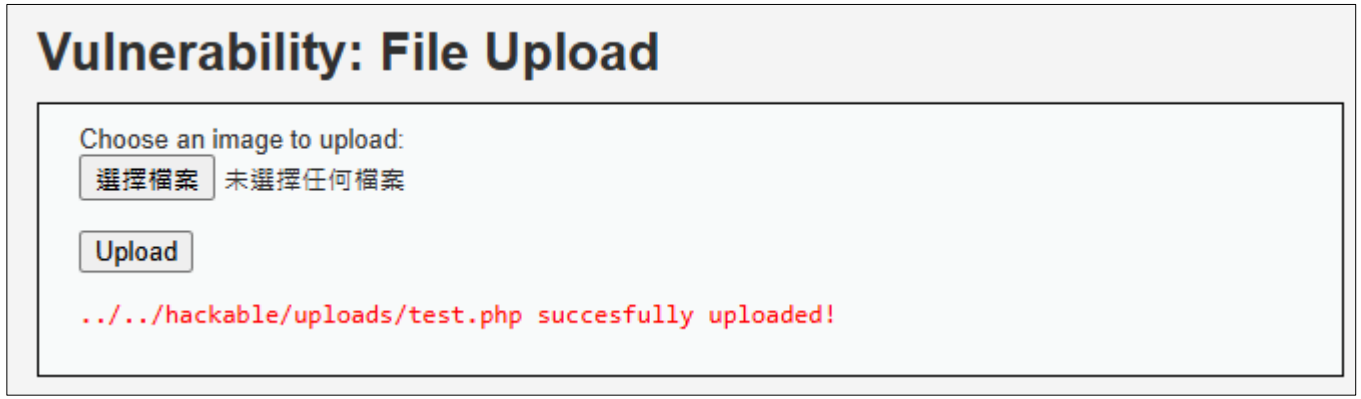
XSS (Stored)

- 題目:與前面手法不同的是不需要使用社交工程釣魚的技巧，也能使 User 受到攻擊，攻擊的方式是將 Javascript 惡意程式碼儲存在伺服器的資料庫中(例如:論壇討論區)，進而讓使用者遭受攻擊。
-

- 題目: 如何利用檔案上傳，破解資料庫中使用者帳號密碼?
- Hint: 用Notepad 儲存 test.php，然後上傳

LAB 8:

File Upload



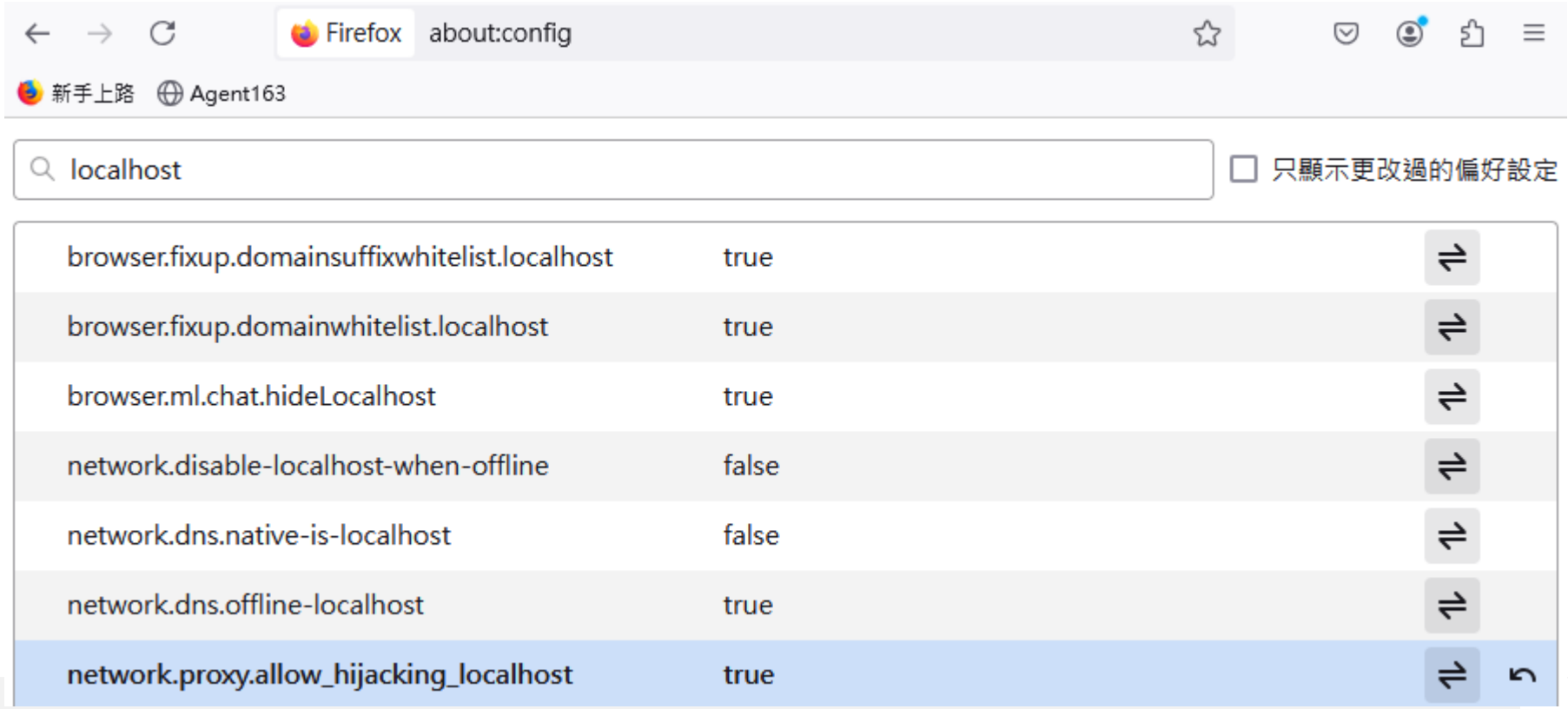
LAB 9:
Brute Force

- 題目: 請試著使用工具，暴力輸入帳號密碼找到使用者密碼?

解題步驟一:
利用Burp Suite執行網站攻擊

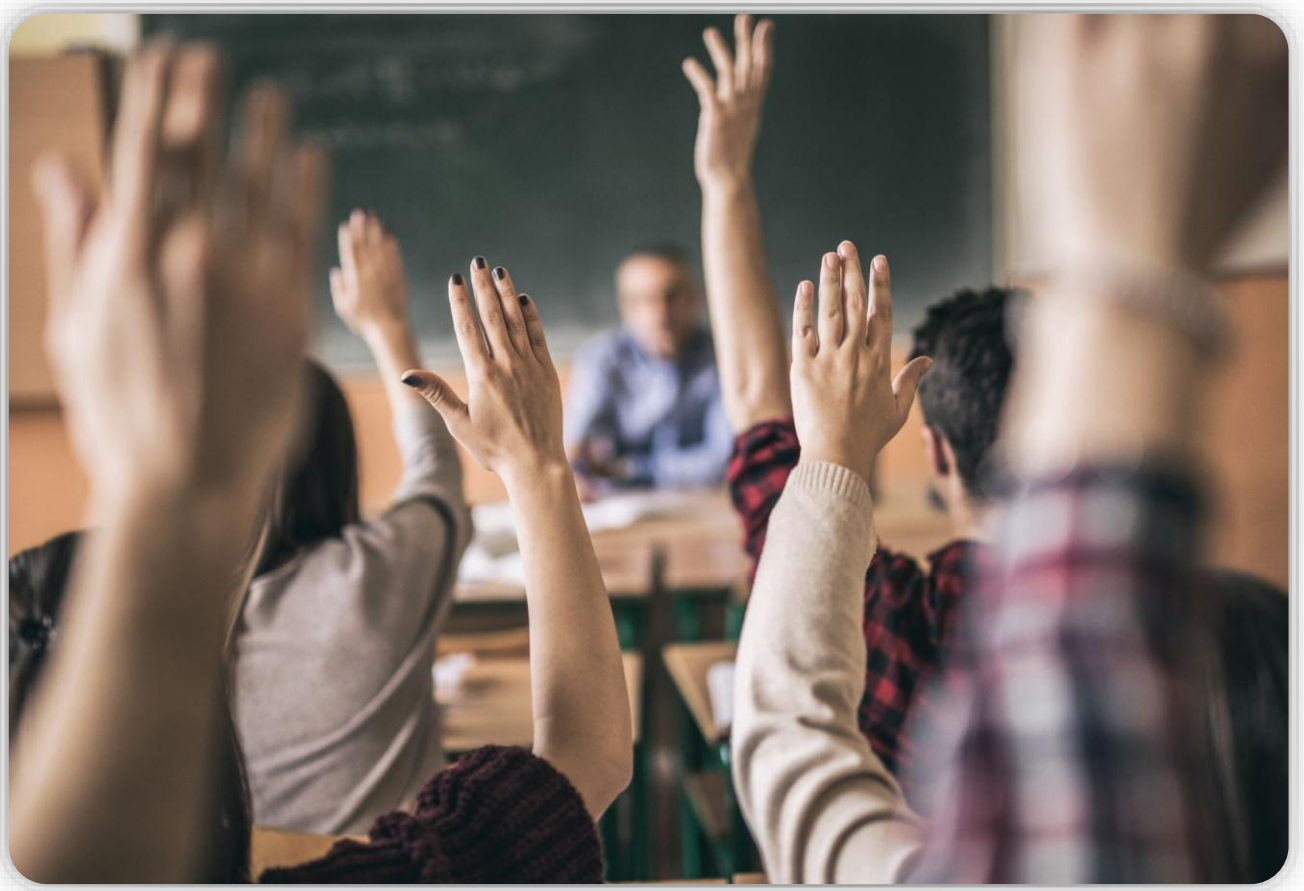
- 下載 Burp Suite Community
  - https://portswigger.net/burp/releases/professional-community-2025-2-3
  - Proxy -> Setting -> Proxy Listeners
    設定127.0.0.1:8080，
- Firefox瀏覽器中設定Local Proxy
  - 網路設定 -> 手動設定 Proxy
    127.0.0.1作為HTTP Proxy，port 8080

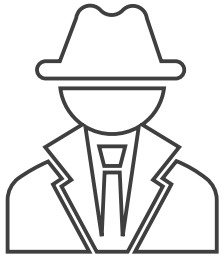
解題步驟二:
Firefox設定 攔截 127.0.0.1封包



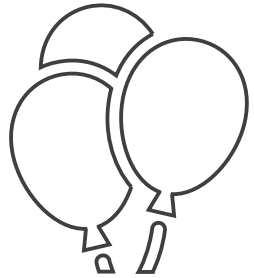
第四部分：總結

- 學習回顧
- Q&A





重視軟體系統開發安全



落實管理和安全的要求