中華資安國際
CHT Security

常見的網頁弱點利用以及防護

講師:蕭子修

聲明

未獲授權而發動駭客攻擊將涉及**刑法**，本訓練案僅供研究學習，請勿任意嘗試以身試法。

蕭子修

在職　中華資安國際/檢測工程師

專長　網頁滲透/原始碼檢測

證照　OSCP、LPT

大綱

- 滲透的手法以及工具
- 滲透常見的網頁弱點
- 淺談防禦

中華資安國際
CHT Security

滲透的手法以及工具

滲透的手法以及工具-情境

擬真程度高 ⟷ 覆蓋範圍高

- 無任何資訊
- 模擬外部使用者或攻擊者

適合驗證外部防護措施不同組件運作

- 有部分資訊
- 模擬不同權限使用者

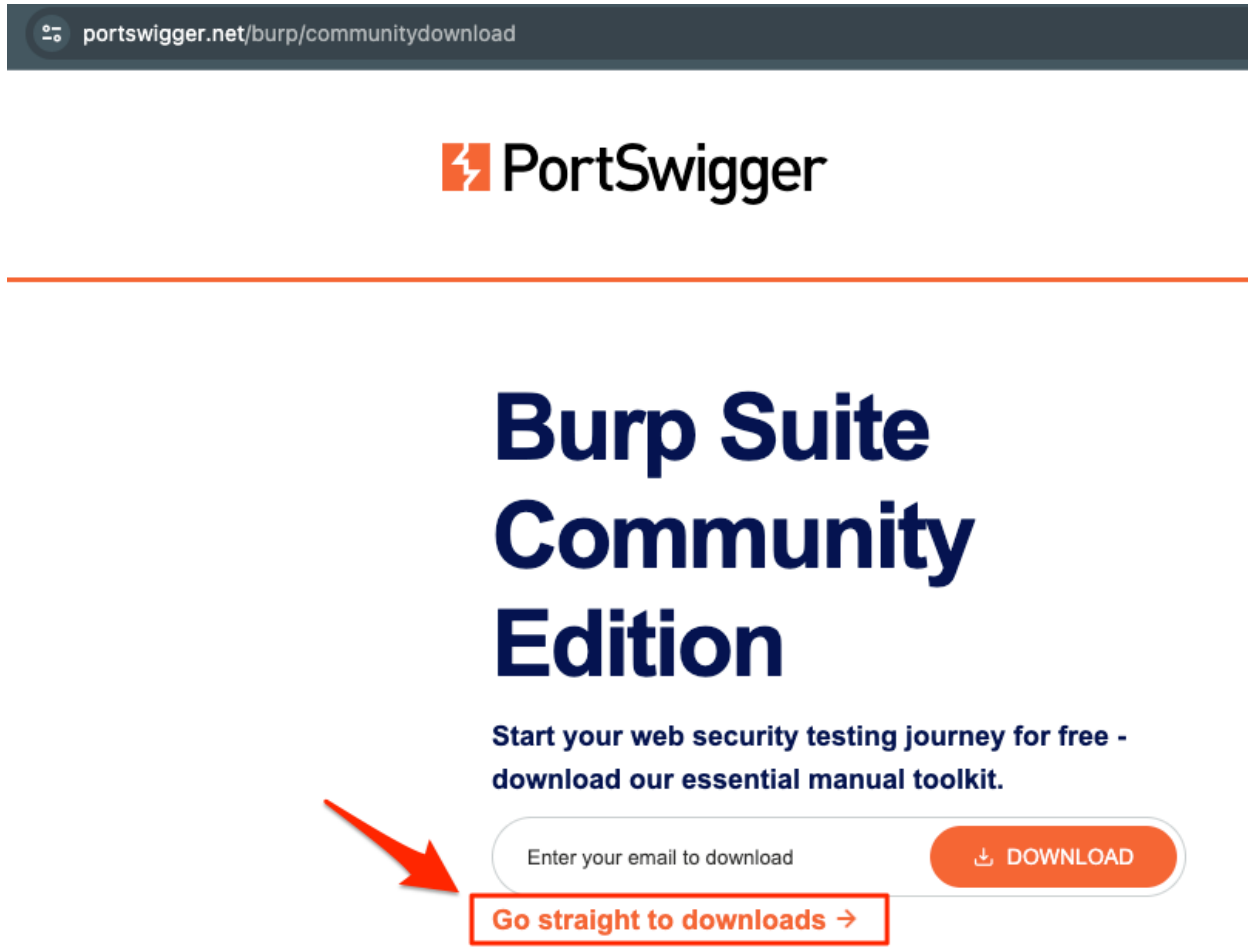
適合定期驗證檢視系統可否應對新攻擊手法

- 具備所有資訊
- 包含原始碼、框架、網路架構等
- 能夠觀察伺服器反應

適合新開發系統或功能可確保程式運作正確

滲透的手法以及工具-Burp

- 免費版:https://portswigger.net/burp/communitydownload



滲透的手法以及工具-Burp

- 攔封包
  - 攔截請求封包，看請求內容或者修改請求內容
- Repeater
  - 把有趣封包儲存，可以重複再利用
- Intruder
  - 執行暴力破解
- Decode/encode
  - 編碼解碼的輔助工具

中華資安國際
CHT Security

滲透常見的網頁弱點

中華資安國際
CHT Security

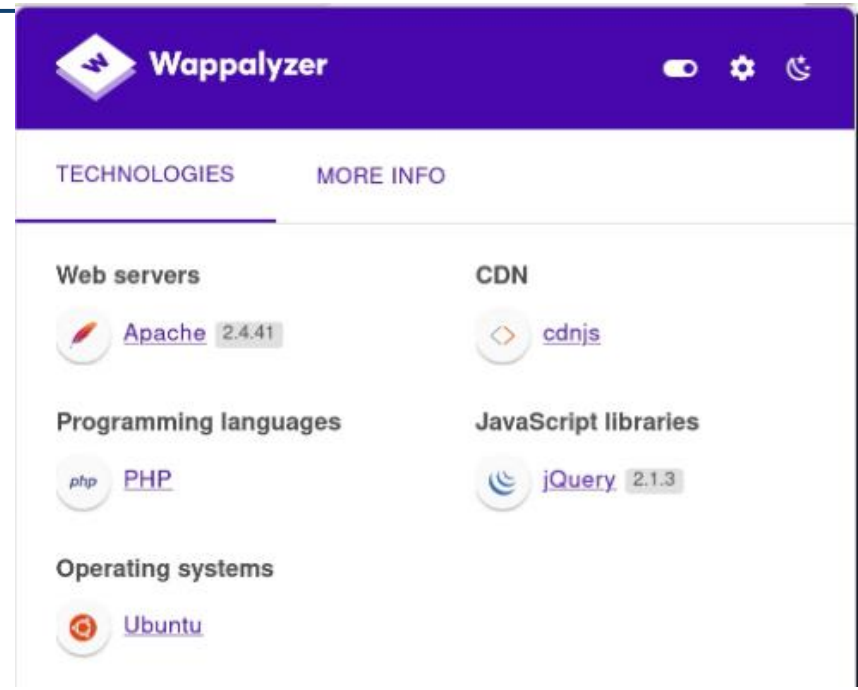
檔案上傳漏洞

攻擊者可以上傳...

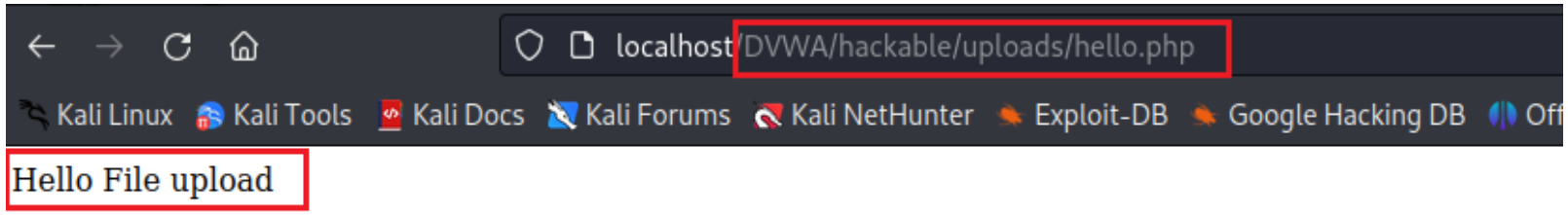
- 惡意病毒程式
- 網站後門程式
- 覆蓋系統檔案
  - 網站首頁
  - .htaccess、config等設定檔
  - /etc/passwd等系統檔案

檔案上傳漏洞

- 辨別web框架
  - 副檔名
  - 工具(Wappalyzer)
- 判斷漏洞
  - 上傳一個無害的檔案，看網頁是否能夠執行



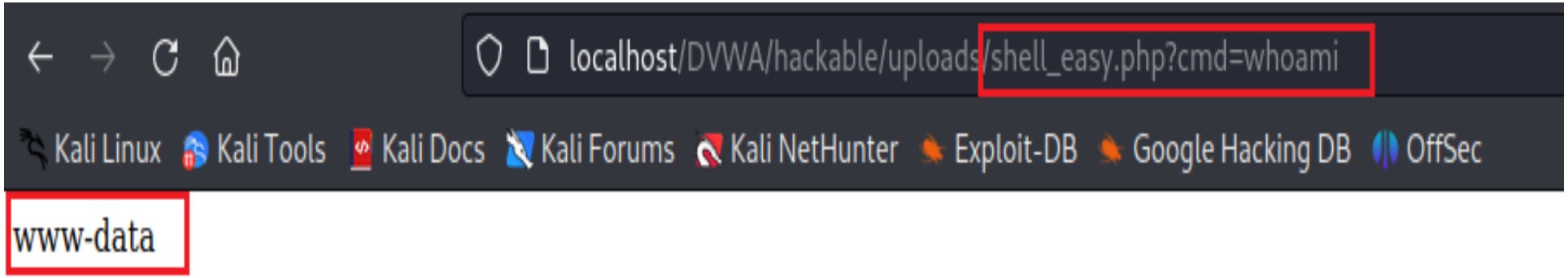

```
<?php
echo "Hello File upload";
?>
```



檔案上傳漏洞

- 能夠執行，下一步呢?
  - 如果今天上傳的不是hello，是一個後門程式呢?

```
┌──(kali㉿kali)-[~/dvwa_test_folder]
└─$ cat shell_easy.php
<?php system($_REQUEST['cmd']); ?>
```



檔案上傳-上傳的封包內容

```
POST /upload.php HTTP/1.1
Host: 127.0.0.1
Content-Length: 192
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryb4S0llL0lBfBPB4U

------WebKitFormBoundaryb4S0llL0lBfBPB4U
Content-Disposition: form-data; name="filename"; filename="test.txt"
Content-Type: text/plain

hello

------WebKitFormBoundaryb4S0llL0lBfBPB4U--
```

檔案類型

檔案名稱

檔案內容

參數內容

檔案上傳漏洞-常見的攻擊方法

- 繞過上傳不確實的上傳檢查，包含以下
  - 繞過檔名過濾
    - 前端過濾
    - 黑名單過濾
    - 白名單過濾
  - type 過濾
  - magic byte檢查繞過

檔案上傳漏洞-前端過濾

- 前端過濾 like this:

檔案上傳漏洞-前端過濾

- 前端過濾：
  - 將過濾程式碼寫在JS裡面，使用者可以直接刪除

```
function validate() {
  var file = $("#uploadFile")[0].files[0];
  var filename = file.name;
  var extension = filename.split('.').pop();

  if (extension !== 'jpg' && extension !== 'jpeg' && extension !== 'png') {
    $('#error_message').text('Only images are allowed!');
    File.form.reset();
    $("#submit").attr("disabled", true);
    return false;
  } else {
    return true;
  }
```

檔案上傳漏洞-黑名單過濾

- 黑名單防不勝防，應以白名單為原則

Type	Alternate Extensions
php	pht, phtml, pwml, php3, php4, php5, inc
asp	asp, aspx, asa, asax, ashx, asmx, aspq, axd, ascx, shtml……
perl	pl, pm, cgi, lib
jsp	jspx, jspf, jsw, jsv

檔案上傳漏洞- 黑名單 + ASP.NET + Windows

- 副檔名皆為空白
  - test.asp.
  - test.asp.....
  - test.asp.[空格].[空格].[空格]

- 副檔名不等於 "asp"
  - test.asp[空格]
  - test.asp[空格][空格][空格]

- 以上檔名在寫檔時都會變成 "test.asp"

檔案上傳漏洞-白名單過濾

- 白名單過濾
  - 優先的防護機制
  - 但也有一些眉角在

```
if (!preg_match('/\.(jpg|jpeg|png|gif)/i', $fileName))
```

```
if (!preg_match('/\.(jpg|jpeg|png|gif)$/i', $fileName))
```

檔案上傳漏洞-白名單過濾

- 白名單過濾
  - 優先的防護機制
  - 但也有一些眉角在

```
if (!preg_match('/\.(jpg|jpeg|png|gif)/i', $fileName))
```

檔案名稱需含有jpg、jpeg、png、gif之一

```
if (!preg_match('/\.(jpg|jpeg|png|gif)$/i', $fileName))
```

檔案名稱結尾是jpg、jpeg、png、gif之一

Shell.jpg.php?

檔案上傳漏洞-白名單過濾

- Windows 檔名限制
  - 檔名結尾不允許 "."
  - 檔名結尾不能是空白

- 如果嘗試儲存 "test.asp." 檔案，在存檔後會變成 "test.asp"
- 如果嘗試儲存 "test.asp " 檔案，在存檔後會變成 "test.asp"

檔案上傳漏洞-白名單過濾

- 白名單過濾
  - Web server/程式語言版本本身的缺陷
    - shell.php%00.jpg
    - 上傳檔名後方的[空格]、[點]會被忽略

    =>版本更新的重要性!

檔案上傳漏洞-白名單過濾

- 為了預防XSS，將角括號刪除
  - shell.php< >
  - shell.php<. jpg

- 先檢查副檔名再刪除角括號

=>考量過濾順序

檔案上傳漏洞-Content-Type

- 仍屬於使用者輸入，可以被修改

```
POST /upload.php HTTP/1.1
Host: 127.0.0.1
Content-Length: 192
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryb4S0llL0lBfBPB4U

------WebKitFormBoundaryb4S0llL0lBfBPB4U
Content-Disposition: form-data; name="filename"; filename="test.txt"
Content-Type: text/plain

hello

------WebKitFormBoundaryb4S0llL0lBfBPB4U--
```

檔案上傳漏洞-File Magic Byte

- 檔案標頭、File Signatures、Magic Bytes
- 仍屬於使用者輸入，可以被修改

```
23 ------WebKitFormBoundaryBrwFktXkvflinWAq
24 Content-Disposition: form-data; name="filename"; filename="
   test.gif"
25 Content-Type: image/gif
26
27 GIF87aæÄ
   +++333<<<CCCKKKSSS[[[cccllsss|||„„„ŠŠ
   ËËÿÿÿ!ù,æÿ`
     di  hª®lë¾p,Ïtmßx®ï|ïÿÀ pH,  È¤rÉl:ÿ
```

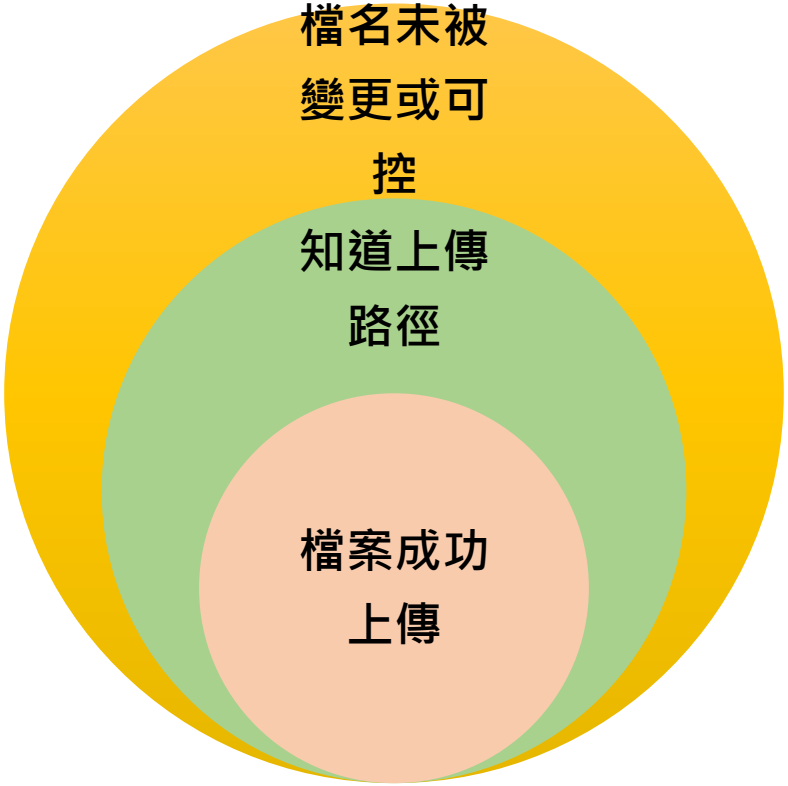
```
23 ------WebKitFormBoundaryBrwFktXkvflinWAq
24 Content-Disposition: form-data; name="filename"; filename="
   test.php"
25 Content-Type: image/gif
26
27 GIF87a
28 <?php phpinfo(); ?>
29 ------WebKitFormBoundaryBrwFktXkvflinWAq--
30
```

常見Magic Byte

檔名	Magic Byte(hex)	
gif	47 49 46 38 37 61 47 49 46 38 39 61	可轉為ASCII **GIF87a** 可轉為ASCII **GIF89a** 所以最常被使用
jpg, jpeg	FF D8 FF DB	
exe, dll	4D 5A	
zip	50 4B 03 04	zip家族的zip, docx, jar等都有此特徵
elf	7F 45 4C 46	Linux執行檔
pdf	25 50 44 46 2D	可轉為ASCII **%PDF-**
doc, xls, ppt	D0 CF 11 E0 A1 B1 1A E1	
7z	37 7A BC AF 27 1C	

檔案上傳漏洞

- 從上述最簡單的例子，可以觀察到檔案上傳弱點利用成功的條件和步驟
  1.需要知道web server使用的語言或可執行的程式
  2.要能成功上傳
  3.需要知道檔案上傳後的位置、以及檔案名稱



檔案上傳漏洞-防禦方法

- 隱藏檔案上傳路徑，若無法隱藏，將檔名隨機命名
  - 避免檔案路徑被攻擊者猜測
    - 資料庫控管
  - 隨機檔名(UUID)
    - 無法猜測
    - 無法修改
- 上傳目錄權限控管
  - 上傳目錄不可執行、執行目錄不可上傳
- 檢查文件類型
  - Content-Type/副檔名/MIME並行
- 上傳檔案掃毒
  - 利用防毒軟體作為基礎防護

中華資安國際
CHT Security

SQL injection

SQL injection 造成的後果

- 機敏資料被偷光光
- 拿到主機權限

SQL injection 可能發生的地方

- 身分驗證
- 查詢

……

- 任何可能與資料庫有連接的地方

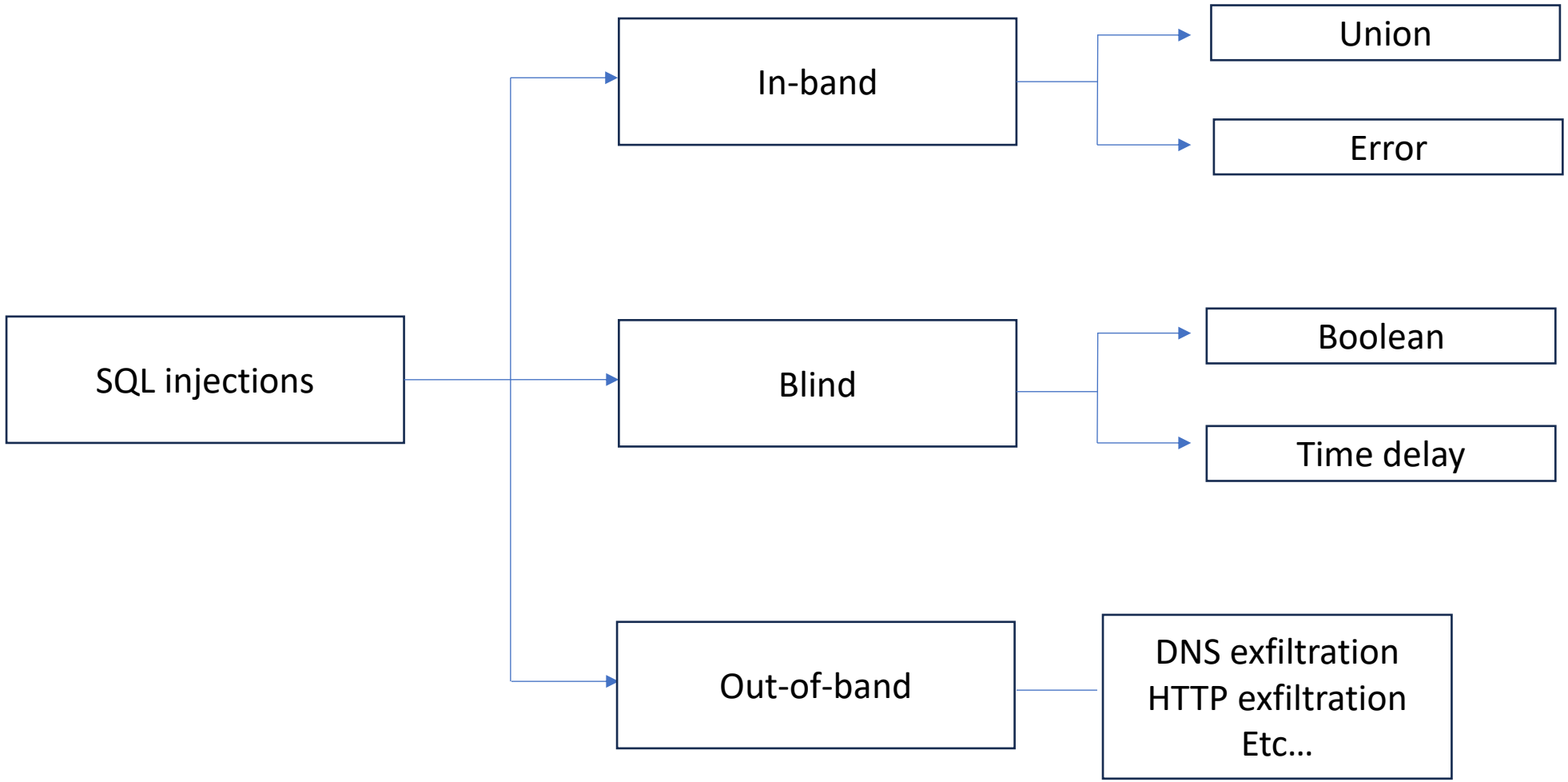
SQL injection-發生原因

- 使用者的輸入造成原本SQL查詢語句邏輯被破壞，並且執行非預期內的語法

SQL injection-判斷漏洞

- 判斷漏洞
  - 輸入特殊符號，觀察網頁回應
    - 回應出SQL語法的錯誤
    - 回應伺服器錯誤(500 status code)
    - 注入不同查詢條件時，網頁是否回應不同
    - 回應時間是否可以控制

SQL injection-常見的類型及對應的攻擊手法



SQL injection-初步認識SQLi

- 根據右圖，可以猜測後端查詢語句可能為:
  - SELECT * FROM logins WHERE username= 'something' AND password = 'something';
- 如果我們
  - 使用者輸入: admin' or '1'='1

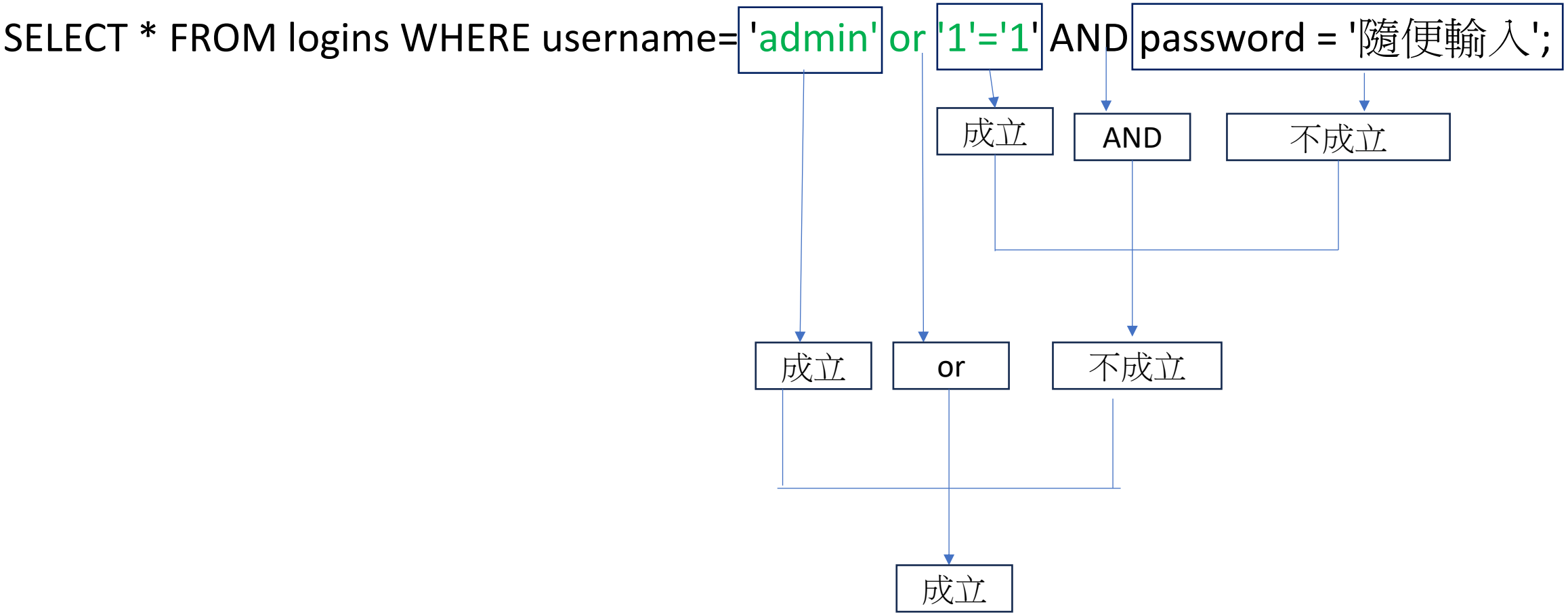
  會發生啥事?

SQL injection-初步認識SQLi

- SELECT * FROM logins WHERE username='admin' or ('1'='1' AND password = 'something ';)
- 其中'1'='1' AND password = 'something'; = > False
- SELECT * FROM logins WHERE username='admin' or (False);
- 又'admin' or (False); =>True
- SELECT * FROM logins WHERE username='admin';

SQL injection-初步認識SQLi

- AND的運算子比OR更優先執行



所以最後SQL語法:SELECT * FROM logins WHERE username= 'admin'

SQL injection-初步認識SQLi

- 使用註解

  在程式語言中，註解可以不讓程式執行，SQL中也一樣

  - --+-
  - #
  - /**/
- 註解的目的
  - SQL 註解可以**忽略原始查詢中不必要的部分**，避免語法錯誤或干擾我們的查詢。

SQL injection-union注入

- 如果今天是一個查詢資料的場景
- 與登入頁面的差異:資料會回傳給使用者

SQL injection-union注入

- 猜測後端語法可能是這樣
  - SELECT * FROM ???? WHERE id=輸入的值;
- 可以這樣利用
  - 1 UNION select 1,@@version,3,4-- -

  =>SELECT * FROM ???? WHERE id=1 UNION select 1,@@version,3,4;-- -

  =>回傳DB的版本

SQL injection-union注入

- 一般union注入的step:
  - 確認有無SQLi(觀察網頁回應)
  - 確認行數
  - 找出注入的位置
  - 依序列出DB的資訊
  - 撈出感興趣的資料

SQL injection-Blind

- Boolean-based
  - maria' AND SUBSTRING(username,1,1))=A-- -
- Time-based
  - ';WAITFOR DELAY '0:0:10'-- -

SQL injection-自動化工具

- 懶人包-SQLMAP
  - 自動化測試SQL injection的工具
- 範例
  - sqlmap "http://URL/case2.php" --data "id=1"

SQL injection-SQL只是代表

- 系統命令
  - OS (Command) Injection
- Cross-Site Scripting(XSS)
- Etc…

SQL Injection-防禦

- 參數化查詢
  - 被視為最有效可預防SQL injection的攻擊手法的防禦方式
  - DB不會將參數的內容當作SQL語句
- 驗證輸入
  - 過濾或限制不應該輸入的特殊符號或字元
- 最小權限原則

中華資安國際
CHT Security

帳號密碼相關弱點

帳號密碼相關弱點

- 弱密碼
- 帳號列舉
- 驗證碼失效
- 暴力破解

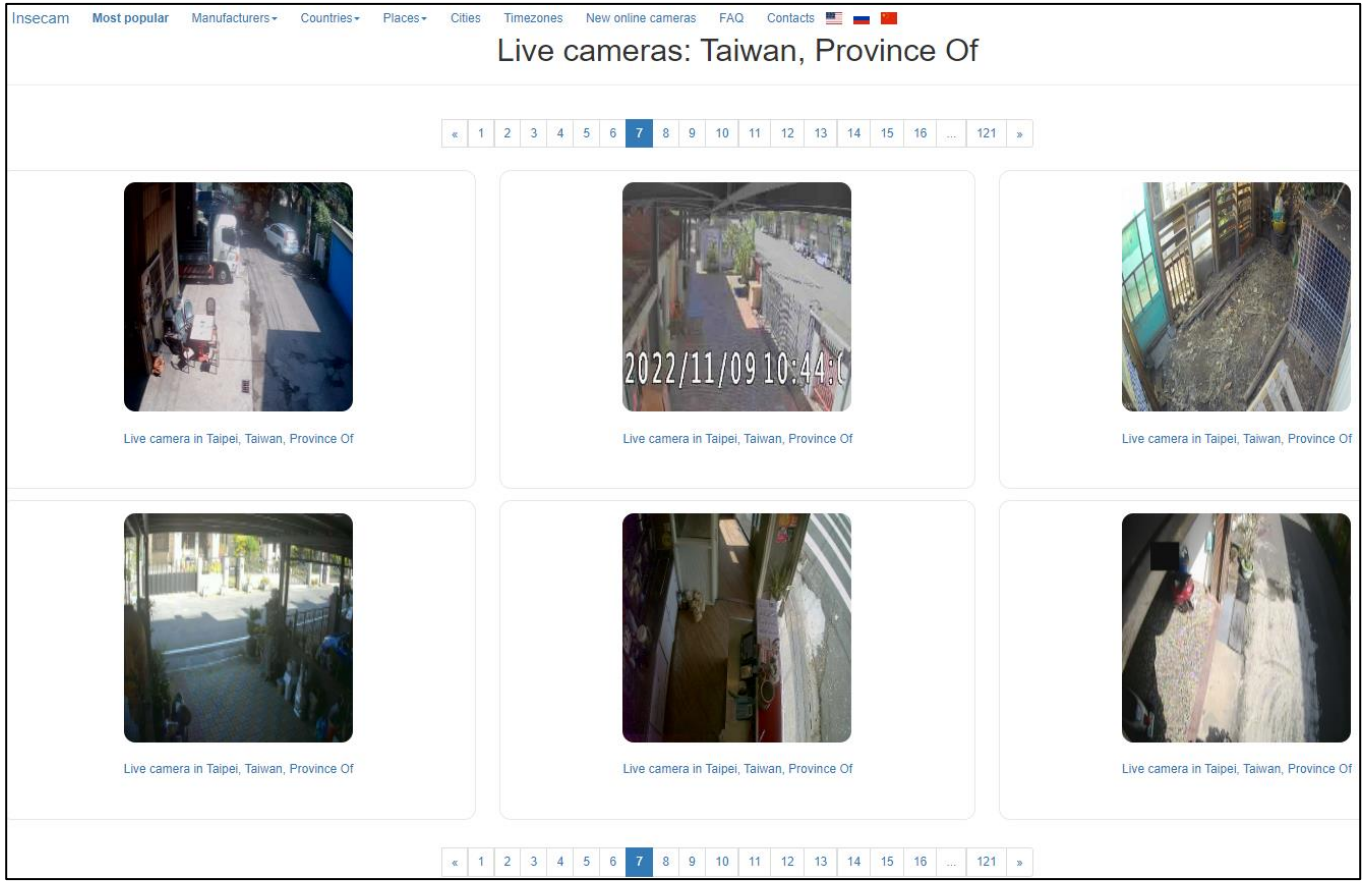
帳號密碼相關-弱密碼

- 技術含量極低，但殺傷力也是屬一屬二
- 密碼只要在rockyou裡面，就是弱密碼
- 不出意外的話，滾鍵盤的密碼就要出意外了
- 如果框架或者預設的後台頁面權限沒設定好，又使用預設密碼
  - Wordpress-login
  - IBM lotus-webadmin.nsf
  - Apache tomcat
  - 族繁不及備載


```
1420    1qaz2wsx
13892   !QAZ2wsx
14344391   EOT*ETX7¡Vamos!ETX
```

帳號密碼相關-弱密碼

- 你知道大家在看你嗎
  - 預設的帳號密碼未修改 - IP cam



ref: http://www.insecam.org/en/bycountry/tw/

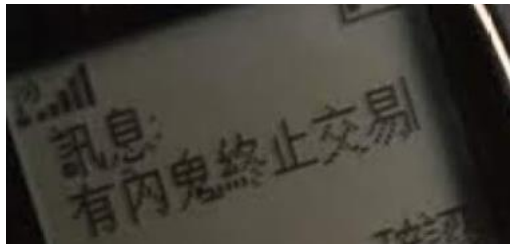
帳號密碼相關-弱密碼-防禦方式

- 資安政策的規定(密碼原則、定期更改密碼)
- 重要系統密碼回收(需要第三方工具)
- 重要網頁設定訪問權限
- 不要使用預設密碼

帳號密碼相關-帳號列舉

- 輸入帳密，網頁檢查後的回應是內鬼嗎?
  - 帳號輸入錯誤!
  - 密碼輸入錯誤!
  - 驗證碼輸入錯誤!

- 上面三個誰是內鬼???



帳號密碼相關-帳號列舉-防禦方式

- 輸入帳密，網頁檢查後的回應要是
  - 驗證碼輸入錯誤!
  - 帳號/密碼輸入錯誤!

帳號密碼相關-驗證碼失效

- 有要求輸入驗證碼，但根本沒有驗證
- 只使用數字(非圖片)，且位數過少
- 驗證錯誤次數門檻

帳號密碼相關-暴力破解

- 將上述弱點集於一身的攻擊
  - 已知帳號猜密碼
  - 已知密碼猜帳號(密碼噴灑)
  - 已知帳密try服務(Credential Stuffing)

帳號密碼相關-暴力破解-防禦

- 已知帳號猜密碼
  - 帳號鎖定、難以繞過的驗證碼機制、MFA
- 已知密碼猜帳號(密碼噴灑)
  - 鎖IP、MFA
- 已知帳密try服務(Credential Stuffing)
  - 鎖IP、MFA

加密失效

加密失效-雜湊和加密

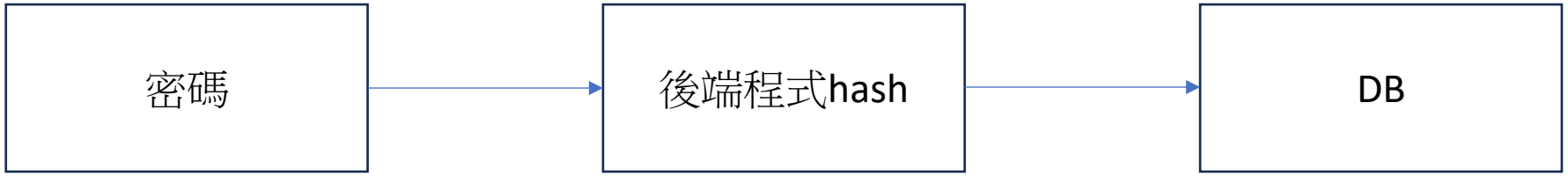
- 雜湊
  - 將任何長度的輸入轉化成固定長度的字符串或字串
  - 相同輸入一定相同輸出
  - 不可逆

- 加密
  - 將輸入上鎖
  - 有鑰匙就能解開
  - 對稱和非對稱兩種型態

加密失效-雜湊和加密

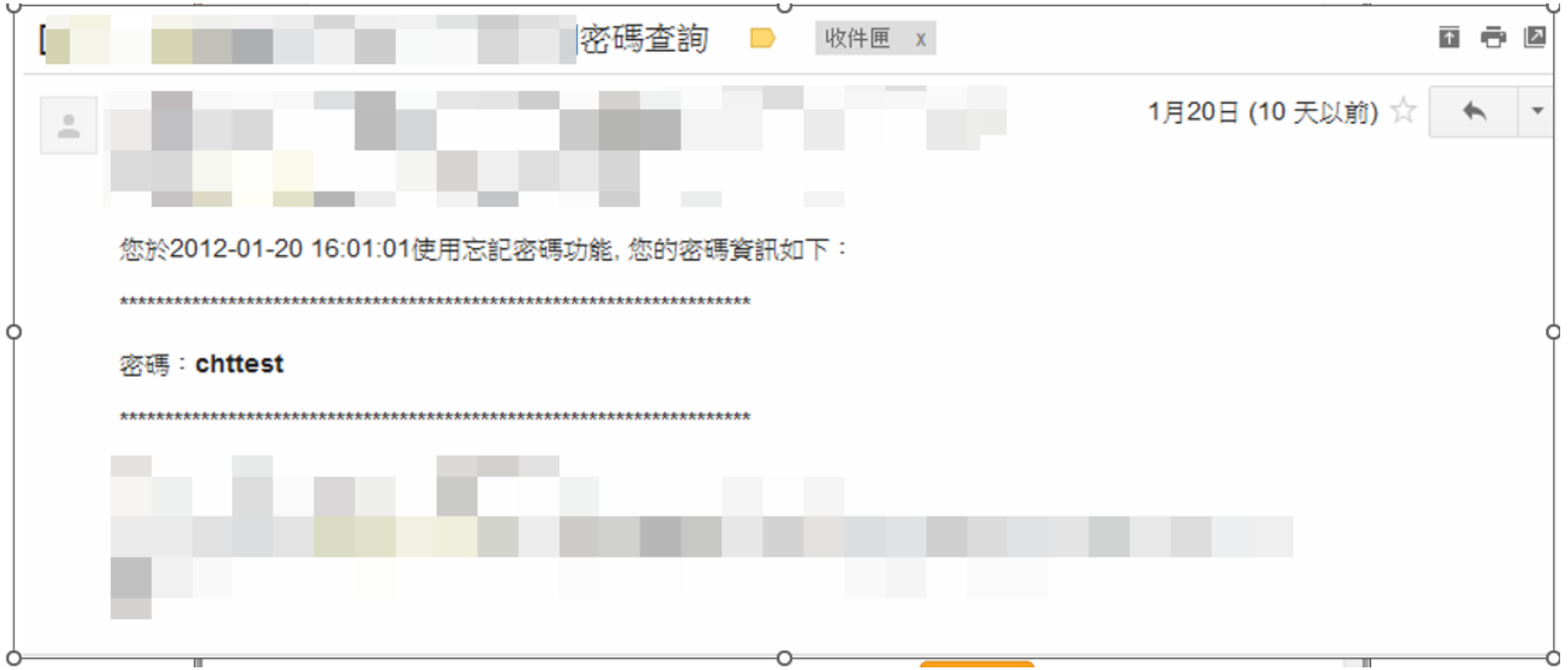
- 雜湊
  - 將任何長度的輸入轉化成固定長度的字符串或字串
  - 相同輸入一定相同輸出
  - 不可逆
    - 雜湊碰撞
    - 駭客都有蒐集癖，彩虹表應孕而生，雜湊裡的rockyou

加密失效-密碼儲存的機制

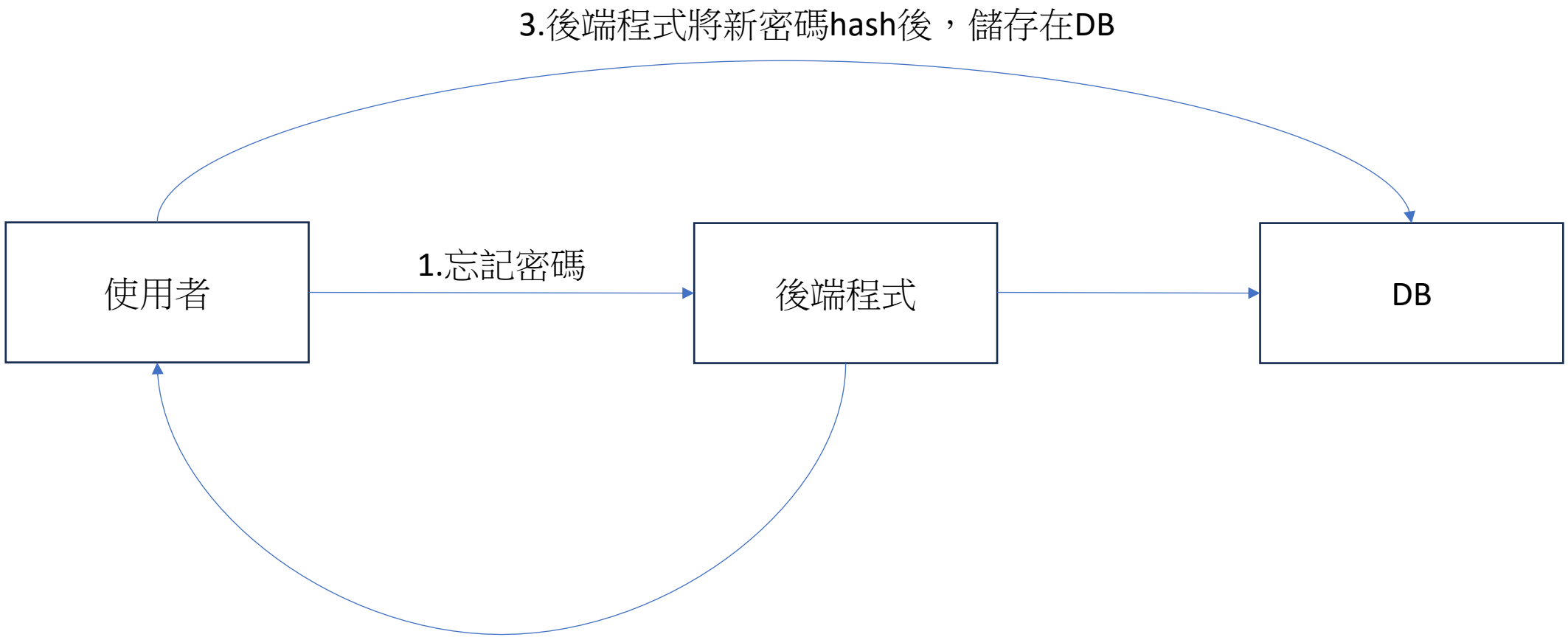


加密失效

- **情境 #1**: 密碼以明碼儲存



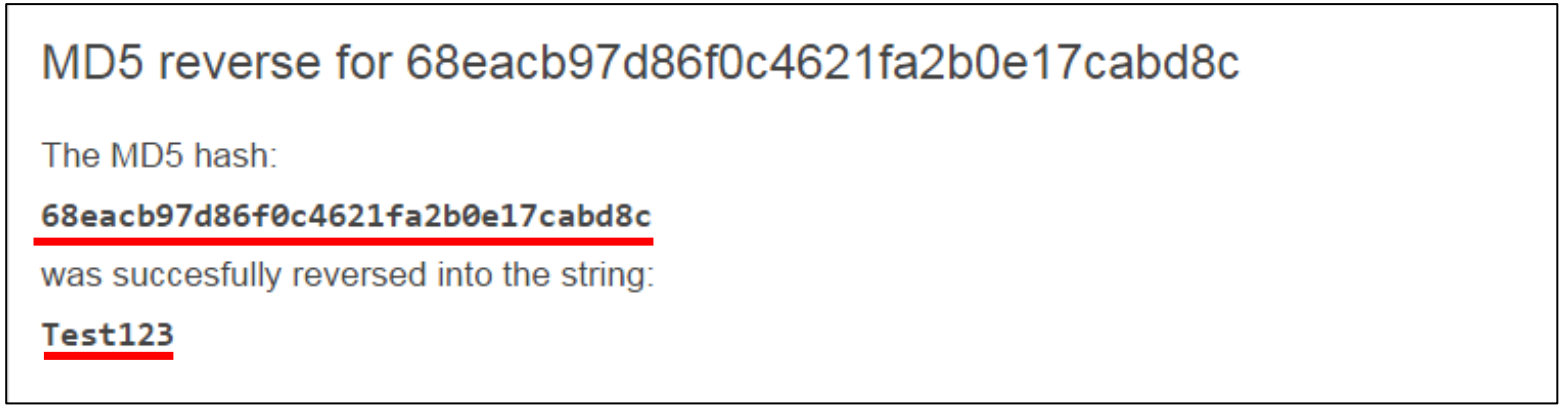
加密失效-忘記密碼處理流程



加密失效

- **情境 #2**: 不安全的儲存方式

Test123 → 68EACB97D86F0C4621FA2B0E17CABD8C
Test1234 → 2C9341CA6CF3D87B9E4EB905D6A3EC45
Test12345 → 662AF1CD1976F09A9F8CECC868CCC0A2



加密失效

- **情境 #2**: 不安全的儲存方式

正確雜湊方式
hash(passwd+salt)

Test123 → 68EACB97D86F0C4621FA2B0E17CABD8C
Test1234 → 2C9341CA6CF3D87B9E4EB905D6A3EC45
Test12345 → 662AF1CD1976F09A9F8CECC868CCC0A2

MD5 reverse for 68eacb97d86f0c4621fa2b0e17cabd8c

The MD5 hash:

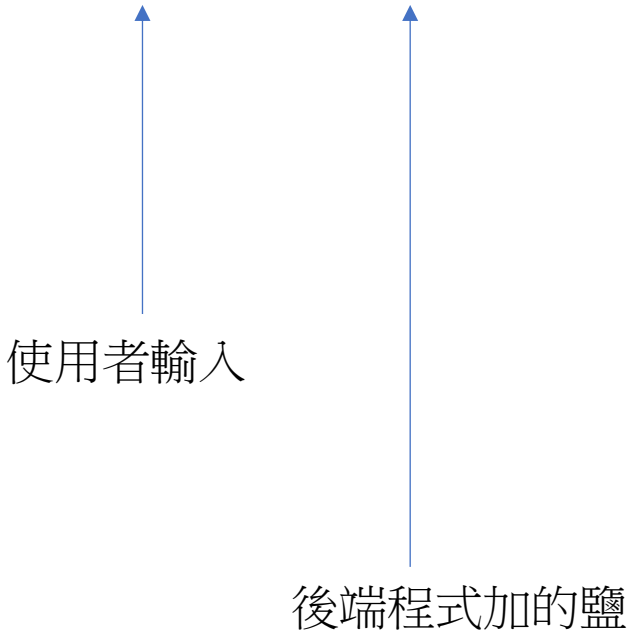
68eacb97d86f0c4621fa2b0e17cabd8c

was succesfully reversed into the string:

Test123

加密失效 –雜湊也要調味

- 正確雜湊方式=>hash(passwd+salt)
- 什麼是salt
  - passwd=1234
  - Salt = cht
  - Hash(1234cht)



加密失效 –

- **情境 #3**: 編碼 != 加密、編碼 != 加密、編碼 != 加密
- 編碼: Base64 (admin => YWRtaW4=)
  - Encode 與 Decode 不需要秘鑰，任何人都可解開
- 加密: AES、RSA
  - 對稱式: 加解密使用同一把秘鑰
  - 非對稱式: 加解密時分別使用一把公鑰一把私鑰
- 雜湊(Hash): MD5、SHA
  - Hash 後的明文是不可逆的，但可以透過彩虹表暴力破解。

加密失效– 防禦

- 不要用明文儲存任何機敏資料
- 使用最新版且標準的強演算法、協定及金鑰
- 使用安全的協定加密傳輸中的資料
- 使用加鹽雜湊法來儲存密碼 hash(passwd+salt)防範彩虹表
- 不要把編碼當作加密

中華資安國際
CHT Security

權限跨越

權限跨越-驗證和授權

- 驗證 Authentication
  - 你是誰
- 授權 Authorization
  - 你能做甚麼

權限跨越

- 水平跨越
  - 訪問同事的資料、做只有隔壁同事能做的事情
- 垂直跨越
  - 把自己當主管
- 未授權跨越
  - 未登入狀態，可以直接訪問登入後的頁面

權限跨越

- 使用者進行預期權限之外的行為
  - 通過修改URL、內部應用程式狀態或HTML頁面，或僅使用自定義API攻擊工具來繞過存取控制檢查
- 特權提升
  - 未登入即成為站台用戶，或以站台用戶身份登入即成為管理員 (垂直/水平)
  - 未經身份驗證的用戶身份強制瀏覽應驗證的頁面或以標準用戶身份存取特權頁面
- 權限控制失效危害
  - 資訊洩露、修改或損壞所有資料，或執行超出用戶權限的業務功能

權限跨越

- **情境 ：** 攻擊者可直接瀏覽某些目標網址(管理員功能)!

```
https://example.com/app/getappInfo
https://example.com/app/admin_getappInfo
https://example.com/app/admin/user_list
```

- 如果未經身份驗證的用戶可以存取任一頁面，就是一個缺陷。
- 如果一個非管理員可以存取管理頁面，這也是一個缺陷。

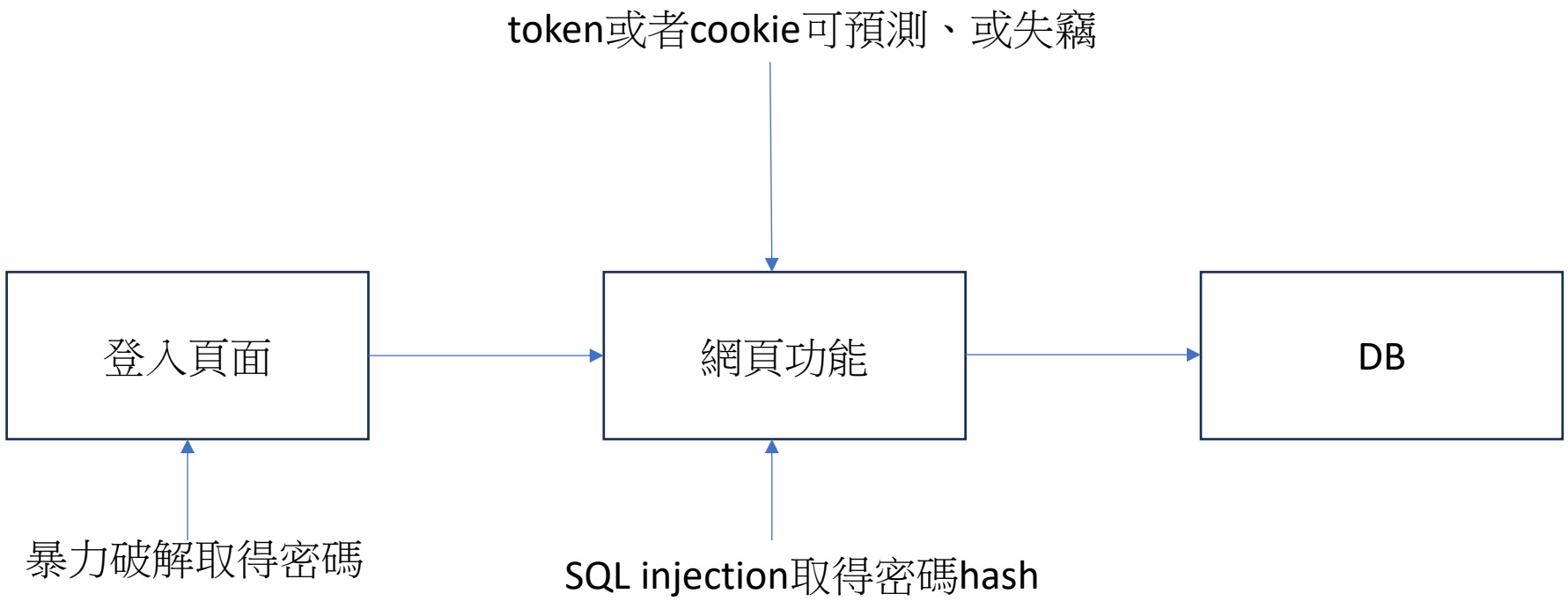
權限跨越-原因

- 權限跨越發生的原因?
  - 驗證失效
  - 拿到別人的身分

權限跨越-拿到別人的身份

- Session Weakness
  - 預測 Session Prediction
    - token或者cookie，使用可預測的方式產生
    - JWT沒有進行簽名驗證
  - 竊取 Session Hijacking
    - 釣魚或者XSS偷竊
  - 固定 Session Fixation
    - 登出後session未失效

權限跨越-拿到別人的身份



權限跨越– 防禦

- 伺服器端確實進行驗證
- session於登出後，在伺服器端應使其失效 (設定最短存取時間)
- JWT應驗證簽名
- Token或者cookie不要使用明文或者可預測的方法產生
- 使用好的雜湊演算法儲存密碼

中華資安國際
CHT Security

其他常見漏洞

Command Injection

- 測試方式
  - ; touch /tmp/test ;
  - && touch /tmp/test &&
  - | touch /tmp/test |
- 常出現弱點的地方
  - 使用到系統指令的功能，例如: 解壓縮檔案、用 ping 檢查連線、curl 取得資訊
- 修補方式
  - 嚴格檢查輸入的參數值

跨站腳本(XSS)

- 造成衝擊
  - 流失你的客戶
  - 網路釣魚/偷你的資料
- 造成原因
  - 允許客戶端執行程式非預期的JS語法

跨站腳本(XSS)-預防

- 檢查/過濾輸入值
  - 關鍵特殊符號
- 參數使用POST
  - 防釣魚URL
- 使用html/URL encode
- 伺服器設定(第二道防線)
  - cookie的安全性設定
  - Header安全性設定

中華資安國際
CHT Security

淺談防禦

淺談防禦

- ~~零風險~~
- 資安產品
  - 工欲善其事，必先利其器
- 監控機制
  - 有些事情是可以預防的
- 資安治理的推動
  - 有些事情還是得來硬的

淺談防禦-監控失效

- 缺乏或不足記錄及監控無法偵測資安事件發生
- 允許使用者或攻擊者讀取日誌或告警事件可能造成資訊洩漏

淺談防禦-監控失效

- 缺乏或不足記錄及監控無法偵測資安事件發生
  - 收什麼樣設備的log
  - 儲存位置
  - Log的內容
  - 可視化
  - 保存多久
  - 利用Log-撰寫規則

淺談防禦-構想規則撰寫

- 想像駭客攻擊流程
- 公司部門業務的合理性
- 以員工行為面構想規則

淺談防禦-規則撰寫例子

- 帳密盜用
- 跳板機
- 防毒大量刪除檔案
- 遠端登入監控
- 離職員工帳號控管
- WAF觸發

淺談防禦-log的保存

- 實作下列控制項：
  - log存取權限
  - 不存在機敏資料(帳密、密鑰)
  - 無法竄改
  - 異地儲存
  - 備份

淺談防禦-資安治理

- 資安治理的推動
  - 定期執行弱掃、滲透、源碼檢測、甚至是紅隊演練
  - 公司內部的規範

中華資安國際
CHT Security

Thank you.