「好的開始是成功的一半」，滲透測試的第一步是蒐集情資，nmap 則是主動情蒐的要角，開源的掃描工具中，筆者找不出可與之匹敵的對象，但選對工具並不代表已成功一半，還得正確有效執行命令才能找出合理的受測對象。

nmap 不僅功能強大，且老少咸宜，適合初學者到老手各個階層，底下是 nmap 最簡單的兩條應用方式（假設掃描對象是 192.168.100.10）：

基本式：

```
nmap 192.168.100.10
# 可以得到端口的狀態
```

套餐式：

```
nmap -A 192.168.100.10
# 除了端口狀態，還能找出端口對應的服務及版本，以及目標主機運行的作業系統
```

這兩條指令幾乎能涵蓋滲透測試 70% 的主動掃描任務。但就像 MS Word，九成的人只用 10% 的基本功能（打字、排版）就能滿足日常作業，但只用到進階功能的人才會被稱為「高手」，對吧！當你拿起這本書時，肯定不滿足於上面兩條指令，本書或許無法讓你的 nmap 功力達到爐火純青、登峰造極境界，但絕對稱得上「高手」。

網路掃描是 nmap 的核心功能，要深入 nmap，必須先了解其本質，本書亦不免俗，將用一章的篇幅介紹 nmap 網路掃描應用。本書不打算從頭談論 nmap，筆者假設你已經用過 nmap，而渴望提升 nmap 的應用技巧及進一步調用 Nmap Script Engine（nmap 腳本引擎，簡稱 NSE）。

## 2.1 掃描的基本觀念

扣除一些行政作業階段，滲透測試的第一步就是資料蒐集，包括來自公開的情資來源，如搜尋引擎、客戶 Web 網站、協力廠商、人力銀行、社群、Email...，這類收集行為因為不會直接碰觸目標主機，有人稱它是被動式收集；反之，直接探測目標主機取得資料的作為則稱為主動式收集或主動式掃描，是在探測當下的最新狀態。本書的重點在於介紹主動式掃描。

碁峰 www.gotop.com.tw

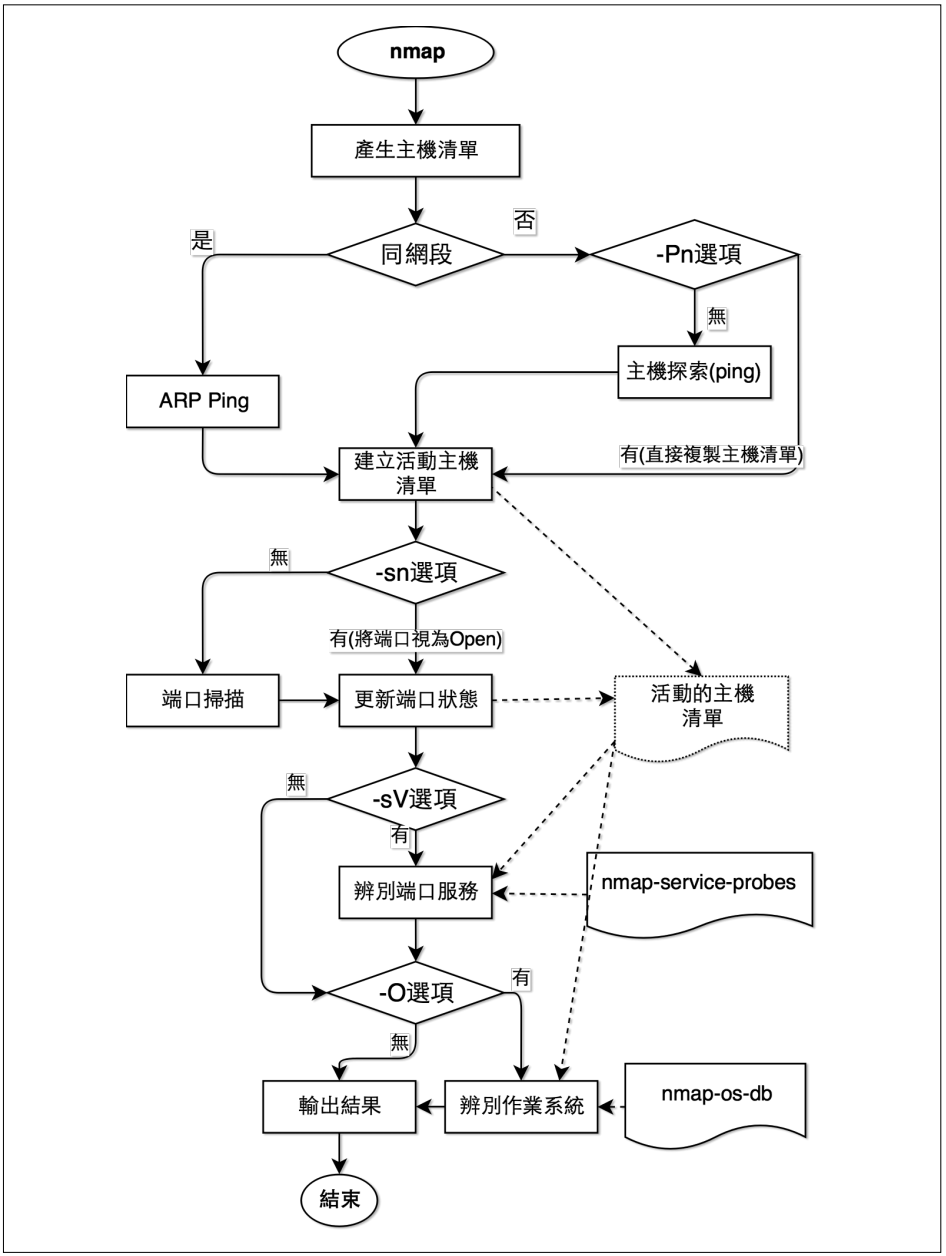


圖 2-3　nmap 掃描過程的概念圖

碁峰 www.gotop.com.tw

茲將上面介紹的常用描掃選項及預期回應整理如下表：

表 3-1：nmap 端口掃描的類型與選項

| 選項 | 封包傳送方式 | Opened | Closed | 說明 |
|---|---|---|---|---|
| -sT | 實作三向交握連線 | 完成連線 | 拒絕連線或逾時 | 非特權之基本掃描類型 |
| -sS | TCP SYN 封包 | SYN/ACK | RST | 預設具 root 權限之掃描類型 |
| -sN | 不設任何旗標之 TCP 封包（請不要和 -sn 選項混淆） | 連線逾時 | RST | 用來繞過 nonstateful 的防火牆 |
| -sF | 只設 FIN 旗標之 TCP 封包 | 連線逾時 | RST | 用來繞過 nonstateful 的防火牆 |
| -sX | 設置 FIN、PSH、URG 旗標之 TCP 封包 | 連線逾時 | RST | 用來繞過 nonstateful 的防火牆 |
| -sA | 設置 ACK 之 TCP 封包 | RST | RST | 只能證明主機存活 |
| -sW | 設置 ACK 之 TCP 封包 | RST | RST | 利用 TCP 的資料窗口大小 ( 大於或等於 0) 檢測被過濾的端口真實狀態是開啟或關閉 |
| -sM | 設置 FIN/ACK 之 TCP 封包 | 連線逾時 | RST | 只對某些 BSD 作業系統有作用 |
| -sI | 設置 SYN 之 TCP 封包 | SYN/ACK | RST | 藉由殭屍主機進行描掃 |
| -sO | 只傳送 IP 封包表頭 | 任何一種協定都會回應 | ICMP Unreachable (Type 3,Code2) | 用來找出主機使用的 IP 位址 |
| -b | 作業系統實作之三向交握連線 | 建立連線 | 拒絕連線或逾時 | 利用 FTP 綁定方式隱藏描掃來源 |
| -sU | 空的 UDP 表頭 | ICMP Unreachable (Type 3，Code 1、2、9、10 或 13) | ICMP Port Unreachable (Type 3，Code 3) | 用於 UDP 掃描，利用逾時的型態可以測試端口是開啟或受到保護 |
| -sV | 特殊協定探測 (SMTP、FTP、HTTP 等 ) | ( 非此用途 ) | ( 非此用途 ) | 檢測在開啟的端口上之服務類型 |
| -O | 利用 TCP 與 UDP 探測封包檢測標的主機的作業系統 | ( 非此用途 ) | ( 非此用途 ) | 利用多重方法檢測目標主機的作業系統／韌體版本 |

碁峰 www.gotop.com.tw

如果該項探測資料有指定適用端口（ports 指示詞），則不論 rarity 的值為何，只要目標主機的開啟端口符合 ports 指示的清單時都會執行該筆探測，例如 TCP ms-sql-s 的適用端口為 1433，而 rarity 8，大於預設強度 7，照說是不會被執行，不過，若 nmap 發現主機開啟端口 1433，則仍會對此端口執行 TCP ms-sql-s 探測。

### --version-light

等同於 --version-intensity 2。

### --version-all

等同於 --version-intensity 9。

## 3.5.4 端口掃描重點整理

一、 有關 TCP、SCTP、UDP 掃描選項及回應重點整理如表 3-2 至 3-4。

表 3-2：TCP 協定的掃描選項及回應的封包格式

| nmap 選項 | 測試方式 | TCP 旗標 | 受測端的回應 | |
|---|---|---|---|---|
| | | | open | closed |
| -sT | TCP 連線 ( 完成三向交握 ) | SYN | SYN/ACK | RST |
| -sS | SYN 連線 | SYN | SYN/ACK | RST |
| -sA / -sW | ACK 連線 | ( 未受防火牆阻擋 ) | RST | RST |
| | | ( 受防火牆阻擋 ) | ( 無回應 )<br>ICMP 不可達 | ( 無回應 )<br>ICMP 不可達 |
| -sM | ACK/FIN | ( 無防火牆 ) | RST | RST |
| | | ( 態狀型防火牆 ) | ( 無回應 ) | ( 無回應 ) |
| -sX | XMAS(Unix) | FIN/PSH/URG | ( 無回應 ) | RST |
| -sF | FIN 連線 (Unix) | FIN | ( 無回應 ) | RST/ACK |
| -sN | NULL(Unix) | all flags OFF | ( 無回應 ) | RST/ACK |

碁峰 www.gotop.com.tw

口掃描資訊；視窗 2 使用 -d，除了提供較詳細的掃描結果外，還可看到視窗 2 中①和③處執行主機探測、②執行 DNS 解析，④發現主機上的端口 80；視窗 3 使用 -d3，可得到更詳細的掃描過程資訊，包括發送及接收的封包格式。當懷疑掃描結果時，可以啟用 -d 選項查看收、發內容，判斷是否因網路某個環節影響最終結果。

視窗1

```
root@kali:~# nmap 192.168.1.199 -p80

Starting Nmap 7.31 ( https://nmap.org ) at
Nmap scan report for 192.168.1.199
Host is up (0.00075s latency).
PORT   STATE SERVICE
80/tcp open  http
Nmap done: 1 IP address (1 host up) scanne
root@kali:~#
```

基本掃描結果

視窗2

```
root@kali:~# nmap 192.168.1.199 -p80 -d

Starting Nmap 7.31 ( https://nmap.org ) at 2017-04-29 02:05 CST
--------------- Timing report ---------------
  hostgroups: min 1, max 100000
  rtt-timeouts: init 1000, min 100, max 10000
  max-scan-delay: TCP 1000, UDP 1000, SCTP 1000
  parallelism: min 0, max 0
  max-retries: 10, host-timeout: 0
  min-rate: 0, max-rate: 0
---------------------------------------------------
Initiating Ping Scan at 02:05
Scanning 192.168.1.199 [4 ports]
Packet capture filter (device eth0): dst host 192.168.84.133 and (icmp
or ((tcp or udp or sctp) and (src host 192.168.1.199)))
We got a TCP ping packet back from 192.168.1.199 port 80 (trynum = 0)
Completed Ping Scan at 02:05, 0.06s elapsed (1 total hosts)
Overall sending rates: 69.95 packets / s, 2658.13 bytes / s.
mass_rdns: Using DNS server 192.168.84.2
Initiating Parallel DNS resolution of 1 host. at 02:05
mass_rdns: 0.06s 0/1 [#: 1, OK: 0, NX: 0, DR: 0, SF: 0, TR: 1]
Completed Parallel DNS resolution of 1 host. at 02:05, 0.06s elapsed
DNS resolution of 1 IPs took 0.06s. Mode: Async [#: 1, OK: 0, NX: 1, D
0, TR: 1, CN: 0]
Initiating SYN Stealth Scan at 02:05
Scanning 192.168.1.199 [1 port]
Packet capture filter (device eth0): dst host 192.168.84.133 and (icmp
or ((tcp or udp or sctp) and (src host 192.168.1.199)))
Discovered open port 80/tcp on 192.168.1.199
Completed SYN Stealth Scan at 02:05, 0.06s elapsed (1 total ports)
Overall sending rates: 15.46 packets / s, 680.36 bytes / s.
Nmap scan report for 192.168.1.199
Host is up, received reset ttl 128 (0.00056s latency).
Scanned at 2017-04-29 02:05:13 CST for 0s
PORT   STATE SERVICE REASON
80/tcp open  http    syn-ack ttl 128
Final times for host: srtt: 562 rttvar: 4046  to: 100000

Read from /usr/bin/../share/nmap: nmap-payloads nmap-services.
Nmap done: 1 IP address (1 host up) scanned in 0.30 seconds
           Raw packets sent: 5 (196B) | Rcvd: 2 (84B)
root@kali:~#
```

還有掃描過程的訊息

比較詳細的掃描結果

視窗3

```
root@kali:~# nmap 192.168.1.199 -p80 -d3

Starting Nmap 7.31 ( https://nmap.org ) at
Fetchfile found /usr/bin/../share/nmap/nma
Fetchfile found /usr/bin/../share/nmap/nma
The max # of sockets we are using is: 0
--------------- Timing report ---------------
  hostgroups: min 1, max 100000
  rtt-timeouts: init 1000, min 100, max 10
  max-scan-delay: TCP 1000, UDP 1000, SCTP
  parallelism: min 0, max 0
  max-retries: 10, host-timeout: 0
  min-rate: 0, max-rate: 0
---------------------------------------------
Fetchfile found /usr/bin/../share/nmap/nma
Initiating Ping Scan at 02:05
Scanning 192.168.1.199 [4 ports]
Packet capture filter (device eth0): dst h
or ((tcp or udp or sctp) and (src host 192
SENT (0.1026s) ICMP [192.168.84.133 > 192.
 id=18724 seq=0] IP [ver=4 ihl=5 tos=0x00
 csum=0x95e8]
SENT (0.1028s) TCP [192.168.84.133:37791 >
k=0 off=6 res=0 win=1024 csum=0xF8AE urp=0
iplen=44 id=781 foff=0 ttl=41 proto=6 csum
SENT (0.1029s) TCP [192.168.84.133:37791 >
34 off=5 res=0 win=1024 csum=0x11C8 urp=0]
58249 foff=0 ttl=44 proto=6 csum=0xd3a9]
SENT (0.1030s) ICMP [192.168.84.133 > 192.
ode=0) id=25884 seq=0 orig=0 recv=0 trans=0] IP [ver=4 ihl=5 tos=0x00 iplen=40 i
d=38669 foff=0 ttl=49 proto=1 csum=0x1b2b]
**TIMING STATS** (0.1031s): IP, probes active/freshportsleft/retry_stack/outstan
ding/retranwait/onbench, cwnd/ssthresh/delay, timeout/srtt/rttvar/
   Groupstats (1/1 incomplete): 4/*/*/*/*/* 10.00/75/* 1000000/-1/-1
Current sending rates: 103.58 packets / s, 3935.99 bytes / s.
Overall sending rates: 103.58 packets / s, 3935.99 bytes / s.
RCVD (0.1032s) TCP [192.168.1.199:80 > 192.168.84.133:37791 R seq=3637073334 ack
=0 off=5 res=0 win=32767 csum=0x95D4 urp=0] IP [ver=4 ihl=5 tos=0x00 iplen=40 id

PORT   STATE SERVICE REASON
80/tcp open  http    syn-ack ttl 128
Final times for host: srtt: 392 rttvar: 3888  to: 100000

Read from /usr/bin/../share/nmap: nmap-payloads nmap-services.
Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds
           Raw packets sent: 5 (196B) | Rcvd: 2 (84B)
root@kali:~#
```

部分輸出已裁切

圖 3-6　不同除錯詳細度選項的差異

碁峰 www.gotop.com.tw

### --iflist

列出本機網路卡資訊。這個選項在 Windows 系統尤其好用，在 Windows 環境使用 -e 選項指定網卡時，如果使用圖 3-7 左方 ipconfig 查到的名稱「區域連線」，會得到「I cannot figure out what source address to use for device 區域連線 , does it even exist?」的錯誤訊息，利用 nmap 的 --iflist 選項查詢（圖 3-7 右方）得知 nmap 認定此網路卡的名稱是「eth2」，所以指定網路卡時應該用「-e eth2」，而不是「-e 區域連線」。

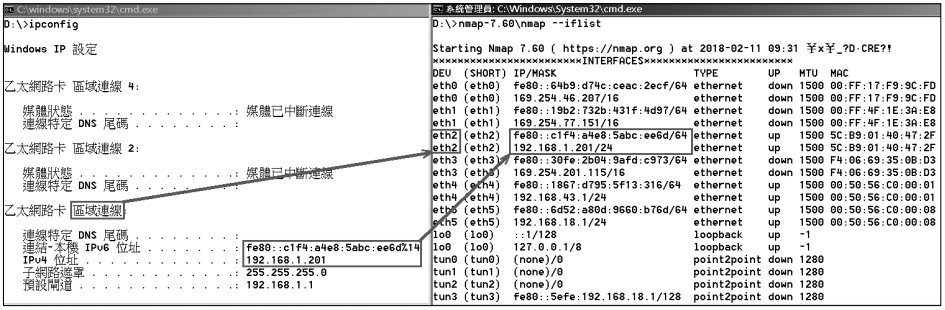


圖 3-7　Windows 網路卡名稱與 nmap 認定的名稱之差異

## 3.12　分散式掃描

常規滲透測試的範圍都不大，十來臺伺服器就了不得了，用到分散掃描的機會並不多，至少在筆者的專案上都還派不上用場。但今日若打算探索使用者的個人電腦是否啟用不當的服務（例如被安置木馬或後門程式），由於電腦數量多，就適合進行分散式掃描。

想像一種情境，某機構的網段分割成資訊部門、業務部門、行政部門及兩家外點單位，範例網路拓撲如圖 3-8 所示。每個網段最多可以有 255 臺主機。

碁峰 www.gotop.com.tw

是的，dnmap 的確可以執行分散掃描，但它分派指令是隨機的，哪一臺 Client 空閒就分派一條指令給它處理，但無法保證 nmap -A 192.168.100.0/24 這條指令可以交給 A2 的工作站處理，同理，其他指令亦無法保證發送給適當的工作站。最差情況如圖 3-10 所示，造成工作站向不同網段交叉掃描，雖然 5 臺工作站可以平行處理，但並無法減少經過路由器 R 的網路流量，依舊會出現 DoS 現象，甚至更像 DDoS。

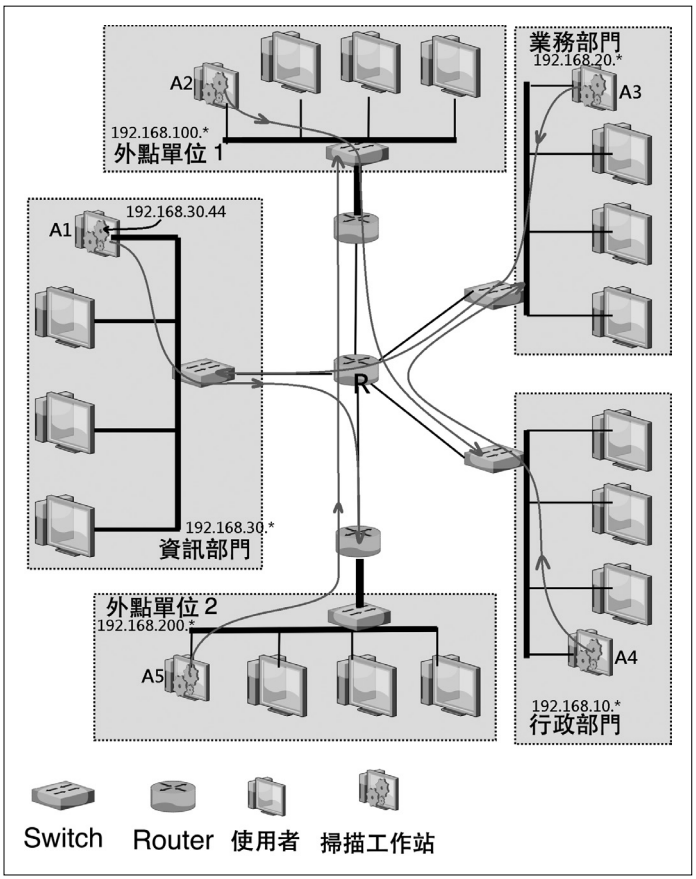


圖 3-10　分散掃描的最差情況

碁峰 www.gotop.com.tw

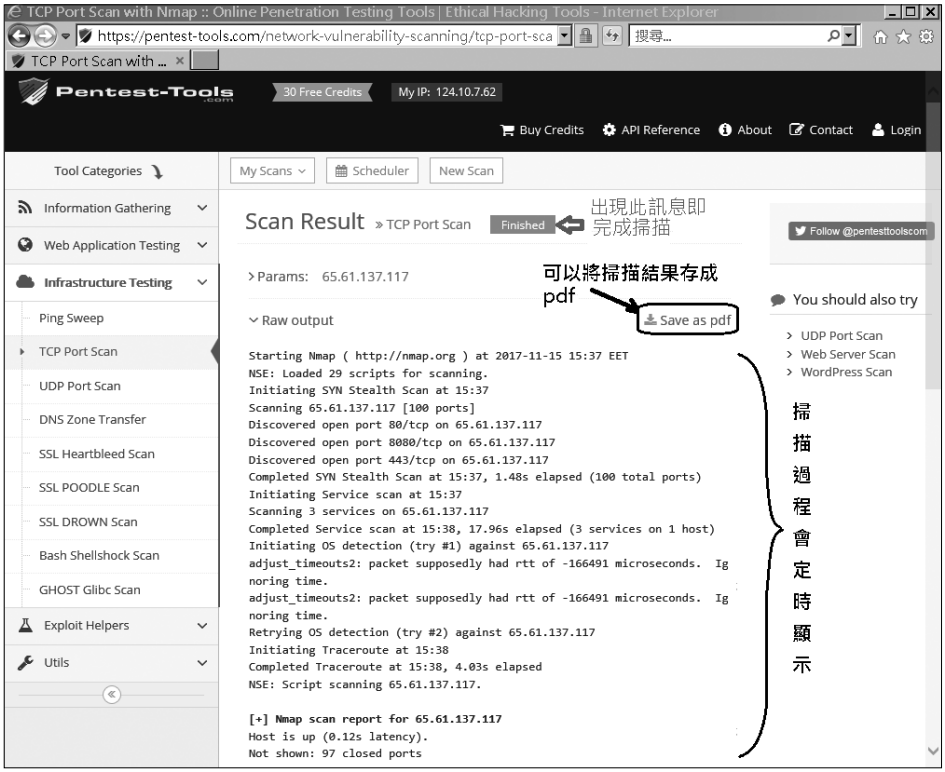


圖 4-2　Pentest-Tools 顯示 TCP Port Scan 結果

## 4.1.2　suip

網址：https://suip.biz/?act=nmap

此網站的功能琳琅滿目，而且都是免費的，包括 IP 位址的分配資訊、弱點掃描、資訊收集、電子郵件分析等，我們在意的是端口掃描功能，它真的是利用 nmap 執行掃描，而且可以掃描自定範圍的主機及端口。此網站的缺點就是內嵌太多廣告，這些廣告可是會佔用頻寬。

碁峰 www.gotop.com.tw

從圖 4-8 可看出原本電腦繞送的第一關是 192.168.1.1，啟用 SecurityKISS VPN 之後會變成 10.11.0.1，表示 Gateway 已改由 VPN 代理，而且 Traceroute 的關卡也變多了。

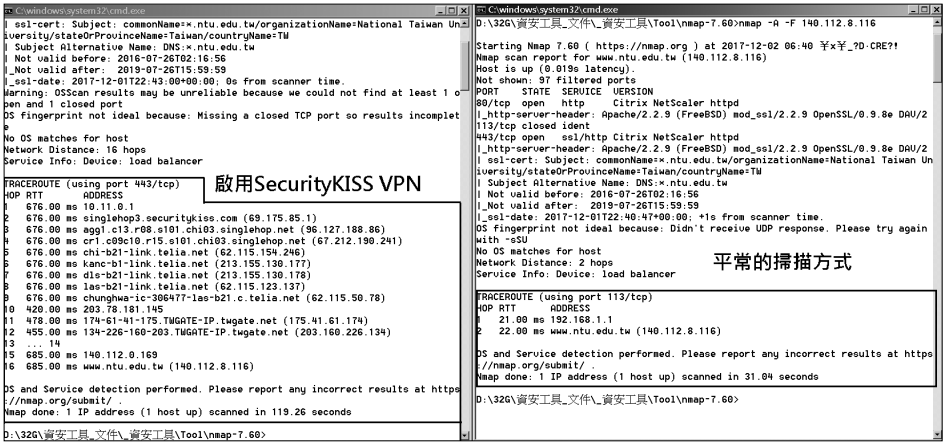


圖 4-8　有無啟用 SecurityKISS 之比較

前面提到免費的 SecurityKISS 只支援 80 及 443，掃描時可能會遇到圖 4-9 的訊息，因為免費的 SecurityKISS 不支援 ping 封包，掃描時若遇到此訊息，請在命令中加入 -Pn 選項。

```
D:\32G\資安工具_文件\_資安工具\Tool\nmap-7.60>nmap -A 163.22.109.1

Starting Nmap 7.60 ( https://nmap.org ) at 2017-12-02 08:25 ¥x¥_?D·CRE?!
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn

Nmap done: 1 IP address (0 hosts up) scanned in 8.33 seconds
```

圖 4-9　免費 VPN 可能不支援某些 Ping 封包

碁峰 www.gotop.com.tw

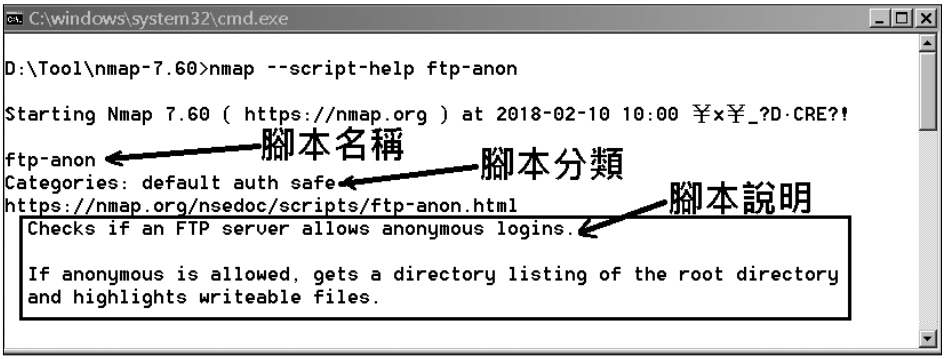


圖 5-1　查看腳本的說明文字

若指定 -oX 選項時，nmap 會將輸出結果寫到指定的 XML 檔案。

### --script-trace

此選項和 --packet-trace 類似，但 --script-trace 是作用在應用層，而 --packet-trace 則是處理封包追蹤，若指定此選項時，所有入站及出站的通訊資訊都會輸出到終端機，包含通訊協定、來源及目的 IP、承載的資料等，如果輸出的資料中有 5% 以上是不可見的內容，就會以十六進制格式輸出。

### --script-updatedb

更新 scripts/script.db 的內容，除非手動在 scripts 目錄添加或移除 NSE 腳本，否則不用去更新 script.db 的內容。使用此選項時不需其他參數，語法為：

`nmap --script-updatedb`。

### --script-timeout TIME

有些腳本執行時間過長，影響整體的掃描作業，可以利用本選項規範每一支腳本可執行的時限，若逾期未完成，nmap 會逕中止該腳本。

碁峰 www.gotop.com.tw

上述雖然以 Excel 為例子，但 Google 線上試算表亦適用。底下就以幾個例子說明如何使用此腳本篩選檔。

## 6.1 Apache Struts

早在 2016 年就爆發的 Apache Struts 漏洞（CVE-2016-3081），可能讓駭客從遠端控制網頁伺服器，雖然經過修補，到了 2017 年還是發現另一個漏洞（CVE-2017-5638；參考 http://tdoh.logdown.com/posts/2017/03/08/1523847）。當研究單位發布漏洞時，機關主管大都會要求系統負人員進行修補，但修補作業是否確實呢？利用 NSE 試試有幾部伺服器還存在 Struts2 的漏洞，會比用人工逐臺檢查來得快速。

利用「文字篩選」「**腳本名稱**」中「包含」apache 或 5638 的腳本（圖 6-5），結果找到三筆腳本，其中 http-vuln-cve2017-5638 是我們想要用來探索漏洞的腳本。

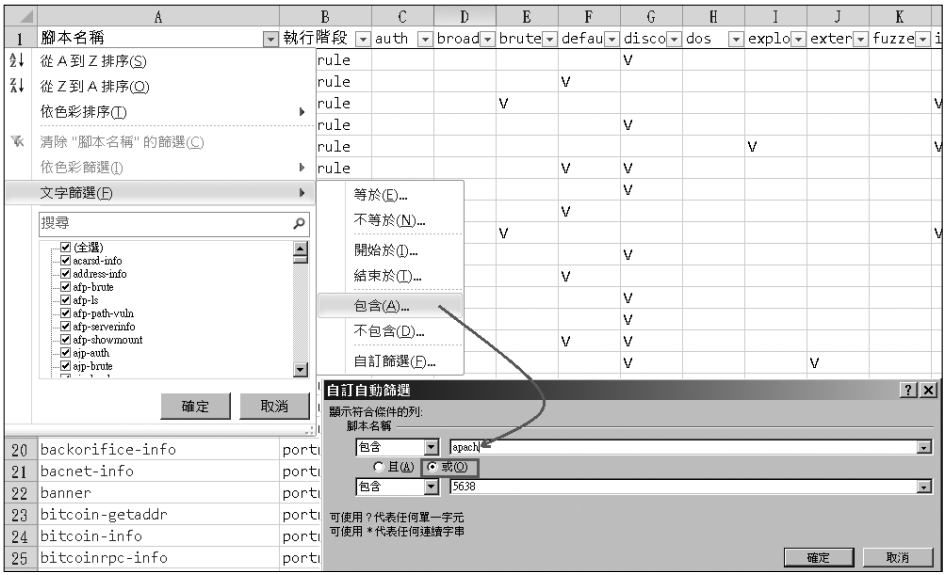


圖 6-5　利用文字篩選有關 apache 的腳本

碁峰 www.gotop.com.tw

## 6.2 SSL-HeartBleed

再來看看經典的 SSL-HeartBleed 例子。一個原本用來保護資料傳輸安全的功能，卻因為記憶體內容滲漏的弱點，造成機密資料外洩，HeartBleed 漏洞並非 SSL/TLS 協定的問題，而是 OpenSSL 實作上的疏忽。

和上一個例子類似，利用「文字篩選」從「腳本名稱」中選擇「包含」ssl 或 heart 的腳本，找到「ssl-heartbleed」，從試算表右邊複製範例語法「nmap <HOST> -p443 --script ssl-heartbleed」，修改並執行如下命令：

```
nmap 192.168.18.* -p443 --script --script ssl-heartbleed
```

由圖 6-7 可知 192.168.18.133 有 SSL-HeartBleed 弱點。

```
C:\windows\system32\cmd.exe
D:\tool\nmap7.6>nmap 192.168.18.* -p443 --script --script ssl-heartbleed

Starting Nmap 7.60 ( https://nmap.org ) at 2017-12-30 04:50 ￥x￥_?D·CRE?!
Nmap scan report for 192.168.18.133
Host is up (0.00s latency).

PORT    STATE SERVICE
443/tcp open  https
| ssl-heartbleed:
|   VULNERABLE:
|   The Heartbleed Bug is a serious vulnerability in the popular OpenSSL cryptog
raphic software library. It allows for stealing information intended to be prote
cted by SSL/TLS encryption.
|     State: VULNERABLE
|     Risk factor: High
|       OpenSSL versions 1.0.1 and 1.0.2-beta releases (including 1.0.1f and 1.0
.2-beta1) of OpenSSL are affected by the Heartbleed bug. The bug allows for read
ing memory of systems protected by the vulnerable OpenSSL versions and could all
ow for disclosure of otherwise encrypted confidential information as well as the
 encryption keys themselves.
|
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0160
|       http://www.openssl.org/news/secadv_20140407.txt
|_      http://cvedetails.com/cve/2014-0160/

Nmap scan report for 192.168.18.145
Host is up (0.00s latency).

PORT    STATE  SERVICE
443/tcp closed https

Nmap done: 256 IP addresses (2 hosts up) scanned in 84.44 seconds

D:\tool\nmap7.6>
```

圖 6-7　藉由 NSE 發現 heartbleeded 漏洞

碁峰 www.gotop.com.tw