

# 第四章

## 掃瞄

# 掃描

- 掃描（Scanning）與列舉（Enumeration）是攻擊者入侵的第一步驟。
- 掃描完畢通常就接著進行列舉，許多的工具軟體也將這兩個步驟結合在一起。

# 掃描

- 目的地範圍內特定的IP，尤其是活著的主機IP。
- 目的主機的作业系統。
- 系統架構及每一台電腦上執行的服務。
- 該主機上安裝的應用程式。

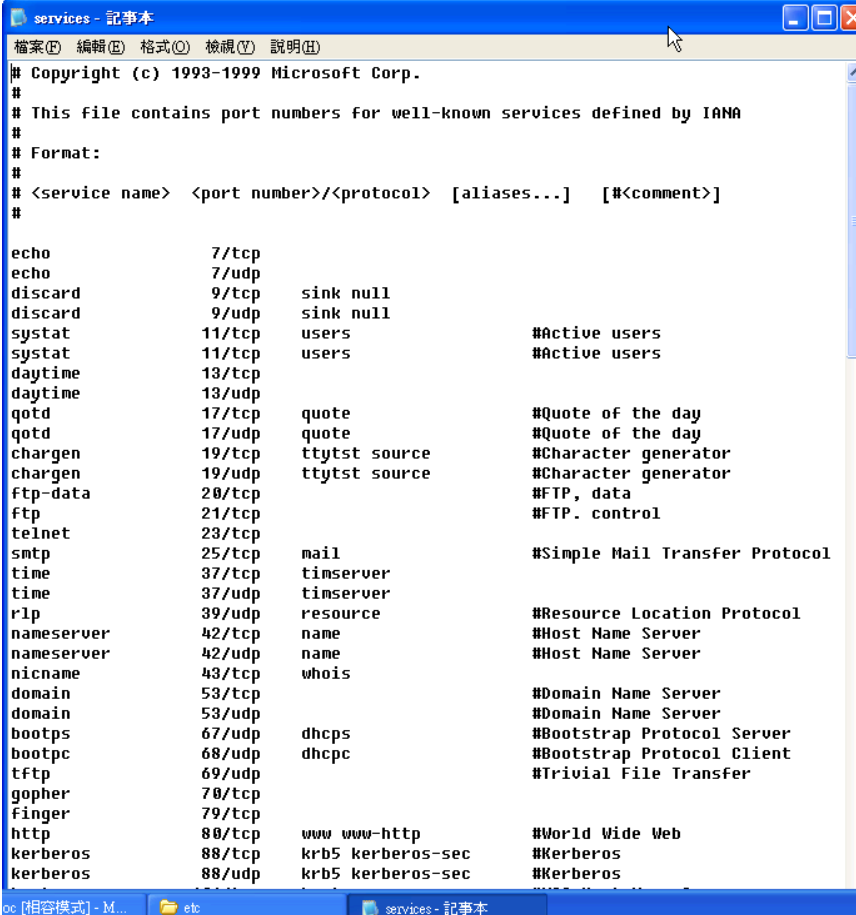
# 掃描的類型

- 網路的掃描（Network Scanning）
- Port 的掃描（Port Scanning）
- 掃描各主機所使用的作業系統
- 弱點的掃描（Vulnerability Scanning）

# Well-Known Port

- Well-Known Port就是我們所熟知的一些Port，例如Port 80就是http，port 443也就是https。
- 在各個作業系統中都有定義Well-Known Port，例如在Window XP作業系統中，Well-Known Port的號碼與服務對應就放在C:\WINDOWS\System32\drivers\etc目錄下有一個services檔案，使用Notepad來開啟該檔案。

# Well-Known Port



The screenshot shows a Notepad window titled "services - 記事本" (services - Notepad). The window contains the following text:

```
# Copyright (c) 1993-1999 Microsoft Corp.
#
# This file contains port numbers for well-known services defined by IANA
#
# Format:
#
# <service name> <port number>/<protocol> [aliases...] [#<comment>]
#
echo          7/tcp
echo          7/udp
discard      9/tcp      sink null
discard      9/udp      sink null
sysstat      11/tcp      users          #Active users
sysstat      11/tcp      users          #Active users
daytime      13/tcp
daytime      13/udp
qotd         17/tcp      quote          #Quote of the day
qotd         17/udp      quote          #Quote of the day
chargen      19/tcp      ttytst source  #Character generator
chargen      19/udp      ttytst source  #Character generator
ftp-data     20/tcp
ftp          21/tcp      #FTP, data
ftp          21/tcp      #FTP, control
telnet       23/tcp
smtp         25/tcp      mail           #Simple Mail Transfer Protocol
time         37/tcp      tinserver
time         37/udp      tinserver
rtp         39/udp      resource       #Resource Location Protocol
nameserver   42/tcp      name           #Host Name Server
nameserver   42/udp      name           #Host Name Server
nicname      43/tcp      whois
domain       53/tcp      #Domain Name Server
domain       53/udp      #Domain Name Server
bootps      67/udp      dhcps          #Bootstrap Protocol Server
bootpc      68/udp      dhcps          #Bootstrap Protocol Client
tftp        69/udp      #Trivial File Transfer
gopher       70/tcp
finger       79/tcp
http         80/tcp      www www-http   #World Wide Web
kerberos     88/tcp      krb5 kerberos-sec #Kerberos
kerberos     88/udp      krb5 kerberos-sec #Kerberos
```

# 掃描的步驟

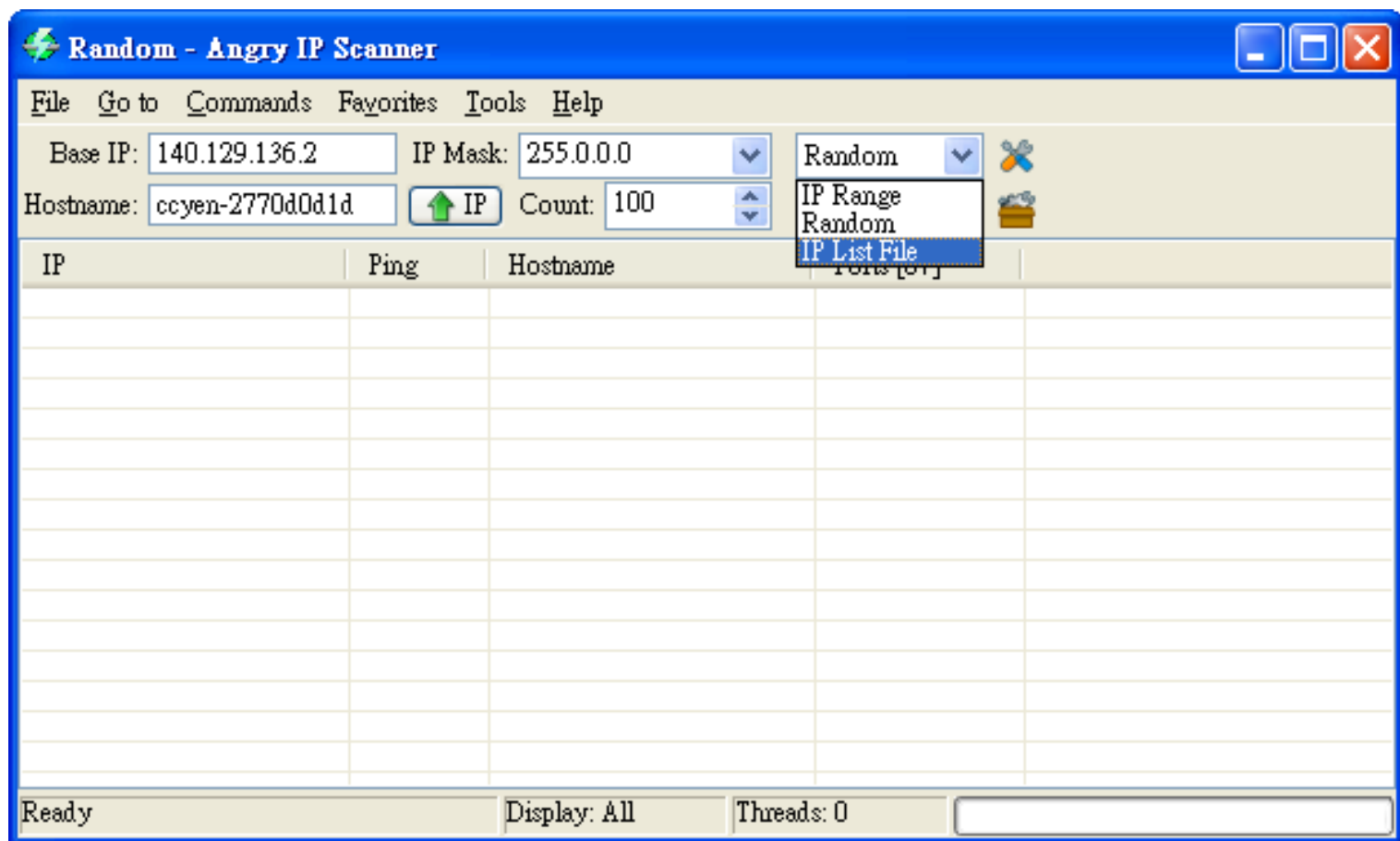
- 檢查活著的系統
- 檢查開啟的 Port
- 服務識別
- 作業系統的足跡（標誌抓取）
- 弱點掃描
- 弱點主機的網路圖
- 準備代理伺服器

# 檢查活著的系統

- 檢查主機是否活著，最常用的方法就是使用 ICMP 協定，ICMP 會送出需求封包，並等待目的地的主機回應。
- ICMP 可以同時進行多個主機是否存活的測試（這種平行測試稱之為 Ping-Sweep）。



# 工具程式：Angry IP Scanner



## 綠色軟體 (Greenware)

- 綠色軟體是免費的軟體，是一種在網路上以無條件的方式發放的小型軟體。
- 優點是檔案比較小、不用安裝、刪除方便和只佔用少量系統資源，甚至可以放在MP3播放器或USB記憶體中讀取。
- 大部分綠色軟體更是開放程式碼，並且不設權限，歡迎任何人參與修改程式或增加功能。

# 檢查開啟的Port

- TCP（傳輸層）在通訊時，在TCP的封包表頭中有多個旗標（Flag），這些Flag的用途是為了識別及控制的用途，每個Flag在封包中其實只佔用了1 bit。
- 因為先天流程上的缺陷，駭客可以利用這些Flag及不完整的程序，來進行主機或Port的掃描，卻不容易被目的地發覺。

# 檢查開啟的Port

這些TCP的旗標中，比較常見的包括：

- Synchronoze (SYN)：用來初始化主機間的連接。
- Acknowledgement (ACK)：用來確認主機間的連接。
- PUSH (PSH)：立即送出緩衝的資料。
- Urgent (URG)：封包中的資料必須立即被處理。
- Finish (FIN)：通知遠端系統傳輸結束。
- Reset (RST)：重新啟動連線。

# 檢查開啟的Port

- 網路伺服器開啟的服務，就是在某一個特定的TCP Port上傾聽（Listen），以等待使用者的連結，例如Http Server在Port 80上傾聽。
- 任何一個程式在某一個特定的Port上傾聽，等待使用者的連結，就稱這個Port的狀態為Open，反之則稱這個Port的狀態為Close。

# 檢查開啟的Port

## 「握手」基本程序

- 電腦A會先向電腦B先送出一個SYN，電腦B如果收到一個SYN，會回應一個SYN/ACK（這代表SYN及ACK都被設成1）。
- 電腦A收到電腦B發送的SYN/ACK封包，會再回應一個ACK。如此，兩台電腦間的連線動作就算完成。

### Three Way Handshake

```
192.168.1.2:2342-----Syn----->192.168.1.3:80
192.168.1.2:2342<-----Syn/Ack-----192.168.1.3:80
192.168.1.2:2342-----Ack----->192.168.1.3:80
```

連線建立

# 檢查開啟的Port

- 如果一個Port的狀態為Open，目的地主機會傳回一個SYN/ACK的封包。但是如果該Port的狀態為Close，則目的地會傳回一個RST的封包。
- 如果前述的「握手」不依照規定進行，會發生什麼狀況？

# 檢查開啟的Port

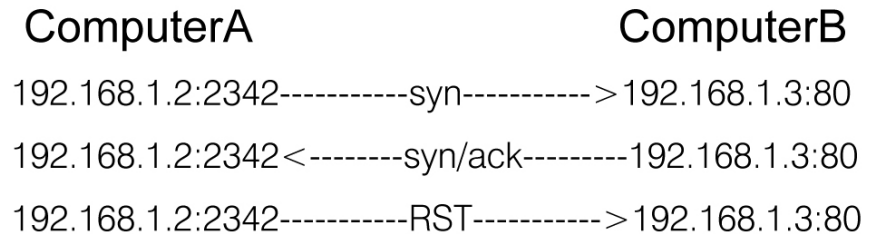
幾種利用不完整的握手程序達到掃描目的的方法：

- SYN Stealth
- Xmas Scan
- FIN Scan
- NULL Scan
- IDLE Scan
- TCP Connect
- RPC Scan



# SYN Stealth / Half Open Scan

這種掃瞄方式通常當作是半開掃描（half open scan），因為它不會開啟一個完整的TCP連線，圖中主機A代表攻擊者。



# SYN Stealth / Half Open Scan

- 首先A發送一個SYN封包送到目的主機B的一個Port，以要求一個連線，同時等待其回應。
- 假如B主機送回一個SYN/ACK封包給來源A，就代表這個Port的服務是在傾聽（而且這個主機B是活著的）。假如來源A只收到主機B的RST封包，就代表目的主機B的這個Port不是活著，或者不是在傾聽狀態。
- 當主機A若收到SYN/ACK封包時，一個RST封包會被主機A送出，這取代了原本應該送出的ACK封包，已終止這個連線，於是完整的連線並沒有成功。因為連線不成功，所以只有少數的目的地主機會記錄這個掃描連線動作。

```
ComputerA                ComputerB
192.168.1.2:2342-----syn----->192.168.1.3:80
192.168.1.2:2342<-----syn/ack-----192.168.1.3:80
192.168.1.2:2342-----RST----->192.168.1.3:80
```

# Xmas Scan

- 若一個關閉的Port收到諸如FIN、X-mas、*Null Scan*的封包，則必須回應一個RST封包，若是開放的Port，則會忽略這些封包。

ComputerA

ComputerB

Xmas scan directed at open port :

192.5.5.92:4031-----FIN/URG/PSH-----> 192.5.5.110:23

192.5.5.92:4031 <-----NO RESPONES-----192.5.5.110:23

Xmas scan directed at closed port :

192.5.5.92:4031-----FIN/URG/PSH-----> 192.5.5.110:23

192.5.5.92:4031 <----- RST/ACK -----192.5.5.110:23

# Xmas Scan

- 首先A發送一個FIN/URG/PSH封包，送到目的主機B的一個Port。
- 假設主機B的Port是開啟的，主機B應該不會有任何回應（由此判斷主機是活著）。
- 假設主機B的Port是關閉的，主機B會送回一個RST/ACK封包給來源A，就代表這個Port的服務是關閉的。

# FIN Scan

- 首先A發送一個FIN封包，送到目的主機B的一個Port。
- 假設主機B的Port是開啟的，主機B應該不會有任何回應（由此判斷主機是活著）。假設主機B的Port是關閉的，主機B會送回一個RST/ACK封包給來源A，就代表這個Port的服務是關閉的。

ComputerA

ComputerB

FIN scan directed at open port :

192.5.5.92:4031-----FIN----->192.5.5.110:23

192.5.5.92:4031 <-----NO RESPONES----- 192.5.5.110:23

FIN scan directed at closed port :

192.5.5.92:4031-----FIN----->192.5.5.110:23

192.5.5.92:4031 <----- RST/ACK -----192.5.5.110:23

# NULL Scan

- 首先A發送一個沒有任何Flag被設定的封包，送到目的主機B的一個Port。
- 假設主機B的Port是開啟的，主機B應該不會有任何回應（由此判斷主機是活著）。假設主機B的Port是關閉的，主機B會送回一個RST/ACK封包給來源A，就代表這個Port的服務是關閉的。

ComputerA

ComputerB

NULL scan directed at open port :

```
192.5.5.92:4031-----NO FLAGS SET----->192.5.5.110:23
192.5.5.92:4031<-----NO RESPONSES----- 192.5.5.110:23
```

NULL scan directed at closed port :

```
192.5.5.92:4031-----NO FLAGS SET----->192.5.5.110:23
192.5.5.92:4031<----- RST/ACK -----192.5.5.110:23
```

# IDLE Scan

- Idle Scan是一種非常難以察覺的Port Scanning的方法，攻擊者完全不需使用真實的IP來傳送封包給被攻擊的目標主機，而IDS（Intrusion Detection System）也會誤以為無辜的僵屍電腦才是攻擊的來源。

# IDLE Scan

## 殭屍電腦（Zombie computer）

- 簡稱「殭屍（zombie）」，有些人稱之為「肉雞」，指的是一部已經被駭客、電腦病毒、或木馬入侵的電腦，殭屍電腦通常都是連上網際網路。
- 殭屍電腦的真正擁有者一般都不會察覺到自己的電腦系統已經被「殭屍化」，所以暗喻這些電腦與電影情節中的殭屍無異。

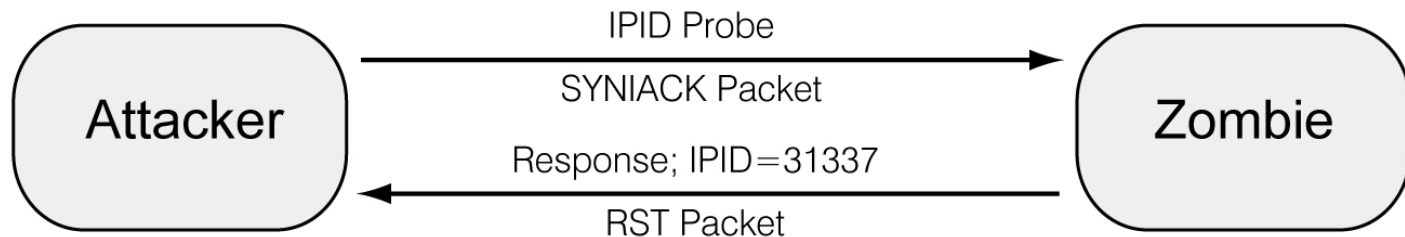


# IDLE Scan

- 每一個IP封包都有一個IPID欄位，大部分的作業系統針對這個欄位只是循序的編號。
- 任何電腦若只收到SYN/ACK將會回覆一個RST（因為沒發出SYN），但是同時這封包會包含這台主機的IPID，而每次的SYN/ACK連線都會讓IPID會加1。

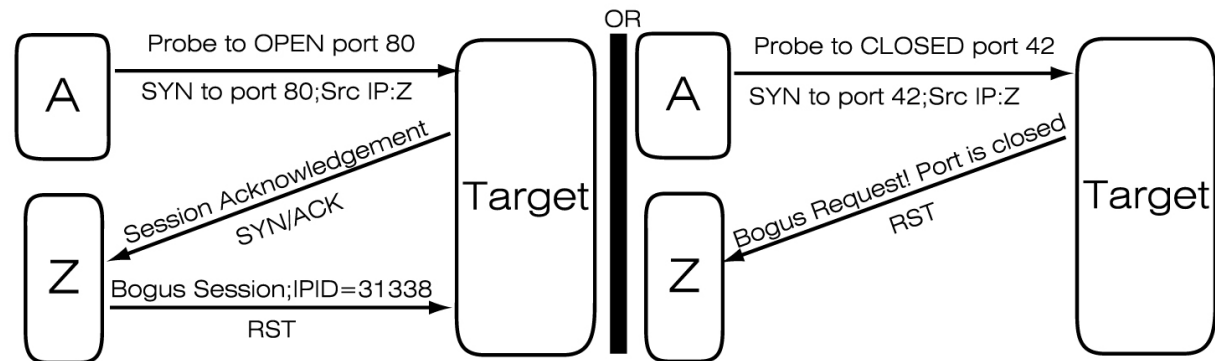
# IDLE Scan

- 1. 攻擊者的主機A先選擇一台殭屍電腦Z，同時送出一個SYN/ACK封包，這時送回來的RST封包會包含IPID，假設是31337。



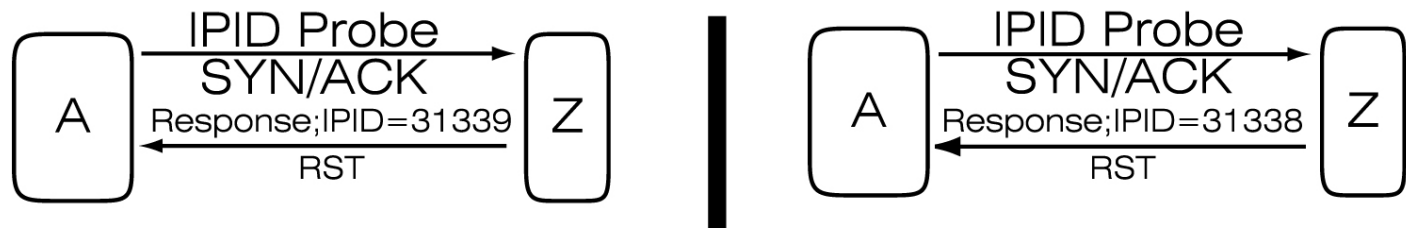
# IDLE Scan

- 2. 接下來攻擊者送出一個SYN封包給目標主機，但是來源IP偽造成Z的IP，若目標主機有回應，將會回應給Z。目標主機的Port如果是開啟的，就會回應一個SYN/ACK給Z，Z收到這個SYN/ACK，就會將IPID加1成為31338，並且送回一個包含IPID的RST給目標主機，如下圖左。目標主機的Port如果是關閉的，目標主機就會回應一個RST給Z。



# IDLE Scan

- 3. 主機A再送出一個SYN/ACK給Z，如果前面的偵測中目標主機的Port是開啟的，主機Z回應的應該是31338加1的結果，也就是31339。如果主機A收到的是31338，就代表這個Port在之前的測試中沒有被加1，也就是沒有開啟。



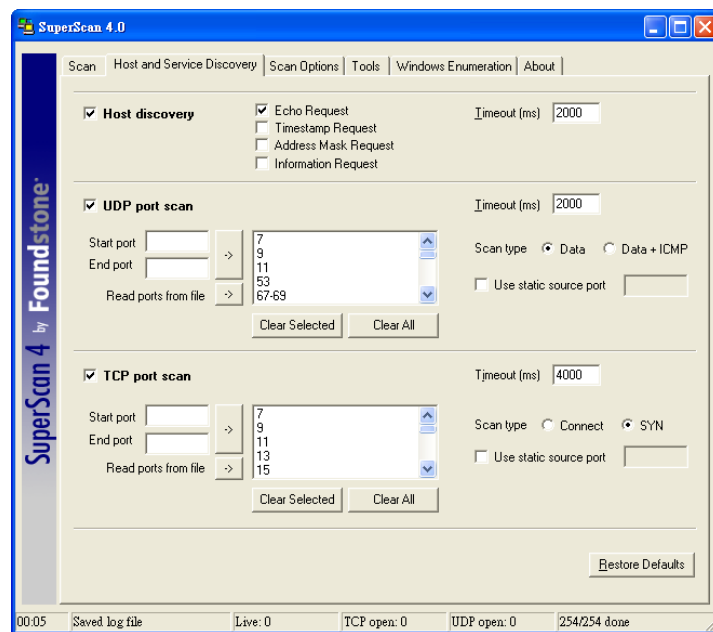
# Port Scan 工具程式

- NMAP
- NMAP 是一個開程式碼的工具，目前有 UNIX (Linux) 版及 Windows 版 (稱為 Zenmap GUI)，此工具程式被設計用來快速掃描大範圍的網路主機，並且可以使用前面敘述的特殊方法來進行偵測，這些方法大多數是為了讓目標主機無法察覺曾經被掃描過。

# Port Scan工具程式

## ■ SuperScan

- SuperScan具有以下功能：
  1. 透過Ping檢驗IP是否在線上。
  2. IP和領域名稱相互轉換。
  3. 檢驗目標電腦開啟的 Port。
  4. 檢驗目標電腦提供的服務。



# Port Scan工具程式

## ■ MegaPing

- MegaPing為一付費軟體，但具有三十天試用期，是個多功能網路診斷工具，它包含了許多實用的測試工具，除了能監控本機的各项訊息，也能使用ping及traceroute功能，也能掃描主機的Port及網段內所有未進行安全保護的共享資源和可見資源，例如：它可以掃描到其他網段內的共享印表機及共享文件夾。

# 服務識別

- 「服務識別」即檢視在某個Port上所執行的服務。
- 通常執行完Port掃描後，接著就會進行「服務識別」。所以一般都會將「Port掃描」與「服務識別」設計成同一組工具程式。



# 標誌抓取

- 標誌抓取就是使用方法去擷取目標系統的指紋（Fingerprinting），以判斷目標主機是使用哪種作業系統，通常是有價值的目標才會進行此步驟。
- `telnet xxx.xxx.xxx.xxx 110` 或 `telnet xxx.xxx.xxx.xxx 25`

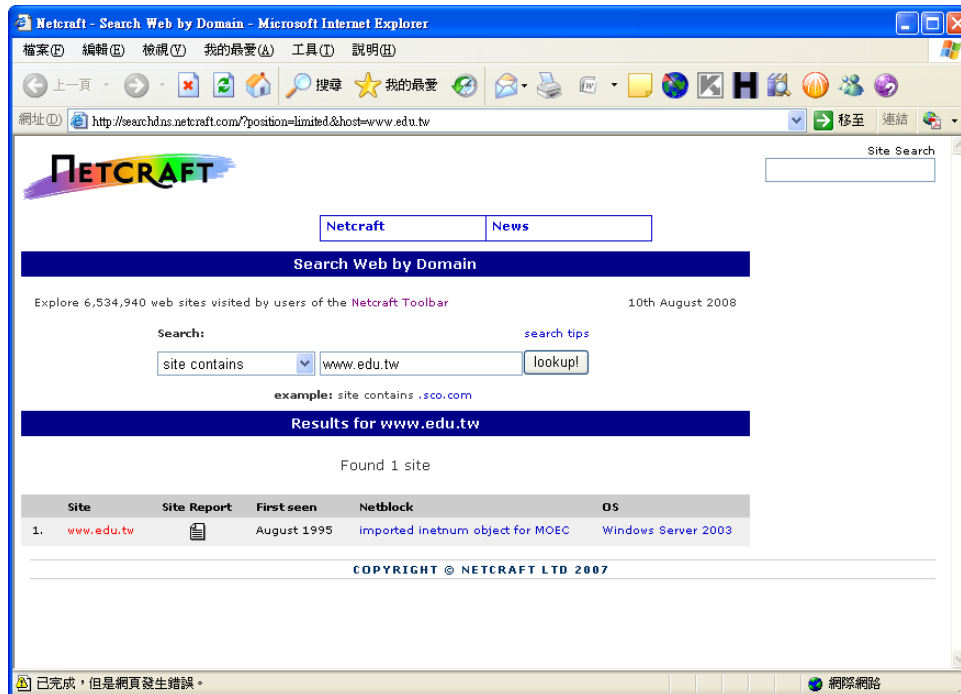
# 系統指紋

獲得方法：

- 主動式：若作業系統不同，則其系統中實作的TCP堆疊形態也就不同。只要取得TCP堆疊形態，再與資料庫進行比對，即可判定目標主機是哪一種作業系統。
- 被動式：使用監聽（sniffing）技術取代掃描技術，較不精確。

# Netcraft

- 此網頁可被動式的識別某台主機不同時期使用的作業系統。



# 對策

- 關閉該服務的標誌（Banner）訊息。
- Apache Server 2.x：修改httpd.conf檔案，加入Header set Server “Banner 的名稱”。
- IIS Server：使用工具程式IIS Lockdown Tool 或ServerMask。

# 弱點掃描

- **Nessus** (<http://zh-tw.tenable.com/>)
- Nessus是弱點掃描器，能檢查系統或軟體的bug或漏洞，是管理者非常有用的工具程式，但是駭客也可以拿來危害安全。
- Nessus有Windows及Linux兩種版本，採用Plug-in架構及NASL，同時可以測試無限制的主機數，並使用Client-server架構及即時更新的資料庫。

# 工具程式

## ■ Friendly Pinger

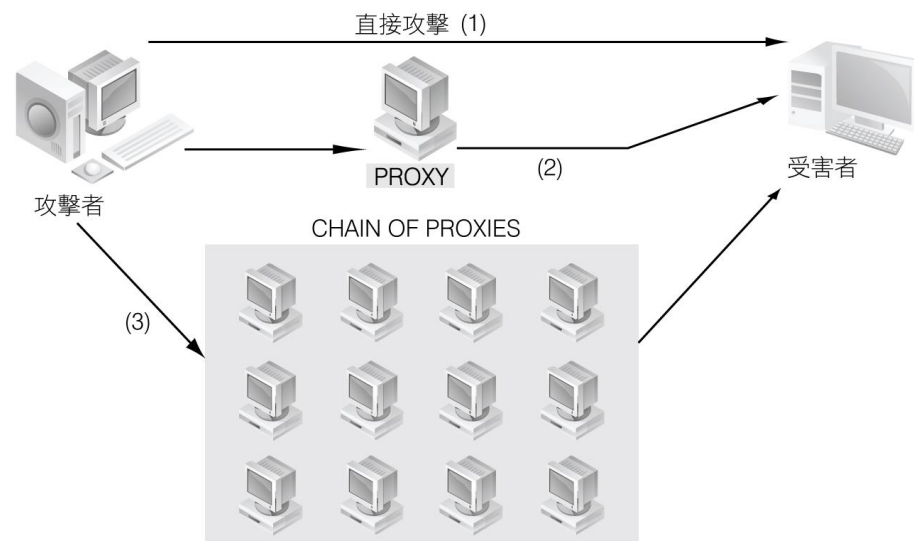
- 有30天的試用版可以使用。Friendly Pinger是圖形化的網路監督與管理工具，可以利用外部的命令同時Ping所有定義的裝置，並且以圖形化直覺的方式顯示。

# 準備代理伺服器

- 攻擊前的最後一個步驟，攻擊者通常不會用自己的主機直接攻擊目標，而會利用代理伺服器（Proxy Servers）。
- 代理伺服器通常用於以下之目的：
  1. 當做Firewall（防火牆）。
  2. 當做IP位址的多工器。
  3. 可以過濾掉不想要的內容。

# 代理伺服器

- 攻擊者可以透過單一的代理伺服器進行攻擊。攻擊者也可以建立一串的代理伺服器，這樣將會讓追蹤更困難。

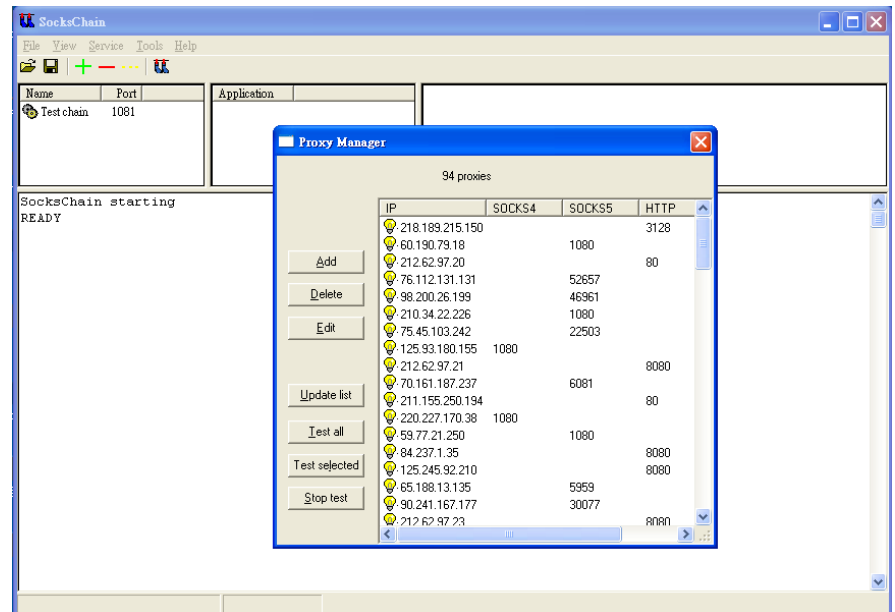




# 工具程式

## ■ SocksChain

- SocksChain是一支能讓任何的網際網路服務透過一串的SOCKS或HTTP代理伺服器以隱藏真實的IP。



# 工具程式

## ■ HTTPort

- HTTPort (client) 及 HTTPHost (server) 是免費的工具，可以用來將任何的TCP流量透過HTTP協定建立通道 (tunnel)。

# 通道（Tunneling）技術

- 許多公司或單位會使用防火牆或路由器將大部分的Port關閉，只留下HTTP（Port 80）及HTTPS（Port 443），如果想要使用遠端伺服器上的其他的服務（例如FTP），可以使用HTTP通道技術，將資料由HTTP協定送出。Httpunnel可建立雙向的虛擬連線，使用者若被限制在防火牆後面，仍然可以突破。

# 對策

- 適當的安全架構，例如裝設防火牆、IDS或IPS。
- 只開啟有必要的Port，其他的Port都關閉。可以利用前述的工具進行自我檢查。
- 敏感資訊不要揭露於網路上，想辦法不顯示。
- 加強組織成員在系統使用的教育訓練。