7. 主動資訊偵搜

勞動部 產業人才投資計畫
中國文化大學 推廣教育部
張耀鴻 副教授
2022年 暑期班

綱要

- DNS列舉
- 通訊埠掃描
- SMB列舉
- NFS列舉
- SMTP列舉
- SNMP列舉

DNS列舉

- 網域名稱系統 (Domain Name System, DNS)是是一個分散式資料庫，負責將使用者友善的網域名稱轉換為 IP 地址
- Internet的分層結構從頂級的根區域開始解析主機名稱（如 www.megacorpone.com）
- 當主機名輸入瀏覽器或其他應用程式時，瀏覽器將主機名稱傳給作業系統的 DNS client端
- 然後作業系統再將請求轉發給它配置使用的外部 DNS 伺服器

DNS recursor

- 名稱解析過程中的第一個伺服器稱為 DNS recursor，負責與 DNS 基礎設施互動並將結果傳回給 DNS 客戶端
- DNS recursor聯繫 DNS 根區域中的一台伺服器，然後根伺服器以負責包含頂層網域 (Top Level Domain, TLD) 區域的伺服器位址進行回應，在本例中的TLD為 .com
- 一旦 DNS recursor收到 TLD DNS 伺服器的位址，就會查詢它以獲取 megacorpone.com 網域的授權名稱伺服器的位址

Authoritative Name Server (ANS)

- 授權名稱伺服器(ANS)是 DNS 查找流程的最後一步，它將DNS記錄儲存在稱為區域檔案(zone file)的本地資料庫中
- 通常為每個網域託管兩個zone，用於查找特定主機名稱的 IP 地址的正向查找zone和用於查找特定 IP 地址的主機名稱的反向查找zone
- 一旦 DNS recursor向 DNS 客戶端提供 www.megacorpone.com 的 IP 地址，瀏覽器就可以透過其 IP 地址聯繫正確的 Web 伺服器並載入網頁

DNS Cache

- 為了提高 DNS 的效能和可靠性，DNS 快取在查找過程的各個階段儲存了 DNS 記錄的本地副本
- 正是這個原因，一些現代應用程式（例如 Web 瀏覽器）保留了單獨的 DNS 快取
- 此外，作業系統的本地 DNS 客戶端還與查找流程中的每個 DNS 伺服器一起維護自己的 DNS 快取
- 網域所有者還可以透過 DNS 記錄的生存時間 (Time to Live, TTL) 欄位來控制伺服器或客戶端快取 DNS 記錄的時間

與 DNS 伺服器互動

- 每個網域可以使用不同類型的 DNS 記錄
- 最常見的 DNS 記錄類型包括：
  - ✓ NS-Nameserver 記錄包含託管網域 DNS 記錄的授權伺服器的名稱。
  - ✓ A – 也稱為主機記錄，“A 記錄”包含主機名稱的 IP 位址（例如 www.megacorpone.com）。
  - ✓ MX – 郵件交換(Mail Exchange)記錄包含負責處理網域電子郵件的伺服器的名稱。一個網域可以包含多個 MX 記錄
  - ✓ PTR – 指標記錄(Pointer Record)用於反向查找區域，用於查找與 IP 位址關聯的記錄。
  - ✓ CNAME – 規範名稱(Canonical Name)記錄用於為其他主機記錄創建別名。
  - ✓ TXT-文字(Text)記錄可以包含任意資料，並且可以用於各種目的，例如網域所有權驗證。

用 host 命令查找 IP 位址

- 由於 DNS 中包含大量資訊，因此它通常是主動資訊收集的有利可圖的目標。
- 為了證明這一點，我們將使用 host 命令查找www.megacorpone.com 的 IP 位址：

```
└─$ host www.megacorpone.com
www.megacorpone.com has address 149.56.244.87
```

- host 命令預設會查找 A 記錄，但我們也可以用-t選項來查詢其他欄位，例如 MX 或 TXT 記錄：

```
└─$ host -t mx megacorpone.com
megacorpone.com mail is handled by 10 fb.mail.gandi.net.
megacorpone.com mail is handled by 50 mail.megacorpone.com.
megacorpone.com mail is handled by 20 spool.mail.gandi.net.
megacorpone.com mail is handled by 60 mail2.megacorpone.com.

┌──(kali㉿kali)-[~]
└─$ host -t txt megacorpone.com
megacorpone.com descriptive text "Try Harder"
```

前向暴力查找 (Forward Lookup Brute Force)

- 暴力查找是一種嘗試錯誤法，旨在尋找有效資訊，包括網路伺服器上的目錄、使用者名稱和密碼組合，或者有效的 DNS 記錄
- 透過使用常見主機名稱的詞彙表，可以嘗試猜測 DNS 記錄並檢查回應中的有效主機名稱
- 首先建立一些可能的主機名稱清單：

```
└─$ cat list.txt
www
ftp
mail
owa
proxy
router
```

前向暴力查找（續）

- 下一步，可以用一行 Bash 命令來嘗試解析每個主機名稱：

```
└─$ for ip in $(cat list.txt); do host $ip.megacorpone.com; done
www.megacorpone.com has address 149.56.244.87
Host ftp.megacorpone.com not found: 3(NXDOMAIN)
mail.megacorpone.com has address 51.222.169.212
Host owa.megacorpone.com not found: 3(NXDOMAIN)
Host proxy.megacorpone.com not found: 3(NXDOMAIN)
router.megacorpone.com has address 51.222.169.214
```

- 透過這個簡化的詞彙表，可以發現“www·”、“mail”和"router"為有效主機名稱，而“ftp”、“owa”和“proxy”為無效主機名稱
- 更全面的詞彙表。這些單詞表可以使用“sudo apt install seclists”命令安裝到“/usr/share/seclists”目錄

反向暴力查找 (Reverse Lookup Brute Force)

- DNS 前向暴力查找列出一群 IP 位址所成的集合
- 仔細觀察可發現它們位於相同的大致範圍 (51.222.168.X) 內
- 如果 megacorpone.com 的 DNS 管理員為網域配置了 PTR 記錄，就可以透過反向查找掃描大致範圍以請求每個 IP 的主機名稱
- 以下命令利用迴圈掃描 IP 位址 51.222.169.200 到 51.222.169.255
- 並透過僅顯示不包含 “not found” 和"ip-51"的項目（使用 grep -v來過濾掉無效結果）：

```
└─$ for ip in $(seq 200 255); do host 51.222.169.$ip; done | grep -v "not found"
 |grep -v "ip-51"
208.169.222.51.in-addr.arpa domain name pointer admin.megacorpone.com.
209.169.222.51.in-addr.arpa domain name pointer beta.megacorpone.com.
210.169.222.51.in-addr.arpa domain name pointer fs1.megacorpone.com.
211.169.222.51.in-addr.arpa domain name pointer intranet.megacorpone.com.
212.169.222.51.in-addr.arpa domain name pointer mail.megacorpone.com.
```

DNS 區域傳輸

- 區域傳輸(Zone transfer)基本上是相關 DNS 伺服器之間的資料庫複製
- 其中區域檔從主要 DNS 伺服器複製到次要伺服器
- 區域檔包含為該區域配置的所有 DNS 名稱列表
- 區域傳輸應該只允許到授權的從 DNS 伺服器，但許多系統管理員錯誤地配置了他們的 DNS 伺服器，以致於任何請求 DNS 伺服器區域副本的人通常都能收到這個副本
- 這相當於把企業網路佈局交給了駭客，伺服器的所有名稱、位址和功能都可能被窺探

Zone transfer指令

- 成功的區域傳輸不會直接導致網路被破壞，不過它確實有助於破壞網路
- 執行區域傳輸的host命令語法如下：
  - ✓ host -l <domain name> <dns server address>
- 先找出DNS servers：

```
└─$ for ip in $(seq 1 9); do host ns$ip.megacorpone.com; done | grep -v "not fou
nd"
ns1.megacorpone.com has address 51.79.37.18
ns2.megacorpone.com has address 51.222.39.63
ns3.megacorpone.com has address 66.70.207.180
```

- 用host -l 試試每一個nameserver，結果發現第2個nameserver就可以允許區域傳輸，並為megacorpone.com 網域提供區域文件的完整轉存，提供IP 位址和相對應 DNS 主機名稱的方便列表！

```
└─$ host -l megacorpone.com ns1.megacorpone.com
Using domain server:
Name: ns1.megacorpone.com
Address: 51.79.37.18#53
Aliases:

Host megacorpone.com not found: 5(REFUSED)
; Transfer failed.

┌──(kali㉿kali)-[~]
└─$ host -l megacorpone.com ns2.megacorpone.com
Using domain server:
Name: ns2.megacorpone.com
Address: 51.222.39.63#53
Aliases:

megacorpone.com name server ns1.megacorpone.com.
megacorpone.com name server ns2.megacorpone.com.
megacorpone.com name server ns3.megacorpone.com.
```

取得給定網域的名稱伺服器

- megacorpone.com 網域只有少數可檢查的 DNS 伺服器
- 但是，一些較大的組織可能會託管許多 DNS 伺服器
- 或者我們可能希望針對給定網域中的所有 DNS 伺服器嘗試區域傳輸請求
- 要嘗試使用 host 命令進行區域傳輸，我們需要兩個參數：名稱伺服器位址和網域名稱，可以用以下命令取得給定網域的名稱伺服器：

```
└─$ host -t ns megacorpone.com | cut -d " " -f 4
ns1.megacorpone.com.
ns2.megacorpone.com.
ns3.megacorpone.com.
```

自動取得每個nameserver的zone transfer資料

- 更進一步，可以寫一個 Bash script來自動識別相關名稱伺服器並嘗試從每個名稱伺服器進行區域傳輸：

```
└─$ cat dns-axfr.sh
#!/bin/bash
if [ -z "$1" ]; then
  echo "[*] Simple Zone transfer script"
  echo "[*] Usage: $0 <domain name"
  exit 0
fi

for server in $(host -t ns $1 | cut -d " " -f4); do
  host -l $1 $server | grep "has address"
done

┌──(kali㉿kali)-[~]
└─$ chmod +x dns-axfr.sh

┌──(kali㉿kali)-[~]
└─$ ./dns-axfr.sh megacorpone.com
admin.megacorpone.com has address 51.222.169.208
beta.megacorpone.com has address 51.222.169.209
fs1.megacorpone.com has address 51.222.169.210
```

DNS列舉相關工具

- DNSRecon:
  - 用 Python 所撰寫的進階的現代 DNS 列舉腳本
  - -d 選項指定網域名稱
  - -t 指定要執行的列舉類型（例如區域傳輸）
- DNSenum:
  - 目的是儘可能收集一個網域的資訊
  - 能夠利用google或者字典檔猜測可能存在的網域名稱，以及對一個網段進行反向查詢
  - 可以查詢網站的主機位址資訊、網域名稱伺服器、mx record（郵件交換記錄）
  - 可在dns server上執行axfr請求，通過google script得到延伸的網域名稱的資訊（google hacking），計算C類位址並執行whois查詢，執行反向查詢等

DNSRecon

- 對 megacorpone.com 執行 dnsrecon，產生以下輸出：

```
└─$ dnsrecon -d megacorpone.com -t axfr
[*] Checking for Zone Transfer for megacorpone.com name servers
[*] Resolving SOA Record
[+]      SOA ns1.megacorpone.com 51.79.37.18
[*] Resolving NS Records
[*] NS Servers found:
[+]      NS ns1.megacorpone.com 51.79.37.18
[+]      NS ns3.megacorpone.com 66.70.207.180
[+]      NS ns2.megacorpone.com 51.222.39.63
[*] Removing any duplicate NS server IP Addresses ...
[*]
[*] Trying NS server 51.222.39.63
[+] 51.222.39.63 Has port 53 TCP Open
[+] Zone Transfer was successful !!
[*]      NS ns1.megacorpone.com 51.79.37.18
[*]      NS ns2.megacorpone.com 51.222.39.63
[*]      NS ns3.megacorpone.com 66.70.207.180
```

暴力破解其他主機名稱

- 接下來可嘗試用之前所建立用於前向查找的 list.txt 來暴力破解其他主機名稱
- -d 選項指定網域名稱
- -D 指定包含潛在子網域字串的檔名
- -t 指定要執行的列舉類型（在本例中為 “brt” 表示使用brute force方式）：

```
└─$ dnsrecon -d megacorpone.com -D ~/list.txt -t brt                    130 ✕
[*] Using the dictionary file: /home/kali/list.txt (provided by user)
[*] brt: Performing host and subdomain brute force against megacorpone.com ...
[+]      A www.megacorpone.com 149.56.244.87
[+]      A router.megacorpone.com 51.222.169.214
[+] 2 Records Found
```

DNSenum

- 以下針對 zonetransfer.me 網域執行dnsenum
- zonetransfer.com由 DigiNinja所擁有並專門允許區域傳輸

```
└─$ dnsenum zonetransfer.me
dnsenum VERSION:1.2.6

-----   zonetransfer.me   -----


Host's addresses:
__________________

zonetransfer.me.                         7200     IN    A        5.196.105.14


Name Servers:
______________

nsztm2.digi.ninja.                       10800    IN    A        34.225.33.2
nsztm1.digi.ninja.                       10800    IN    A        81.4.108.41
```

Exercise 7-1

1. 查找 megacorpone.com 網域的 DNS 伺服器
2. 寫一個Bash script，嘗試從 megacorpone.com 進行區域傳輸
3. 使用dnsrecon重做以上練習，嘗試從 megacorpone.com 進行區域傳輸

通訊埠掃描

- 通訊埠掃描(Port scanning)是檢查遠端主機上的 TCP 或 UDP 通訊埠的過程
- 目的是檢測目標上正在運行哪些服務以及可能存在哪些潛在攻擊向量
- 請注意，通訊埠掃描並非正常的使用者活動，在某些司法管轄區可能被視為非法。因此，未經目標網路所有者的直接書面許可，不得在實驗室外進行
- 了解通訊埠掃描的含義以及特定通訊埠掃描可能產生的影響至關重要

Port scanning的副作用

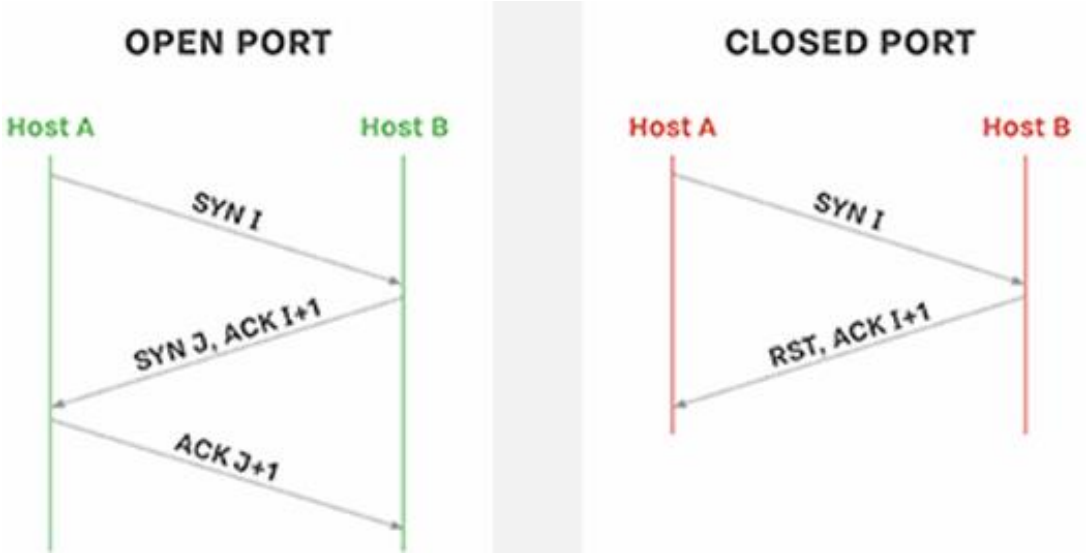
- 由於某些掃描會產生大量流量以及它們的侵入性，盲目地運行通訊埠掃描可能會對目標系統或客戶端網路產生不良影響
- 例如伺服器和網路連結過載或觸發 IDS
- 運行錯誤的掃描可能會導致客戶端當機
- 使用適當的通訊埠掃描方法可以顯著提高滲透測試的效率，同時還可以限制許多風險
- 根據參與的範圍，我們可以從僅掃描通訊埠 80 和 443 開始，而不是對目標網路運行所有通訊埠的掃描

通訊埠掃描為動態過程

- 有了可能的 Web 伺服器列表，我們可以對這些伺服器運行全通訊埠掃描在後台執行其他列舉
- 完成全通訊埠掃描後，我們可以進一步縮小掃描範圍，以便在每次後續掃描中探測越來越多的資訊
- 通訊埠掃描應被視為一個動態的過程，每個參與都是獨一無二的。一次掃描的結果決定了下一次掃描的類型和範圍

TCP / UDP scanning

- 最簡單的 TCP 通訊埠掃描技術通常稱為 CONNECT 掃描，依賴於 TCP 三次握手機制
- 這種機制的設計是為了讓試圖通訊的兩台主機可以在傳輸任何資料之前協商網路 TCP socket連接的參數
- 主機將 TCP SYN 封包發送到目標通訊埠上的伺服器
- 如果目的通訊埠是開放的，伺服器用一個 SYN-ACK 封包回應，客戶端主機則發送一個 ACK 封包來完成握手
- 如果握手成功完成，則通訊埠被認為是打開的。



Example: TCP port scanning

- 對metasploitable2通訊埠 78-80 運行 TCP Netcat 通訊埠掃描
- -w 選項指定連接超時 (以秒為單位)
- -z 用於指定Zero-I/O模式，該模式將不發送資料，只要掃描即可：

```
└─$ nc -nvv -w 1 -z 10.0.2.7 78-80
(UNKNOWN) [10.0.2.7] 80 (http) open
(UNKNOWN) [10.0.2.7] 79 (finger) : Connection refused
(UNKNOWN) [10.0.2.7] 78 (?) : Connection refused
 sent 0, rcvd 0
```

- 輸出結果顯示通訊埠 80 已打開，而通訊埠 78 和 79 上的連接超時

Example: TCP port scanning (續)

- 以下截圖顯示了Wireshark 捕獲這次port scanning的封包：

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	10.0.2.7	TCP	74	49626 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460
2	0.000978072	10.0.2.7	10.0.2.15	TCP	74	80 → 49626 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=
3	0.001009757	10.0.2.15	10.0.2.7	TCP	66	49626 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TS
4	0.001262343	10.0.2.15	10.0.2.7	TCP	66	49626 → 80 [FIN, ACK] Seq=1 Ack=1 Win=64256 Len
5	0.001640146	10.0.2.15	10.0.2.7	TCP	74	49412 → 79 [SYN] Seq=0 Win=64240 Len=0 MSS=1460
6	0.002666702	10.0.2.7	10.0.2.15	TCP	60	79 → 49412 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
7	0.003192352	10.0.2.7	10.0.2.15	TCP	66	80 → 49626 [FIN, ACK] Seq=1 Ack=2 Win=5824 Len=
8	0.003204794	10.0.2.15	10.0.2.7	TCP	66	49626 → 80 [ACK] Seq=2 Ack=2 Win=64256 Len=0 TS
9	0.003530760	10.0.2.15	10.0.2.7	TCP	74	59488 → 78 [SYN] Seq=0 Win=64240 Len=0 MSS=1460
10	0.004333919	10.0.2.7	10.0.2.15	TCP	60	78 → 59488 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
11	5.229070003	PcsCompu_…	PcsCompu_…	ARP	42	Who has 10.0.2.7? Tell 10.0.2.15
12	5.231202803	PcsCompu_…	PcsCompu_…	ARP	60	10.0.2.7 is at 08:00:27:4a:3f:e3

- 上圖顯示，Netcat 分別在第 1、5 和 9 行向通訊埠 80、79 和 78 發送了幾個 TCP SYN 封包
- 由於多種因素，包括時間問題，封包在 Wireshark 中可能出現亂序
- 請注意，伺服器在第 2 行從通訊埠 80 發送了一個 TCP SYN·ACK 封包，表示該通訊埠是開放的
- 其他通訊埠沒有回復類似的 SYN-ACK 封包，因此我們可以推斷它們沒有打開
- 最後，在第 8 行，Netcat 發送一個 FIN-ACK 封包關閉了這個連線

Example: UDP port scanning

- 由於 UDP 是無狀態的並且不涉及三次握手，因此 UDP 通訊埠掃描背後的機制與 TCP 不同
- 對不同目標上的通訊埠 68-70 運行 UDP Netcat 通訊埠掃描（-u）：

```
└─$ nc -nv -u -z -w 1 10.0.2.7 68-70
(UNKNOWN) [10.0.2.7] 69 (tftp) open
(UNKNOWN) [10.0.2.7] 68 (bootpc) open
```

Example: UDP port scanning (續)

- 用Wireshark截取UDP scan封包如下:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	10.0.2.7	UDP	43	57564 → 70 Len=1
2	0.000930490	10.0.2.7	10.0.2.15	ICMP	71	Destination unreachable (Port unreachable)
3	1.005242737	10.0.2.15	10.0.2.7	TFTP	43	45266 → 69 Len=1[Malformed Packet]
4	2.009992140	10.0.2.15	10.0.2.7	TFTP	43	45266 → 69 Len=1[Malformed Packet]
5	2.010550932	10.0.2.15	10.0.2.7	BOOTP	43	[Malformed Packet]
6	3.016703023	10.0.2.15	10.0.2.7	BOOTP	43	[Malformed Packet]

- 如上圖所示，空的 UDP 封包會被發送到特定通訊埠（封包1、3、4）
- 如果目標 UDP 通訊埠是開放的，封包將被傳到應用層，收到的回應將取決於應用程式如何回應空的封包
- 在本例中，應用程式不發送回應
- 但是，如果目標 UDP 通訊埠是關閉的，則目標應以無法存取(unreachable)的 ICMP 通訊埠進行回應（如封包 2 所示），該通訊埠由目標機器的 UDP/IP 堆疊發送。

常見的通訊埠掃描陷阱

- 出於多種原因，UDP 掃描可能會出現問題
- 首先，UDP 掃描通常不可靠，因為防火牆和路由器可能會丟棄 ICMP 封包
- 這可能導致誤報和通訊埠顯示為開放狀態，而實際上它們是關閉的
- 其次，許多通訊埠掃描器不會掃描所有可用通訊埠，並且通常具有預先設置要掃描的“感興趣的通訊埠”列表
- 這意味著開放的 UDP 通訊埠可能會被忽視
- 使用特定於協議的 UDP 通訊埠掃描器可能有助於獲得更準確的結果
- 最後，滲透測試人員經常忘記掃描開放的 UDP 通訊埠，而是專注於更令人興奮的 TCP 通訊埠
- 儘管 UDP 掃描可能不可靠，但開放的 UDP 通訊埠背後隱藏著大量攻擊vector

Nmap port scanning

- Nmap 是最常用、通用和強大的通訊埠掃描器之一
- 除了通訊埠掃描之外還具有許多功能
- 預設的 Nmap TCP 掃描將掃描給定機器上 1000 個最常用的通訊埠。在我們開始盲目地運行掃描之前，讓我們檢查一下這種掃描所產生的流量
- 我們將掃描其中metasploitable2機器，同時用 iptables 監控發送到目標主機的流量

iptables設定

- 我們將使用幾個 iptables 選項
- -I 選項將新規則插入給定鏈(chain)，在本例中，該鏈包括 INPUT（Inbound）和 OUTPUT（Outbound）鏈，後面跟著規則編號
- -s 指定來源 IP 位址
- -d 指定目標 IP 位址
- -j 來ACCEPT流量
- -Z 選項將所有鏈中的封包和位元組計數器歸零

```
└─$ sudo iptables -I INPUT 1 -s 10.0.2.7 -j ACCEPT
[sudo] password for kali:

┌──(kali㉿kali)-[~]
└─$ sudo iptables -I OUTPUT 1 -d 10.0.2.7 -j ACCEPT

┌──(kali㉿kali)-[~]
└─$ sudo iptables -Z
```

用nmap產生流量

- 掃描完成並顯示了一些開放的通訊埠

```
└─$ nmap 10.0.2.7
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-18 11:25 EDT
Nmap scan report for 10.0.2.7
Host is up (0.0024s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
```

iptables統計資料

- 看看 iptables 統計資料，可了解掃描產生了多少流量
- -v 選項表示要輸出詳細資訊
- -n 啟用數字輸出
- -L 列出所有鏈中存在的規則：

```
└─$ sudo iptables -vn -L
Chain INPUT (policy ACCEPT 8 packets, 4176 bytes)
 pkts bytes target     prot opt in     out     source               destination
 1010 42229 ACCEPT     all  --  *      *       10.0.2.7             0.0.0.0/0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 8 packets, 2237 bytes)
 pkts bytes target     prot opt in     out     source               destination
 1052 62708 ACCEPT     all  --  *      *       0.0.0.0/0            10.0.2.7
```

進一步掃描所有通訊埠

- 用“iptabes -Z”將所有鏈中的封包和位元組計數器歸零
- 並使用-p 運行另一個nmap 掃描來指定所有TCP通訊埠：

```
└─$ nmap -p 1-65535 10.0.2.7
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-18 11:36 EDT
Nmap scan report for 10.0.2.7
Host is up (0.028s latency).
Not shown: 65505 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
```

進一步掃描所有通訊埠（續）

- 探測所有 65535 個通訊埠，產生了大約 4 MB 的流量，明顯更高
- 但是，此完整通訊埠掃描發現了預設 TCP 掃描未找到的新通訊埠

```
└─$ sudo iptables -vn -L
[sudo] password for kali:
Chain INPUT (policy ACCEPT 90 packets, 51408 bytes)
 pkts bytes target     prot opt in     out     source               destination
66724 2688K ACCEPT     all  --  *      *       10.0.2.7             0.0.0.0/0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 90 packets, 27657 bytes)
 pkts bytes target     prot opt in     out     source               destination
66754 4004K ACCEPT     all  --  *      *       0.0.0.0/0            10.0.2.7
```

Stealth / SYN scanning

- Nmap 最常用的掃描技術是 SYN scan，或者叫做Stealth scan (隱身掃描)
- 使用 SYN 掃描有很多好處，因此這是在 nmap 命令中未指定掃描技術而且使用者具有raw socket權限時預設的掃描方式
- SYN 掃描是一種 TCP 通訊埠掃描方法，它涉及將 SYN 封包發送到目標機器上的各個通訊埠，而無需完成 TCP 握手
- 如果 TCP 通訊埠開放，則應從目標機器發回 SYN-ACK，通知我們該通訊埠已開放
- 此時，通訊埠掃描器不會費心發送最終的 ACK 來完成三次握手

執行SYN掃描

- 由於三次握手從未完成，因此資訊不會傳到應用層，因此不會出現在任何應用程式日誌中
- SYN 掃描也因此更快、更有效，因為發送和接收的封包更少

```
└─$ sudo nmap -sS 10.0.2.7
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-18 19:05 EDT
Nmap scan report for 10.0.2.7
Host is up (0.0035s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
```

TCP 連線掃描

- 當運行 nmap 的使用者沒有raw socket權限時，Nmap 預設會使用 TCP 連線掃描(TCP connect scan)技術
- 由於 Nmap TCP 連線掃描使用 Berkeley socket API 來執行三次握手，因此無需提升權限
- 但是，由於 Nmap 必須等待連線完成，API 才會傳回連線狀態，因此連線掃描比 SYN 掃描需要更長的時間才能完成
- 有時需要特別使用 nmap 執行連線掃描，例如，透過某些類型的代理進行掃描時

TCP 連線掃描執行例

- 執行nmap -sT 10.0.2.7

```
└─$ nmap -sT 10.0.2.7
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-18 19:34 EDT
Nmap scan report for 10.0.2.7
Host is up (0.014s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
```

UDP掃描

- 在執行 UDP 掃描時，Nmap 組合兩種不同方法的組合來確定通訊埠是開放還是關閉
- 對於大多數通訊埠，它使用前述的標準“ICMP port unreachable”方法，透過向給定通訊埠發送一個空封包
- 但是，對於常用的通訊埠，例如 SNMP 使用的通訊埠 161，則發送協定所專用的 SNMP 封包，以嘗試從綁定到該通訊埠的應用程式獲得回應

- 使用-sU選項來執行 UDP 掃描，並且需要 sudo 才能存取 raw socket
- Nmap會降低檢測速率，以避免目標機器丟棄的無用封包充斥網路，以致於掃描65535個通訊埠有時需要18 hrs以上

```
└─$ sudo nmap -sU 10.0.2.7
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-19 19:15 EDT
Nmap scan report for 10.0.2.7
Host is up (0.0026s latency).
Not shown: 993 closed udp ports (port-unreach)
PORT     STATE         SERVICE
53/udp   open          domain
68/udp   open|filtered dhcpc
69/udp   open|filtered tftp
111/udp  open          rpcbind
137/udp  open          netbios-ns
138/udp  open|filtered netbios-dgm
2049/udp open          nfs
MAC Address: 08:00:27:4A:3F:E3 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1071.53 seconds
```

UDP掃描（續）

- UDP 掃描 (-sU) 也可以跟 TCP SYN 掃描 (-sS) 選項結合使用，以建構更完整的目標圖：

```
└─$ sudo nmap -sS -sU 10.0.2.7
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-19 19:47 EDT
Nmap scan report for 10.0.2.7
Host is up (0.0025s latency).
Not shown: 993 closed udp ports (port-unreach), 977 closed tcp ports (reset)
PORT    STATE         SERVICE
21/tcp  open          ftp
22/tcp  open          ssh
```

```
Nmap done: 1 IP address (1 host up) scanned in 997.25 seconds
```

Network sweeping

- 對於大型網路上的主機，或者想要節省網路流量，可以嘗試使用網路掃描(Network Sweeping)技術探測目標
- 一開始先廣泛掃描，然後再對感興趣的主機使用更具體的掃描
- 使用 Nmap的-sn選項執行網路掃描時，主機探索過程不僅僅包括發送 ICMP 回應請求，還使用了其他幾個探測
- Nmap 還會向通訊埠443發送一個 TCP SYN 封包，向通訊埠80發送一個 TCP ACK 封包，並請求一個 ICMP 時間戳記以驗證主機是否可用

- nmap -sn 掃描結果：

```
└─$ nmap -sn 10.0.2.1-20
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-19 19:52 EDT
Nmap scan report for 10.0.2.1
Host is up (0.0025s latency).
Nmap scan report for 10.0.2.2
Host is up (0.0046s latency).
Nmap scan report for 10.0.2.7
Host is up (0.0020s latency).
Nmap scan report for 10.0.2.8
Host is up (0.0037s latency).
Nmap scan report for 10.0.2.10
Host is up (0.0033s latency).
Nmap scan report for 10.0.2.15
Host is up (0.00015s latency).
Nmap done: 20 IP addresses (6 hosts up) scanned in 14.45 seconds
```

nmap的-oG選項

- 在標準 nmap 輸出上使用 grep 命令搜尋即時機器可能很麻煩
- 可使用 Nmap 的"greppable" 輸出參數 -oG， 將這些結果儲存為更易於管理的格式：

```
└─$ nmap -v -sn 10.0.2.1-20 -oG ping-sweep.txt
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-19 19:56 EDT
Initiating Ping Scan at 19:56
Scanning 20 hosts [2 ports/host]
Completed Ping Scan at 19:56, 1.41s elapsed (20 total hosts)
Initiating Parallel DNS resolution of 6 hosts. at 19:56
Completed Parallel DNS resolution of 6 hosts. at 19:56, 13.01s elapsed
Nmap scan report for 10.0.2.1
Host is up (0.039s latency).
```

```
└─$ grep Up ping-sweep.txt | cut -d " " -f 2
10.0.2.1
10.0.2.2
10.0.2.7
```

掃描特定通訊埠

- 掃描整個網路特定的 TCP 或 UDP 通訊埠，可探測常見的服務，以找出可能有用的系統，或以其他方式存在已知漏洞的系統
- 這種掃描往往比 ping 掃描更準確：

```
└─$ nmap -p 80 10.0.2.1-20 -oG web-sweep.txt
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-19 20:03 EDT
Nmap scan report for 10.0.2.1
Host is up (0.0017s latency).

PORT   STATE  SERVICE
80/tcp closed http
```

```
└─$ grep open web-sweep.txt | cut -d " " -f2
10.0.2.2
10.0.2.7
10.0.2.8
10.0.2.10
10.0.2.15
```

OS Fingerprinting

- 使用 -O 選項可啟用Nmap 的 OS 指紋辨識(OS Fingerprinting)內建功能
- 此功能會檢查傳回的封包，來猜測目標的作業系統
- 由於作業系統的 TCP/IP 堆疊實作（例如不同的預設 TTL 值和 TCP window 大小）通常會稍微不一樣，而這些細微的差異給了 Nmap 可辨識的指紋
- Nmap 會檢查從目標機器接收到的流量，並嘗試將指紋與已知清單進行比對

```
└─$ sudo nmap -O 10.0.2.7
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-19 20:44 EDT
Nmap scan report for 10.0.2.7
Host is up (0.0022s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
```

Nmap Scripting Engine (NSE)

- Nmap 腳本引擎 (NSE)可啟動使用者自建的腳本，以自動執行各種掃描任務
- 這些腳本執行廣泛的功能，包括 DNS 列舉、暴力攻擊，甚至漏洞識別
- NSE 腳本位於 /usr/share/nmap/scripts 目錄
- 例如，smb-os-discovery 腳本嘗試連接到目標系統上的 SMB 服務並確定其作業系統

```
└─$ nmap 10.0.2.7 --script=smb-os-discovery
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-19 20:56 EDT
Nmap scan report for 10.0.2.7
Host is up (0.012s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
```

```
Host script results:
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: metasploitable
|   NetBIOS computer name:
|   Domain name: localdomain
|   FQDN: metasploitable.localdomain
|   System time: 2022-05-19T20:55:37-04:00
```

Example: NSE Zone Transfer

- 利用NSE進行DNS Zone Transfer:

```
└─$ nmap --script=dns-zone-transfer -Pn -p 53 ns2.megacorpone.com
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-19 21:08 EDT
Nmap scan report for ns2.megacorpone.com (51.222.39.63)
Host is up (0.21s latency).

PORT   STATE SERVICE
53/tcp open  domain
| dns-zone-transfer:
| megacorpone.com.          SOA  ns1.megacorpone.com. admin.megac
| megacorpone.com.          TXT  "Try Harder"
| megacorpone.com.          TXT  "google-site-verification=U7B_b0
```

Exercise 7-2

1. 使用 Nmap 對目標 IP 範圍進行 ping 掃描並將輸出儲存到檔案中，再用 grep 顯示在線(Up)的機器IP
2. (承上題)掃描在上一題中找到的 IP 位址以查找開放的網頁伺服器通訊埠。使用 Nmap 查找網頁伺服器和作業系統版本。
3. 使用 NSE 腳本掃描實驗環境中運行 SMB 服務的機器。
4. 使用 Wireshark 捕獲 Nmap TCP連線掃描並將其與 Netcat 通訊埠掃描進行比較。它們是相同的還是不同的？
5. 使用 Wireshark 捕獲 Nmap SYN 掃描並將其與 TCP連線掃描進行比較並確定它們之間的差異。

SMB列舉

- 伺服器訊息區塊 (SMB)協定由於其複雜的實現和開放性，多年來的安全記錄一直很差
- 包括從 Windows 2000 和 XP 中未經身份驗證的 SMB null session，到多年來的大量 SMB 錯誤和漏洞等
- 不過SMB 協定也隨著 Windows 作業系統版本更新而有所改進

掃描 NetBIOS 服務

- NetBIOS服務監聽 TCP 通訊埠 139 以及幾個 UDP 通訊埠
- 需要注意的是，SMB（TCP 通訊埠 445）和 NetBIOS 是兩個獨立的協定
- NetBIOS 是一個獨立的session層協定和服務，允許本地網路上的電腦相互通訊
- 雖然現在 SMB 的實作可以在沒有 NetBIOS 的情況下工作，但 TCP 上的 NetBIOS (NetBIOS over TCP, NBT) 是向後兼容所必需的，並且通常一起啟用
- 出於這個原因，這兩個服務的列舉通常是齊頭並進的。這些可以使用 nmap 進行掃描：

```
└─$ nmap -v -p 139,445 -oG smb.txt 10.0.2.1-20
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-20 00:07 EDT
Initiating Ping Scan at 00:07
Scanning 20 hosts [2 ports/host]
Completed Ping Scan at 00:07, 1.42s elapsed (20 total hosts)
```

Nmap SMB NSE Scripts

- Nmap 包含許多有用的 NSE scripts，可用於發現和列舉 SMB 服務
- 這些腳本可以在 /usr/share/nmap/scripts 目錄中找到：

```
└─$ ls -l /usr/share/nmap/scripts/smb*
-rw-r--r-- 1 root root  3753 Oct 26  2021 /usr/share/nmap/scripts/smb2-capabilities.nse
-rw-r--r-- 1 root root  2689 Oct 26  2021 /usr/share/nmap/scripts/smb2-security-mode.nse
-rw-r--r-- 1 root root  1408 Oct 26  2021 /usr/share/nmap/scripts/smb2-time.nse
-rw-r--r-- 1 root root  5269 Oct 26  2021 /usr/share/nmap/scripts/smb2-vuln-uptime.nse
-rw-r--r-- 1 root root 45138 Oct 26  2021 /usr/share/nmap/scripts/smb-brute.nse
```

Example: 使用 nmap SME執行作業系統探索

```
└─$ nmap -v -p 139,445 --script=smb-os-discovery 10.0.2.7
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-20 00:13 EDT
```

```
PORT    STATE SERVICE
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds

Host script results:
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: metasploitable
|   NetBIOS computer name:
|   Domain name: localdomain
|   FQDN: metasploitable.localdomain
|_  System time: 2022-05-20T00:12:50-04:00
```

Example：掃描已知漏洞

- 要檢查已知的 SMB 協定漏洞，我們可以調用“smb-vuln” NSE 腳本之一
- 例如 smb-vuln-ms08-067，它透過“--script-args”選項傳送參數到NSE 腳本

```
└─$ nmap -v -p 139,445 --script=smb-vuln-ms08-067 --script-args=unsafe=1 10.0.2.7
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-20 00:21 EDT
NSE: Loaded 1 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 00:21
Completed NSE at 00:21, 0.00s elapsed
```

Exercise 7-3

1. 使用 Nmap 列出實驗室中運行 Windows 的 SMB 伺服器。
2. 使用 NSE 腳本掃描這些系統中的 SMB 漏洞。

NFS列舉

- 網路檔案系統(Network File System, NFS) 通常是用在 UNIX 作業系統上，而且NFS的實作是不安全的
- 安全設置可能有些困難，因此發現 NFS 共享(NFS share)資訊向全世界開放的情況並不少見
- 這些共享資訊可用來收集敏感資訊、提升權限等

掃描 NFS 共享

- Portmapper 和 RPCbind 都在 TCP 通訊埠 111 上運行
- RPCbind 將 RPC 服務映射到它們所監聽的通訊埠
- RPC 程序在啟動時會通知 rpcbind，註冊它們正在監聽的通訊埠以及它們希望獲得服務的 RPC 程序編號
- 然後，客戶端系統使用特定的 RPC 程序編號聯繫伺服器上的 rpcbind
- rpcbind 服務將客戶端重定向到正確的通訊埠號（通常是 TCP 通訊埠 2049），因此它可以跟
- 所請求的服務溝通

使用rpcinfo

- 以下命令使用nmap掃描通訊埠111：

```
└─$ nmap -v -p 111 10.0.2.1-20
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-20 01:35 EDT
Initiating Ping Scan at 01:35
Scanning 20 hosts [2 ports/host]
Completed Ping Scan at 01:35, 1.32s elapsed (20 total hosts)
```

- 也可以用NSE 腳本" rpcinfo"來查找可能已向 rpcbind 註冊的服務：

```
└─$ nmap -sV -p 111 --script=rpcinfo 10.0.2.1-20
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-20 01:38 EDT
Nmap scan report for 10.0.2.1
Host is up (0.00094s latency).
```

SMTP列舉

- 易受攻擊的郵件伺服器也可以收集有關主機或網路的資訊
- 簡單郵件傳輸協定 (SMTP) 支援幾個有趣的命令，例如 VRFY 和 EXPN
- VRFY要求伺服器驗證電子郵件位址，而 EXPN 則要求伺服器提供郵件列表的成員資格
- 這些通常可以被濫用來驗證郵件伺服器上的現有使用者

```
└─$ nc -nv 10.0.2.7 25
(UNKNOWN) [10.0.2.7] 25 (smtp) open
220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
421 4.4.2 metasploitable.localdomain Error: timeout exceeded
```

SNMP列舉

- 這麼多年來，許多網路管理員還是沒有很好的理解簡單網路管理協定 (Simple Network Management Protocol, SNMP)
- 這通常會導致 SNMP 配置錯誤，進而導致大量資訊洩漏
- SNMP 是基於 UDP的簡單無狀態協定，因此容易受到 IP 欺騙和重放攻擊
- 此外，常用的 SNMP 協定 1、2 和 2c 不提供流量加密，這意味著可以透過本地網路輕鬆攔截 SNMP 資訊和機敏資訊
- 傳統SNMP 協定的身份驗證方式較不安全，並且通常配置有預設的公開和私有社群字串

SNMP MIB Tree

- SNMP 管理資訊庫 (MIB) 是一個包含通常與網路管理相關的資訊的階層式資料庫
- MIB資料庫的結構像樹一樣，其中分支機構代表不同的組織或網路功能
- 樹的葉子（最終端點）對應於特定的變數值，外部使用者可以存取和探測這些變數值
- IBM Knowledge Center包含有關 MIB 樹的大量資訊

Microsoft Windows SNMP 參數

- 以下 MIB 值對應於特定的 Microsoft Windows SNMP 參數，其中所包含的資訊遠不止基於網路的資訊：
  - ✓ 1.3.6.1.2.1.25.1.6.0 **System Processes**
  - ✓ 1.3.6.1.2.1.25.4.2.1.2 **Running Programs**
  - ✓ 1.3.6.1.2.1.25.4.2.1.4 **Processes Path**
  - ✓ 1.3.6.1.2.1.25.2.3.1.4 **Storage Units**
  - ✓ 1.3.6.1.2.1.25.6.3.1.2 **Software Name**
  - ✓ 1.3.6.1.4.1.77.1.2.25 **User Accounts**
  - ✓ 1.3.6.1.2.1.6.13.1.3 **TCP LocalPorts**

SNMP掃描

- 要掃描開放的 SNMP 通訊埠，可以運行 nmap
- -sU 選項用於執行 UDP 掃描
- --open 選項用於限制輸出僅顯示開放的通訊埠：

```
└─$ sudo nmap -sU --open -p 161 10.0.2.1-20 -oG open-snmp.txt
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-20 02:37 EDT
Nmap scan report for 10.0.2.2
Host is up (0.00085s latency).

PORT    STATE         SERVICE
161/udp open|filtered snmp
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)

Nmap done: 20 IP addresses (7 hosts up) scanned in 15.03 seconds
```

onesixityone example

- 也可以使用 onesixtyone 之類的工具，它會嘗試對 IP 位址列表進行暴力攻擊
- 首先，必須建構包含社群字串和我們希望掃描的 IP 位址的文字檔：

```
└─$ echo public > community

┌──(kali㉿kali)-[~]
└─$ echo private >> community

┌──(kali㉿kali)-[~]
└─$ echo manager >> community

┌──(kali㉿kali)-[~]
└─$ for ip in $(seq 1 20); do echo 10.0.2.$ip; done > ips

┌──(kali㉿kali)-[~]
└─$ onesixtyone -c community -i ips
Scanning 20 hosts, 4 communities
```

Summary

- 對於任何給定的情況，從來沒有一個“最好的”工具
- 特別是因為 Kali Linux 中的許多工具在功能上重疊
- 最好讓自己盡可能多熟悉一些工具，了解它們的細微差別
- 並且儘可能衡量結果以了解幕後發生的事情
- 在某些情況下，“最好的工具”是掌握在最有經驗的專業人士手上


END