



UNIVERSIDADE DA CORUÑA

La "trilogía"

LSI - 2019/2020

José Manuel Vázquez Naya
jose@udc.es

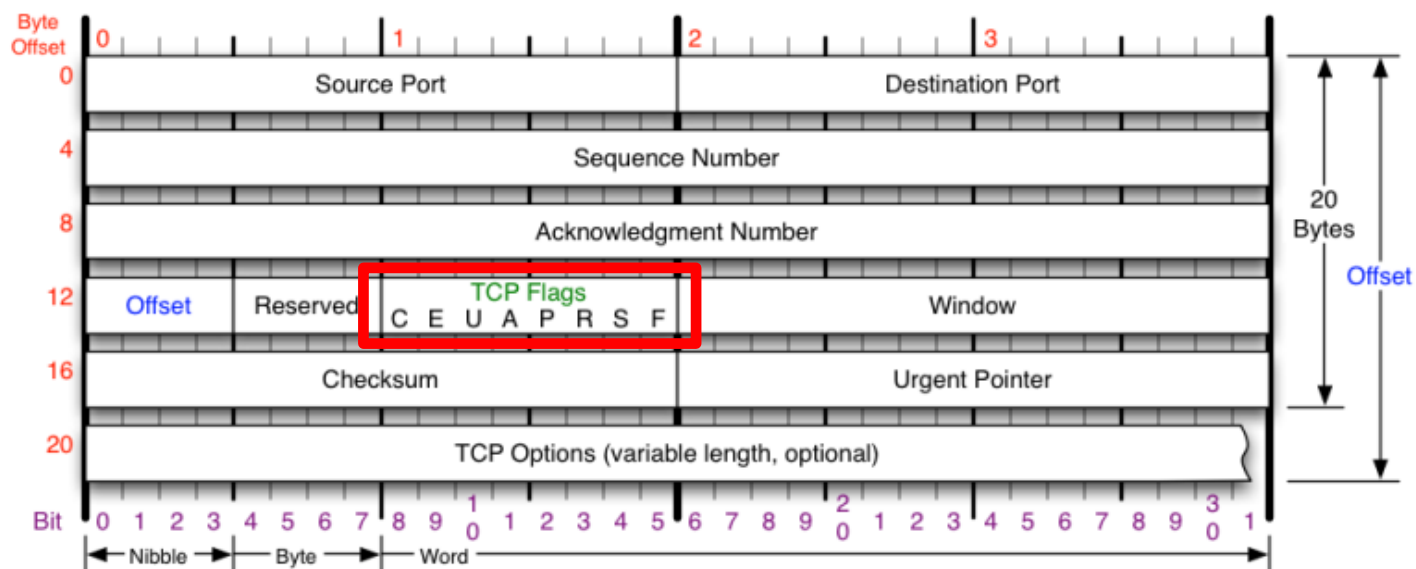
Contenido

- Fases previas de un ataque:
 - *Host discovery*
 - *Port scanning*
 - *Fingerprinting*



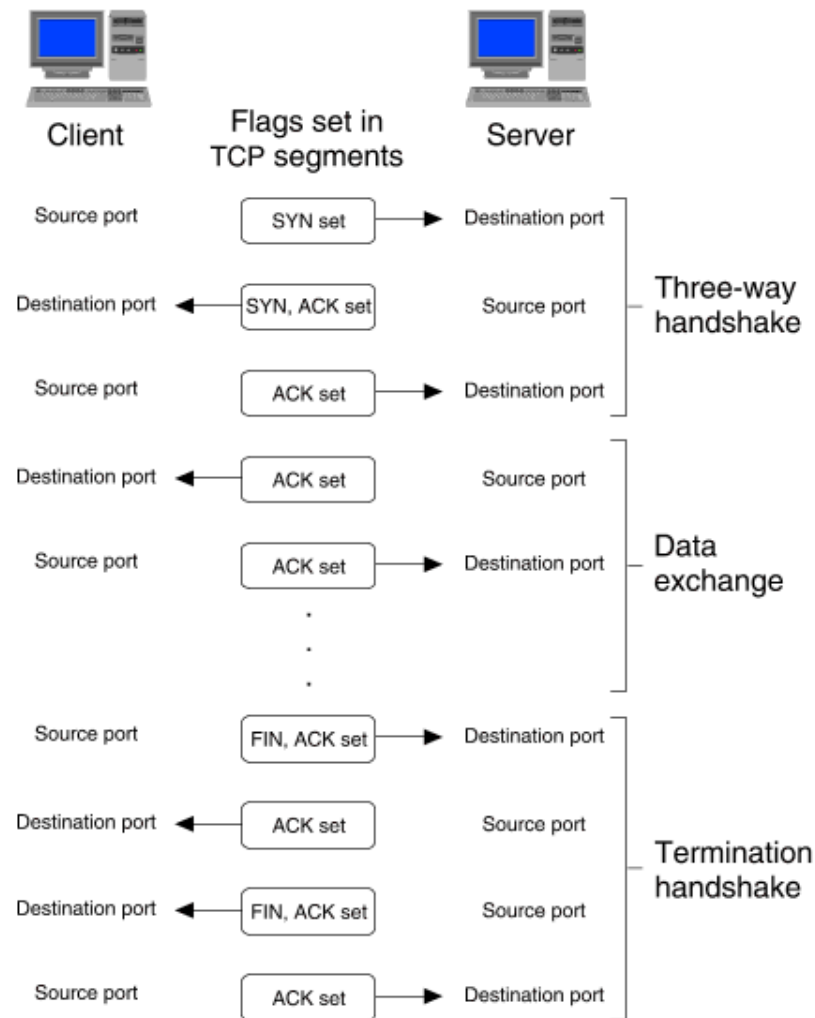
ALGUNOS CONCEPTOS DE REDES

Cabecera TCP

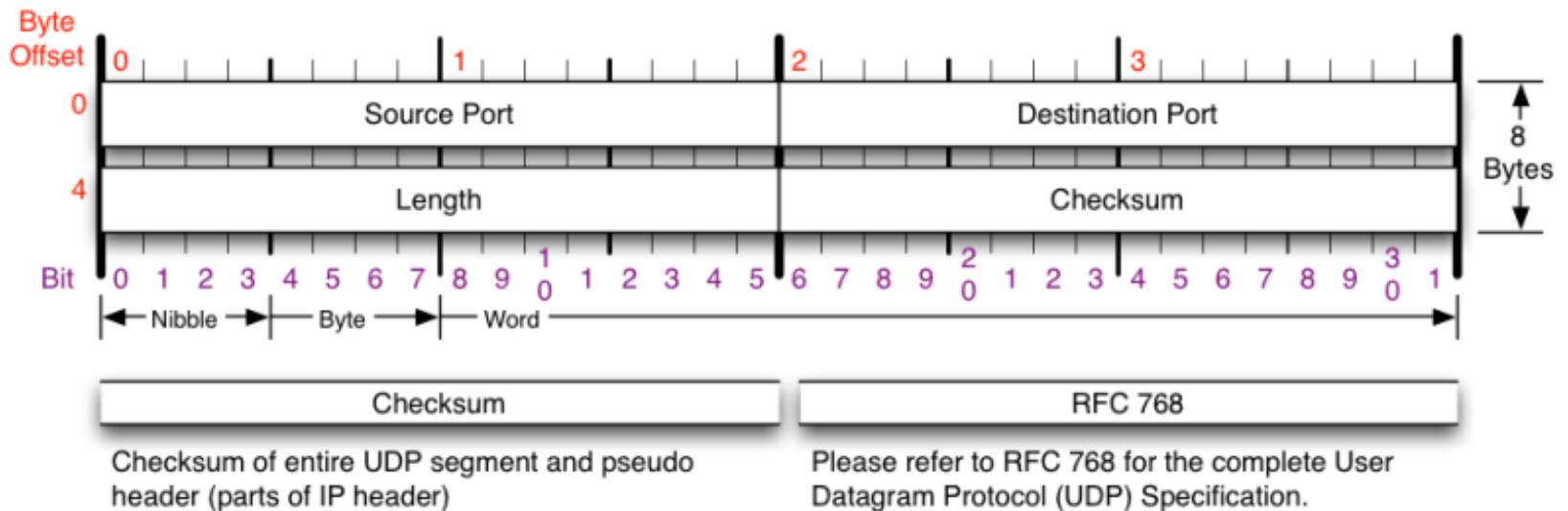


TCP Flags	Congestion Notification	TCP Options	Offset																											
C E U A P R S F	ECN (Explicit Congestion Notification). See RFC 3168 for full details, valid states below.	0 End of Options List 1 No Operation (NOP, Pad) 2 Maximum segment size 3 Window Scale 4 Selective ACK ok 8 Timestamp	Number of 32-bit words in TCP header, minimum value of 5. Multiply by 4 to get byte count.																											
Congestion Window C 0x80 Reduced (CWR) E 0x40 ECN Echo (ECE) U 0x20 Urgent A 0x10 Ack P 0x08 Push R 0x04 Reset S 0x02 Syn F 0x01 Fin	<table><tr><th>Packet State</th><th>DSB</th><th>ECN bits</th></tr><tr><td>Syn</td><td>0 0</td><td>1 1</td></tr><tr><td>Syn-Ack</td><td>0 0</td><td>0 1</td></tr><tr><td>Ack</td><td>0 1</td><td>0 0</td></tr><tr><td>No Congestion</td><td>0 1</td><td>0 0</td></tr><tr><td>No Congestion</td><td>1 0</td><td>0 0</td></tr><tr><td>Congestion</td><td>1 1</td><td>0 0</td></tr><tr><td>Receiver Response</td><td>1 1</td><td>0 1</td></tr><tr><td>Sender Response</td><td>1 1</td><td>1 1</td></tr></table>	Packet State	DSB	ECN bits	Syn	0 0	1 1	Syn-Ack	0 0	0 1	Ack	0 1	0 0	No Congestion	0 1	0 0	No Congestion	1 0	0 0	Congestion	1 1	0 0	Receiver Response	1 1	0 1	Sender Response	1 1	1 1	Checksum Checksum of entire TCP segment and pseudo header (parts of IP header)	RFC 793 Please refer to RFC 793 for the complete Transmission Control Protocol (TCP) Specification.
Packet State	DSB	ECN bits																												
Syn	0 0	1 1																												
Syn-Ack	0 0	0 1																												
Ack	0 1	0 0																												
No Congestion	0 1	0 0																												
No Congestion	1 0	0 0																												
Congestion	1 1	0 0																												
Receiver Response	1 1	0 1																												
Sender Response	1 1	1 1																												

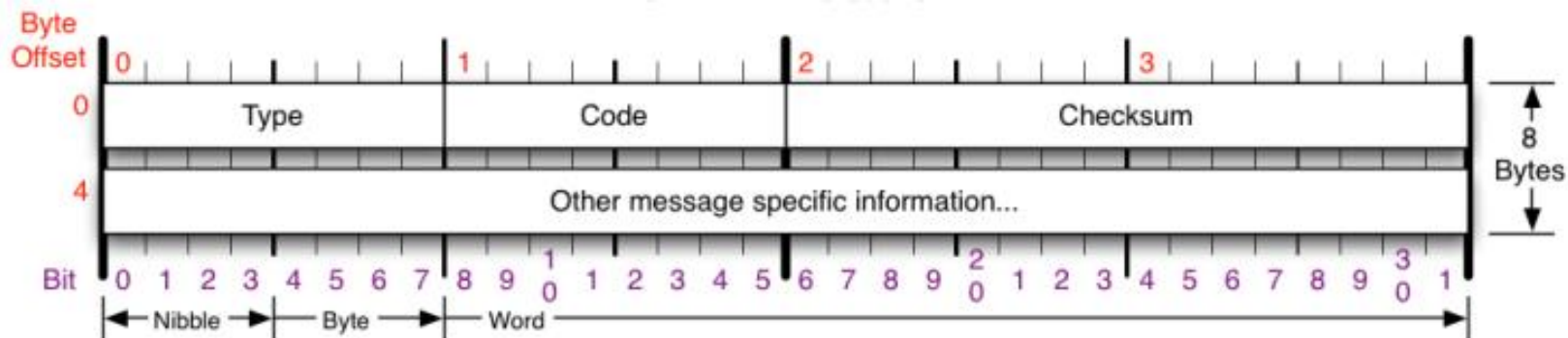
TCP handshake



Cabecera UDP



Cabecera ICMP



ICMP Message Types

Type Code/Name

- 0 Echo Reply
- 3 Destination Unreachable
 - 0 Net Unreachable
 - 1 Host Unreachable
 - 2 Protocol Unreachable
 - 3 Port Unreachable
 - 4 Fragmentation required, and DF set
 - 5 Source Route Failed
 - 6 Destination Network Unknown
 - 7 Destination Host Unknown
 - 8 Source Host Isolated
 - 9 Network Administratively Prohibited
 - 10 Host Administratively Prohibited
 - 11 Network Unreachable for TOS

Type Code/Name

- 3 Destination Unreachable (continued)
 - 12 Host Unreachable for TOS
 - 13 Communication Administratively Prohibited
- 4 Source Quench
- 5 Redirect
 - 0 Redirect Datagram for the Network
 - 1 Redirect Datagram for the Host
 - 2 Redirect Datagram for the TOS & Network
 - 3 Redirect Datagram for the TOS & Host
- 8 Echo
- 9 Router Advertisement
- 10 Router Selection

Type Code/Name

- 11 Time Exceeded
 - 0 TTL Exceeded
 - 1 Fragment Reassembly Time Exceeded
- 12 Parameter Problem
 - 0 Pointer Problem
 - 1 Missing a Required Operand
 - 2 Bad Length
- 13 Timestamp
- 14 Timestamp Reply
- 15 Information Request
- 16 Information Reply
- 17 Address Mask Request
- 18 Address Mask Reply
- 30 Traceroute

Checksum

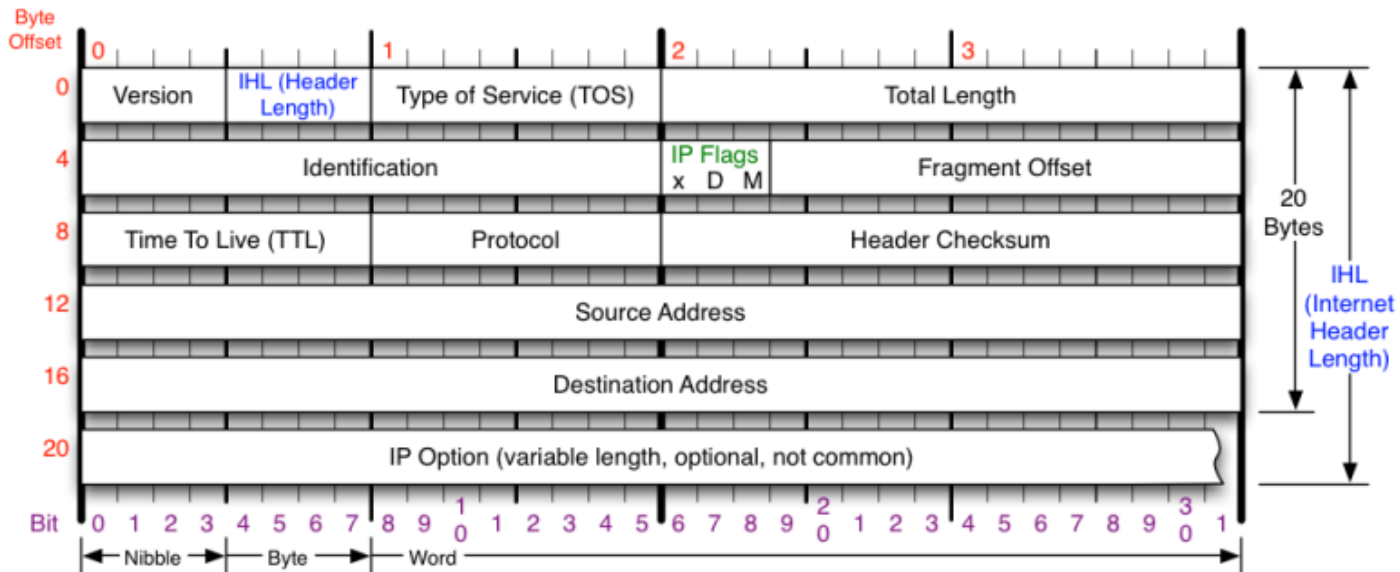
Checksum of ICMP header

RFC 792

Please refer to RFC 792 for the Internet Control Message protocol (ICMP) specification.



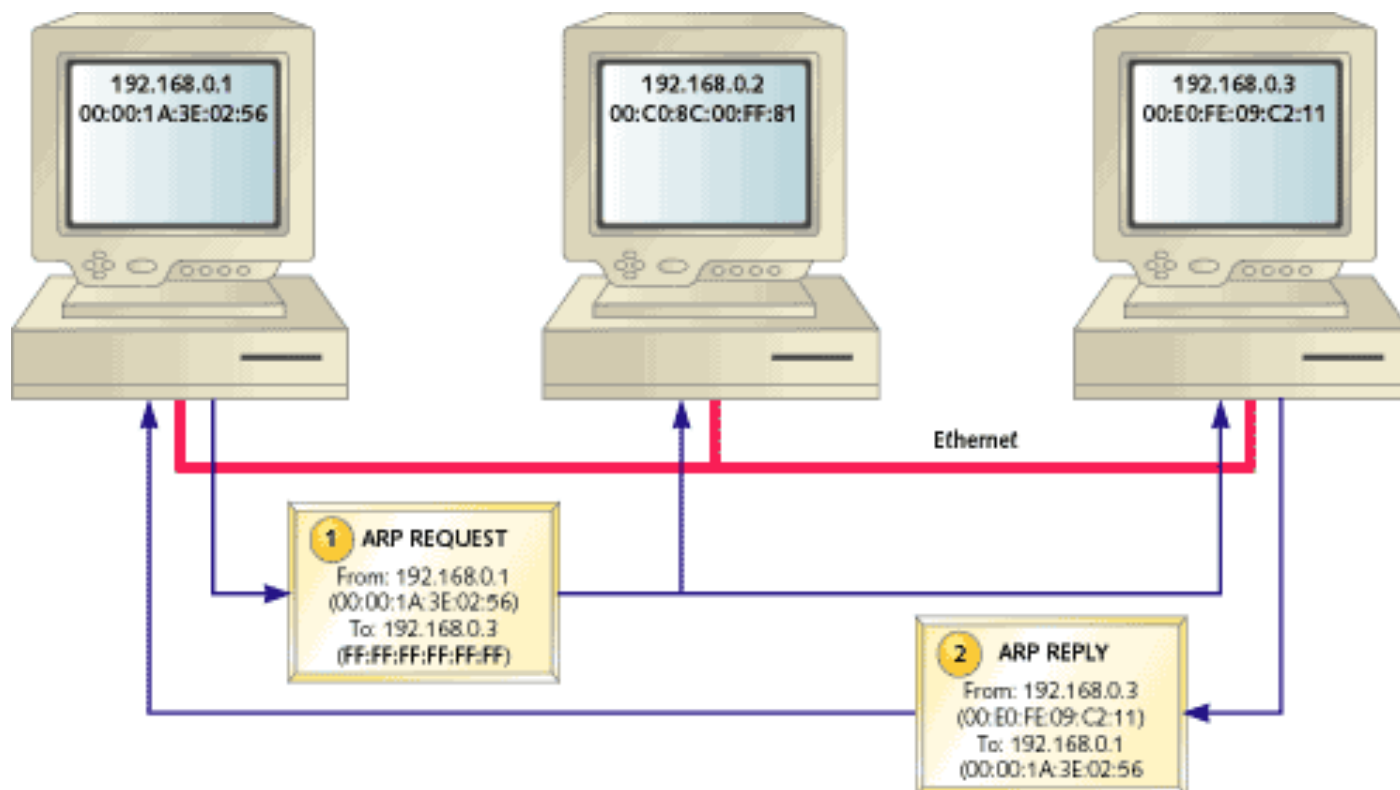
Cabecera IP



Version Version of IP Protocol. 4 and 6 are valid. This diagram represents version 4 structure only.	Protocol IP Protocol ID. Including (but not limited to): 1 ICMP 17 UDP 57 SKIP 2 IGMP 47 GRE 88 EIGRP 6 TCP 50 ESP 89 OSPF 9 IGRP 51 AH 115 L2TP	Fragment Offset Fragment offset from start of IP datagram. Measured in 8 byte (2 words, 64 bits) increments. If IP datagram is fragmented, fragment size (Total Length) must be a multiple of 8 bytes.	IP Flags x D M x 0x80 reserved (evil bit) D 0x40 Do Not Fragment M 0x20 More Fragments follow
Header Length Number of 32-bit words in TCP header, minimum value of 5. Multiply by 4 to get byte count.	Total Length Total length of IP datagram, or IP fragment if fragmented. Measured in Bytes.	Header Checksum Checksum of entire IP header	RFC 791 Please refer to RFC 791 for the complete Internet Protocol (IP) Specification.

ARP (*Address Resolution Protocol*)

- El protocolo de resolución de direcciones (ARP) es la forma en que las redes TCP/IP “averiguan” las direcciones MAC basándose en las direcciones IP de destino



HOST DISCOVERY

¿Qué es *host discovery*?

- Descubrir las máquinas que hay en una red (PCs, impresoras, routers, firewalls, dispositivos de almacenamiento, etc.)
- Es el primer paso
 - Una vez sabemos qué máquinas están “vivas”, podemos realizar “port scanning” sólo sobre éstas (o un subconjunto), en lugar de hacerlo sobre todo el rango.
 - Realizar *port scanning* sobre todas las IPs de un rango sería muy costoso, especialmente en redes basadas en direccionamiento privado.
 - Las redes basadas en direccionamiento privado RFC1918, como la 10.0.0.0/8 tienen más de 16 millones de direcciones IP

¿Qué es *host discovery*?

- Mejor si es rápido
- Mejor si es silencioso
- También se conoce como **escaneo ping** o **descubrimiento de sistemas**

Host discovery

- Una forma básica de *host discovery* es enviar un ping a cada dirección IP de un rango y ver qué direcciones responden.

Problemas:

- ☐ No todas las máquinas responden al ping
- ☐ Proceso lento

Nmap

- Es la herramienta por excelencia para escaneo de puertos (*port scanning*)
- También permite **host discovery** y *fingerprinting*
- Desarrollada por Gordon "Fyodor" Lyon
- Con *Nmap Scripting Engine* (NSE) puede incluso averiguar las aplicaciones concretas (nombre y número de versión) que se están ejecutando en un puerto
- Código fuente disponible
- Utilizaremos Nmap para ilustrar la mayoría de los ejemplos
- No es la única herramienta!
 - hping3, scapy, ... <- puedes buscar más en SecTools.Org



Nmap. Instalación en Debian

- `apt-get install nmap`
- Se puede instalar la última versión desde nmap.org

Aspectos importantes sobre Nmap

- La función por defecto de nmap es **escaneo de puertos** (*port scanning*)
 - Es decir, salvo que se especifique lo contrario, nmap ejecutará *port scanning*
 - `nmap 10.10.102.0/24`
 - Realizará *host discovery* sobre cada máquina
 - Sobre cada host "vivo" encontrado, realizará *port scanning*
 - En muchas ocasiones, interesa ejecutar sólo *host discovery* en una primera fase
 - Hay que indicarlo: opción **-sP** (lo veremos más adelante)

```
# Sólo host discovery  
nmap -sP 10.10.102.0/24
```



Aspectos importantes sobre Nmap

- Para poder aprovechar toda la funcionalidad de nmap, es necesario ejecutarlo con **privilegios de administrador**
 - Capacidad para "fabricar" *raw packets*
 - Si se ejecuta sin privilegios, nmap delegará en llamadas al SO. Algunas funcionalidades no estarán soportadas
- Se comporta de forma diferente si detecta que el escaneo se está efectuando en una red local
 - La opción **--packet-trace** permite ver los paquetes que envía nmap. Muy útil para comprender cómo funciona

Opciones de *host discovery* con Nmap

■ -sL (escaneo de lista o **List Scan**):

- Únicamente lista cada equipo de la/s red/es especificada/s
- **No envía paquetes** de ningún tipo a los objetivos
- Por defecto, realiza **resolución DNS inversa** en los equipos para obtener sus nombres
 - Se puede obtener información muy útil del nombre de un sistema (fw.acme.com, mail.acme.com,...)
- Paso importante, para asegurarse de escanear las máquinas adecuadas
- Esta opción no puede combinarse con opciones de mayor nivel de funcionalidad (análisis de puertos, detección de sistema operativo o escaneo ping)

Opciones de *host discovery* con Nmap

```
# nmap -sL 10.10.102.50/28
```

```
Starting Nmap 5.00 ( http://nmap.org ) at 2012-09-13 16:46 CEST
```

```
Host 10.10.102.48 not scanned
```

```
Host 10.10.102.49 not scanned
```

```
Host 10.10.102.50 not scanned
```

```
Host 10.10.102.51 not scanned
```

```
Host 10.10.102.52 not scanned
```

```
Host 10.10.102.53 not scanned
```

```
Host 10.10.102.54 not scanned
```

```
Host 10.10.102.55 not scanned
```

```
Host 10.10.102.56 not scanned
```

```
Host 10.10.102.57 not scanned
```

```
Host 10.10.102.58 not scanned
```

```
Host 10.10.102.59 not scanned
```

```
Host 10.10.102.60 not scanned
```

```
Host 10.10.102.61 not scanned
```

```
Host 10.10.102.62 not scanned
```

```
Host 10.10.102.63 not scanned
```

```
Nmap done: 16 IP addresses (0 hosts up) scanned in 0.01 seconds
```



Opciones de *host discovery* con Nmap

```
# nmap -sL www.udc.es/24

Starting Nmap 5.00 ( http://nmap.org ) at 2012-09-13 16:46 CEST
Host 193.144.48.0 not scanned
...
Host 193.144.48.90 not scanned
Host v3.viveros.udc.es (193.144.48.91) not scanned
Host afi.udc.es (193.144.48.92) not scanned
Host webafi.udc.es (193.144.48.93) not scanned
Host pafi.udc.es (193.144.48.94) not scanned
Host v3-2.viveros.udc.es (193.144.48.95) not scanned
Host 193.144.48.96 not scanned
Host 193.144.48.97 not scanned
Host 193.144.48.98 not scanned
Host 193.144.48.99 not scanned
Host zape2.udc.es (193.144.48.100) not scanned
Host 101.viveros.udc.es (193.144.48.101) not scanned
Host 102.viveros.udc.es (193.144.48.102) not scanned
Host sicrede.udc.es (193.144.48.103) not scanned
Host 193.144.48.104 not scanned
Host mentor.udc.es (193.144.48.105) not scanned
...
Host 193.144.48.255 not scanned
Nmap done: 256 IP addresses (0 hosts up) scanned in 4.37 seconds
```



Opciones de *host discovery* con Nmap

■ -sP (escaneo ping, barrido ping o **Ping Scan**)

- Indica a nmap que **sólo** realice *host discovery*
- Emite listado de equipos que respondieron (*host h is up*)
- Intrusivo (envía paquetes a los objetivos)
- Permite reconocimiento ligero de la red sin llamar mucho la atención
- Más fiable que hacer ping a la dir. de broadcast (algunos equipos no responden)

Opciones de *host discovery* con Nmap

- **-sP (escaneo ping, barrido ping o *Ping Scan*) (Cont.):**
 - Envía:
 - ICMP echo request
 - Paquete TCP SYN al puerto 443
 - Paquete TCP ACK al puerto 80
 - Solicitud ICMP timestamp
 - Si usuario no tiene privilegios: paquetes SYN a los puertos 80 y 443 utilizando la llamada al sistema connect()

Opciones de *host discovery* con Nmap

- Una de las formas de uso más comunes de Nmap es el sondeo de una red de área local Ethernet
- En la mayoría de las redes locales hay muchas direcciones IP sin usar en un momento determinado (especialmente en las que utilizan rangos de direcciones privadas)
- Cuando Nmap intenta enviar un paquete IP crudo (*raw*), como pudiera ser una solicitud de echo ICMP, el SO debe determinar primero la dirección MAC (ARP) correspondiente a la IP objetivo para poder dirigirse a ella en la trama Ethernet
 - Si recibe respuesta, significa que está vivo => no tiene sentido seguir enviando paquetes (ya hemos conseguido el objetivo)



Opciones de *host discovery* con Nmap

- Esta opción, que se puede especificar con **-PR** (Ping ARP), es el comportamiento por defecto, cuando se analizan sistemas Ethernet, si Nmap detecta que están en la red local, incluso aunque se especifique **-PE** o **-PS**
 - Se puede evitar con **--send-ip**
- El proceso de enviar múltiples ARP requests se convierte en problemático (y lento) si necesitamos hacer muchas consultas ARP en un corto período de tiempo
 - S.O. no está pensado para esto
- Nmap se ocupa él mismo de las solicitudes ARP, con un algoritmo optimizado
 - Mucho más rápido y fiable que los escaneos basados en IP

Opciones de *host discovery* con Nmap

```
# nmap -sP 10.10.102.50/26
```

```
Starting Nmap 5.00 ( http://nmap.org ) at 2012-09-13 18:24 CEST
Host 10.10.102.4 is up (0.00031s latency).
MAC Address: 00:90:FB:22:FF:95 (Portwell)
Host 10.10.102.5 is up (0.00031s latency).
MAC Address: 00:90:FB:22:FF:95 (Portwell)
Host 10.10.102.9 is up (0.00040s latency).
MAC Address: 00:50:56:91:2F:52 (VMWare)
Host 10.10.102.22 is up (0.00023s latency).
MAC Address: 00:15:17:12:B2:98 (Intel Corporate)
Host 10.10.102.27 is up (0.00022s latency).
MAC Address: 00:1D:09:14:1E:7C (Dell)
Host 10.10.102.29 is up (0.00025s latency).
MAC Address: 00:50:56:91:30:02 (VMWare)
Host 10.10.102.30 is up (0.00039s latency).
MAC Address: 00:50:56:91:4E:18 (VMWare)
Host 10.10.102.34 is up (0.00032s latency).
MAC Address: 00:50:56:91:30:76 (VMWare)
Host 10.10.102.44 is up (0.00030s latency).
MAC Address: 00:50:56:91:2F:58 (VMWare)
Nmap done: 64 IP addresses (10 hosts up) scanned in 0.95 seconds
```





Modificando el test de host discovery...

Opciones de *host discovery* con Nmap

- **-PS [lista de puertos]** (Ping TCP SYN)
 - Envía un paquete TCP vacío con el flag SYN activo.
 - El puerto destino por defecto es el 80 (man nmap)
 - También se puede especificar una lista de puertos:
 - **-PS22,23,25,80,113,1050,35000** <- envía escaneos en paralelo
 - El flag SYN indica al sistema remoto que quiere establecer una conexión
 - Si el puerto destino está cerrado se recibirá un RST
 - Si el puerto está abierto: el equipo remoto responderá con un TCP SYN/ACK (segundo paso del *Three-way handshake*)
 - El equipo local abortará la conexión con un RST (lo hace el SO)
 - Aquí no importa si el puerto está abierto o cerrado. Si llega respuesta, la máquina está "viva"

Opciones de *host discovery* con Nmap

- **-PS [lista de puertos]** (Ping TCP SYN) (Cont.)
 - Usuario no privilegiado, no puede fabricar *TCP raw packets* -> Nmap utiliza la llamada al sistema `connect()` contra el puerto destino
 - Si respuesta rápida (éxito o ECONNREFUSED):
 - El sistema está "vivo"
 - Si el intento de conexión se mantiene parado (vence temporizador)
 - El sistema se marca como no disponible

Opciones de *host discovery* con Nmap

- **-PA [lista de puertos]** (Ping TCP ACK)
 - Similar a Ping TCP SYN, pero con *flag* ACK en lugar de SYN
 - Recordar: ACK indica que se han recibido datos de una conexión TCP establecida
 - En este caso la conexión no existe
 - Los sistemas deberían responder con RST -> está "vivo"
 - Usuario sin privilegios -> llamada a `connect()` <- envía SYN, no ACK :(

Opciones de *host discovery* con Nmap

■ -PA [lista de puertos] (Ping TCP ACK) (Cont.)

□ ¿Qué aporta?

- Muchos administradores configuran routers y FW *stateless* para que bloqueen paquetes SYN con el objetivo de bloquear conexiones entrantes y permitir conexiones salientes

- En este caso pasarán los ACK

- Los FW actuales suelen ser *stateful* -> son capaces de descartar paquetes no esperados

- En este caso es más probable que funcione un paquete SYN dirigido a un puerto abierto

□ Entonces -PS o -PA?

- Las dos!

Opciones de *host discovery* con Nmap

- **-PU [lista de puertos]** (Ping UDP)
 - Envía paquete UDP vacío, por defecto al puerto ~~31338~~ 40125
 - ¿Por qué ese puerto?
 - Si se alcanza un puerto abierto, la mayoría de los servicios UDP descartarán el paquete y no devolverán respuesta (UDP no confirma)
 - Interesa un puerto que probablemente esté cerrado, ya que esto si generará respuesta: un error ICMP
 - Si se recibe ICMP de puerto no alcanzable (es la respuesta esperada para un puerto UDP cerrado)-> "vivo"
 - Otros errores ICMP o no respuesta -> no disponible o no alcanzable
 - Ventaja: algunos FW sólo filtran TCP

Opciones de *host discovery* con Nmap

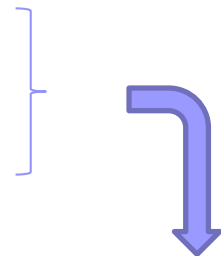
- **-PE; -PP; -PM** (Tipos de ping ICMP)

- **-PE** : solicitud de echo (consulta estándar de ping ICMP)


- Muchos sistemas y FW bloquean esos paquetes -> Escaneos que sólo utilizan ICMP no son muy fiables

- **-PP** : consulta de marca de tiempo

- **-PM** : consulta de máscara de red



Útil cuando los administradores bloquean paquetes de "solicitud de echo" explícitamente y se olvidan de otros tipos de mensajes ICMP



¿Y si el test de *host discovery* falla y quiero escanear la máquina igualmente?

Opciones de *host discovery* con Nmap

■ -PN (**skip ping**):

- No realiza la etapa de *host discovery*
- Trata a todos los hosts como si estuviesen "vivos"
- Por defecto, Nmap sólo realiza escaneos agresivos (análisis de puertos, detección de versión o SO) contra equipos "vivos", pero si se omite la etapa de descubrimiento (*host discovery*), el escaneo se realizará sobre todas las direcciones IP especificadas
 - Cuidado con el rango que se especifica!
- Fuerza el análisis de máquinas que de otra forma no se realizaría
- No es compatible con -sP (Ping Scan)

Opciones de *host discovery* con Nmap

- **-n** (No realizar resolución DNS inversa)
 - Las consultas DNS suelen ser lentas. Esto permite ganar un poco de tiempo
- **-R** (Realizar resolución de nombres con todos los objetivos)
 - Normalmente se realiza esto sólo si se descubre que el objetivo se encuentra vivo
- **--dns-servers <servidor1[,servidor2],...>** (Servidores a utilizar para las consultas DNS)
 - Nmap determina DNS a partir de `resolv.conf` (UNIX) o registro (Win32) y realiza él mismo las consultas directamente a los DNS, en paralelo, para mejorar el rendimiento
 - Esta opción permite especificar otros DNS. Utilizar más de uno suele ser más rápido

Formatos de salida

- -oN <filename>: normal
- -oG <filename>: grepable (salida para cada host en una línea)
- -oX <filename>: XML



Introducción al escaneo de puertos

PORT SCANNING

¿Qué es *Port Scanning*?

- Es el acto de testear de forma remota varios puertos para determinar en qué estado están
- La mayor parte de las herramientas distinguen entre "open" y "closed"

Por qué escanear puertos?

- Por seguridad

- Principio de seguridad de redes:

- "reducir el número y complejidad de los servicios ofrecidos reduce la oportunidad de que los atacantes entren"

- Los atacantes escanearan tu red en busca de vulnerabilidades -> hazlo tu primero y soluciona los problemas antes de que ellos los encuentren

- *Port Scanning* es la funcionalidad principal de Nmap -> Será la herramienta que usaremos en los ejemplos

¿Qué es un puerto?

- Abstracción, usada para distinguir distintos canales de comunicación
 - 16bits -> 0..65535
- Direcciones IP identifican máquinas en una red -> Puertos identifican aplicaciones en una máquina
- En TCP y UDP, conexión se identifica por:

IP+Puerto Origen / IP+Puerto Destino

Clasificación de puertos según la *Internet Assigned Numbers Authority* (IANA).

- Puertos bien conocidos (*well-known ports*)
 - Son puertos reservados en el rango 1 a 1023 que han sido registrados con la IANA para un servicio determinado
 - Algunos sistemas exigen privilegios especiales para asociar aplicaciones a estos puertos (e.g. Unix)
 - e.g. 22, 25 y 80 para SSH, SMTP y HTTP respectivamente
- Puertos registrados (*registered ports*)
 - Rango 1024 a 49151. También han sido registrados con la IANA
 - La diferencia con los anteriores es que no se exigen privilegios especiales
- Puertos dinámicos y/o privados
 - 49152 a 65535
 - Se usan dinámicamente o bien por servicios privados de una compañía

20 Puertos TCP más populares

- 1) 80 (HTTP) - Hypertext Transfer Protocol
- 2) 23 (Telnet) - Todavía se usa (puerto de administración en algunos routers/switches), a pesar de que es muy inseguro
- 3) 443 (HTTPS) - HTTP Secure
- 4) 21 (FTP) - File Transfer Protocol (inseguro)
- 5) 22 (SSH) - Secure Shell, un reemplazo cifrado para el Telnet (y, en algunos casos, FTP)
- 6) 25 (SMTP) - Standard Mail Transfer Protocol (inseguro)
- 7) 3389 (ms-term-server) - puerto de administración de Microsoft Terminal Services
- 8) 110 (POP3) - Post Office Protocol versión 3 para recuperación de correo-e (inseguro)
- 9) 445 (Microsoft-DS) - Para comunicación SMB sobre IP con MS Windows services (como compartir archivos o impresoras)
- 10) 139 (NetBIOS-SSN) - NetBIOS Session Service para comunicación con MS Windows service.

20 Puertos TCP más populares (cont.)

- 11) 143 (IMAP) - Internet Message Access Protocol version 2. Protocolo de recuperación de correo-e (inseguro)
- 12) 53 (Domain) - Domain Name System (DNS), un sistema inseguro para convertir entre nombres de host/dominio y direcciones IP
- 13) 135 (MSRPC) - Otro puerto común para MS Windows services
- 14) 3306 (MySQL) - Para comunicación con MySQL
- 15) 8080 (HTTP-Proxy) - Usado normalmente para proxies HTTP o como un puerto alternativo para servidores web
- 16) 1723 (PPTP) - Point-to-point tunneling protocol. Un método de implementar VPNs
- 17) 111 (RPCBind) - SUN Remote Procedure Call
- 18) 995 (POP3S) - POP3 con SSL
- 19) 993 (IMAPS) - IMAPv2 con SSL
- 20) 5900 (VNC) - Escritorio remoto (inseguro)

20 Puertos UDP más populares

- 1) 631 (IPP) - Internet Printing Protocol
- 2) 161 (SNMP) - Simple Network Management Protocol
- 3) 137 (NETBIOS-NS) - Uno de muchos puertos UDP para Windows services (como compartir archivos o impresoras)
- 4) 123 (NTP) - Network Time Protocol
- 5) 138 (NETBIOS-DGM) - Otro puerto Windows Services
- 6) 1434 (MS-SQL-DS) - Microsoft SQL Server
- 7) 445 (Microsoft-DS) - Otro puerto Windows Services
- 8) 135 (MSRPC) - Otro puerto Windows Services
- 9) 67 (DHCP) - Dynamic Host Configuration Protocol Server (reparte IPs a los clients cuando se unen a la red)
- 10) 53 (Domain) - Domain Name System (DNS) server

20 Puertos UDP más populares (cont.)

- 11) 139 (NETBIOS-SSN) - Otro puerto Windows Services
- 12) 500 (ISAKMP)-Internet Security Association and Key Management Protocol. Se usa para configurar VPNs con IPsec
- 13) 68 (DHCP) - puerto cliente DHCP
- 14) 520 (Route) - Routing Information Protocol (RIP)
- 15) 1900 (UPNP) - Microsoft Simple Service Discovery Protocol, permite descubrir dispositivos plug-and-play
- 16) 4500 (nat-t-ike) - Para permitir el funcionamiento de IPsec a través de NAT.
- 17) 514 (Syslog) - Demonio de log estándar (UNIX)
- 18) 49152 (Varía) - El primero de los puertos dinámicos/privados especificados por IANA. Existe una probabilidad alta de que algún servicio se asocie a este puerto
- 19) 162 (SNMPTrap) - Simple Network Management Protocol trap port
- 20) 69 (TFTP) - Trivial File Transfer Protocol



nmap-services

- Además de los puertos registrados, incluye otros...

```
#
# Fields in this file are: Service name, portnum/protocol, open-frequency, optional comments
#

ftp      21/tcp    0.197667      # File Transfer [Control]
ftp      21/udp    0.004844      # File Transfer [Control]
ftp      21/sctp   0.000000      # File Transfer [Control]
ssh      22/tcp    0.182286      # Secure Shell Login
ssh      22/udp    0.003905      # Secure Shell Login
ssh      22/sctp   0.000000      # Secure Shell Login
...
http     80/tcp    0.484143      # World Wide Web HTTP
http     80/udp    0.035767      # World Wide Web HTTP
http     80/sctp   0.000000      # World Wide Web HTTP
...
zincite-a 1034/tcp    0.001064      # Zincite.A backdoor
...
netbus   12345/tcp   0.000527      # NetBus backdoor trojan or Trend Micro Office Scan
netbus   12346/tcp   0.000088      # NetBus backdoor trojan
...
bo2k     14141/tcp   0.000038      # Back Orifice 2K BoPeep mouse/keyboard input
...
bo2k     15151/tcp   0.000013      # Back Orifice 2K BoPeep video output
...
```

Estados de un puerto reconocidos por Nmap

- open
 - una aplicación esta aceptando conexiones TCP o paquetes UDP en este puerto -> posibilidad de ataque
- closed
 - el puerto es accesible, pero no hay aplicación escuchando
- filtered
 - Nmap no puede determinar el estado del puerto porque el filtrado de paquetes (FW, reglas de un router, ...) impide que las pruebas alcancen el puerto
- unfiltered
 - el puerto es accesible, pero Nmap no es capaz de determinar si está abierto o cerrado
- open | filtered
 - Nmap no es capaz de determinar si un puerto está abierto o filtrado
- closed | filtered
 - Nmap no es capaz de determinar si un puerto está cerrado o filtrado

Escaneo de puertos con Nmap

- Nmap ofrece muchas opciones para escaneos avanzados, pero la opción por defecto es sencilla:

```
nmap <target>
```

```
# nmap scanme.nmap.org
```

```
Starting Nmap 5.00 ( http://nmap.org ) at 2012-08-19 16:43 CEST
```

```
Interesting ports on scanme.nmap.org (74.207.244.221):
```

```
Not shown: 997 filtered ports
```

PORT	STATE	SERVICE
------	-------	---------

21/tcp	closed	ftp
--------	--------	-----

80/tcp	open	http
--------	------	------

443/tcp	closed	https
---------	--------	-------

```
Nmap done: 1 IP address (1 host up) scanned in 25.29 seconds
```


Métodos de escaneo de puertos soportados por Nmap

- TCP SYN (Stealth) Scan (-sS)
 - TCP Connect Scan (-sT)
 - UDP Scan (-sU)
 - TCP ACK Scan (-sA)
 - TCP FIN, NULL y Xmas (-sF, -sN, -sX)
 - TCP Idle Scan (-sI)
 - ...
-
- NOTA: la mayoría de los tipos de escaneo necesitan enviar y recibir *raw IP packets* => usuario privilegiado

TCP SYN (Stealth) Scan (-sS)

- Escaneo por defecto
- Relativamente discreto y sigiloso, ya que nunca completa las conexiones TCP
 - No obstante, IDSs y firewalls personales podrían detectar escaneos SYN!
- Permite diferenciación clara y fiable entre: open, closed y filtered
- Requiere privilegios "*raw-packet*"
- SYN scan también se conoce como *half-open scanning*

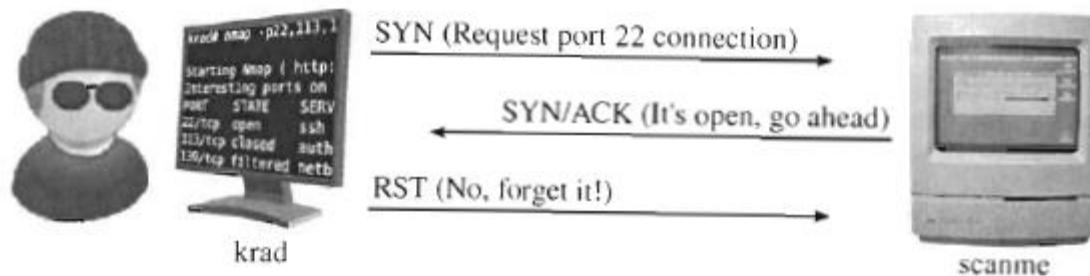
```
krad# nmap -p22,113,139 scanme.nmap.org
```

```
Starting Nmap ( http://nmap.org )  
Interesting ports on scanme.nmap.org (64.13.134.52):  
PORT      STATE      SERVICE  
22/tcp    open       ssh  
113/tcp   closed     auth  
139/tcp   filtered   netbios-ssn
```

```
Nmap done: 1 IP address (1 host up) scanned in 1.35 seconds
```

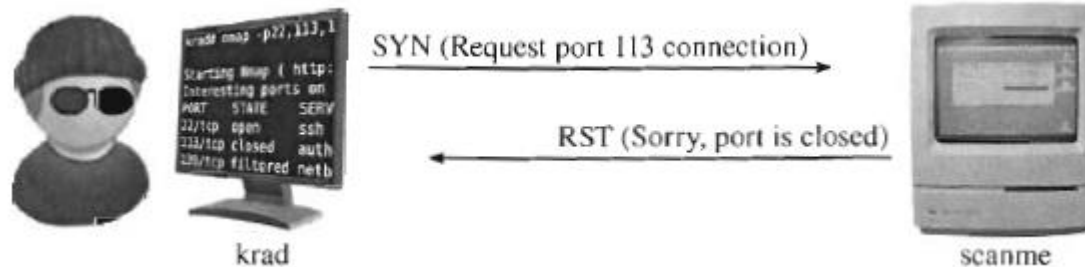
TCP SYN (Stealth) Scan (-sS)

- SYN scan del puerto 22 (**abierto**)



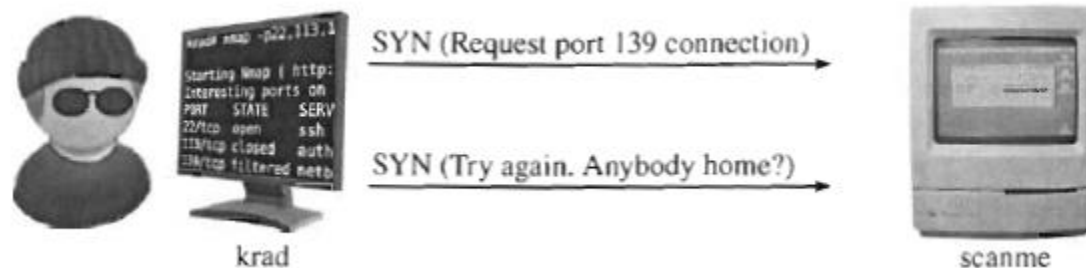
TCP SYN (Stealth) Scan (-sS)

- SYN scan del puerto 113 (**cerrado**)



TCP SYN (Stealth) Scan (-sS)

- SYN scan del puerto 139 (**filtrado**)



- Un puerto que no responde, normalmente es un puerto filtrado (bloqueado por un firewall), pero este test no es decisivo
 - Puede que el host esté apagado, o que la prueba o la respuesta se hayan perdido
 - Nmap también marcará un puerto como filtrado si recibe ciertos mensajes de error ICMP

TCP SYN (Stealth) Scan (-sS)

Respuesta	Estado asignado
TCP SYN/ACK	open
TCP RST	closed
Sin respuesta (incluso después de retransm.)	filtered
Error ICMP unreachable (tipo 3, cód. 1, 2, 3, 9,10 o 13)	filtered

Asignación de estados a puertos basándose en las respuestas a una prueba SYN

TCP SYN (Stealth) Scan (-sS)

```
krad# nmap -d --packet-trace -p22,113,139 scanme.nmap.org

Starting Nmap ( http://nmap.org )
SENT (0.0130s) ICMP krad > scanme echo request (type=8/code=0) ttl=52 id=1829
SENT (0.0160s) TCP krad:63541 > scanme:80 A iplen=40 seq=91911070 ack=99850910
RCVD (0.0280s) ICMP scanme > krad echo reply (type=0/code=0) iplen=28
We got a ping packet back from scanme: id = 48821 seq = 714 checksum = 16000
massping done:  num_hosts: 1  num_responses: 1
Initiating SYN Stealth Scan against scanme.nmap.org (scanme) [3 ports] at 00:53
SENT (0.1340s) TCP krad:63517 > scanme:113 S iplen=40 seq=10438635
SENT (0.1370s) TCP krad:63517 > scanme:22 S iplen=40 seq=10438635
SENT (0.1400s) TCP krad:63517 > scanme:139 S iplen=40 seq=10438635
RCVD (0.1460s) TCP scanme:113 > krad:63517 RA iplen=40 seq=0 ack=10438636
RCVD (0.1510s) TCP scanme:22 > krad:63517 SA iplen=44 seq=75897108 ack=10438636
SENT (1.2550s) TCP krad:63518 > scanme:139 S iplen=40 seq=10373098 win=3072
```

Ejercicio: a partir de los paquetes recibidos (RCVD) deduce el estado de los puertos



TCP SYN (Stealth) Scan (-sS)

```
krad# nmap -d --packet-trace -p22,113,139 scanme.nmap.org

Starting Nmap ( http://nmap.org )
SENT (0.0130s) ICMP krad > scanme echo request (type=8/code=0) ttl=52 id=1829
SENT (0.0160s) TCP krad:63541 > scanme:80 A iplen=40 seq=91911070 ack=99850910
RCVD (0.0280s) ICMP scanme > krad echo reply (type=0/code=0) iplen=28
We got a ping packet back from scanme: id = 48821 seq = 714 checksum = 16000
massping done:  num_hosts: 1  num_responses: 1
Initiating SYN Stealth Scan against scanme.nmap.org (scanme) [3 ports] at 00:53
SENT (0.1340s) TCP krad:63517 > scanme:113 S iplen=40 seq=10438635
SENT (0.1370s) TCP krad:63517 > scanme:22 S iplen=40 seq=10438635
SENT (0.1400s) TCP krad:63517 > scanme:139 S iplen=40 seq=10438635
RCVD (0.1460s) TCP scanme:113 > krad:63517 RA iplen=40 seq=0 ack=10438636
RCVD (0.1510s) TCP scanme:22 > krad:63517 SA iplen=44 seq=75897108 ack=10438636
SENT (1.2550s) TCP krad:63518 > scanme:139 S iplen=40 seq=10373098 win=3072
The SYN Stealth Scan took 1.25s to scan 3 total ports.
Interesting ports on scanme.nmap.org (64.13.134.52):
PORT      STATE      SERVICE
22/tcp    open       ssh
113/tcp   closed     auth
139/tcp   filtered  netbios-ssn

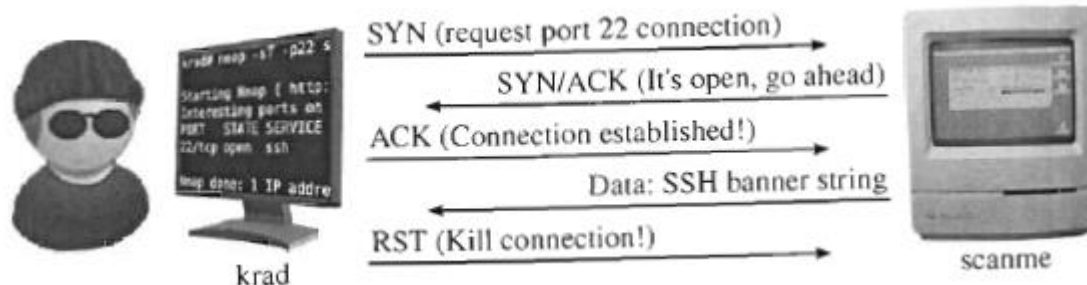
Nmap done: 1 IP address (1 host up) scanned in 1.40 seconds
```

Ejercicio: a partir de los paquetes recibidos (RCVD) deduce el estado de los puertos



TCP Connect Scan (-sT)

- Es el escaneo por defecto cuando SYN scan no se puede realizar (usuario no tiene privilegios para crear paquetes o está escaneando redes IPv6)
- Nmap pide al SO subyacente que establezca conexión con la máquina y puerto objetivos a través de la llamada al sistema connect()
- La llamada al sistema completa la conexión para abrir los puertos objetivo en lugar de realizar el "*half-open reset*" que hace SYN scan
 - ⇒ Más tiempo, más paquetes, mayor probabilidad de que el acceso quede registrado
 - Sólo en el caso de puertos abiertos. Puertos cerrados o filtrados generan el mismo tráfico que con SYN scan



UDP Scan (-sU)

- Escaneo UDP es más lento y difícil que TCP
- Se envía una cabecera UDP vacía a cada puerto objetivo. Basándose en la respuesta (o ausencia) el puerto se asigna a uno de cuatro estados

Respuesta	Estado asignado
Cualquier respuesta UDP desde el puerto objetivo (inusual)	open
Sin respuesta (incluso después de retransm.)	open filtered
Error ICMP port unreachable (tipo 3, cód. 3)	closed
Otros errores ICMP unreachable (tipo 3, cód. 1, 2, 9,10 o 13)	filtered

Cómo Nmap interpreta las respuestas a una prueba UDP

UDP Scan (-sU)

- El ejemplo muestra dos características del escaneo UDP:
 - Ambigüedad open | filtered
 - Tiempo de escaneo

```
krad# nmap -sU -v felix
```

```
Starting Nmap ( http://nmap.org )  
Interesting ports on felix.nmap.org (192.168.0.42):  
(The 997 ports scanned but not shown below are in state: closed)  
PORT      STATE          SERVICE  
53/udp    open|filtered  domain  
67/udp    open|filtered  dhcpserver  
111/udp    open|filtered  rpcbind  
MAC Address: 00:02:E3:14:11:02 (Lite-on Communications)  
  
Nmap done: 1 IP address (1 host up) scanned in 999.25 seconds
```

Cómo saber si un puerto UDP está abierto o filtrado?

- En el caso de un sitio fuertemente filtrado, el escaneo no es capaz de reducir los puertos abiertos
 - Se necesita nueva estrategia

```
krad# nmap -sU -T4 scanme.nmap.org
```

```
Starting Nmap ( http://nmap.org )
```

```
All 1000 scanned ports on scanme.nmap.org (64.13.134.52) are open|filtered
```

```
Nmap done: 1 IP address (1 host up) scanned in 5.50 seconds
```

Los 1000 puertos están en estado "open | filtered"



Cómo saber si un puerto UDP está abierto o filtrado?

- Los servicios UDP normalmente definen su propia estructura de paquete
 - Un paquete SNMP es completamente diferente de un paquete DHCP o DNS
- Para enviar el paquete adecuado para cada servicio se necesitaría una gran base de datos que definiese sus formatos de prueba
 - nmap-service-probes
 - Es parte de "service and version detection", `-sV`

TCP ACK Scan (-sA)

- No determina si un puerto está open (ni open | filtered)
- Se usa para averiguar las reglas de los firewalls, determinando si son *stateful* o no y qué puertos se filtran
- El paquete de prueba tiene sólo el flag ACK activo
- Al escanear sistemas no filtrados, los puertos abiertos y cerrados devolverán un paquete RST. Esto significa que son alcanzables por un paquete ACK, pero no se sabe si son open o closed
- Los puertos que no responden o envían ciertos mensajes de error ICMP se etiquetan como filtered

Respuesta	Estado asignado
respuesta TCP RST	unfiltered
Sin respuesta (incluso después de retransm.)	filtered
Errores ICMP unreachable (tipo 3, cód. 1, 2, 3, 9,10 o 13)	filtered

TCP ACK Scan (-sA)

```
krad# nmap -sA -T4 scanme.nmap.org
```

```
Starting Nmap ( http://nmap.org )
```

```
Interesting ports on scanme.nmap.org (64.13.134.52):
```

```
Not shown: 994 filtered ports
```

PORT	STATE	SERVICE
22/tcp	unfiltered	ssh
25/tcp	unfiltered	smtp
53/tcp	unfiltered	domain
70/tcp	unfiltered	gopher
80/tcp	unfiltered	http
113/tcp	unfiltered	auth

```
Nmap done: 1 IP address (1 host up) scanned in 4.01 seconds
```



Escaneos TCP FIN, NULL y Xmas ($-sF$, $-sN$, $-sX$)

- Estos tipos de escaneo tienen por objetivos
 - atravesar FW que no hagan inspección de estado o routers que hagan filtrado de paquetes
 - averiguar si un puerto está cerrado (si un puerto está cerrado y recibe un paquete como los siguientes, debe responder con un RST)
 - $-sF$
 - Se fija el flag TCP FIN
 - $-sN$
 - No se fija ningún flag TCP
 - $-sX$
 - Se fijan los flags FIN, PSH, y URG

TCP Idle Scan (-sI)

- Técnica publicada en 1998 por Antirez (autor de hping2)
- Características:
 - Muy sigiloso
 - Necesita un equipo "Zombie"
 - Permite averiguar relaciones de confianza entre las máquinas
 - Obtiene lista de puertos desde punto de vista del "zombie"
 - Muy lento
 - 15 seg SYN scan -> 15 min Idle scan

TCP Idle Scan (-sI)

■ Bases de funcionamiento

- Una forma de determinar si un puerto TCP está abierto es enviar un paquete SYN:
 - Si se recibe SYN/ACK: el puerto está abierto
 - Si se recibe RST: el puerto está cerrado
- Una máquina que recibe un SYN/ACK no solicitado responderá con un RST. Un RST no solicitado se ignorará
- Cada paquete IP en Internet tiene un número de identificación de fragmento: **IP ID**
 - Muchos S.O.'s simplemente incrementan este número para cada paquete que envían
 - Averiguar el IP ID permite saber cuántos paquetes envió el equipo desde la última consulta

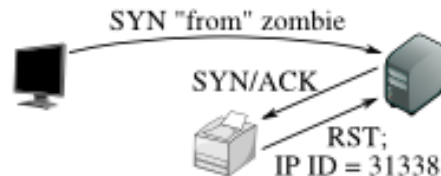
TCP Idle Scan (-sI). Puerto abierto

Step 1: Probe the zombie's IP ID.



The attacker sends a SYN/ACK to the zombie. The zombie, not expecting the SYN/ACK, sends back a RST, disclosing its IP ID.

Step 2: Forge a SYN packet from the zombie.



The target sends a SYN/ACK in response to the SYN that appears to come from the zombie. The zombie, not expecting it, sends back a RST, incrementing its IP ID in the process.

Step 3: Probe the zombie's IP ID again.



The zombie's IP ID has increased by 2 since step 1, so the port is open!

TCP Idle Scan (-sI). Puerto cerrado

Step 1: Probe the zombie's IP ID.



The attacker sends a SYN/ACK to the zombie. The zombie, not expecting the SYN/ACK, sends back a RST, disclosing its IP ID. This step is always the same.

Step 2: Forge a SYN packet from the zombie.



The target sends a RST (the port is closed) in response to the SYN that appears to come from the zombie. The zombie ignores the unsolicited RST, leaving its IP ID unchanged.

Step 3: Probe the zombie's IP ID again.



The zombie's IP ID has increased by only 1 since step 1, so the port is not open.

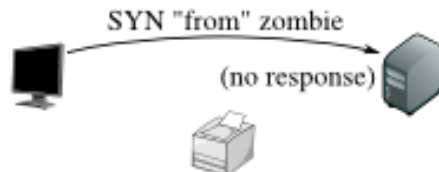
TCP Idle Scan (-sI). Puerto filtrado

Step 1: Probe the zombie's IP ID.



Just as in the other two cases, the attacker sends a SYN/ACK to the zombie. The zombie discloses its IP ID.

Step 2: Forge a SYN packet from the zombie.



The target, obstinately filtering its port, ignores the SYN that appears to come from the zombie. The zombie, unaware that anything has happened, does not increment its IP ID.

Step 3: Probe the zombie's IP ID again.



The zombie's IP ID has increased by only 1 since step 1, so the port is not open. From the attacker's point of view this filtered port is indistinguishable from a closed port.

=> No puede distinguir entre puertos cerrados y puertos filtrados

Más info en: <http://nmap.org/book/idlescan.html>

Combinando distintos tipos de escaneo

- Algunas veces una combinación de tipos de escaneo puede usarse para recoger información extra de un sistema
- Si un tipo de escaneo identifica un puerto como (open | filtered) y otro lo identifica como (unfiltered), la lógica dicta que debe estar open
 - No siempre se cumple. Los escaneos no siempre devuelven estados consistentes para un mismo puerto:
 - SYN scan puede considerar puerto SSH filtrado y ACK scan considerarlo no filtrado

“interpretar los resultados de Nmap es un arte que se beneficia de la experiencia y la intuición” -- Fyodor

Control del tiempo de escaneo

■ Modos:

- ☐ paranoid (0): evasión IDS
- ☐ sneaky (1): evasión IDS
- ☐ polite (2): consume menos ancho de banda
- ☐ normal (3)
- ☐ aggressive (4): acelera escaneo, asumiendo red rápida y fiable
- ☐ insane (5): red muy rápida y fiable o velocidad frente a precisión

■ Ejemplos:

```
# nmap -T4 insecure.org
```

```
# nmap -T aggressive insecure.org
```

FINGERPRINTING

¿Qué es *Fingerprinting*?

- Uso de técnicas para determinar de forma remota las aplicaciones que hay en una máquina remota
 - Sistema Operativo (*OS fingerprinting*)
 - Servicios

Razones

- Determinar la vulnerabilidad de las máquinas objetivo
- Confeccionar *exploits*
- Inventario de la red
- Detectar dispositivos no autorizados
- Ingeniería social

Tipos de *Fingerprinting*

- Pasivo: no interroga directamente a la máquina
 - Escucha la red y analiza la información
 - Puede utilizar solicitudes legítimas para obtener tráfico. Por ejemplo, la solicitud de una página web

Tipos de *Fingerprinting*

- Pasivo: no interroga directamente a la máquina
 - Escucha la red y analiza la información
 - Puede utilizar solicitudes legítimas para obtener tráfico. Por ejemplo, la solicitud de una página web

- Activo: interroga a la máquina
 - Puede ser más o menos agresivo
 - Se basa en enviar paquetes a la máquina remota y analizar sus respuestas

OS fingerprinting

- Se basa en el principio de que la pila de protocolos de cada SO tiene su propia idiosincrasia
 - Cada SO responde de forma diferente a una variedad de paquetes
 - Necesitamos una BD sobre cómo diferentes SOs responden a diferentes paquetes

OS fingerprinting muy básico :-)

```
$ ping 192.168.1.215
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.056 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.041 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.029 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.049 ms
64 bytes from localhost (127.0.0.1): icmp_seq=5 ttl=64 time=0.040 ms
^C
--- localhost ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.029/0.043/0.056/0.009 ms
```

```
$ ping 10.112.12.30
PING 10.112.12.30 (10.112.12.30): 56 data bytes
64 bytes from 10.112.12.30: icmp_seq=0 ttl=128 time=0.622 ms
64 bytes from 10.112.12.30: icmp_seq=1 ttl=128 time=0.786 ms
64 bytes from 10.112.12.30: icmp_seq=2 ttl=128 time=0.704 ms
64 bytes from 10.112.12.30: icmp_seq=3 ttl=128 time=0.510 ms
64 bytes from 10.112.12.30: icmp_seq=4 ttl=128 time=0.913 ms
^C
--- 10.112.12.30 ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.510/0.707/0.913/0.138 ms
```



OS fingerprinting muy básico :-)

- TTL para Windows suele ser 128
- TTL para Linux suele ser 64 o 255

OS fingerprinting muy básico :-)

- TTL para Windows suele ser 128
- TTL para Linux suele ser 64 o 255
- En Linux, se puede consultar/modificar en

- `/proc/sys/net/ipv4/ip_default_ttl`

```
# cat /proc/sys/net/ipv4/ip_default_ttl  
64
```

- Para que la modificación sea persistente, es necesario
 - Editar: `/etc/sysctl.conf`
 - Añadir: `net.ipv4.ip_default_ttl = 64`
 - Validar: `sysctl -p /etc/sysctl.conf`

OS fingerprinting muy básico :-)

```
# ping -c3 www.debian.org
PING www.debian.org (86.59.118.148) 56(84) bytes of data.
64 bytes from englund.debian.org (86.59.118.148): icmp_req=1 ttl=48 time=57.5 ms
64 bytes from englund.debian.org (86.59.118.148): icmp_req=2 ttl=48 time=52.7 ms
64 bytes from englund.debian.org (86.59.118.148): icmp_req=3 ttl=48 time=54.0 ms

--- www.debian.org ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 52.708/54.751/57.519/2.047 ms
```

■ Y ttl 48?



OS fingerprinting muy básico :-)

```
# ping -c3 www.debian.org
PING www.debian.org (86.59.118.148) 56(84) bytes of data.
64 bytes from englund.debian.org (86.59.118.148): icmp_req=1 ttl=48 time=57.5 ms
64 bytes from englund.debian.org (86.59.118.148): icmp_req=2 ttl=48 time=52.7 ms
64 bytes from englund.debian.org (86.59.118.148): icmp_req=3 ttl=48 time=54.0 ms

--- www.debian.org ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 52.708/54.751/57.519/2.047 ms
```

```
# traceroute www.debian.org
traceroute to www.debian.org (86.59.118.148), 30 hops max, 60 byte packets
[...]
9  te8-1.ccr01.par03.atlas.cogentco.com (154.54.58.142)  207.192 ms te2-
1.ccr01.par03.atlas.cogentco.com (130.117.51.141)  207.126 ms te3-
1.ccr01.par03.atlas.cogentco.com (154.54.58.134)  207.098 ms
10  cbv-core-2.gigabiteth4-4.tele2.net (130.244.200.37)  35.657 ms  35.624 ms  35.579 ms
11  cbv-core-3.tengige0-0-0-0.tele2.net (130.244.52.230)  36.895 ms  37.028 ms  41.849 ms
12  wen3-core-1.pos9-0.tele2.net (130.244.49.110)  53.969 ms  53.945 ms  55.958 ms
13  130.244.49.118 (130.244.49.118)  55.893 ms  52.283 ms  52.395 ms
14  212.152.193.89 (212.152.193.89)  52.350 ms  52.502 ms  52.477 ms
15  81.189.132.90 (81.189.132.90)  52.422 ms  52.383 ms  52.507 ms
16  86.59.118.145 (86.59.118.145)  53.104 ms  53.242 ms  53.185 ms
17  englund.debian.org (86.59.118.148)  56.468 ms  56.431 ms  54.822 ms
```



TCP/IP stack fingerprinting

- Algunos campos TCP/IP que pueden variar de un SO a otro son:
 - TTL
 - Window size
 - Max segment size, MSS
 - Window scaling value
 - Don't fragment
 - Selective Acknowledgment Permitted, "sackOK"
 - No operation, NOP
 - Initial packet size

- Pueden combinarse en una firma (*fingerprint*) para identificar al SO
- Los SOs no solo usan diferentes opciones por defecto, también listan las opciones en diferente orden

TCP/IP stack fingerprinting

Operating System (OS)	IP Initial TTL	TCP window size
Linux (kernel 2.4 and 2.6)	64	5840
Google's customized Linux	64	5720
FreeBSD	64	65535
Windows XP	128	65535
Windows 7, Vista and Server 2008	128	8192
Cisco Router (IOS 12.4)	255	4128

Extraído de <http://www.netresec.com/>

OS fingerprinting con Nmap (-O) (OS detection)

```
# nmap -O -v scanme.nmap.org

Starting Nmap ( http://nmap.org )
Nmap scan report for scanme.nmap.org (74.207.244.221)
Not shown: 994 closed ports
PORT      STATE      SERVICE
22/tcp    open      ssh
80/tcp    open      http
646/tcp    filtered  ldap
1720/tcp   filtered  H.323/Q.931
9929/tcp   open      nping-echo
31337/tcp  open      Elite
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:kernel:2.6.39
OS details: Linux 2.6.39
Uptime guess: 1.674 days (since Fri Sep  9 12:03:04 2011)
Network Distance: 10 hops
TCP Sequence Prediction: Difficulty=205 (Good luck!)
IP ID Sequence Generation: All zeros

Read data files from: /usr/local/bin/../../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 5.58 seconds
      Raw packets sent: 1063 (47.432KB) | Rcvd: 1031 (41.664KB)
```



OS fingerprinting con Nmap (-O) (OS detection)

■ Device type

- Todos los fingerprints están clasificados con uno o más tipos de dispositivo de alto nivel como router, printer, firewall o general purpose
- Es posible que se muestren varios tipos de dispositivos:
 - "Device type: router | firewall"

■ Running

- Muestra la familia del SO (e.g. Linux) y la generación, si está disponible (e.g. 2.6.x).

■ OS details / Aggressive OS guesses (si no hay coincidencia exacta)

- Datos en formato libre "Linux 2.6.20-1 (Fedora Core 5)"

■ Uptime guess (si --verbose)

- Estimación calculada a partir del *timestamp* de los paquetes TCP

OS fingerprinting con Nmap (-O) (OS detection)

- Network distance
 - Número de routers entre la máquina y el objetivo
- TCP Sequence Prediction
 - Algunos sistemas son vulnerables a "blind TCP spoofing attacks", al usar números de inicio de secuencia predecibles
 - Class describe el algoritmo de ISN (Initial Sequence Number)
 - Difficulty es una estimación de la dificultad de realizar el ataque: "Trivial joke", "Easy", "Medium", "Formidable", "Worthy challenge" y "Good luck"
- IP ID Sequence Generation (si --verbose)
 - Algunos sistemas proporcionan información sensible sobre sus niveles de tráfico, en función de cómo generan este campo

Fingerprinting de aplicaciones

- Detrás de un puerto abierto no tiene por qué estarse ejecutando el servicio esperado
- Un servicio puede ejecutarse en un puerto diferente al que le correspondería
 - A veces, se eligen puertos en base a los que tiene abiertos el FW

Técnicas no automáticas de *fingerprinting*

■ **Direct Banner Grabbing**

```
root@nostromo# telnet mail.fh-hagenberg.at 143
Trying 193.170.124.96...
Connected to postman.fh-hagenberg.at.
Escape character is '^]'.
* OK Microsoft Exchange Server 2003 IMAP4rev1 server version
6.5.7226.0 (postman.fhs-hagenberg.ac.at) ready.
```

```
root@nostromo# telnet nostromo.joeh.org 80
Trying 193.170.32.26...
[...]
HEAD / HTTP/1.0
[...]
Server: Microsoft-IIS/8.1
[...]
The analysis would reveal: Server: Microsoft-IIS/8.1: It is a
Microsoft IIS, version 8.1.
```

Técnicas no automáticas de *fingerprinting*

■ ***Indirect Banner Grabbing***

- Las cabeceras del correo suelen contener la versión del cliente de correo utilizada por el usuario
 - Banners de los FW que atraviesa
 - Marcas de los antivirus que escanean el mensaje
 - ...

Return-Path: sender.perez@udc.es
Received: from smtpout2.correo.udc.es (LHLO smtpout2.correo.udc.es)
(193.144.48.102) by ms2.correo.udc.es with LMTP; Fri, 2 Sep 2016 16:48:35
+0200 (CEST)
Received: from localhost (localhost.localdomain [127.0.0.1])
by smtpout2.correo.udc.es (Postfix) with ESMTP id C5E682071B
for <receipt.suarez@udc.es>; Fri, 2 Sep 2016 16:48:35 +0200 (CEST)
X-Spam-Flag: NO
X-Spam-Score: -1.394
X-Spam-Level:
X-Spam-Status: No, score=-1.394 tagged_above=-10 required=6.2
tests=[ALL_TRUSTED=-1, BAYES_00=-1.9, SUBJ_ALL_CAPS=1.506]
autolearn=no autolearn_force=no
Received: from smtpout2.correo.udc.es ([127.0.0.1])
by localhost (smtpout2.correo.udc.es [127.0.0.1]) (amavisd-new, port 10032)
with ESMTP id udJw0FkzaIDW for <receipt-suarez@udc.es>;
Fri, 2 Sep 2016 16:48:35 +0200 (CEST)
Received: from localhost (localhost.localdomain [127.0.0.1])
by smtpout2.correo.udc.es (Postfix) with ESMTP id 6E9842071F
for <receipt-suarez@udc.es>; Fri, 2 Sep 2016 16:48:35 +0200 (CEST)
X-Virus-Scanned: amavisd-new at smtpout2.correo.udc.es
Received: from smtpout2.correo.udc.es ([127.0.0.1])
by localhost (smtpout2.correo.udc.es [127.0.0.1]) (amavisd-new, port 10026)
with ESMTP id NGqxeHEvNC30 for <receipt-suarez@udc.es>;
Fri, 2 Sep 2016 16:48:35 +0200 (CEST)
Received: from [192.168.0.33] (130.XX.YY.213.dynamic.reverse-mundo-r.com [213.YY.XX.130])
by smtpout2.correo.udc.es (Postfix) with ESMTPSA id 3AFBA2071B
for <receipt-suarez@udc.es>; Fri, 2 Sep 2016 16:48:35 +0200 (CEST)
From: Pepito Perez <sender.perez@udc.es>
Content-Type: text/plain; charset=utf-8
Content-Transfer-Encoding: quoted-printable
Subject: =?utf-8?Q?E-mail header example?=
Message-Id: <E08896AF-XXXX-YYYY-9559-885585D1572D@udc.es>
Date: Fri, 2 Sep 2016 16:48:12 +0200
To: Sr. Suarez <receipt-suarez@udc.es>
Mime-Version: 1.0 (Mac OS X Mail 9.3 \ (3124\))
X-Mailer: Apple Mail (2.3124)
X-EsetId: 37403A499DC073656D7061



Fingerprinting de aplicaciones con Nmap: Service and Application Version Detection (-sV)

- Pasos principales de -sV
 - Si puerto TCP, se conecta
 - Escucha 5 seg. aprox. ("NULL probe")
 - FTP, SSH, SMTP, Telnet, POP3, IMAP,... se identifican a sí mismos en una cabecera de bienvenida
 - Si se reciben datos, se comparan con firmas en `/usr/share/nmap/nmap-service-probes`
 - Servicio completamente identificado: OK
 - Servicio parcialmente identificado ("soft match"), se intenta de nuevo con pruebas específicas del servicio
 - Escaneo UDP

Fingerprinting de aplicaciones con Nmap: Service and Application Version Detection (-sV)

```
# nmap -sV -T4 insecure.org

Starting Nmap ( http://nmap.org )
Nmap scan report for insecure.org (74.207.254.18)
Host is up (0.016s latency).
rDNS record for 74.207.254.18: web.insecure.org
Not shown: 95 filtered ports
PORT      STATE  SERVICE  VERSION
22/tcp    open   ssh      OpenSSH 4.3 (protocol 2.0)
25/tcp    open   smtp     Postfix smtpd
80/tcp    open   http     Apache httpd 2.2.3 ((CentOS))
113/tcp   closed auth
443/tcp   open   ssl/http Apache httpd 2.2.3 ((CentOS))
Service Info: Host:  web.insecure.org

Nmap done: 1 IP address (1 host up) scanned in 14.82 seconds
```



Fingerprinting de aplicaciones con Nmap:

Service and Application Version Detection (-sV)

```
# nmap -sV -T4 localhost
```

```
Starting Nmap ( http://nmap.org )
```

```
Nmap scan report for felix (127.0.0.1)
```

```
(The 1640 ports scanned but not shown below are in state: closed)
```

PORT	STATE	SERVICE	VERSION
21/tcp	open	ftp	WU-FTPD wu-2.6.1-20
22/tcp	open	ssh	OpenSSH 3.1p1 (protocol 1.99)
53/tcp	open	domain	ISC BIND 9.2.1
79/tcp	open	finger	Linux fingerd
111/tcp	open	rpcbind	2 (rpc #100000)
443/tcp	open	ssl/http	Apache httpd 2.0.39 ((Unix) mod_perl/1.99_04-dev)
515/tcp	open	printer	
631/tcp	open	ipp	CUPS 1.1
953/tcp	open	rndc?	
5000/tcp	open	ssl/ftp	WU-FTPD wu-2.6.1-20
5001/tcp	open	ssl/ssh	OpenSSH 3.1p1 (protocol 1.99)
5002/tcp	open	ssl/domain	ISC BIND 9.2.1
5003/tcp	open	ssl/finger	Linux fingerd
6000/tcp	open	X11	(access denied)
8000/tcp	open	http-proxy	Junkbuster webproxy
8080/tcp	open	http	Apache httpd 2.0.39 ((Unix) mod_perl/1.99_04-dev)
8081/tcp	open	http	Apache httpd 2.0.39 ((Unix) mod_perl/1.99_04-dev)

```
Nmap done: 1 IP address (1 host up) scanned in 42.494 seconds
```



-sV o -O?

- Ambas!
 - Opción -A activa ambos tipos de escaneo
- En algunos casos, como un proxy firewall reenviando a una aplicación en otro host, las respuestas pueden diferir
 - TCP/IP fingerprinting identificará el proxy
 - Escaneo de versión generalmente detectará el servidor que está ejecutando la aplicación en el proxy (aplicación "proxificada")
- Incluso cuando no hay proxy o reenvío de puertos, usar ambas técnicas es beneficioso
 - Si devuelven el mismo valor eso hace los resultados más creíbles
 - Si devuelven resultados muy diferentes investigar para determinar qué está pasando antes de confiar en cualquiera de los dos

Leer lista de hosts/redes desde un archivo (IPv4)

- Creamos el archivo lista.txt (puede ser a partir de un escaneo previo)

```
# cat > lista.txt  
  
scanme.nmap.org  
localhost
```

- Le pasamos la lista a nmap

```
# nmap -iL lista.txt
```


Excluir hosts/redes (IPv4)

- Ejemplo:

```
# nmap 192.168.1.0/24 --exclude 192.168.1.1,192.168.1.4
```

- Ejemplo:

```
nmap -iL lista.txt --excludefile excluir.txt
```

Protección contra el fingerprinting

- Ofuscadores (*fingerprint scrubbing*)
- Limitando el tipo y la cantidad de tráfico de respuesta
 - P.ej.: Bloquear tráfico ICMP saliente no necesario

Otras herramientas

- Xprobe2 (activo)
- p0f (pasivo)
- NetworkMiner (pasivo)
- Ettercap. TCP/IP stack fingerprinting (pasivo)
- ZMap

- Pruébalas!

Especificación de objetivos

Direcciones IP, nombres de sistemas, redes, etc

Ejemplo: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254

-iL fichero lista en fichero -iR n elegir objetivos aleatoriamente, 0 nunca acaba
--exclude --excludefile fichero excluir sistemas desde fichero

Descubrimiento de sistemas

-PS n tcp syn ping -PA n ping TCP ACK -PU n ping UDP
-PM Netmask Req -PP Timestamp Req -PE Echo Req
-sL análisis de listado -PO ping por protocolo -PN No hacer ping
-n no hacer DNS -R Resolver DNS en todos los sistemas objetivo
--traceroute: trazar ruta al sistema (para topologías de red)
-sP realizar ping, igual que con -PP -PM -PS443 -PA80

Técnicas de análisis de puertos

-sS análisis TCP SYN -sT análisis TCP CONNECT -sU análisis UDP
-sY análisis SCTP INIT -sZ COOKIE ECHO de SCTP -sO protocolo IP
-sW ventana TCP -sN -sF -sX NULL, FIN, XMAS -sA TCP ACK

Especificación de puertos y orden de análisis

-p n-m rango -p- todos los puertos -p n,m,z especificados
-p U:n-m,z T:n,m U para UDP, T para TCP -F rápido, los 100 comunes
--top-ports n analizar los puertos más utilizados -r no aleatorio

Duración y ejecución

-T0 paranoico -T1 sigiloso -T2 sofisticado
-T3 normal -T4 agresivo -T5 locura
--min-hostgroup --max-hostgroup
--min-rate --max-rate
--min-parallelism --max-parallelism
--min-rtt-timeout --max-rtt-timeout --initial-rtt-timeout
--max-retries --host-timeout --scan-delay

Ejemplos

Análisis rápido nmap -T4 -F
Análisis rápido (puerto 80) nmap -T4 --max_rtt_timeout 200 --initial_rtt_timeout 150 --min_hostgroup 512 --max_retries 0 -n -P0 -p80
Análisis de ping nmap -sP -PE -PP -PS21,23,25,80,113,31339 -PA80,113,443,10042 --source-port 53 -T4
Exhaustivo lento nmap -sS -sU -T4 -A -v -PE -PP -PS21,22,23,25,80,113,31339 -PA80,113,443,10042 -PO --script all
Trazado de ruta rápido nmap -sP -PE -PS22,25,80 -PA21,23,80,3389 -PU -PO --traceroute

Detección de servicios y versiones

-sV: detección de la versión de servicios --all-ports no excluir puertos
--version-all probar cada exploración
--version-trace rastrear la actividad del análisis de versión

-O activar detección del S. Operativo --fuzzy adivinar detección del SO
--max-os-tries establecer número máximo de intentos contra el sistema objetivo

Evasión de Firewalls/IDS

-f fragmentar paquetes -D d1,d2 encubrir análisis con señuelos
-S ip falsear dirección origen -g source falsear puerto origen
--randomize-hosts orden --spoof-mac mac cambiar MAC de origen

Parámetros de nivel de detalle y depuración

-v Incrementar el nivel de detalle --reason motivos por sistema y puerto
-d (1-9) establecer nivel de depuración --packet-trace ruta de paquetes

Opciones interactivas

v/V aumentar/disminuir nivel de detalle del análisis
d/D aumentar/disminuir nivel de depuración
p/P activar/desactivar traza de paquetes

Otras opciones

--resume file continuar análisis abortado (tomando formatos de salida con -oN o -oG)
-6 activar análisis IPV6
-A agresivo, igual que con -O -sV -sC --traceroute

Scripts

-sC realizar análisis con los scripts por defecto --script file ejecutar script (o todos)
--script-args n=v proporcionar argumentos
--script-trace mostrar comunicación entrante y saliente

Formatos de salida

-oN normal -oX XML -oG programable -oA todos



Nmap 5
cheatsheet

Bibliografía

- Lyon, G. F. (2008). ***Nmap Network Scanning: Official Nmap Project Guide to Network Discovery and Security Scanning***. Sunnyvale, CA: Insecure.Com LLC.
 - Versión on-line (no completa): <http://nmap.org/book/toc.html>



Enlaces de interés

- NMAP6 CHEATSHEET (Security by Default):
<http://www.securitybydefault.com/2013/12/nmap6-cheatsheet.html>
- hping Project: www.hping.org
- Lyon, G. F. (2012). **SecTools.Org: Top 125 Network Security Tools**. Disponible en:
<http://sectools.org>
- Lyon, G. F. (2008). Nmap: Scanning the Internet. DEFCON 16.
<http://www.youtube.com/watch?v=Hk-21p2m8YY> #video
- Stanger, J. Security testing with hping
www.linuxpromagazine.com/index.php/content/download/62147/484014/file/038-041_hping.pdf

