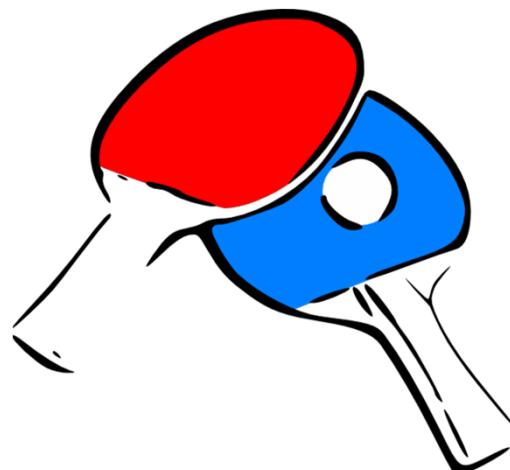


## 7. ESCANEO

# PING

Ping es un comando existente en cualquier sistema operativo que nos permite saber si un dispositivo IP esta conectado a internet.

Para lograr su objetivo, utiliza el protocolo ICMP, el cual envía una paquete y espera una respuesta, dicha respuesta puede o no llegar, dependiendo del contenido de la respuesta se define si un dispositivo esta conectado o no.



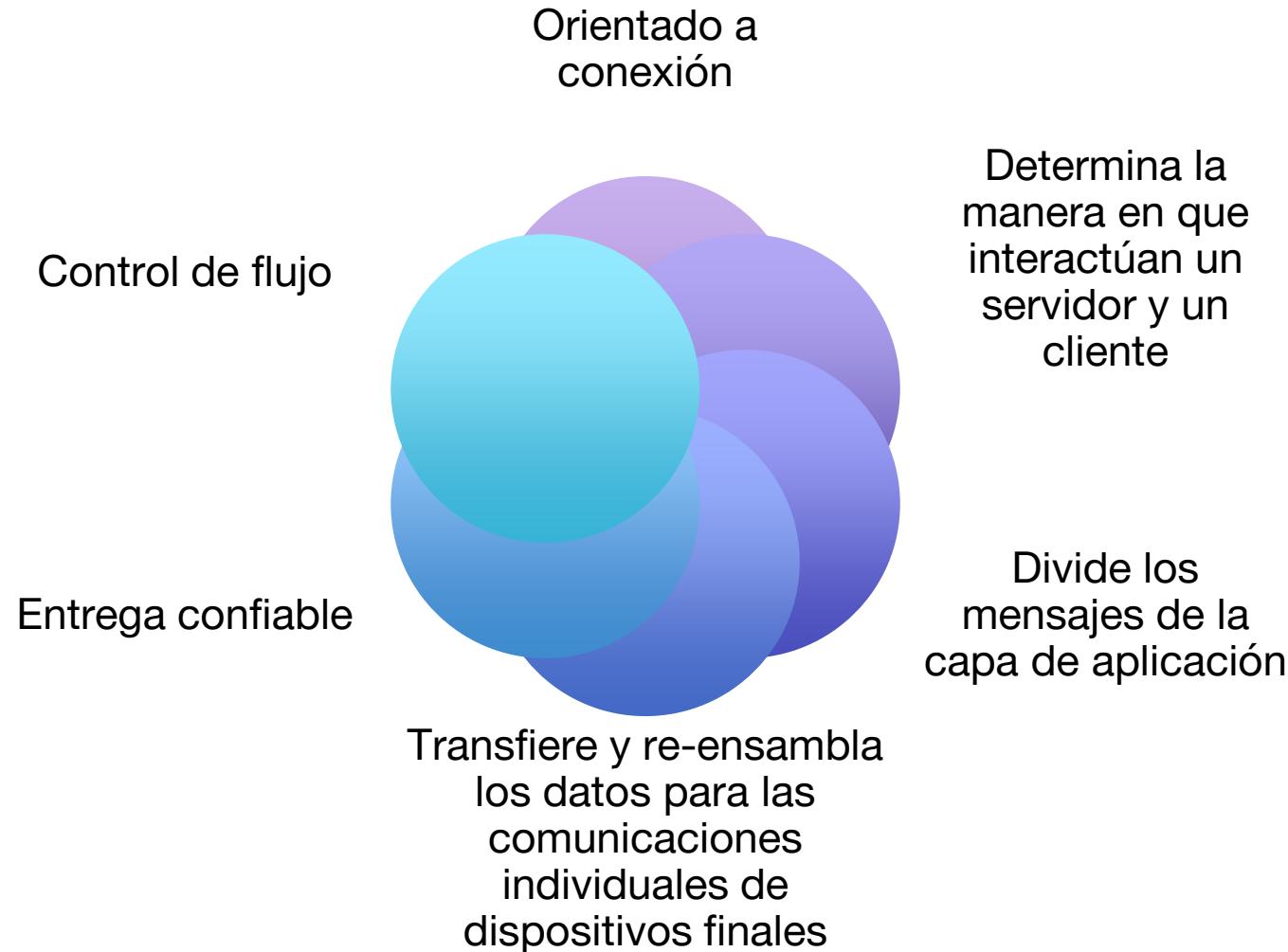
# ¿Qué nos puede decir PING?

Cuando se envía un comando ping, este tiene dentro del encabezado del protocolo IP metadatos que indican ciertas características especiales, como el TTL, o time to live. El TTL es el número de saltos que un paquete dará antes de que sea descartado por algún router. El TTL nos puede indicar el sistema operativo de nuestro objetivo

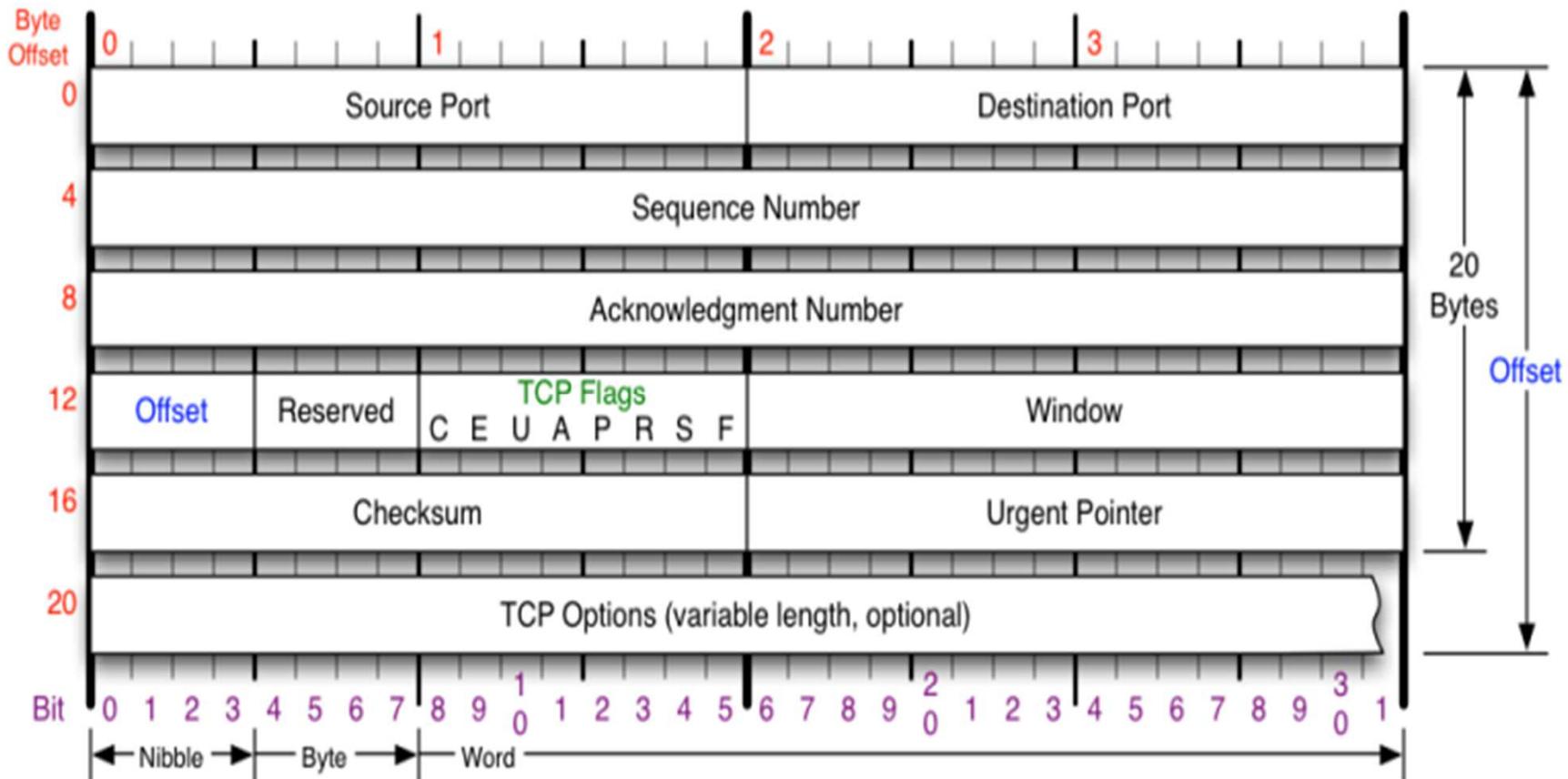
- Windows: TTL 128
- Linux: TTL 66 o 255
  - vm1chaac.ccg.unam.mx
  - google.com



# TCP



# Cabecera TCP



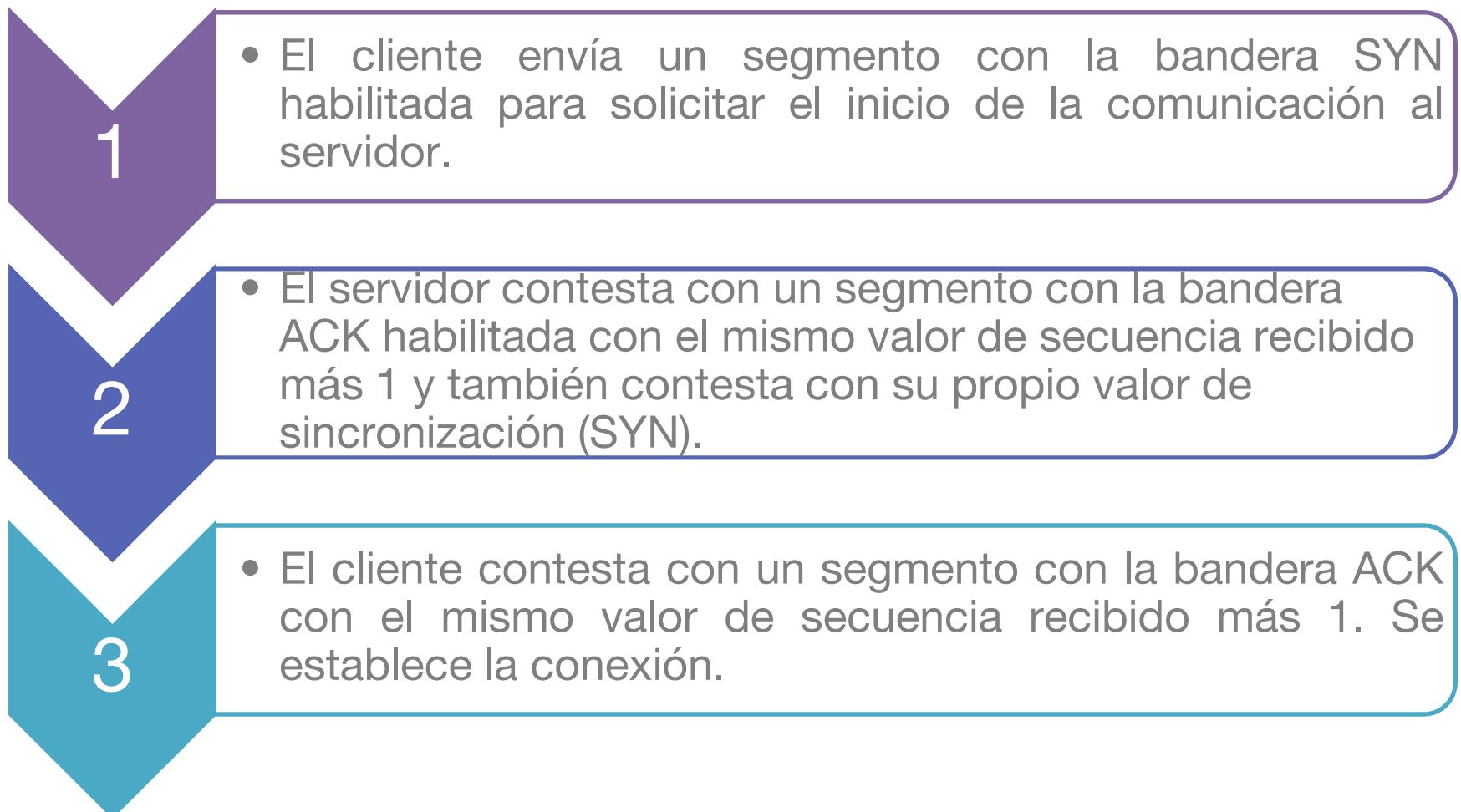
(nmap.org, 2016)

# TCP

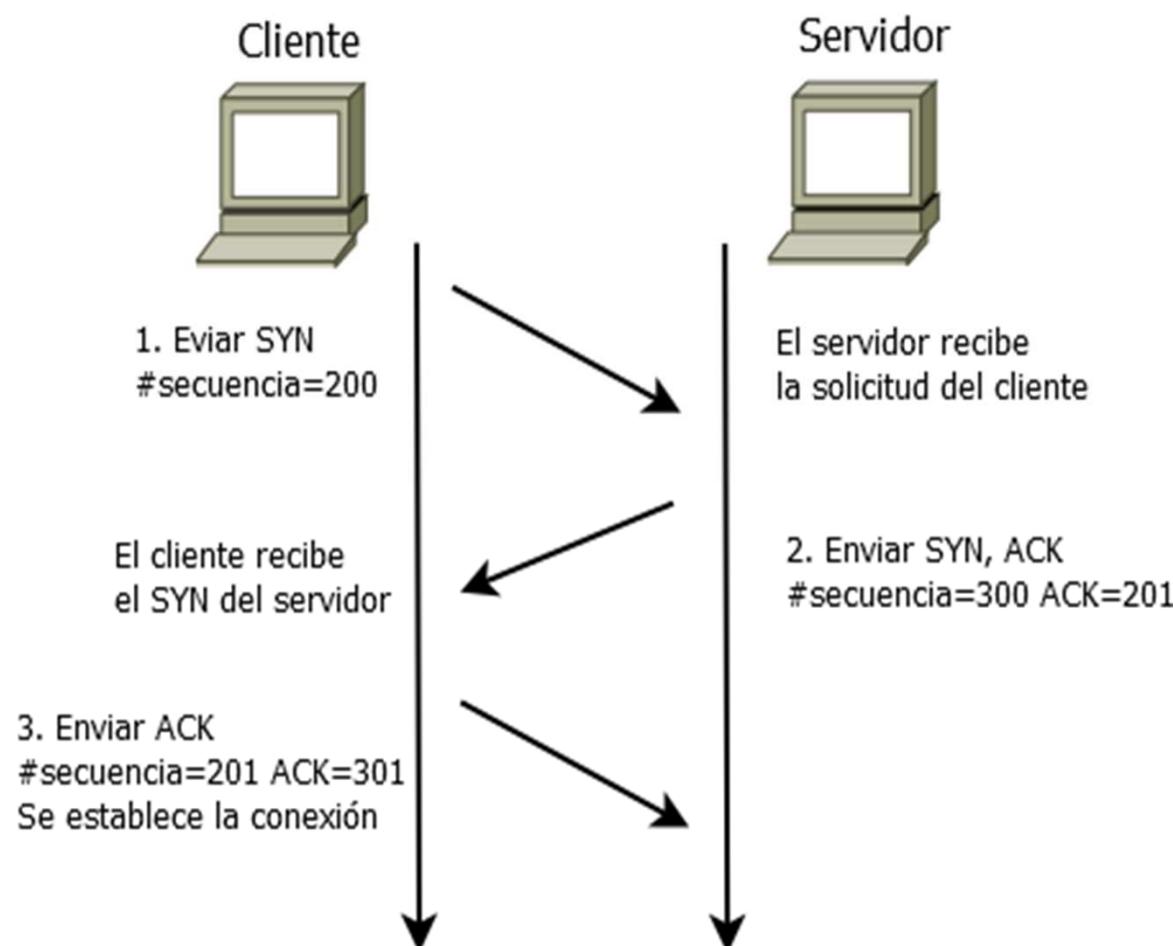
Las banderas TCP se especifican en 1 byte de la cabecera, cada bit es una bandera y se utilizan para gestionar los procesos TCP:

Bandera	Representación en hexadecimal	Descripción de bandera	Descripción
C	0x80	CWR	Indica una congestión en la ventana
E	0x40	ECN Echo	Notificación de congestión
U	0x20	Urgent	Indica que el segmento es de prioridad urgente
A	0x10	Ack	Conocimiento
P	0x08	Push	Push de datos
R	0x04	Reset	Cierre de conexión forzoso
S	0x02	Syn	Sincronización
F	0x01	Fin	Cierre de conexión normal

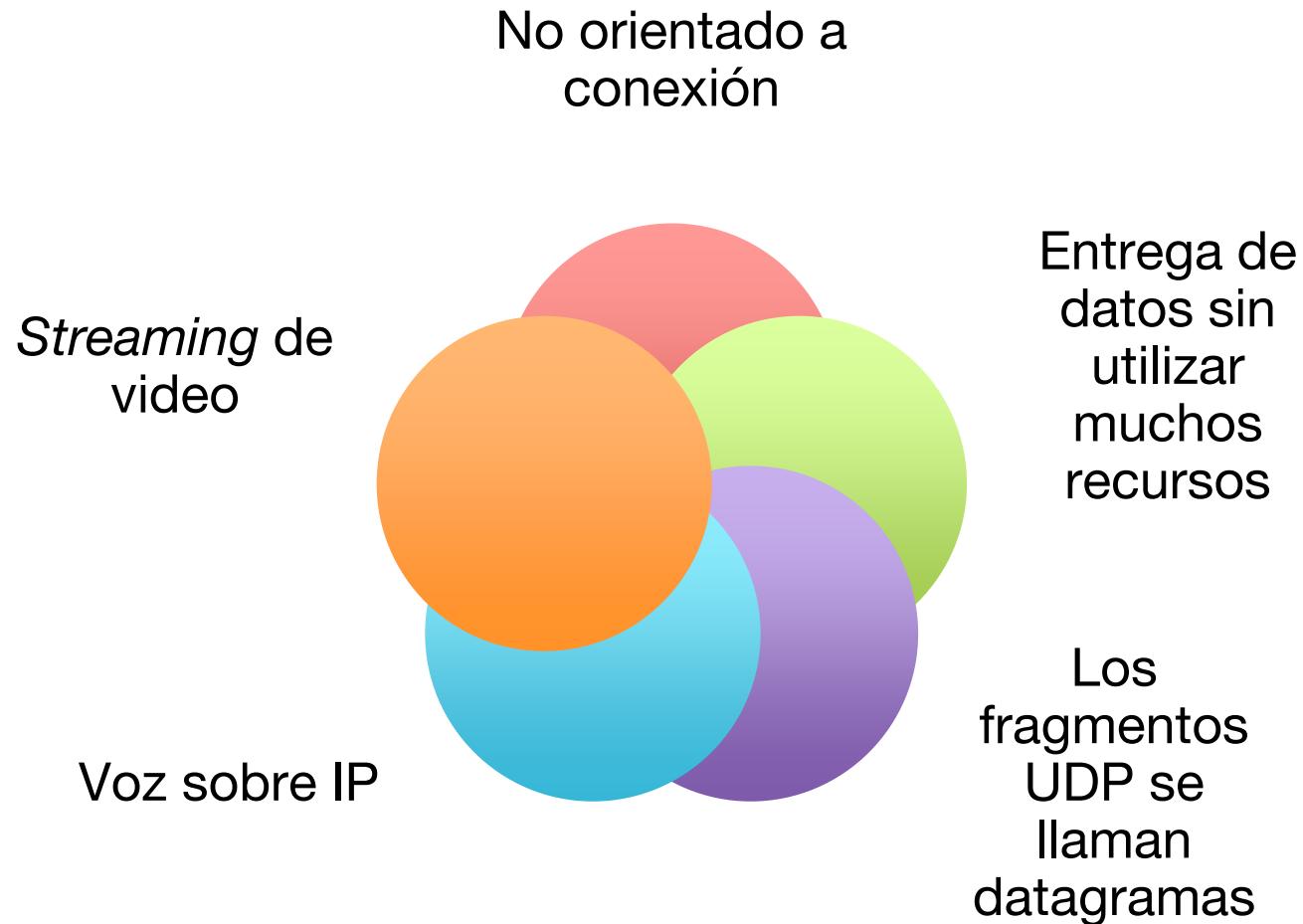
# Comunicación de tres vías



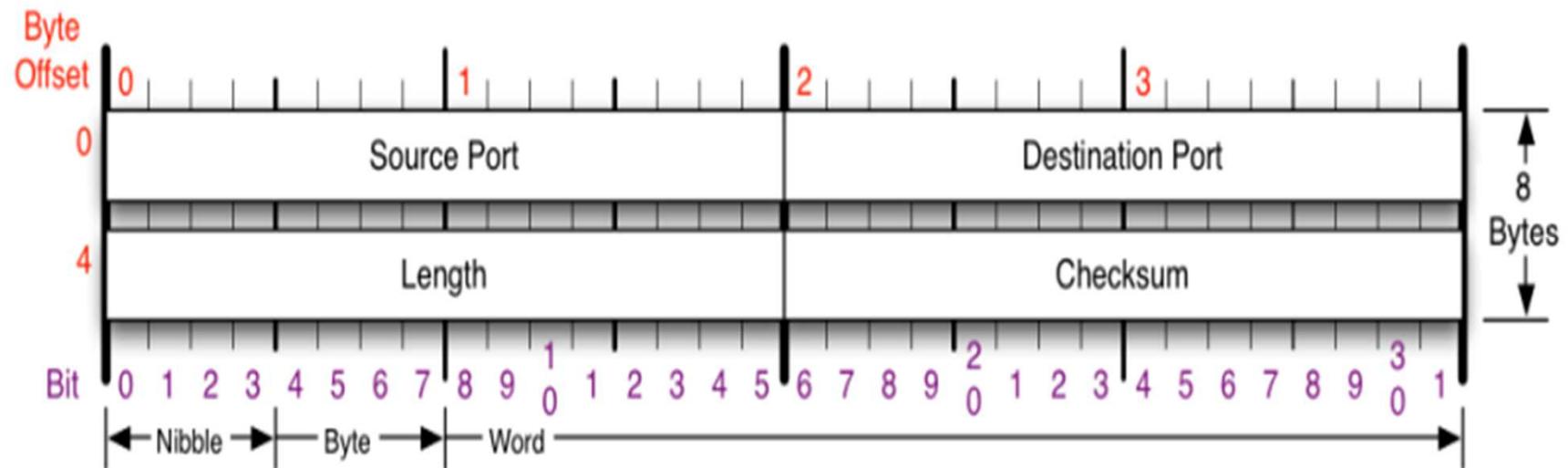
# Comunicación de tres vías (continuación)



# UDP



# Cabecera udp

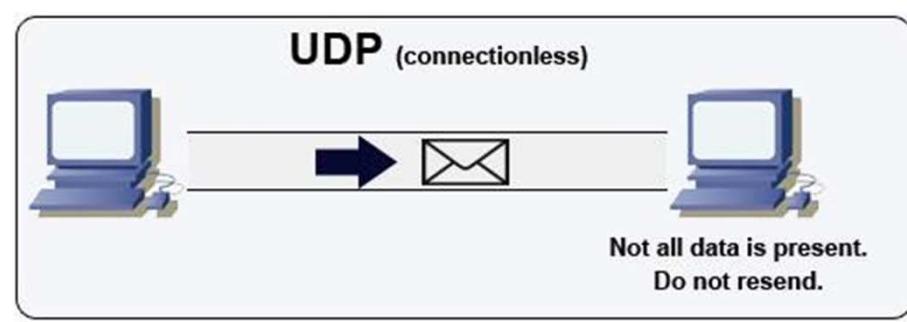
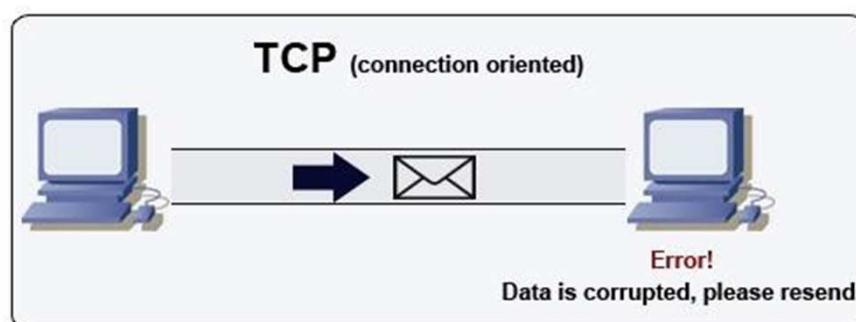


(nmap.org, 2016)

# TCP vs UDP

Se dice TCP está orientado a conexión, mientras que UDP no lo esta.

Lo que la afirmación anterior significa es que el protocolo TCP, lleva a cabo un proceso de verificación de recepción de información entre fuente y receptor, para que puedan llevar a cabo la transmisión de información. UDP solo envía la información, no requiere verificación de conexión, esto debido a que existen servicios que requieren velocidad.



# Internet Protocol

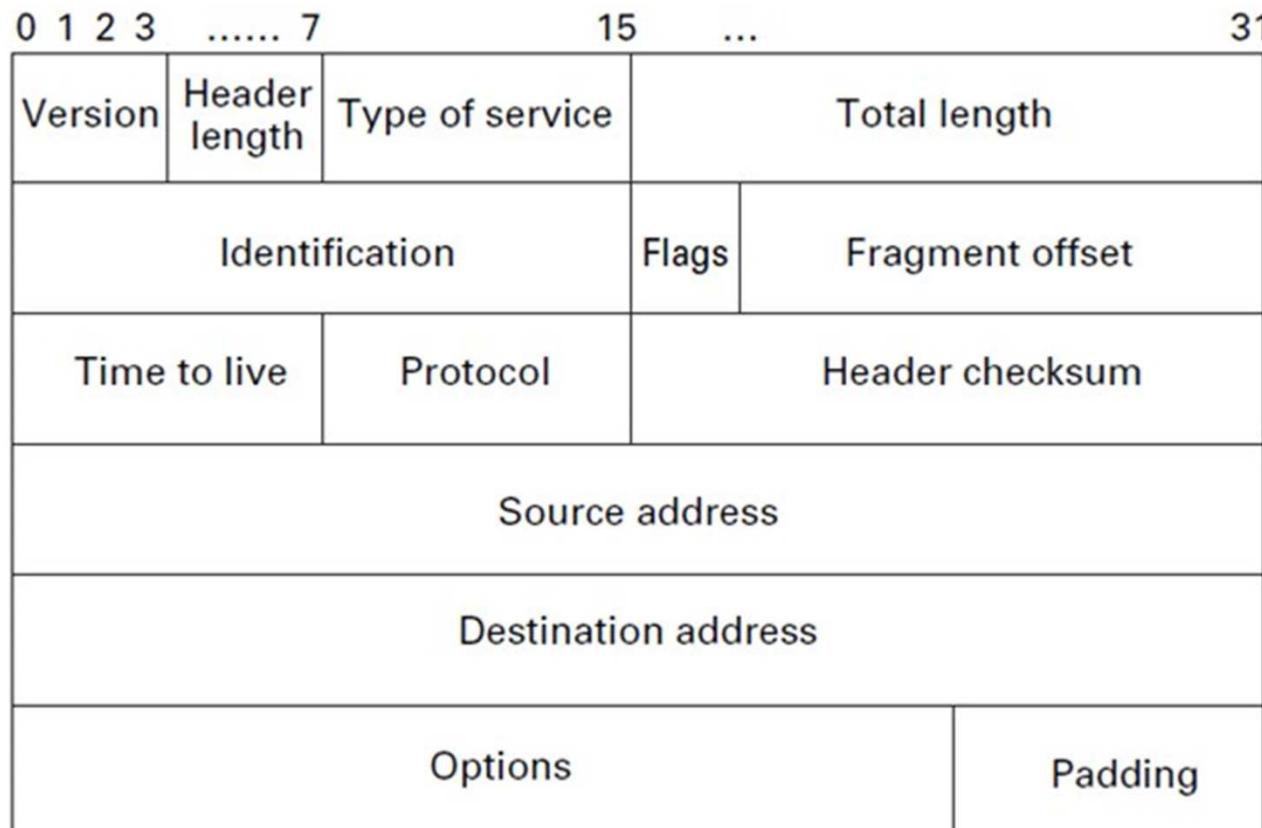
La dirección IP es un identificador que permite que los paquetes de datos sean encaminados a su destino correcto.

Existen rangos de Ips públicas y privadas.

Class	Private Address Range
A	10.0.0.0 to 10.255.255.255
B	172.16.0.0 to 172.31.255.255
C	192.168.0.0 to 192.168.255

# Internet Protocol Header

RFC 791



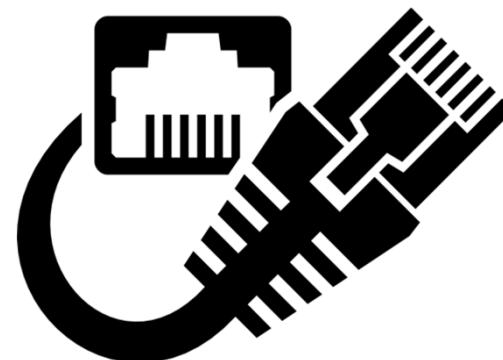
IP header

# Sockets

¿Cómo logran los protocolos de red hacer su trabajo y conectarse a otros equipos?

La respuesta son sockets. Formalmente un socket se describe como un descriptor de archivo dentro de un sistema operativo, esto es la abstracción de un dispositivo o archivo que interactúa con el kernel del sistema operativo con el fin de leer información como flujos de bits.

Un socket, en términos simples, es el punto terminal de una conexión.



# Puertos

Dado que un equipo puede generar varias conexiones por servicio, se crean varios sockets para que se pueda transmitir o leer información.

Estos sockets necesitan un identificador cuando se recibe o se leer información, de lo contrario no habría una especificación del punto preciso del cual leer datos. Por ejemplo, por defecto un servidor web utiliza los puertos 443 y 80 para transmitir información por el protocolo HTTP, si no se tuviera esa especificación numérica el servidor solo podría realizar una conexión a la vez.

Un puerto es un número que permite multiplexar conexiones al identificar sockets para transmitir o recibir información.

# Clasificación

Existen 65535 puertos posibles. Se clasifican en:

Puertos bien conocidos. (1-1023)

Puertos registrados. (1024-49151)

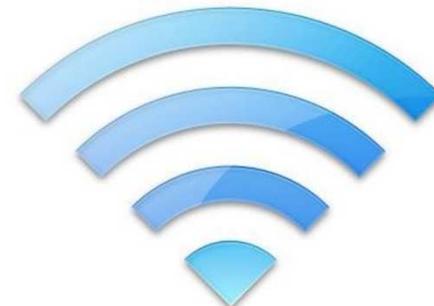
Puertos dinámicos o privados. (49152 - 65535)

Se puede cambiar el puerto por defecto de una aplicación, esto puede ser una buena práctica de seguridad, a esto se le conoce como seguridad por obscuridad

# MAC Address

El protocolo ARP es un protocolo que asocia la dirección MAC de un dispositivo con una dirección IP, esto para tener el control de una red interna y que los paquetes lleguen a su destino a través de una dirección física. Una dirección física o MAC, es única por lo que se puede identificar un dispositivo por esta dirección en cualquier parte del mundo.

El comando ARP en Windows, nos puede dar una idea de como funciona además de “mapear” direcciones físicas a direcciones IP que se encuentran en el mismo segmento de red.



aa:bb:cc:dd:ee:ff

# Escaneo



(nmap.org, 2016)

Fase donde se aplican técnicas para conocer más acerca de los objetivos de evaluación.

Permite identificar equipos activos en la red, sistema operativo, estado de los puertos, servicios en ejecución y vulnerabilidades asociadas.

# Sobre escaneo

Puede llegar a generar una gran cantidad de alertas de seguridad, sobre todo con tipos específicos de escaneo, por lo que se debe tener cuidado si mantenerse anónimo es una prioridad.

Es fundamental para las pruebas de penetración, es una de las etapas que más información nos dará, es importante tener en cuenta que generar un reporte, puede ser de gran ayuda también para el pentester, ya que si un vector de ataque no es el adecuado, se puede probar con otro.

# Practica

Crear la máquina virtual metasploitable2as virtuales.

Averiguar IP de KaliLinux con el comando ifconfig

# Tipos de escaneo

## *Network Sweep*

- Identificar dispositivos IP en una red.

## Escaneo de puertos

- Determinar el estado de los puertos TCP o UDP de un dispositivo en red.

## *Fingerprinting SO*

- Identificar el sistema operativo de un dispositivo de acuerdo a su comportamiento.

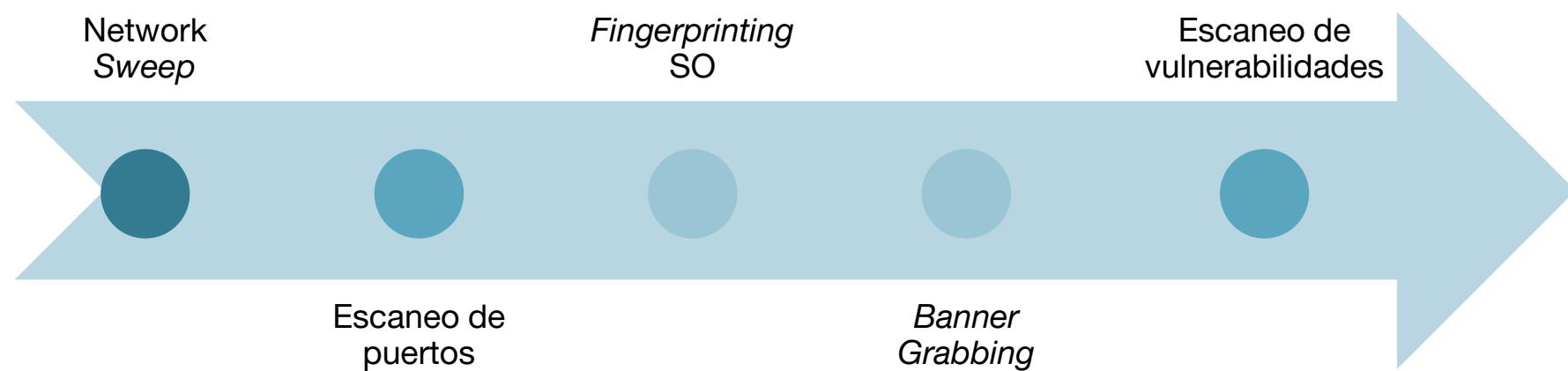
## *Banner grabbing*

- Descubrir los servicios y su versión mediante mensajes desplegados y contenidos en aplicaciones.

## Escaneo de vulnerabilidades

- Obtener una lista de posibles vulnerabilidades, malas configuraciones, software desactualizado, entre otros.

# Flujo de escaneo



NETCAT

# Netcat

Netcat es considerado una navaja suiza debido a la amplia versatilidad en sus opciones para conexiones de red.

Permitiendo establecer conexiones con protocolos TCP/UDP, redirigir la Entrada/Salida de comandos a través de red, usarlo como servidor o cliente entre otras características.



*(github.io 2017)*

# Netcat - Sintaxis básica (servidor)

Netcat es incluido en diversas distribuciones de GNU/Linux, puede ser ejecutado bajo el nombre de nc o netcat, ambos hacen referencia al mismo binario.

Para usarlo como servidor y asociarlo un puerto en específico el comando debe tener la siguiente sintaxis.

```
nv -lvp 23456
```

Opciones:

- l Listen: Modo escucha.
- v Verbose: Más información.
- p Port: Puerto donde espera una conexión.

# Netcat - Sintaxis básica

El binario netcat iniciará un servicio de red asociado al puerto 23456, esperando a la conexión con algún cliente.

```
root@kali:~# nc -lvp 23456
listening on [any] 23456 ...
```

# Netcat - Sintaxis básica (cliente)

Es posible establecer una comunicación de red con un equipo conociendo su dirección IP y un puerto asociado a un servicio, para esto se emplea la siguiente sintaxis.

```
nc -v <Dirección_IP_Servidor> <Puerto>
```

```
root@kali:~# nc -v localhost 23456
localhost [127.0.0.1] 23456 (?) open
```

# Netcat- Sintaxis básica

Se ha establecido una comunicación del cliente al servidor, esta arquitectura básica funciona como un chat, se envían cadenas de texto de una terminal a otra.

```
root@kali:~# nc -lvp 23456
listening on [any] 23456 ...
connect to [127.0.0.1] from localhost [127.0.0.1] 50072
hola, eres un servidor?
no, soy un netcat
```

Servidor.

```
root@kali:~# nc -v localhost 23456
localhost [127.0.0.1] 23456 (?) open
hola, eres un servidor?
no, soy un netcat
```

Cliente.

# Netcat – Escáner de puertos

Otra característica de Netcat es la identificación de puertos abiertos en un host, automatiza la conexión a un host de acuerdo a un rango:

```
nc -vz <Dirección_IP_Host> <Li>-<Ls>
```

Opciones:

- v      Verbose: Más información
- z      zero-I/O: Si encuentra un puerto abierto no envía información, se conecta y cierra la conexión inmediatamente.
- Li      Límite inferior: Puerto hasta donde terminará la conexión
- Ls      Límite superior: Puerto donde inicia el intento de conexión

# Netcat – Escáner de puertos

Se hizo un barrido de conexiones desde el puerto 85 al puerto 75, donde sólo identificó el puerto 80 abierto (http).

```
root@kali:~# nc -vvz localhost 75-85
localhost [127.0.0.1] 85 (?) : Connection refused
localhost [127.0.0.1] 84 (?) : Connection refused
localhost [127.0.0.1] 83 (?) : Connection refused
localhost [127.0.0.1] 82 (?) : Connection refused
localhost [127.0.0.1] 81 (?) : Connection refused
localhost [127.0.0.1] 80 (http) : Connection refused
localhost [127.0.0.1] 79 (finger) : Connection refused
localhost [127.0.0.1] 78 (?) : Connection refused
localhost [127.0.0.1] 77 (?) : Connection refused
localhost [127.0.0.1] 76 (?) : Connection refused
localhost [127.0.0.1] 75 (?) : Connection refused
sent 0, rcvd 0
```

# Netcat – Transferencia de archivos

Enviar:

```
|└ $nc 127.0.0.1 -q 0 -lvp 9999 < archivo  
listening on [any] 9999 ...  
connect to [127.0.0.1] from localhost [127.0.0.1] 59330
```

Recibir:

```
|└ $nc 127.0.0.1 9999 > archivo
```

# Netcat – exec

Puerto en escucha:

```
|└ $ nc 127.0.0.1 -c /bin/bash -lvp 8888  
listening on [any] 8888 ...  
connect to [127.0.0.1] from localhost [127.0.0.1]
```

Cliente:

```
|└ $ nc 127.0.0.1 8888  
pwd  
/home/chaos
```

# Netcat – exec 2

Puerto en escucha:

```
[prueba@oink]$ mkfifo /tmp/f  
[prueba@oink]$ cat /tmp/f | sh -i 2>&1 | nc -lvp 12345 192.168.100.35 > /tmp/f  
Listening on [192.168.100.35] (family 0, port 12345)
```

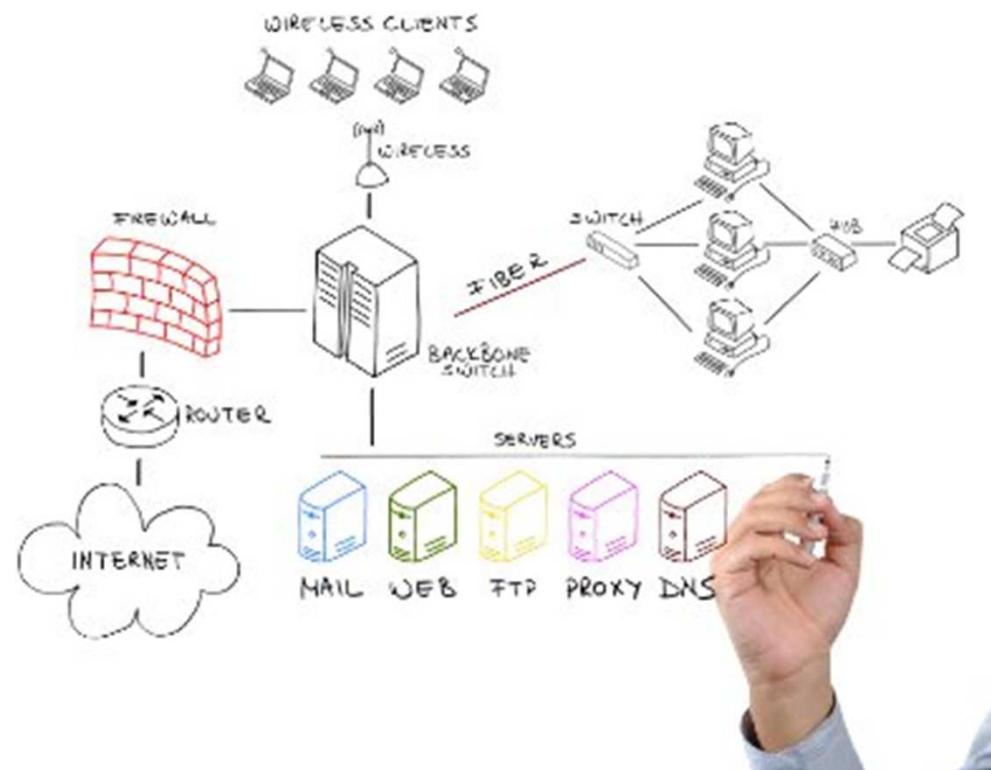
Cliente:

```
[otro@srv]$ nc 192.168.100.35 12345  
[prueba@oink]$ ls  
1.eml  
2.eml  
procesados  
[prueba@oink]$
```

NETDISCOVER

# Netdiscover

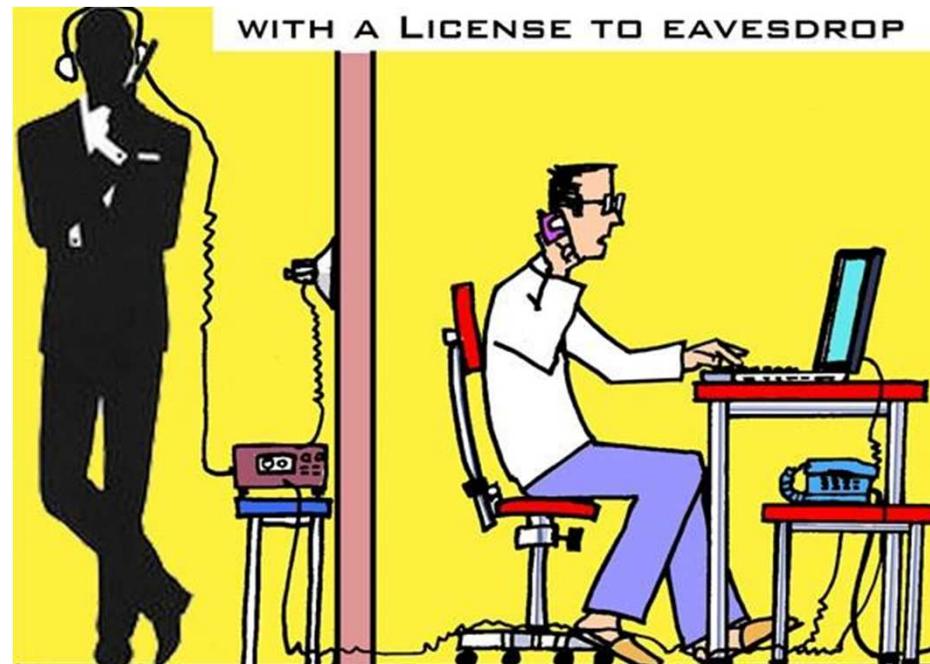
Dentro de la etapa de reconocimiento uno de los objetivos es identificar equipos en un segmento de red, una forma de hacer esto es usar la herramienta netdiscover.



(media.licdn.com, 2016)

# Netdiscover - Pasiva

Netdiscover trabaja de forma pasiva y activa usando el protocolo ARP, de forma pasiva entra en modo escucha y como un sniffer ve el tráfico que circula por la red (sólo paquetes ARP).



(media.cagle.com, 2016)

# Netdiscover - Activa

De forma activa, netdiscover envía un broadcast de paquetes ARP en busca de una dirección IP, las direcciones IP se pueden definir con una máscara de red.

29711 4... Dell_4e:76:55	Broadcast	ARP	42 who has 127.0.106.209? Tell 127.0.106.67
29712 4... Dell_4e:76:55	Broadcast	ARP	42 who has 127.0.106.210? Tell 127.0.106.67
29713 4... Dell_4e:76:55	Broadcast	ARP	42 who has 127.0.106.211? Tell 127.0.106.67
29714 4... Dell_4e:76:55	Broadcast	ARP	42 who has 127.0.106.212? Tell 127.0.106.67
29715 4... Dell_4e:76:55	Broadcast	ARP	42 who has 127.0.106.213? Tell 127.0.106.67
29716 4... Dell_4e:76:55	Broadcast	ARP	42 who has 127.0.106.214? Tell 127.0.106.67
29717 4... Dell_4e:76:55	Broadcast	ARP	42 who has 127.0.106.215? Tell 127.0.106.67
29718 4... Dell_4e:76:55	Broadcast	ARP	42 who has 127.0.106.216? Tell 127.0.106.67
29719 4... Dell_4e:76:55	Broadcast	ARP	42 who has 127.0.106.217? Tell 127.0.106.67
29720 4... Dell_4e:76:55	Broadcast	ARP	42 who has 127.0.106.218? Tell 127.0.106.67

# Netdiscover

Debemos conocer la dirección IP y máscara de red que asigna el servidor DHCP, con el comando siguiente se obtiene esa información:

`ifconfig <interfaz de red>`

```
root@kali:~# ifconfig eth1
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.14 netmask 255.255.255.0 broadcast 10.0.2.255
        ether 08:00:27:11:c3:36 txqueuelen 1000 (Ethernet)
        RX packets 2 bytes 1180 (1.1 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 18 bytes 1860 (1.8 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    inet6 fe80::a00:27ff:fe11:c336 prefixlen 64 scopeid 0x20<link>
```

Segmento: 10.0.2.0/24

# Netdiscover

De manera predeterminada se identifican host de forma activa empleando la siguiente sintaxis.

```
netdiscover -i eth1 -r 10.0.2.0/24
```

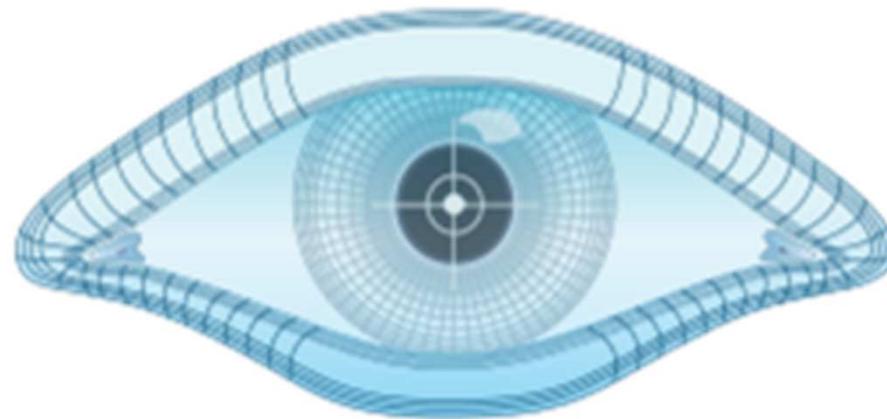
Opciones:

-i      Device:      Interface de red

-r      range: Segmento de red con direcciones IP a probar.

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
-----				
10.0.2.1	52:54:00:12:35:00	1	60	Unknown vendor
10.0.2.2	52:54:00:12:35:00	1	60	Unknown vendor
10.0.2.3	08:00:27:7b:fe:50	1	60	PCS Systemtechnik GmbH
10.0.2.16	08:00:27:77:d3:f7	1	60	PCS Systemtechnik GmbH

# Nmap



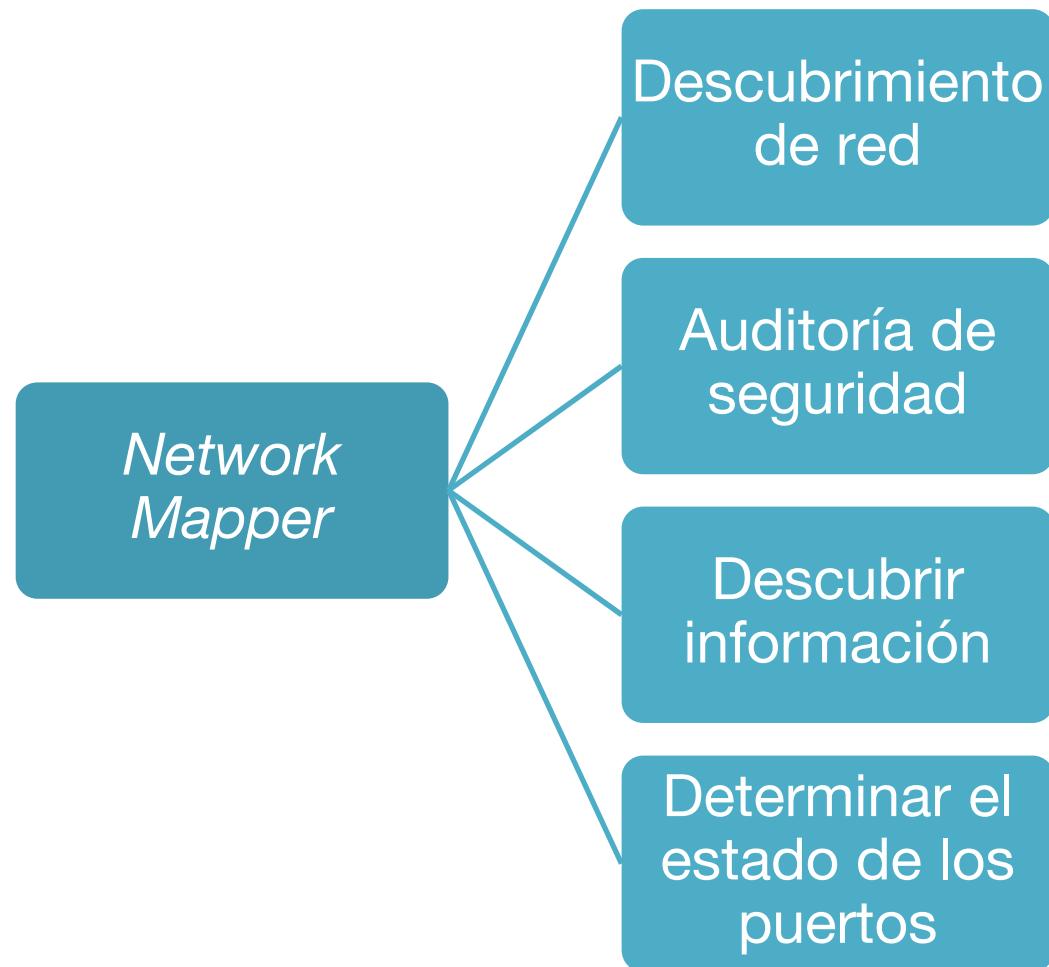
← → C

ⓘ No es seguro | <http://ftp.cerias.purdue.edu/pub/tools/unix/scanners/strobe/strobe.c>

```
/*
 * Strobe (c) 1995-1997 Julian Assange (proff@suburbia.net),
 * All rights reserved.
 *
 * $ cc strobe.c -o strobe
 */

#define VERSION "1.05"
```

# nmap



# Nmap (continuación)

- Ayuda a determinar los dispositivos activos en una red
- Identifica servicios y sus versiones que se encuentran ejecutándose en los dispositivos
- Identifica el sistema operativo de los hosts (*fingerprinting OS*)
- Identifica si los puertos están abiertos, cerrados o filtrados

Otras características



# Estado de puertos

Open

- Significa que una aplicación esta en escucha para aceptar conexiones.

Filtered

- Significa que un firewall u otro mecanismo de seguridad está bloqueando el puerto, por tanto nmap no puede determinar si el puerto está abierto o cerrado.

Closed

- Los puertos cerrados no tienen una aplicación o servicios activos.

Unfiltered

- Es cuando los puertos responden a nmap pero no se puede determinar el estado del puerto.

Open|Filtered

- Nmap no puede determinar si el puerto está filtrado o abierto.

Closed|Filtered

- Nmap no puede determinar si el puerto está cerrado o filtrado.

# Nmap (continuación)

¿Cómo nmap determina el estado de los puertos cuando se utiliza TCP?

# Escenario 1

Se envía una señal de sincronización (SYN).

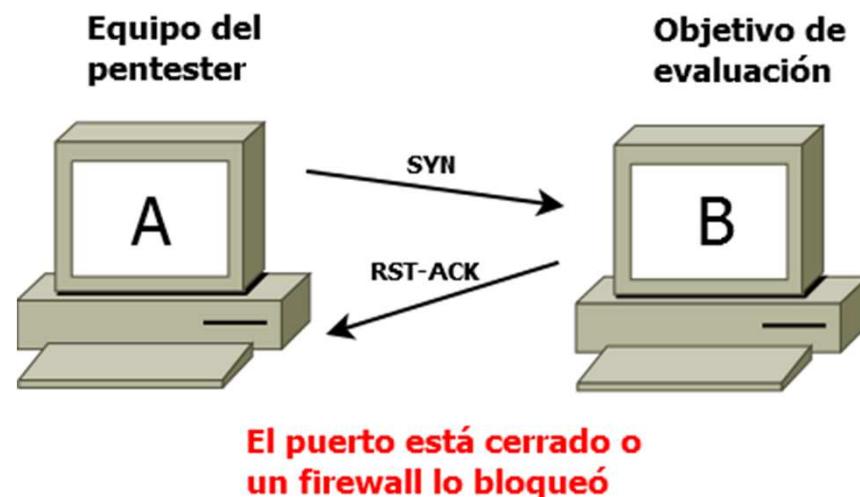
El puerto está abierto si la respuesta del objetivo contiene las banderas SYN y ACK.



## Escenario 2

Se envía una señal de sincronización (SYN).

El puerto está cerrado o bloqueado si la respuesta del objetivo contiene las banderas RST y ACK

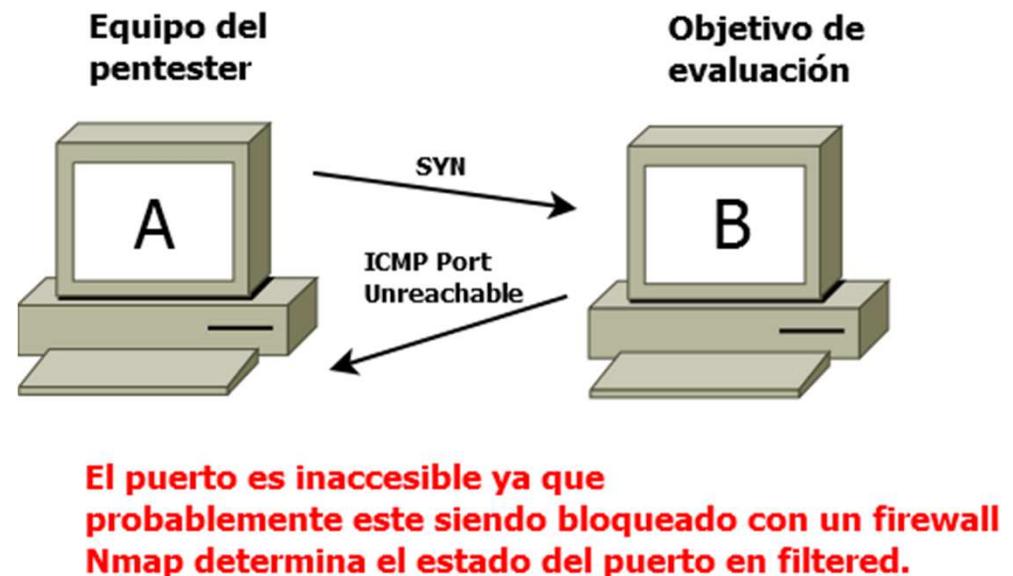


# Escenario 3

Se envía una señal de sincronización (SYN).

El puerto es inaccesible si la respuesta del objetivo es un mensaje ICMP Port Unreachable.

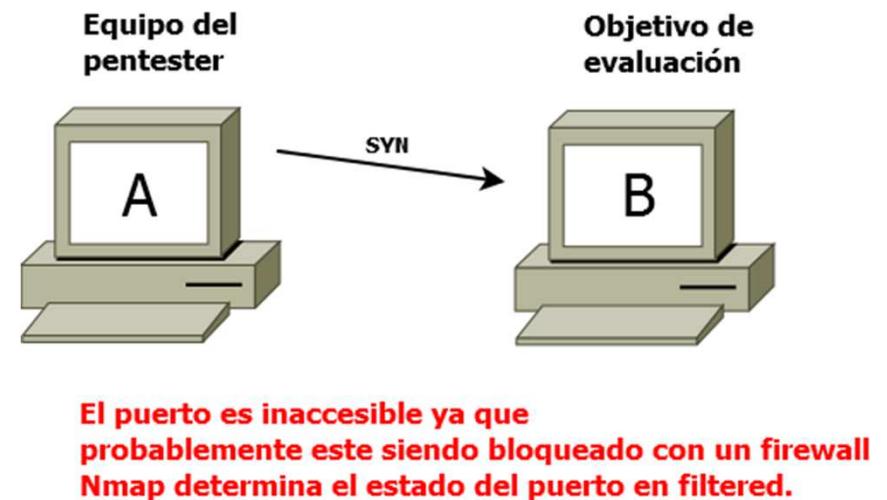
El puerto está siendo bloqueado por un firewall.



# Escenario 4

Se envía una señal de sincronización (SYN).

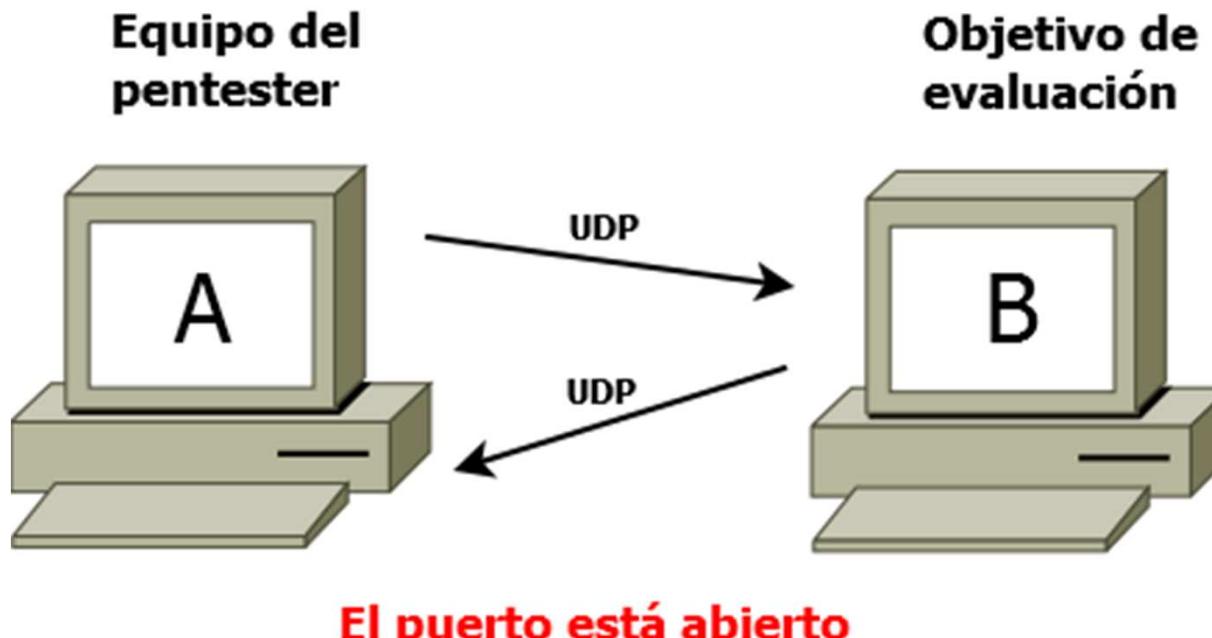
Si la respuesta del objetivo de evaluación no emite ninguna respuesta, significa que el puerto es inaccesible debido a que está siendo bloqueado por un firewall.



# Nmap (continuación)

¿Cómo nmap determina el estado de los puertos cuando se utiliza UDP?

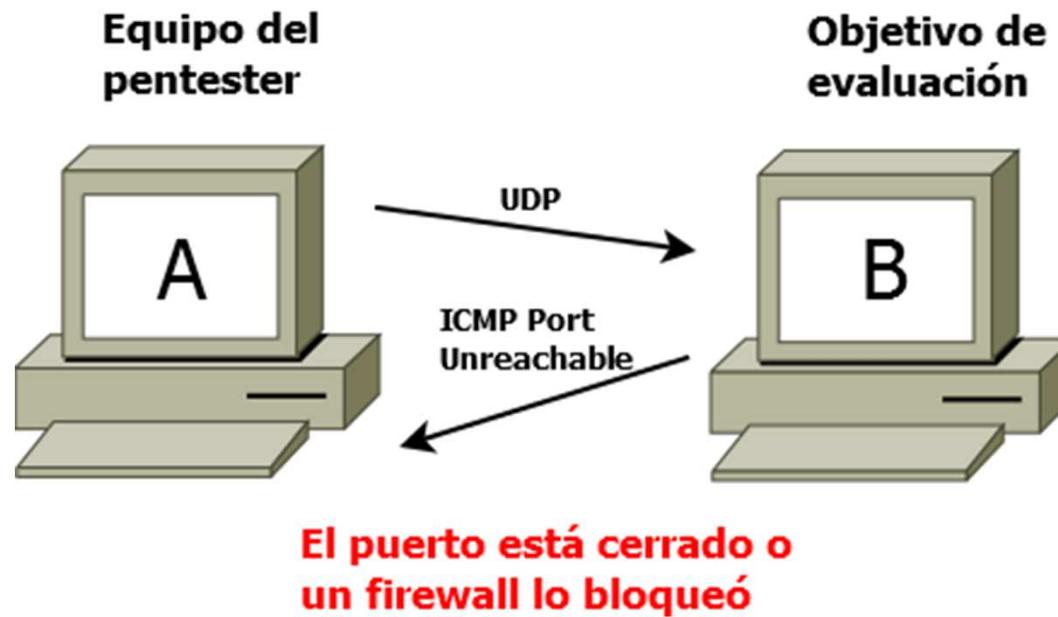
# Escenario 5



Se envía un datagrama UDP.

El puerto está abierto si la respuesta del objetivo de evaluación responde con otro datagrama UDP.

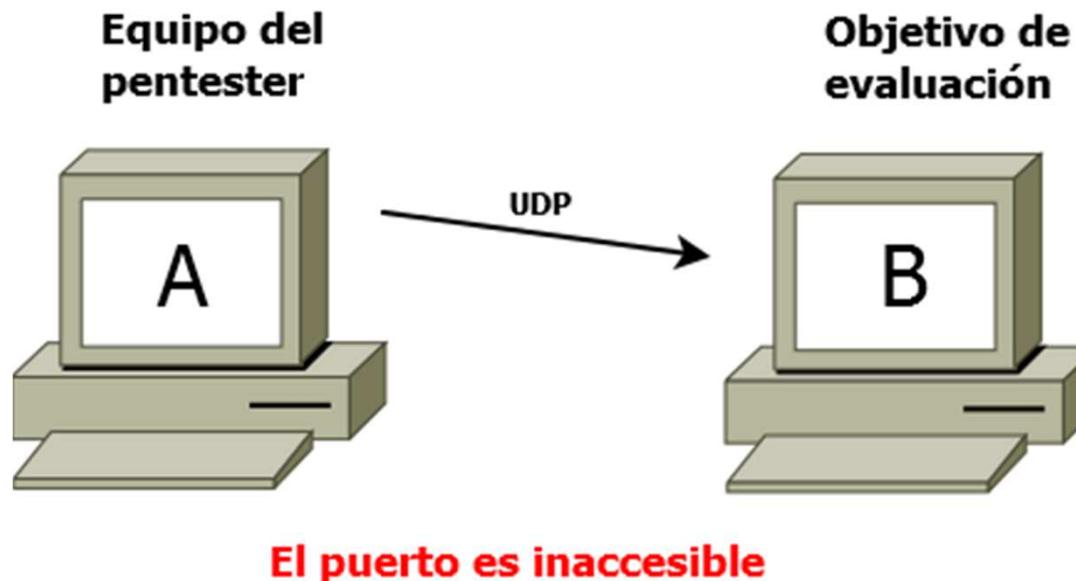
# Escenario 6



Se envía un datagrama UDP.

El puerto está cerrado o bloqueado si la respuesta del objetivo de evaluación responde con ICMP Port Unreachable.

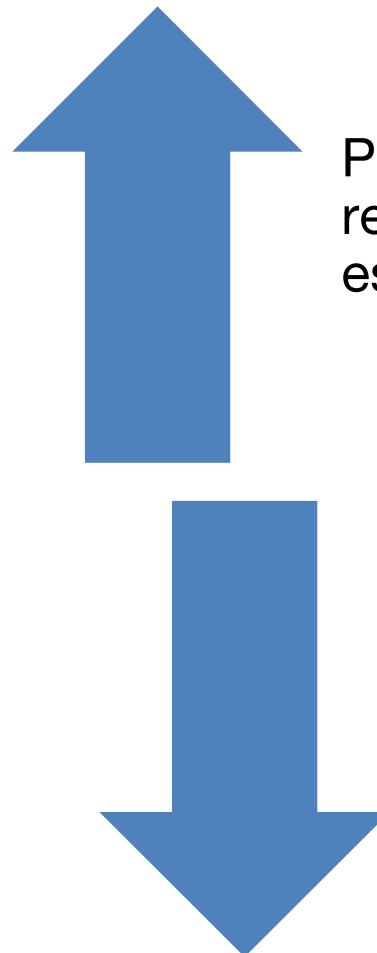
# Escenario 7



Se envía un datagrama UDP.

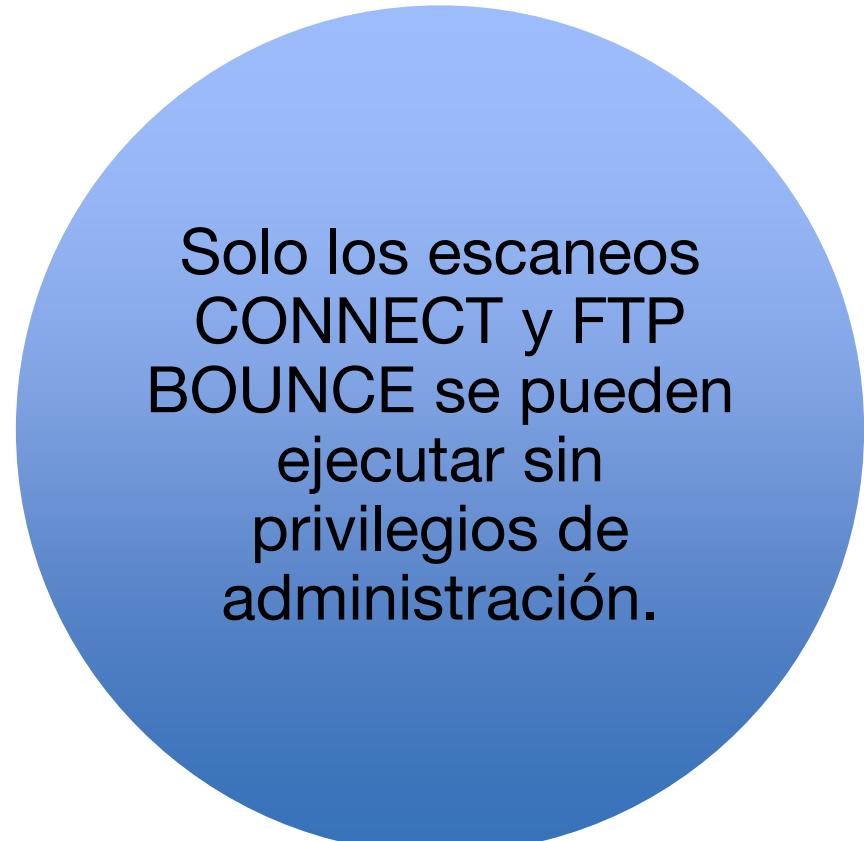
Si el objetivo no emite respuesta, el puerto es inaccesible porque: el puerto está cerrado, un *firewall* está bloqueando paquetes UDP entrantes o respuestas salientes, el puerto está abierto pero solo se buscaron datos especificados en el *payload* UDP.

# Técnicas de escaneo con nmap



Por defecto nmap realiza un escaneo SYN.

En caso de no contar con suficientes permisos para enviar paquetes en crudo, nmap lo intercambia por un escaneo CONNECT.



Solo los escaneos CONNECT y FTP BOUNCE se pueden ejecutar sin privilegios de administración.

# Técnicas de escaneo con nmap (continuación)

-sS (escaneo TCP SYN)

- También conocido como half open porque no completa una conexión TCP.

# Técnicas de escaneo con nmap (continuación)

-sT (escaneo TCP  
CONNECT)

- Cuando no se tienen suficientes privilegios para ejecutar el escaneo SYN, por defecto nmap realiza un escaneo TCP Connect. Toma más tiempo ya que se completa la comunicación TCP.

# Técnicas de escaneo con nmap (continuación)

## -sU (escaneo UDP)

- Servicios como DNS, SNMP y DHCP utilizan UDP.
- Para determinar el estado de los puertos se envía un datagrama UDP.

# Técnicas de escaneo con nmap (continuación)

## -sV (escaneo VERSION)

- Es una de las más utilizadas por los especialistas de seguridad, ya que ayuda a determinar la versión del servicio o aplicación que se encuentra ejecutándose en un puerto.
- Cuando se usa este escaneo se envían varias pruebas de nmap-service-probes a los puertos abiertos.

# Técnicas de escaneo con nmap (continuación)

## -sP (escaneo PING)

- Es un escaneo que sirve para enumerar los dispositivos de una red, utiliza paquetes con el bit de control ACK habilitado.

# Técnicas de escaneo con nmap (continuación)

-sN (escaneo TCP  
NULL)

- No habilita ninguna bandera TCP. El RFC 793 describe que cualquier paquete TCP de solicitud que no contenga los bits de control habilitados SYN, RST o ACK se recibirá una respuesta RST que significa que el puerto está cerrado y si no se genera ninguna respuesta, nmap determina el estado del puerto como open|filtered.

# Técnicas de escaneo con nmap (continuación)

-sF (escaneo TCP FIN)

- Esta técnica tiene el mismo comportamiento que el escaneo TCP Null, solo que ésta habilita la bandera FIN.

# Técnicas de escaneo con nmap (continuación)

-sX (escaneo TCP XMAS)

- Esta técnica tiene el mismo comportamiento que el escaneo TCP Null, solo que ésta habilita las banderas FIN, PSH y URG; también es conocida como *Christmas tree*.

# Técnicas de escaneo con nmap (continuación)

-sW (escaneo TCP  
WINDOW)

- Para establecer el estado de los puertos, inspecciona el campo *Window* de la cabecera TCP; en algunos sistemas los puertos abiertos utilizan un tamaño de ventana positivo, mientras que los puertos cerrados tienen un valor de ventana igual a cero.

# Técnicas de escaneo con nmap (continuación)

## --scanflags (escaneo TCP CUSTOM)

- Permite realizar escaneos personalizados, por lo regular se utiliza para evadir sistemas de detección de intrusos. Proporciona a los usuarios avanzados de nmap la opción de diseñar sus pruebas, habilitando los bits de control TCP que se deseen.

# Técnicas de escaneo con nmap (continuación)

Para realizar escaneos personalizados se puede consultar la siguiente tabla:

Decimal	128	64	32	16	8	4	2	1
Base 2	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
Acrónimo para nmap	CWR	ECN	URG	ACK	PSH	RST	SYN	FIN
Bandera	C	E	U	A	P	R	S	F

# Técnicas de escaneo con nmap (continuación)

## -sl (escaneo IDLE)

- Es una técnica de escaneo avanzada de nmap, permite utilizar un equipo zombi para realizar un escaneo de puertos al objetivo de evaluación.

# Técnicas de escaneo con nmap (continuación)

## -b (escaneo FTP BOUNCE)

- Este tipo de escaneo aprovecha una característica de conexión proxy FTP que permite realizar un escaneo de puertos a través de un servidor FTP, esto podría servir a un atacante a evadir firewalls u otros mecanismos de seguridad y cubrir sus huellas.

# PRÁCTICA #5: FLUJO Y ESCANEOS CON NMAP

# Práctica #5: Desarrollo

Verificar conectividad entre las máquinas virtuales para garantizar que el equipo del *pentester* (Kali linux) tiene comunicación con el equipo que fungirá como el objetivo de evaluación (Metasploitable o cualquier otro).

```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST
      inet 192.168.26.130 netmask 255.255.255.0
      inet6 fe80::20c:29ff:fe77:9c76 prefixlen 64
          brd fe80.20c.29ff.ffe7:9c76
```

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 00:0c:29:ff:fe:69
          inet addr:192.168.26.129 Bcast:192.168.26.255
          inet6 addr: fe80::20c:29ff:fe69:84ff%eth0
          UP BROADCAST RUNNING MULTICAST MTU:1500
          RX packets:127 errors:0 dropped:0
          TX packets:59 errors:0 dropped:0
          collisions:0 txqueuelen:1000
          RX bytes:10218 (9.9 KB)  TX bytes:10218 (9.9 KB)
          Interrupt:19 Base address:0x2000
```

```
root@kali:~# ping 192.168.26.129
PING 192.168.26.129 (192.168.26.129) 56(64) bytes from 192.168.26.129: icmp_seq=1
64 bytes from 192.168.26.129: icmp_seq=2
64 bytes from 192.168.26.129: icmp_seq=3
64 bytes from 192.168.26.129: icmp_seq=4
64 bytes from 192.168.26.129: icmp_seq=5
64 bytes from 192.168.26.129: icmp_seq=6
```

# Práctica #5: Desarrollo

Ejecutar una terminal desde el equipo del *pentester* e ingresar nmap. Observe que se despliega información como la versión de nmap, el modo de uso y las opciones de la herramienta.

```
root@kali:~# nmap
Nmap 7.40 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
      -iL <inputfilename>; Input from list of hosts/networks
      -iR <num hosts>; Choose random targets
      --exclude <host1[,host2][,host3],...>; Exclude hosts/networks
      --excludefile <exclude_file>; Exclude list from file
HOST DISCOVERY:
      -sL: List Scan - simply list targets to scan
      -sn: Ping Scan - disable port scan
      -Pn: Treat all hosts as online -- skip host discovery
      -PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
      -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
```

# Práctica #5: Desarrollo

Visualizar en la terminal el manual de nmap.

`man nmap`

NMAP(1)	Nmap Reference Guide	NMAP(1)
<b>NAME</b>		
	<code>nmap</code> - Network exploration tool and security / port scanner	
<b>SYNOPSIS</b>		
	<code>nmap [Scan Type...] [Options] {target specification}</code>	
<b>DESCRIPTION</b>		
	Nmap (“Network Mapper”) is an open source tool for network exploration and security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. While Nmap is commonly used for security audits, many systems and network administrators find it useful for routine tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime.	

# Práctica #5: Desarrollo

Realizar un barrido de red (*Network Sweep*).

```
nmap -sP <Segmento de IP>/<Mascara>
```

```
Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-20 15:33 EDT
Nmap scan report for 192.168.26.1
Host is up (0.00038s latency).
MAC Address: 00:50:56:C0:00:08 (VMware)
Nmap scan report for 192.168.26.2
Host is up (0.000070s latency).
MAC Address: 00:50:56:EE:94:C0 (VMware)
Nmap scan report for 192.168.26.129
Host is up (0.00016s latency).
MAC Address: 00:0C:29:69:84:50 (VMware)
Nmap scan report for 192.168.26.254
Host is up (0.000054s latency).
MAC Address: 00:50:56:FF:38:37 (VMware)
Nmap scan report for 192.168.26.130
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 7.47 seconds
```

# Práctica #5: Desarrollo

Hacer un escaneo de puertos TCP SYN para identificar puertos abiertos comunes en el objetivo de evaluación. `nmap -sS <IP objetivo>`

```
root@kali:~# nmap -sS 192.168.26.129

Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-20 15:37 EDT
Nmap scan report for 192.168.26.129
Host is up (0.000092s latency).
Not shown: 976 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
```

# Práctica #5: Desarrollo

El escaneo de puertos de tipo SYN requiere permisos administrativos, en caso de no tenerlos, también podemos realizar un escaneo TCP CONNECT.

`nmap -sT <IP objetivo>`

```
root@kali:~# nmap -sT 192.168.26.129

Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-20 15:40 EDT
Nmap scan report for 192.168.26.129
Host is up (0.0024s latency).
Not shown: 976 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
```

# Práctica #5: Desarrollo

Realizar ahora el escaneo, del mismo objeto de evaluación, pero sobre protocolo UDP.

`nmap -sU <IP objetivo>`

```
root@kali:~# nmap -sU 192.168.26.129

Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-20 15:41 EDT
Stats: 0:00:37 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 4.97% done; ETC: 15:54 (0:12:07 remaining)
Warning: 192.168.26.129 giving up on port because retransmission cap hit
Nmap scan report for 192.168.26.129
Host is up (0.00021s latency).

Not shown: 992 closed ports
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
```

# Práctica #5: Desarrollo

De acuerdo al flujo de escaneo, después de realizar el escaneo de puertos se debe proseguir a tratar de descubrir el sistema operativo del objetivo de evaluación, esta técnica es conocida como *fingerprinting OS*.

`nmap -O <objetivo>`

```
root@kali:~# nmap -O 192.168.26.129
Starting Nmap 7.40 ( https://nmap.org ) 
Nmap scan report for 192.168.26.129
Host is up (0.00023s latency).
Not shown: 976 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
```

```
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
52869/tcp open  unknown
MAC Address: 00:0C:29:69:84:50 (VMware)
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
```

# Práctica #5: Desarrollo

Identificar el servicio y la versión correspondiente de los puertos abiertos, con la finalidad de descubrir vulnerabilidades específicas de los mismos, esta técnica es conocida como *banner grabbing*.

`nmap -sV <objetivo>`

```
root@kali:~# nmap -sV 192.168.26.129

Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-20 15:45 EDT
Nmap scan report for 192.168.26.129
Host is up (0.000066s latency).
Not shown: 976 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
```

# Práctica #5: Desarrollo

El último paso durante el flujo es el escaneo de vulnerabilidades, utilizando scripts de nmap podemos visualizar posibles vulnerabilidades en los servicios descubiertos.

```
nmap --script vuln <objetivo>
```

```
root@kali:~# nmap --script vuln 192.168.26.129

Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-20 15:48 EDT
Nmap scan report for 192.168.26.129
Host is up (0.000076s latency).
Not shown: 976 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
|_ftp-vsftpd-backdoor:
|   VULNERABLE:
|     vsFTPD version 2.3.4 backdoor
|       State: VULNERABLE (Exploitable)
|       IDs: OSVDB:73573 CVE:CVE-2011-2523
|         vsFTPD version 2.3.4 backdoor, this was reported on 2011-07-04.
|       Disclosure date: 2011-07-03
|       Exploit results:
|         Shell command: id
|         Results: uid=0(root) gid=0(root)
```

## Práctica #5: Desarrollo

Ahora exploraremos otro tipo de escaneos que soporta nmap, como el escaneo TCP NULL, el cual consiste en el envío de paquetes sin banderas bajo protocolo TCP, si no hay respuesta, el puerto se determina como open|filtered.

```
root@kali:~# nmap -sN 192.168.26.129

Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-20 15:49 EDT
Nmap scan report for 192.168.26.129
Host is up (0.000075s latency).
Not shown: 976 closed ports
PORT      STATE          SERVICE
21/tcp    open|filtered  ftp
22/tcp    open|filtered  ssh
23/tcp    open|filtered  telnet
25/tcp    open|filtered  smtp
53/tcp    open|filtered  domain
80/tcp    open|filtered  http
111/tcp   open|filtered  rpcbind
139/tcp   open|filtered  netbios-ssn
445/tcp   open|filtered  microsoft-ds
```

# Práctica #5: Desarrollo

Escaneo TCP FIN el cual tiene el mismo comportamiento que el TCP NULL, sólo que en éste únicamente se activa la bandera FIN.

```
nmap -sF <objetivo>
```

```
root@kali:~# nmap -sF 192.168.26.129

Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-20 15:51 EDT
Nmap scan report for 192.168.26.129
Host is up (0.00016s latency).
Not shown: 976 closed ports
PORT      STATE          SERVICE
21/tcp    open|filtered  ftp
22/tcp    open|filtered  ssh
23/tcp    open|filtered  telnet
25/tcp    open|filtered  smtp
53/tcp    open|filtered  domain
80/tcp    open|filtered  http
```

# Práctica #5: Desarrollo

Escaneo TCP XMAS habilita las banderas PSH, URG, FIN, también conocido como Christmas tree, tiene el mismo comportamiento que los escaneos TCP NULL y FIN.

`nmap -sX <objetivo>`

```
root@kali:~# nmap -sX 192.168.26.129

Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-20 15:52 EDT
Nmap scan report for 192.168.26.129
Host is up (0.00017s latency).
Not shown: 976 closed ports
PORT      STATE          SERVICE
21/tcp    open|filtered  ftp
22/tcp    open|filtered  ssh
23/tcp    open|filtered  telnet
25/tcp    open|filtered  smtp
53/tcp    open|filtered  domain
80/tcp    open|filtered  http
```

## Práctica #5: Desarrollo

Escaneo TCP ACK habilita la bandera ACK, sirve para la evaluación de reglas de *firewall*, con lo cual determina si un puerto está filtrado o no.

`nmap -sA <objetivo>`

```
root@kali:~# nmap -sA 192.168.26.129

Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-20 15:53 EDT
Nmap scan report for 192.168.26.129
Host is up (0.00021s latency).
All 1000 scanned ports on 192.168.26.129 are unfiltered
MAC Address: 00:0C:29:69:84:50 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.24 seconds
```

## Práctica #5: Desarrollo

Si se llegase a necesitar la determinación de los protocolos que utiliza el objetivo de evaluación se puede utilizar el escaneo IP.

`nmap -sO <objetivo>`

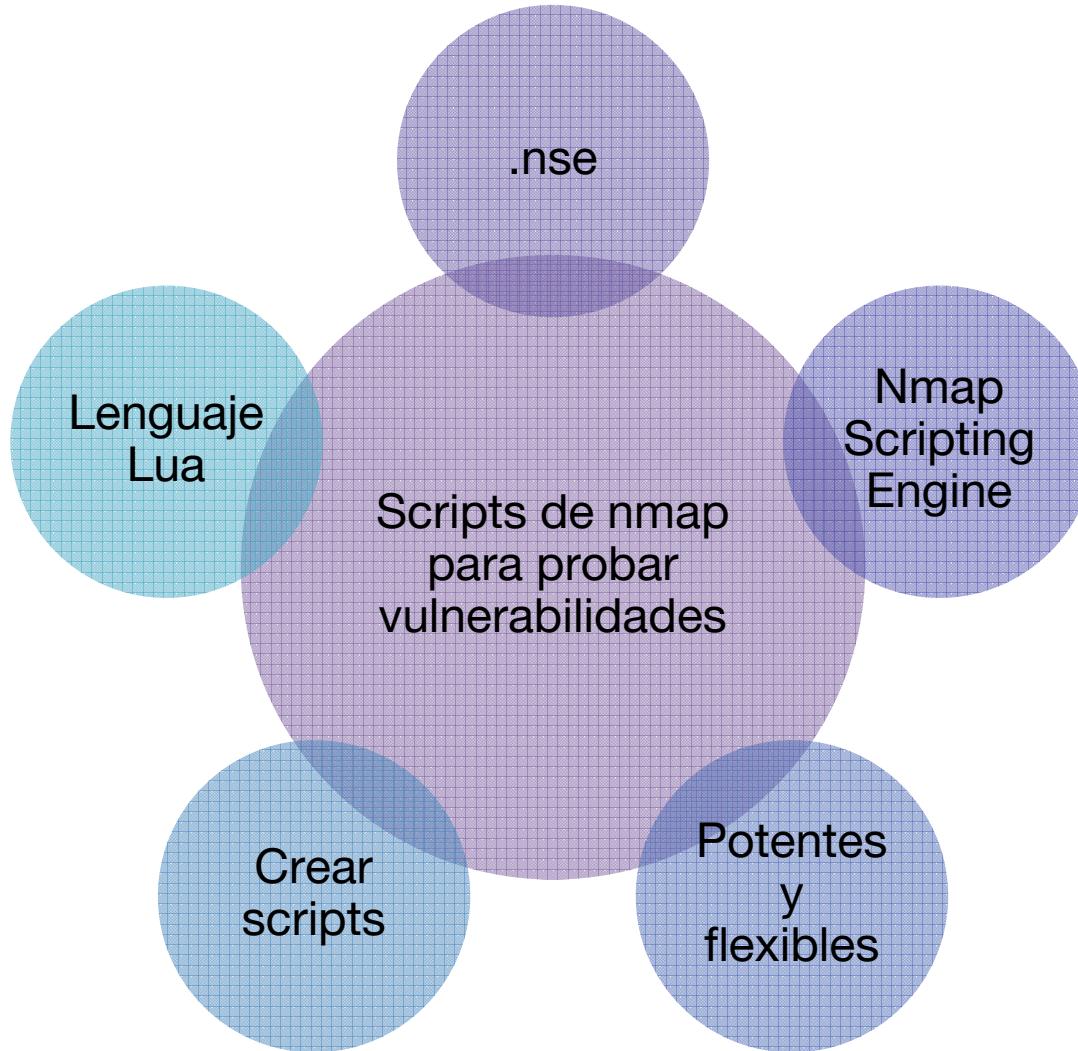
```
root@kali:~# nmap -sO 192.168.26.129

Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-20 15:54 EDT
Warning: 192.168.26.129 giving up on port because retransmission cap hit (10).
Nmap scan report for 192.168.26.129
Host is up (0.00023s latency).
Not shown: 239 closed protocols
PORT      STATE       SERVICE
1         open        icmp
2         open|filtered igmp
6         open        tcp
12        open|filtered pup
17        open        udp
19        open|filtered dcn-meas
51        open|filtered ah
```

# Práctica #5: Desarrollo

- Nmap como herramienta de evaluación de seguridad de dispositivos en red.
- Realización de escaneos con sintaxis sencilla e intuitiva.
- Escaneos personalizados.
- No solo sirve para determinar estados de puertos.
- Análisis de vulnerabilidades.
- Creación de archivos tipo script .nse.

# NSE



## NSE (continuación)

El escaneo de scripts sirve para usar nmap como escáner de vulnerabilidades, el cual utiliza una colección de scripts propios de la herramienta.

La forma más simple de utilizarlo es:

```
nmap -sC dirIP
```

## NSE (continuación)

La manera avanzada de utilizar este escaneo es la siguiente:

```
--script  
<nombre_de_archivo>|<categoría>|<directorío>|<expresión>
```

# Categorías de scripts de nmap

## auth

- Son relacionados con autenticación de usuarios.

## broadcast

- Conseguir información mediante peticiones broadcast.

## brute

- Ayudan a realizar auditoría de fuerza bruta a contraseñas.

## default

- Son aquellos que se ejecutan cuando se realiza un escaneo de scripts (-sC).

## discovery

- Sirven para descubrir servicios.

# Categorías de scripts de nmap (continuación)

## *dos*

- Son aquellos con relación a pruebas de denegación de servicio.

## *exploit*

- Sirven para explotar vulnerabilidades de seguridad.

## *external*

- Son aquellos scripts que dependen de terceras partes.

## *fuzzer*

- Dirigidos a técnicas de fuzzing.

## *intrusive*

- Pueden provocar una falla o generar mucho ruido en la red.

# Categorías de scripts de nmap (continuación)

## *malware*

- Son útiles para detectar malware.

## *version*

- Son scripts utilizados por el escaneo de versión.

## *safe*

- Son scripts que no provocan ningún daño.

## *vuln*

- son scripts utilizados para descubrir vulnerabilidades bien conocidas.

# PRÁCTICA #6: NMAP COMO ESCÁNER DE VULNERABILIDADES Y HERRAMIENTA DE EXPLOTACIÓN

# Práctica #6: Flujo y escaneo con nmap

Una de las características importantes de esta herramienta es que puede ser utilizada como escáner de vulnerabilidades y utilería de explotación.

# Práctica #6: Desarrollo

Para exemplificar las categorías de scripts de Nmap se procederá a ejecutar ejemplos sobre la mayoría de ellas.

En una terminal ejecutar:

```
nmap --script auth <IP>
```

```
Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-19 14:09 CDT
Nmap scan report for 192.168.199.135
Host is up (0.00045s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
|_ ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp  ←
|_ smtp-enum-users:
|   Method RCPT returned a unhandled status code.
53/tcp    open  domain
80/tcp    open  http  ←
|_ http-default-accounts:
111/tcp   open  rpcbind
```

# Práctica #6: Desarrollo

Prueba de los scripts de tipo default.

```
nmap --script default <IP>
```

```
Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-19 14:08 CDT
Nmap scan report for 192.168.199.135
Host is up (0.00054s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp    open  ssh
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|   2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp    open  telnet
25/tcp    open  smtp
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS,
|_ATUSCODES, 8BITMIME, DSN,
```

# Práctica #6: Desarrollo

Prueba de los scripts de tipo *exploit*.

`nmap --script exploit <IP>`

```
Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-19 14:25 CDT
Nmap scan report for 192.168.199.135
Host is up (0.00049s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
|_ftp-vsftpd-backdoor:
  VULNERABLE:
    vsFTPD version 2.3.4 backdoor
      State: VULNERABLE (Exploitable)
      IDs: CVE:CVE-2011-2523 OSVDB:73573
        vsFTPD version 2.3.4 backdoor, this was reported on 2011-07-04.
      Disclosure date: 2011-07-03
      Exploit results:
        Shell command: id
        Results: uid=0(root) gid=0(root)
      References:
        https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523
        http://osvdb.org/73573
        https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/unix/ftp/vsftpd_234_
backdoor.rb
|_ http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html
22/tcp    open  ssh
23/tcp    open  telnet
```

# Práctica #6: Desarrollo

Para probar los scripts de la categoría external a diferencia de las pruebas anteriores, estos *scripts* solo evaluarán el sitio testfire.net y el puerto 80.

```
nmap --script external -p 80 testfire.net
```

```
Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-19 14:38 CDT
Pre-scan script results:
| targets-asn:
|   targets-asn.asn is a mandatory parameter
Nmap scan report for testfire.net (65.61.137.117)
Host is up (0.0052s latency).
PORT      STATE SERVICE
80/tcp    open  http
| http-xssed:
|       UNFIXED XSS vuln.

|       http://demo.testfire.net/search.aspx?txtSearch=%22%3E%3Cscript%3Ealert(%2Fwww.sec-r1z.com%2F)%3C%2Fs<br>cript%3E%22%3E%3C%2Fscript%3E
|_
```

# Práctica #6: Desarrollo

Probando los scripts de la categoría intrusive, sólo se probarán los dirigidos al puerto 25, el cual corresponde al servicio SMTP de correo.

```
nmap --script intrusive -p 25 <IP>
```

```
Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-19 14:45 CDT
Nmap scan report for 192.168.199.135
Host is up (0.00055s latency).

PORT      STATE SERVICE
25/tcp    open  smtp
| smtp-enum-users:
|_ Method RCPT returned a unhandled status code.
|_ smtp-open-relay: Server doesn't seem to be an open relay, all tests failed
|_ smtp-vuln-cve2010-4344:
|_ The SMTP server is not Exim: NOT VULNERABLE
|_ sslv2-drown:
MAC Address: 00:0C:29:4D:65:04 (VMware)

Host script results:
| dns-brute: Can't guess domain of "192.168.199.135"; use dns-brute.domain script argument.

Nmap done: 1 IP address (1 host up) scanned in 40.18 seconds
```

# Práctica #6: Desarrollo

En el caso de los scripts de la categoría safe, se utilizará el script de http-php-version sobre el puerto 80 del objetivo.

```
nmap --script http-php-version -p 80 <IP>
```

```
Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-19 14:48 CDT
Nmap scan report for 192.168.199.135
Host is up (0.0013s latency).
PORT      STATE SERVICE
80/tcp    open  http
| http-php-version: Versions from logo query (less accurate): 5.1.3 - 5.1.6, 5.2.0 - 5.2.17
| Versions from credits query (more accurate): 5.2.3 - 5.2.5, 5.2.6RC3
|_Version from header x-powered-by: PHP/5.2.4-2ubuntu5.10
MAC Address: 00:0C:29:4D:65:04 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.01 seconds
```

# Práctica #6: Desarrollo

Realicemos una prueba de los *scripts* de la categoría vuln.

**nmap --script vuln <IP>**

```
Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-19 14:50 CDT
Nmap scan report for 192.168.199.135
Host is up (0.00030s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
|_ftp-vsftpd-backdoor:
|   VULNERABLE:
|     vsFTPD version 2.3.4 backdoor
|       State: VULNERABLE (Exploitable)
|       IDs: OSVDB:73573 CVE:CVE-2011-2523
|         vsFTPD version 2.3.4 backdoor, this was reported on 2011-07-04.
|       Disclosure date: 2011-07-03
|       Exploit results:
|         Shell command: id
|         Results: uid=0(root) gid=0(root)
|       References:
|         https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/unix/ftp/vsftpd_234_
backdoor.rb
|         http://osvdb.org/73573
|         http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html
|         https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523
|_sslv2-drown:
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
|_smtp-vuln-cve2010-4344:
|   The SMTP server is not Exim: NOT VULNERABLE
|_sslv2-drown:
53/tcp    open  domain
```

# Práctica #6: Desarrollo

Para obtener información acerca de un script podemos utilizar:

```
nmap --script-help smb-system-info
```

```
Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-19 15:05 CDT
smb-system-info
Categories: discovery intrusive
https://nmap.org/nsedoc/scripts/smb-system-info.html
    Pulls back information about the remote system from the registry. Getting all
    of the information requires an administrative account, although a user account
    will still get a lot of it. Guest probably won't get any, nor will anonymous.
    This goes for all operating systems, including Windows 2000.

    Windows Vista disables remote registry access by default, so unless it was enabled,
    this script won't work.

    If you know of more information stored in the Windows registry that could be interesting,
    post a message to the nmap-dev mailing list and I (Ron Bowes) will add it to my todo list.
    Adding new checks to this is extremely easy.

WARNING: I have experienced crashes in <code>regsvc.exe</code> while making registry calls
against a fully patched Windows 2000 system; I've fixed the issue that caused it,
but there's no guarantee that it (or a similar vuln in the same code) won't show
up again. Since the process automatically restarts, it doesn't negatively impact
the system, besides showing a message box to the user.
```

# Práctica #6: Desarrollo

Otro ejemplo por nombre y categoría.

`nmap --script-help http-apache-server-status`

```
Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-19 15:07 CDT
[...]
http-apache-server-status
Categories: discovery safe
https://nmap.org/nsedoc/scripts/http-apache-server-status.html
Attempts to retrieve the server-status page for Apache webservers that
have mod_status enabled. If the server-status page exists and appears to
be from mod_status the script will parse useful information such as the
system uptime, Apache version and recent HTTP requests.

References:
* http://httpd.apache.org/docs/2.4/mod/mod_status.html
* https://blog.sucuri.net/2012/10/popular-sites-with-apache-server-status-enabled.html
* https://www.exploit-db.com/ghdb/1355/
* https://github.com/michenriksen/nmap-scripts
```

# Práctica #6: Desarrollo

Otro aspecto importante es conocer el comportamiento y funcionamiento de las pruebas que realiza un *script*, para eso existe.

`nmap -p 80 --script-trace --script http-php-versión <IP>`

```
Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-19 15:09 CDT
NSE: TCP 192.168.199.130:57870 > 192.168.199.135:80 | CONNECT
NSE: TCP 192.168.199.130:57870 > 192.168.199.135:80 | 00000000: 47 45 54 20 2f 3f 3d 50 48 50 45 39 35 3
6 38 46 GET /?=PHPE9568F
00000010: 33 36 2d 44 34 32 38 2d 31 31 64 32 2d 41 37 36 36-D428-11d2-A76
00000020: 39 2d 30 30 41 41 30 30 31 41 43 46 34 32 20 48 9-00AA001ACF42 H
00000030: 54 54 50 2f 31 2e 31 0d 0a 43 6f 6e 6e 65 63 74 TTP/1.1 Connect
00000040: 69 6f 6e 3a 20 63 6c 6f 73 65 0d 0a 55 73 65 72 ion: close User
00000050: 2d 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f -Agent: Mozilla/
00000060: 35 2e 30 20 28 63 6f 6d 70 61 74 69 62 6c 65 3b 5.0 (compatible;
00000070: 20 4e 6d 61 70 20 53 63 72 69 70 74 69 6e 67 20 Nmap Scripting
00000080: 45 6e 67 69 6e 65 3b 20 68 74 74 70 73 3a 2f 2f Engine; https://
00000090: 6e 6d 61 70 2e 6f 72 67 2f 62 6f 6f 6b 2f 6e 73 nmap.org/book/ns
000000a0: 65 2e 68 74 6d 6c 29 0d 0a 48 6f 73 74 3a 20 31 e.html) Host: 1
000000b0: 39 32 2e 31 36 38 2e 31 39 39 2e 31 33 35 0d 0a 92.168.199.135
000000c0: 0d 0a
```

# Práctica #6: Conclusiones

Nmap como herramienta es

Robusta, flexible y muy intuitiva

No sólo es útil en la fase de escaneo, también lo es en otras, como en la explotación

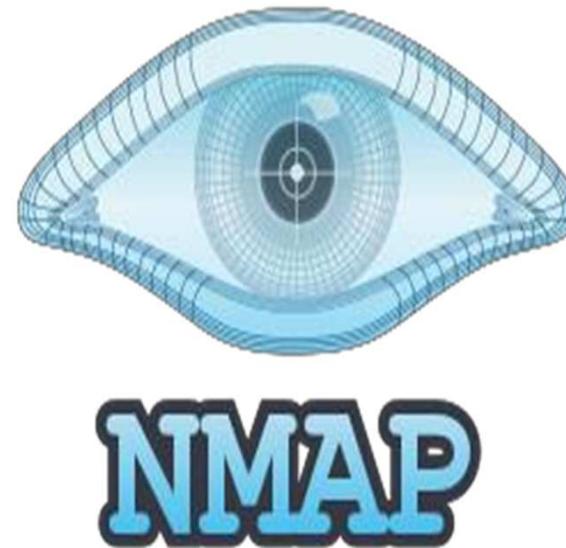
Tiene una amplia variedad de scripts instalados por defecto

Es posible agregar scripts de la comunidad o desarrollos propios

Hay otras herramientas de escaneo de puertos que pueden ser muy útiles a la hora de descubrir los servicios de un sistema.

# Herramientas para escaneo

- Netdiscover
- Masscan
- NMAP
- Nikto
- Nessus
- OpenVAS
- Arachni
- ZAP



## Tener en cuenta

Reconocimiento y enumeración son los pilares de las pruebas de penetración, son lo que guiarán el resto de las pruebas.

Estos procesos deben ser debidamente documentados, con el fin de generar información que pueda ser consultada posteriormente.

El reporte final no será la documentación que se genere para consulta, sino que debe llevar un orden lógico para el cliente

# TAREA

- Hacer programa que obtenga SO de una lista de hosts a partir de TTL
- Programar escáner de puertos
- Hacer script de nmap