



Published on hacktimes.com (<http://www.hacktimes.com>)

## IDENTIFICACIÓN REMOTA DEL SISTEMA OPERATIVO DE UN SERVIDOR

By **vaxman**

Creado 28 Ago 2006 - 11:16

En la actualidad, existen multitud de Sistemas Operativos diferentes (Windows, HP-UX, Linux, Solaris, etc.) y cada uno tiene sus propias características que lo diferencian de los demás: distintas implementaciones de la pila TCP/IP, diferentes comportamientos ante el envío de según qué paquetes especialmente formados, distintas respuestas en función del protocolo utilizado (TCP, ICMP, ARP), etc.

Al realizar una revisión de seguridad, auditoría o test de intrusión, es importante, antes de empezar a enumerar qué servicios hay activos, reconocer el Sistema Operativo del servidor remoto que se está analizando ya que el procedimiento, las herramientas y las técnicas a emplear son diferentes. Esto es lo que se conoce como **Fingerprinting de Sistemas**.

El Fingerprinting de Sistemas es la técnica que usan la mayoría de analizadores de puertos avanzados (como Nmap) para intentar descubrir el Sistema Operativo de un servidor remoto. Esta identificación se basa en los tiempos de respuesta a los diferentes paquetes ACK y SYN al establecer una conexión en el protocolo TCP/IP.

Básicamente, existen dos formas principales de intentar descubrir el Sistema Operativo presente en un host remoto: forma activa y/o pasiva y existen cantidad de herramientas que, usando cualquiera de los dos métodos, permiten realizar el fingerprinting remoto tales como Nmap, Xprobe, el antiguo QueSO, P0f o incluso la más moderna SinFP. Las técnicas tradicionales de reconocimiento de Sistemas mediante la captura del banner de algún servicio activo tipo TELNET, FTP, etc. que proporcionan información acerca del Sistema Operativo son limitadas y obsoletas y están prácticamente en desuso debido a la facilidad para modificar el banner de servicio y ofuscar, de esta manera, la identificación del Sistema Operativo presente en el servidor.

Además, existe un método alternativo de descubrir qué sistema está presente en la mayoría de servidores remotos importantes presentes en Internet, sin hacer ningún tipo de ruido ni ninguna prueba complicada que es utilizar NETCRAFT (<http://www.netcraft.com>). NETCRAFT, compañía dedicada, básicamente, a realizar estadísticas del uso de software en Internet, dispone de un servicio Web en el que con,

simplemente, introducir el nombre del servidor a analizar, en breves segundos proporciona la información acerca del servidor Web, el Sistema Operativo, etc.

En el presente documento se va a realizar un estudio de las principales técnicas de Fingerprinting activo, cómo combinarlas y cómo conseguir determinar con un alto grado de exactitud el Sistema Operativo de un servidor así como también se van a enumerar las principales herramientas utilizadas en la actualidad.

## FINGERPRINTING ACTIVO

Este tipo de identificación del Sistema Operativo se basa en analizar la respuesta del servidor que se quiere revisar cuando se le envían determinados paquetes TCP y UDP.

El Fingerprinting activo aporta mayor variedad de pruebas porque se envían diferentes tipos de paquetes al servidor y se analiza su respuesta, como contrapartida principal está el hecho de que sea más fácil de detectar e interceptar por parte de los dispositivos de seguridad, firewalls principalmente, que estén presentes en la red donde resida el servidor analizado además de que es necesario que exista algún puerto abierto contra el que lanzar los paquetes y analizar el resultado de las diferentes peticiones. Para más información acerca del tipo de peticiones que se realizan se recomienda recurrir al artículo de Fyodor del año 2002:

<http://www.insecure.org/nmap/nmap-fingerprinting-article.html> [1] o a la segunda generación de detección explicada en: <http://insecure.org/nmap/osdetect> [2]

El archiconocido Nmap, Xprobe, etc. son algunos ejemplos de herramientas que se sirven de este método de identificación.

### ▪ NMAP

con las opciones -O y/o con -A, se activa la detección activa del Sistema Operativo basándose en marcas que hacen únicas a las pilas de protocolos TCP/IP de los distintos Sistemas Operativos en los paquetes que envían como respuesta ante determinadas peticiones. Nmap en su última versión disponible a la hora de escribir este artículo, la 4.11, dispone de una base de datos con más de 1500 "huellas" de Sistema Operativo con las que realizar la identificación (fichero nmap-os-fingerprints). Nmap tiene versiones para Windows y Linux. Nmap también permite realizar fingerprinting de cualquier sistema de forma indirecta mediante el reconocimiento de la versión de los diferentes servicios que están activos en el servidor. Dicha opción se activa con el parámetro -sV.

Nmap realiza 9 pruebas, a partir de comprobar que el servidor está activo y que tiene algún puerto abierto, para intentar identificar el Sistema de un host remoto:

**1.- TSeq:** se observa el ISN (número de secuencia inicial) del paquete SYN para buscar pautas de implementaciones diferentes de la pila TCP/IP en el ISN de la cabecera TCP, al responder a una solicitud de conexión. Existen varios grupos de clasificación: 64K (Class=64K) para versiones antiguas de Unix, incrementos aleatorios (Class=RI) para Solaris, IRIX y FreeBSD sobretodo, verdaderamente aleatorio o al azar (Class=TR) para Linux y alguna versión nueva de AIX, dependiente del tiempo (Class=TD) para Sistemas Windows, constante (3Com), etc.

**2.- T1:** en esta prueba se envía un paquete con el flag SYN (petición de inicio de conexión) con una serie de opciones TCP a un puerto abierto. Estas opciones consisten en un valor de escala de ventana de 10, un tamaño máximo del segmento de 265, y un valor del timestamp de 1061109567.

**3.- T2:** se envía un paquete NULL (paquete con todos los flags URG|ACK|PSH|RST|SYN y FIN desactivados en la cabecera TCP) con las mismas opciones del anterior paquete a un puerto abierto.

**4.- T3:** se envía un paquete con los flags SYN|FIN|URG|PSH a un puerto abierto.

**5.- T4:** se envía un paquete con el campo de acuse de recibo ACK activo a un puerto abierto.

**6.- T5:** se envía un paquete con el flag de inicio de conexión SYN a un puerto cerrado.

**7.- T6:** se envía un ACK a un puerto cerrado.

**8.- T7:** se envía un paquete con los campos FIN|PSH|URG (fin de conexión, el receptor no pone los datos en cola sino que los pasa directamente a la aplicación y puntero de urgente) a un puerto cerrado.

**9.- PU:** esta es la prueba de puerto inalcanzable (Port Unreachable). Se envía un paquete UDP a un puerto cerrado y, si el puerto está realmente cerrado (closed) y no hay ningún firewall de por medio, se devuelve un mensaje ICMP de puerto inalcanzable que tiene una longitud de 8 bytes en su cabecera aunque sólo se utilizan los 4 primeros y el resto tienen valor cero.

Ejemplo de "huella" del Sistema Operativo Windows 2003 del Nmap donde se puede observar el resultado de los 9 test anteriores:

```
Fingerprint Microsoft Windows 2003 Server
Class Microsoft | Windows | 2003/.NET | general purpose
TSeq(Class=TR%gcd=<6%IPID=I)
T1(DF=N%W=4000%ACK=S++%Flags=AS%Ops=MNWNNT)
T2(Resp=N)
T3(Resp=Y%DF=N%W=4000%ACK=S++%Flags=AS%Ops=MNWNNT)
T4(DF=N%W=0%ACK=0%Flags=R%Ops=)
T5(DF=N%W=0%ACK=S++%Flags=AR%Ops=)
T6(DF=N%W=0%ACK=0%Flags=R%Ops=)
T7(DF=N%W=0%ACK=S++%Flags=AR%Ops=)
PU(DF=N%TOS=80%IPLen=B0%RIPTL=148%RID=E%RIPCK=F%UCK=F%ULEN=134%
DAT=E)
```

Las dos primeras líneas son la identificación del Sistema Operativo para Windows 2003 y .NET dentro del fichero de base de datos de Nmap (nmap-os-fingerprints).

La prueba de TSeq tiene como resultado "Class=TR%gcd=<6%IPID=I" que significa que es Truly Random (Class=TR, verdaderamente al azar) y está identificando, a priori, el Sistema Operativo como Linux o como algún AIX moderno. Lo siguiente es el atributo IPID que se refiere a los bits de identificación IP en la cabecera IP, IPID=I significa que es un sistema que aumenta el IPID en un incremento estándar con cada paquete enviado.

**T1** indica que el flag o el bit de no fragmentación (DF=N) no está activado, lo siguiente término es la ventana del paquete (W=4000), a continuación se observa el número de ACK (ACK=S++), los flags que tienen que estar activados (Flags=AS), en este caso concreto se necesitan los bits de ACK y SYN y por último, se muestra las opciones que se tienen que recibir ordenadas según la respuesta (Ops=MNWNNT), es decir, MSS (not echoed), NOP, Window scale, NOP, NOP y TimeStamp.

**T2** indica que no se obtiene una respuesta a esa prueba en concreto (Resp=N).

**T3** indica que sí hay respuesta al paquete enviado (Resp=Y) pero no se obtiene ningún valor nuevo de los ya vistos en el test 1 (T1).

La línea de **T4** indica aparte de que el flag DF no está activado, que la ventana del paquete es igual a 0, que el número de ACKs es 0, que sólo tiene que estar activo el flag de reset y que no se requieren opciones del TCP.

**T5** muestra información en su respuesta acerca de los flags que tienen que estar activos (Flags=AR), ACK y el de RST (reset).

**T6** no aporta más datos que los ya obtenidos con el T4.

**T7** muestra la misma información ya descubierta con el T5.

Y el último test, el de PU (Port Unreachable) muestra como resultado DF=N, TOS=80, IPLEN=B0, RIPTL=148, RID=E, RIPCK=F, UCK=F, ULEN=134 y DAT=E:

- DF=N, si el bit de no fragmentación (don't fragment) se encuentra activado el resultado es Y (yes), sino, se obtiene un valor N para esta prueba.
- TOS=80, el byte de la cabecera para el tipo de servicio IP, Type of Service (ToS), obtenido es 80 (en hexadecimal) y se compara con el campo para TOS que nmap tiene guardado como huella de un sistema Windows 2003 Server.
- IPLEN=B0, indica, en hexadecimal, la longitud total del campo IP en la cabecera IP del mensaje recibido.
- RIPTL=148, el tamaño del paquete UDP enviado a un puerto cerrado es de 328 bytes (0x148 en hexadecimal) al obtener este valor en el mensaje de puerto inalcanzable devuelto por el servidor en la cabecera IP se consigue una prueba más de que se está analizando un sistema Windows ya que, para este test, Microsoft se adapta al estándar. Si se hubiera obtenido otro valor como, por ejemplo, 20 bytes se estaría revisando un servidor con OpenBSD como sistema operativo.
- RID=E (copia del ID), si los bytes de identificación IP de la respuesta que envía el servidor analizado son idénticos a los que se enviaron originalmente con la petición del nmap, se obtiene un valor E para este test, si dichos bytes son modificados por el servidor se obtiene un valor F.
- RIPCK=F (Checksum IP), entre que se envía el paquete UDP y entre que se recibe, el TTL puede decrementarse en cada sistema y el checksum es recalculado. Una E como resultado indica que ambos controles coinciden y una F, como en este caso, que el campo del checksum IP de la cabecera IP es diferente al checksum IP de la cabecera IP del paquete devuelto por el servidor.
- UCK=F (Checksum UDP), indica que el control del paquete UDP que se ha obtenido en el mensaje de respuesta del servidor es diferente al original que se envió. UCK=E indicaría que el Checksum UDP se ha hecho de forma correcta y que ambos campos en cada paquete (el paquete de inicio y el de respuesta) son iguales.
- ULEN=134 (UDP Length), indica la longitud del paquete UDP recibido (0x134). Nmap envía originalmente un paquete UDP de tamaño 134 bytes.
- DAT=E, significa que los datos del paquete que devolvió el servidor ante la petición realizada por nmap están correctos. La mayoría de las implementaciones de los diferentes sistemas no devuelven ningún tipo de información UDP y el valor DAT=E es el que se obtiene por defecto, si los datos no fueran correctos se obtendría un valor F.

**Como conclusión, después de analizar el resultado de las 9 pruebas que realiza Nmap, se clasifica el Sistema Operativo del servidor analizado como Windows 2003.**

## ▪ XPROBE2

escrito por Fyodor Yarochkin y Ofir Arkin utiliza sobretodo paquetes ICMP pero la última versión, la 2.0.3, ya es capaz de analizar la respuesta ante determinados paquetes TCP SYN e incluso SNMP y SMB. Xprobe2 realiza 6 pruebas durante el proceso de Fingerprinting de un servidor remoto cuyo resultado comprueba con la base de datos de que dispone donde se almacenan hasta 225 firmas de Fingerprintings que el programa es capaz de reconocer (fichero xprobe2.conf). Xprobe se centra sobre todo en servidores y Sistemas Operativos de estaciones de trabajo y no incluye elementos de red conocidos tipo routers, switches, salvo para Cisco y algún servidor virtual de impresión HP Jetdirect.

Se ejecuta el test de reconocimiento del Sistema Operativo y se envían los paquetes definidos según las opciones disponibles (SNMP, TCP, ICMP, etc.), el programa examina el resultado de las pruebas y asigna un tanto por ciento de aproximación al Sistema Operativo presente en el servidor remoto. Las 6 pruebas realizadas son:

- 1.- Envío de paquete ICMP petición de Echo, es decir, un simple ping (icmp echo request): Módulo A.
- 2.- Envío de paquete ICMP Timestamp request: Módulo B
- 3.- Envío de paquete ICMP de petición de máscara de dirección (address mask request): Módulo C.
- 4.- Envío de paquete ICMP de petición de información (information request): Módulo D.
- 5.- Envío de paquete UDP a un puerto cerrado para obtener el mensaje de port unreachable (mensaje ICMP del tipo 3). Módulo E.
- 6.- Envío de un paquete TCP con el flag SYN activado a un puerto TCP abierto: Módulo F.

Xprobe2 no realiza pruebas con paquetes malformados o especialmente modificados con lo que su detección por parte de algún sistema de firewall y, su posterior bloqueo, es más complicada.

Ejemplo de "huella" del Sistema Operativo Windows 2003 del Xprobe2 donde se puede observar el resultado de los 6 test anteriores:

```
fingerprint {
OS_ID = "Microsoft Windows 2003 Server Standard Edition"
#Entry inserted to the database by: Ofir Arkin (ofir@sys-
security.com)
#Entry contributed by: Ofir Arkin (ofir@sys-security.com)
#Date: 14 July 2003
#Modified: 14 July 2003

#Module A
icmp_echo_reply = y
```

```
icmp_echo_code = 0
icmp_echo_ip_id = !0
icmp_echo_tos_bits = 0
icmp_echo_df_bit = 1
icmp_echo_reply_ttl = < 128

#Module B
icmp_timestamp_reply = y
icmp_timestamp_reply_ttl = <128
icmp_timestamp_reply_ip_id = !0

#Module C
icmp_addrmask_reply = n
icmp_addrmask_reply_ttl = <128
icmp_addrmask_reply_ip_id = !0

#Module D
icmp_info_reply = n
icmp_info_reply_ttl = <128
icmp_info_reply_ip_id = !0

#Module E
#IP Header of the UDP Port Unreachable error message
icmp_unreach_reply = y
icmp_unreach_echoed_dtsize = >64
icmp_unreach_reply_ttl = <128
icmp_unreach_precedence_bits = 0
icmp_unreach_df_bit = 0
icmp_unreach_ip_id = !0

#Original_data_echoed_with_the_UDP_Port_Unreachable_error_message
icmp_unreach_echoed_udp_cksum = OK
icmp_unreach_echoed_ip_cksum = OK
icmp_unreach_echoed_ip_id = OK
icmp_unreach_echoed_total_len = OK
icmp_unreach_echoed_3bit_flags = OK

#Module F [TCP SYN | ACK Module]
#IP header of the TCP SYN | ACK
tcp_syn_ack_tos = 0
tcp_syn_ack_df = 1
tcp_syn_ack_ip_id = !0
tcp_syn_ack_ttl = <128

#Information from the TCP header
tcp_syn_ack_ack = 1
tcp_syn_ack_window_size = 65535,64240,17520
tcp_syn_ack_options_order = "MSS NOP WSCALE NOP NOP TIMESTAMP
NOP NOP SACK"
tcp_syn_ack_wscale = 0
tcp_syn_ack_tsval = 0
tcp_syn_ack_tsecr = 0
```

```

#Module G [TCP RST|ACK]
tcp_rst_reply = y
tcp_rst_df = 0
tcp_rst_ip_id_1 = !0
tcp_rst_ip_id_2 = !0
tcp_rst_ip_id_strategy = I
tcp_rst_ttl = <128

smb_lanman = Windows Server 2003 5.2
smb_nativeos = Windows Server 2003 3790
}

fingerprint {

```

## OTRAS HERRAMIENTAS

### ▪ SinFP

esta nueva herramienta de reciente publicación (en el momento de escribir este documento acababa de salir la versión 2.01-1) se presenta como una alternativa a las limitaciones que incluyen los distintos dispositivos de seguridad presentes habitualmente en una red, IDSs, firewalls, etc. para poder realizar un reconocimiento eficaz del Sistema Operativo de cualquier servidor. SinFP sólo requiere un puerto TCP abierto al que enviar los paquetes estándar y limita el número de pruebas a 1, 2 o como máximo 3. Permite realizar un análisis Fingerprinting en un host determinado de forma activa y pasiva y es la única herramienta, hasta el momento, que puede trabajar sobre IPv6.

SinFP está escrito en perl y está hecho como un módulo fácilmente integrable, incluso el fichero de firmas está en formato SQL para poder ser portado a otras aplicaciones. Como ventajas de este programa cabe destacar que es sumamente rápido y relativamente silencioso y que sólo necesita, como ya se ha destacado, un puerto TCP abierto para funcionar.

- Existen otras herramientas que realizan detección remota del Sistema Operativo como hping2, SING, OSFP (Operating System Fingerprinting Project), SYNSCAN, alguna para Windows, etc. pero no son objeto de estudio de este documento porque se han quedado obsoletas y porque comentarlas todas es prácticamente imposible.

## EJEMPLOS

A continuación se va a tratar de identificar, a modo de ejemplo, el Sistema Operativo del servidor Web de Hacktimes mediante las diferentes herramientas y técnicas descritas a lo largo de este documento. Es importante destacar que el resultado de las pruebas y tests puede variar en función de que exista algún dispositivo de seguridad y de red entre el Servidor a analizar y el equipo cliente desde el que se esté realizando el Fingerprinting, es decir, el resultado es distinto si hay algún firewall, IDS, balanceadores de carga, etc. y

será necesario adecuar las pruebas si se ha detectado la presencia de estos sistemas pero dicha modificación no está contemplada en este documento.

- **Captura del banner de servicio:** Hacktimes tiene el puerto 80 TCP abierto, se va a proceder a capturar el banner del servidor Web para comprobar si facilita información del Sistema Operativo.

```
nc www.hacktimes.com 80

GET / HTTP/1.0

HTTP/1.1 406 Not Acceptable
Date: Thu, XX Jun 2006 09:07:20 GMT
Server: Apache
Last-Modified: Sat, XX XXX 2006 12:03:04 GMT
ETag: "1c8008-97-441bf6f8"
Accept-Ranges: bytes
Content-Length: 151
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

La información obtenida mediante la captura del banner es mínima (servidor Web Apache sin ni siquiera la versión) y no se ha conseguido descubrir qué Sistema Operativo tiene el servidor.

- **Netcraft:** después de hacer la búsqueda mediante la aplicación de NetCraft se obtiene que el servidor Web es un Apache y que el Sistema Operativo es Linux. El resultado es bastante antiguo (data del 2004) y, lógicamente, el Sistema Operativo del servidor puede haber cambiado. Se continúa con las pruebas de fingerprinting.

```
Netblock Owner IP address OS Web Server Last changed
Telefonica de Espana 80.25.154.240 Linux Apache 16-Dec-2004
SAU Red de servicios
IP Spain
```

#### ▪ **NMAP**

se lanza un escaneo con la opción -O del Nmap activada y también con la opción -A para intentar descubrir qué Sistema Operativo tiene el servidor. Si Nmap no fuera capaz de hacer el fingerprinting proporcionaría la "huella" TCP encontrada con el resultado de las 9 pruebas realizadas. En el ejemplo, se reconoce el Sistema Operativo como un sistema Linux y ya se obtienen más detalles acerca del kernel del Sistema Operativo presente en el servidor.

```
nmap -O -A -P0 -p 80 www.hacktimes.com (el P0 se incluye porque el servidor no responde a tráfico ICMP ni a pings)
```

```
Starting Nmap 4.11 (http://www.insecure.org/nmap) at 2006-06-XX
10:23 Hora estandar romance
Warning: OS detection will be MUCH less reliable because we did
not find at least 1 open and 1 closed TCP port
Interesting ports on 172.Red-80-35-173.staticIP.rima-tde.net
(80.35.173.172):
PORT STATE SERVICE VERSION
80/tcp open  http Apache httpd
Device type: general purpose
Running: Linux 2.4.X
OS details: Linux 2.4.18 - 2.4.27

Nmap finished: 1 IP address (1 host up) scanned in 21.016
seconds
```

## ▪ Xprobe2

se lanza el xprobe2 indicando algún puerto abierto y algún puerto cerrado, tanto TCP como UDP, para refinar aún más si cabe, el reconocimiento del Sistema Operativo del servidor. La opción -D 1 indica que no se cargue el módulo ICMP Echo (ping) que ya se ha visto que el servidor de Hacktimes no responde a pings. Xprobe2 reconoce con un 100% de aproximación que el servidor de Hacktimes, de forma similar a como lo ha hecho la herramienta nmap, es un Linux con la versión de kernel comprendida entre la 2.4.6 y la 2.6.4

```
xprobe2 www.hacktimes.com -p TCP:80:open -p TCP:83:closed -p
UDP:53:open -p UDP:55:closed -D 1
```

```
Xprobe2 v2.0.3 Copyright (c) 2002-2005 fyodor@o0o.nu, ofir@sys-
security.com, meder@o0o.nu
```

```
[+] Target is www.hacktimes.com
[+] Loading modules.
[+] Following modules are loaded:
[x][1] ping:tcp_ping - TCP-based ping discovery module
[x][2] ping:udp_ping - UDP-based ping discovery module
[x][3] infogather:ttd_calc - TCP and UDP based TTL distance
calculation
[x][4] infogather:portscan - TCP and UDP PortScanner
[x][5] fingerprinting:icmp_echo - ICMP Echo request
fingerprinting module
[x][6] fingerprinting:icmp_tstamp - ICMP Timestamp request
fingerprinting module
[x][7] fingerprinting:icmp_amask - ICMP Address mask request
fingerprinting module
[x][8] fingerprinting:icmp_info - ICMP Information request
fingerprinting module
[x][9] fingerprinting:icmp_port_unreach - ICMP port unreachable
fingerprinting module
[x][10] fingerprinting:tcp_hshake - TCP Handshake
fingerprinting module
[x][11] fingerprinting:tcp_rst - TCP RST fingerprinting module
[x][12] fingerprinting:smb - SMB fingerprinting module
```

```

[x][13] fingerprinting:dnmp - SNMPv2c fingerprinting module
[+] 13 modules registered
[+] Initializing scan engine
[+] Running scan engine
[+] Host: 80.35.173.172 is up (Guess probability: 100%)
[+] Target: 80.35.173.172 is alive. Round-Trip Time: 0.12506
sec
[+] Selected safe Round-Trip Time value is: 0.25013 sec
[-] fingerprint:smb need either TCP port 139 or 445 to run
[+] Primary guess:
[+] Host 80.35.173.172 Running OS: "Linux Kernel 2.6.0" (Guess
probability: 96%)
[+] Other guesses:
[+] Host 80.35.173.172 Running OS: "Linux Kernel 2.6.1" (Guess
probability: 96%)
[+] Host 80.35.173.172 Running OS: "Linux Kernel 2.6.7" (Guess
probability: 96%)
[+] Host 80.35.173.172 Running OS: "Linux Kernel 2.6.6" (Guess
probability: 96%)
[+] Host 80.35.173.172 Running OS: "Linux Kernel 2.6.5" (Guess
probability: 96%)
[+] Host 80.35.173.172 Running OS: "Linux Kernel 2.6.4" (Guess
probability: 100%)
[+] Host 80.35.173.172 Running OS: "Linux Kernel 2.6.3" (Guess
probability: 100%)
[+] Host 80.35.173.172 Running OS: "Linux Kernel 2.6.2" (Guess
probability: 100%)
[+] Host 80.35.173.172 Running OS: "Linux Kernel 2.4.5" (Guess
probability: 100%)
[+] Host 80.35.173.172 Running OS: "Linux Kernel 2.4.6" (Guess
probability: 100%)
[+] Cleaning up scan engine
[+] Modules deinitialized
[+] Execution completed.

```

#### ▪ SinFP

se lanza un escaneo con la opción -i para indicarle la dirección IP del servidor de Hacktimes que se está analizando y se le marca también un puerto TCP abierto, en este caso el 80. El resultado sigue correspondiéndose con lo encontrado hasta ahora con otras herramientas, el servidor de Hacktimes es una distribución Linux.

```

perl sinfp.pl -i 80.35.173.172 -p 80
T1: B10111 F0x12 W5840 O0204ffff M1452
T2: B10111 F0x12 W5792
O0204ffff0402080affffff4445414401030300 M1452
T3: B00110 F0x04 W0 O0 M0
IPv4: HEURISTIC1/FULL: GNU/Linux: OSS: Linux: 2.4.x (2.4.2,
2.4.7)

```

Mirando la versión real del servidor web de Hacktimes se comprueba como todas las pruebas realizadas han sido bastante acertadas y el Sistema Operativo es una distribución Linux con la versión del kernel 2.4.18. La herramienta

nmap ha sido quizás la que más se ha acercado en el proceso de fingerprinting del servidor web.

```
root@hacktimes:~$ uname -a
Linux hacktimes 2.4.18-bf2.4 #1 Mon XXX XX 09:53:28 CEST 200X
i686 GNU/Linux
```

## CONTRAMEDIDAS

Existen diversos métodos para dificultar el reconocimiento remoto de cualquier sistema, los más obvios pasan por introducir diferentes dispositivos de seguridad, como firewalls, IDSs, etc entre el servidor a reconocer y la red externa, en este caso Internet, pero también hay otras formas que requieren de la actuación del administrador. A continuación se enumeran las más importantes:

### ▪ MODIFICACIÓN DEL BANNER DEL SERVICIO

Los servidores web y otros servicios de red (telnet, FTP, etc) disponibles en cualquier servidor, muestran un banner cuando se conecta a ellos. Este banner puede ser utilizado para sacar información de un servidor y de este modo poder saber qué vulnerabilidades afectan a ese software, si el sistema está correctamente actualizado y parcheado, etc.

Por ejemplo conectando a un servidor web se obtiene la siguiente información:

```
HTTP/1.1 200 OK
Date: Thu, 21 May 2006 21:13:58 GMT
Server: Apache/2.0.41 (Unix) (Red-Hat/Linux) mod_ssl/2.8.12
Connection: close
Content-Type: text/html
```

Se observa que el Sistema Operativo del servidor es un Linux Red Hat, que el servidor Web es un Apache y hasta se ha conseguido la versión del mismo, es decir, gran cantidad de información para realizar una posterior intrusión.

Es muy recomendable alterar el banner por defecto de cualquier servicio que el servidor tenga activo para no informar de la versión del sistema operativo o del software instalado. Es importante remarcar que la modificación del banner no afecta al resultado del Fingerprinting realizado con las herramientas comentadas en este documento, dichas herramientas (nmap, xprobe2, etc) se basan en analizar el comportamiento de la pila TCP/IP de cada Sistema Operativo y no en la información que muestran los banners de servicio de los distintos servidores revisados.

En sistemas Linux es suficiente con modificar los ficheros de configuración del servicio para introducir lo que el administrador del servidor remoto considere oportuno o, incluso, modificar el código fuente de cada servicio antes de instalarlo:

/etc/issue y /etc/issue.net para un servicio telnet.  
/httpd.conf para el servidor Apache.  
/named.conf para un servicio de DNS bajo BIND.  
/sendmail.cf para un servicio de correo con Sendmail.

Además, mod\_security incluye una función que permite cambiar la identidad de un servidor Apache.

En sistemas Windows se puede utilizar URLScan y Servermask (de pago) que permiten cambiar el banner de un servidor web IIS (Internet Information Server). En Windows 2000 se pueden modificar dos claves del registro que permiten ofuscar el Sistema Operativo. Estas claves son:

HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters:

- Clave DefaultTTL, valor DWORD se puede poner a 64
- Clave TcpWindowSize, valor DWORD se puede poner a 24.820 y simular ser un sistema Solaris o a 32.120 y parecer un Linux, por ejemplo.
- Clave GlobalMaxTcpWindowSize, mismos valores que en la anterior clave.

Para un servidor de correo bajo Windows también se puede utilizar una herramienta para editar la metabase, metaedit por ejemplo, y quitar así la información acerca del servicio SMTP que proporciona el banner por defecto. Se busca el siguiente archivo y se siguen los siguientes pasos:

Lm\Smtpsvc\número de servidor virtual

clic en Modificar, clic en Nuevo y, a continuación, clic en Cadena, comprobar que la entrada del cuadro Id. es otros y escribir 36907 (decimal) a la derecha del cuadro Id, en el cuadro Valor, escribir la información que se desea que aparezca y se reinicia el servidor virtual SMTP o el servicio SMTP

## ▪ MODIFICACIÓN DE LA PILA TCP

La pila TCP, como se ha visto, es fundamental para permitir el reconocimiento del Sistema Operativo de un servidor remoto y cada sistema se comporta de diferente manera, es por ello, que existen herramientas y mecanismos para modificar el comportamiento de la pila TCP/IP (tamaño de ventana, TTL,...) de cualquier servidor y dificultar así el proceso de Fingerprinting para confundir a los programas de escaneo vistos en este documento (nmap, xprobe, etc.), se pueden modificar incluso los tiempos de respuesta que tiene por defecto cualquier servidor y así hacer creer que se trata de otro tipo de sistema.

Una de esas herramientas es IP Personality que permite ocultar el comportamiento específico de una implementación TCP/IP en Linux con el

kernel 2.4.20 como máximo pero que ya no se desarrolla actualmente debido a que el código era poco estable. Además, existen numerosos parches para el kernel que ayudan a esconder el Sistema Operativo de un servidor. Uno de los parches para el kernel más conocidos es “stealth patch” que, actuando como un firewall, básicamente, realiza lo siguiente:

- Bloquea paquetes ACK incorrectos.
- Bloquea paquetes con flags activados incorrectamente.
- Bloquea paquetes con el flag SYN y RST.
- Ignora el tráfico ICMP a excepción del Echo Reply (Ping).

Manualmente se pueden hacer algunas modificaciones en la pila TCP/IP que no precisan de ninguna herramienta externa:

## ▪ MODIFICACIÓN DEL TTL

el Time To Live (TTL) tiene un valor predeterminado de 64 en sistemas Linux y 128 en sistemas Windows, cambiando ese valor se consigue confundir a la herramienta de escaneo Xprobe2, a Nmap no porque no utiliza el valor TTL en sus pruebas, tal y como se puede observar a continuación:

```
# uname -a (se comprueba la versión del kernel existente en el sistema)
Linux localhost 2.6.17-1.2139_FC4 #1 Mon XXX XX 09:53:28 CEST 200X i686 GNU/Linux
```

```
C:\ >ver
```

```
Microsoft Windows XP [Versión 5.1.2600]
(se comprueba que la versión del sistema MS Windows instalado)
```

```
# more /proc/sys/net/ipv4/ip_default_ttl
64
(se comprueba que el TTL por defecto es 64 para sistemas Linux)
```

```
c:\>ping localhost
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
Respuesta desde 127.0.0.1: bytes=32 tiempo<1m TTL=128
Respuesta desde 127.0.0.1: bytes=32 tiempo<1m TTL=128
(se comprueba que el TTL por defecto es 128 para sistemas Windows)
```

Se lanza Xprobe2 desde otro sistema adyacente para ver el resultado del proceso de Fingerprinting antes de modificar el TTL por defecto de un sistema Linux y un sistema Windows:

```
#xprobe2 192.168.1.239 -p TCP:22:open -p TCP:80:closed -p UDP:53:open -p UDP:55:closed -D 1
```

```
[+] Primary guess:
[+] Host 192.168.1.239 Running OS: "Linux Kernel 2.6.10" (Guess
probability: 94%)
```

```
#xprobe2 192.168.1.189 -p TCP:445:open -p TCP:80:closed -p
UDP:135:open -p UDP:55:closed -D 1
```

```
[+] Primary guess:
[+] Host 192.168.1.189 Running OS: "Microsoft Windows XP SP2"
(Guess probability: 100%)
```

```
# echo 128 > /proc/sys/net/ipv4/ip_default_ttl (se cambia el
TTL por defecto de un sistema Linux a un valor de 128)
```

c:\>regedit HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\  
se crea o se modifica una nueva variable DWORD con el nombre DefaultTTL  
y se introduce el valor deseado 192, por ejemplo,

Se cambia el TTL por defecto y se añade un valor siempre por debajo de 256  
para no crear inestabilidad en el sistema Windows.

Se vuelve a lanzar Xprobe2 contra ambos sistemas y se comprueba el resultado  
una vez modificado el valor de TTL:

```
[+] Primary guess:
[+] Host 192.168.1.239 Running OS: "Linux Kernel 2.4.29" (Guess
probability: 86%)
```

```
[+] Primary guess:
[+] Host 192.168.1.189 Running OS: "Microsoft Windows XP SP2"
(Guess probability: 90%)
```

El resultado ha variado sensiblemente, se sigue detectando como un Sistema  
Linux pero la versión del kernel ha cambiado relativamente dificultando así el  
proceso de Fingerprinting del Sistema Operativo presente en el servidor Linux  
analizado. Para el sistema Windows se mantiene que es un Windows XP SP2  
pero la probabilidad ha bajado sensiblemente al 90%.Con nmap el resultado es  
el siguiente:

```
Device          type:          general          purpose
Running:                Linux          2.6.X
OS details: Linux 2.6.10 - 2.6.17
```

```
Device          type:          general          purpose
Running:  Microsoft Windows 2003/ .NET|NT/2k/XP
OS details: Microsoft Windows 2003 Server or XP SP2
```

La modificación del TTL no ha influido en absoluto y el Sistema Operativo del  
servidor es casi reconocido a la perfección.

## ▪ MODIFICACIÓN DE TCP TIMESTAMP Y DEL TAMAÑO DE LA VENTANA

en Linux vienen activados por defecto los parámetros de TCP Timestamp y el TCP Window Scale, si se desactivan el resultado del fingerprinting con nmap varía más sensiblemente:

```
# echo 0 > /proc/sys/net/ipv4/tcp_window_scaling (se desactiva el valor del tamaño de la ventana TCP)
# echo 0 > /proc/sys/net/ipv4/tcp_timestamps (se desactiva el valor del TCP Timestamp)
```

```
c:\>regedit
HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Interfaces (se modifica el valor de la ventana en Windows 2000)
HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\ (se modifica el valor de la ventana en Windows 2003)
se crea o se modifica una nueva variable DWORD con el nombre TcpWindowSize y se introduce el valor deseado
```

Se analiza con nmap:

```
Device type: general purpose
Running: Linux 2.4.X
OS details: Linux 2.4.7 (x86)
```

La detección del sistema Windows con nmap permanece inalterable con la modificación realizada en el parámetro del tamaño de ventana.

Se analiza con Xprobe2:

```
[+] Primary guess:
[+] Host 192.168.1.239 Running OS: "Linux Kernel 2.4.30" (Guess probability: 76%)
```

```
[+] Primary guess:
[+] Host 192.168.1.239 Running OS: "Microsoft Windows XP SP2" (Guess probability:87%)
```

El resultado cambia sensiblemente y el tanto por ciento de aproximación va descendiendo desde el 94% inicial, antes de realizar ninguna modificación en los parámetros de la pila TCP/IP del sistema, hasta el actual 76%. Revisando con nmap se obtiene que es un sistema Linux también pero ahora ya no se ajusta tanto con la versión del kernel existente. Para el sistema Windows sigue bajando la probabilidad hasta un 87%.

## ▪ ICMP REDIRECTS

las distintas implementaciones TCP/IP de los sistemas disponen de medidas para controlar e, incluso desactivar, este tipo de paquetes ya que son

innecesarios en equipos finales que no son dispositivos de red. Para cambiar este parámetro de la pila TCP/IP en Windows y en Linux se realiza lo siguiente:

```
c:\>regedit
HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\
se crea o se modifica una nueva variable DWORD con el nombre
EnableICMPRedirects y se introduce el valor 0 (por defecto está
a 1)
```

```
# echo 0 > /proc/sys/net/ipv4/conf/all/accept_redirects (se
desactiva el valor para aceptar este tipo de paquetes ICMP)
# echo 0 > /proc/sys/net/ipv4/conf/all/send_redirects (se
desactiva el valor para enviar paquetes ICMP del tipo
redirects)
```

Se analiza con Xprobe2:

```
[+] Primary guess:
[+] Host 192.168.1.239 Running OS: "Linux Kernel 2.4.19" (Guess
probability: 97%)
```

```
[+] Primary guess:
[+] Host 192.168.1.239 Running OS: "Microsoft Windows XP SP2"
(Guess probability:82%)
```

La detección varía de nuevo sensiblemente con Xprobe2, el sistema sigue siendo Linux y Windows pero han cambiado las versiones del kernel encontradas y el tanto por ciento de la probabilidad del Xprobe2.

Se analiza con nmap:

```
Device type: general purpose
Running: Linux 2.4.X|2.5.X@2.6.X
OS details: Linux 2.4.0 - 2.5.20, Linux 2.4.7 - 2.6.11
```

Con nmap el proceso de fingerprinting del sistema Windows no ha tenido éxito (se ha conseguido camuflar y ocultar totalmente el Sistema Operativo del servidor analizado) y, como es habitual, nos muestra la huella TCP que no ha sido capaz de procesar al compararla con alguno de los resultados que tiene en su base de datos. En el sistema Linux se sigue reconociendo que se trata de un servidor Linux pero cada vez es más inexacto acerca de las versiones del kernel instaladas.

## REFERENCIAS:

- NETCRAFT <http://news.netcraft.com> [3]
- NMAP <http://www.insecure.org> [4]
- XPROBE2 <http://xprobe.sourceforge.net> [5]
- SinFP

<http://www.gomor.org/cgi-bin/index.pl?mode=view;page=sinfp> [6]  
- mod\_security <http://www.modsecurity.org/> [7]  
- URLScan  
<http://www.microsoft.com/technet/security/tools/urlscan.mspix> [8]  
- Servermask  
<http://www.port80software.com/products/servermask/iis> [9]  
- metaedit  
<http://download.microsoft.com/download/iis50/Utility/5.0/NT45/EN-US/MtaEdt22.exe>  
[10]  
- ip-personality <http://ippersonality.sourceforge.net/> [11]  
- stealth patch <http://www.securityfocus.com/tools/1747> [12]

---

**Source URL:**

<http://www.hacktimes.com/?q=node/36>

**Links:**

[1] <http://www.insecure.org/nmap/nmap-fingerprinting-article.html>  
[2] <http://insecure.org/nmap/osdetect/>  
[3] <http://news.netcraft.com/>  
[4] <http://www.insecure.org>  
[5] <http://xprobe.sourceforge.net/>  
[6] <http://www.gomor.org/cgi-bin/index.pl?mode=view;page=sinfp>  
[7] <http://www.modsecurity.org/>  
[8] <http://www.microsoft.com/technet/security/tools/urlscan.mspix>  
[9] <http://www.port80software.com/products/servermask/iis>  
[10] <http://download.microsoft.com/download/iis50/Utility/5.0/NT45/EN-US/MtaEdt22.exe>  
[11] <http://ippersonality.sourceforge.net/>  
[12] <http://www.securityfocus.com/tools/1747>