

SIN CLASIFICAR



**GUÍA DE SEGURIDAD DE LAS TIC
(CCN-STIC-954)**

GUÍA AVANZADA DE NMAP

AGOSTO 2012

SIN CLASIFICAR

Edita:



Centro Criptológico Nacional, 2012

Tirada: 1000 ejemplares

Fecha de Edición: agosto de 2012

José Vila y José Luis Chica del Centro de Seguridad TIC de la Comunidad Valenciana (CSIRT-cv) han participado en la elaboración y modificación del presente documento y sus anexos.

LIMITACIÓN DE RESPONSABILIDAD

El presente documento se proporciona tal cual, rechazando expresamente cualquier tipo de garantía implícita que se pueda encontrar relacionada. En ningún caso, el **Centro Criptológico Nacional** puede ser considerado responsable del daño directo, indirecto, fortuito o extraordinario derivado de la utilización de la información y software que se indican incluso cuando se advierta de tal posibilidad.

AVISO LEGAL

Quedan rigurosamente prohibidas, sin la autorización escrita del **Centro Criptológico Nacional**, bajo las sanciones establecidas en las leyes, la reproducción parcial o total de este documento por cualquier medio o procedimiento, comprendidos la reprografía y el tratamiento informático, y la distribución de ejemplares del mismo mediante alquiler o préstamo públicos.

PRÓLOGO

El uso masivo de las tecnologías de la información y las telecomunicaciones (TIC), en todos los ámbitos de la sociedad, ha creado un nuevo espacio, el ciberespacio, donde se producirán conflictos y agresiones, y donde existen ciberamenazas que atentarán contra la seguridad nacional, el estado de derecho, la prosperidad económica, el estado de bienestar y el normal funcionamiento de la sociedad y de las administraciones públicas.

La Ley 11/2002, de 6 de mayo, reguladora del Centro Nacional de Inteligencia, encomienda al Centro Nacional de Inteligencia el ejercicio de las funciones relativas a la seguridad de las tecnologías de la información en su artículo 4.e), y de protección de la información clasificada en su artículo 4.f), a la vez que confiere a su Secretario de Estado Director la responsabilidad de dirigir el Centro Criptológico Nacional en su artículo 9.2.f).

Partiendo del conocimiento y la experiencia del CNI sobre amenazas y vulnerabilidades en materia de riesgos emergentes, el Centro realiza, a través de su Centro Criptológico Nacional, regulado por el Real Decreto 421/2004, de 12 de marzo, diversas actividades directamente relacionadas con la seguridad de las TIC, orientadas a la formación de personal experto, a la aplicación de políticas y procedimientos de seguridad, y al empleo de tecnologías de seguridad adecuadas.

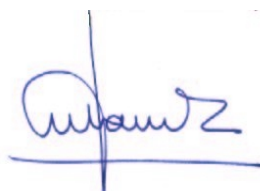
Una de las funciones más destacables del Centro Criptológico Nacional es la de elaborar y difundir normas, instrucciones, guías y recomendaciones para garantizar la seguridad de los sistemas de las tecnologías de la información y las comunicaciones de la Administración, materializada en la existencia de la serie de documentos CCN-STIC.

Disponer de un marco de referencia que establezca las condiciones necesarias de confianza en el uso de los medios electrónicos es, además, uno de los principios que establece la ley 11/2007, de 22 de junio, de acceso electrónico de los ciudadanos a los servicios públicos, en su artículo 42.2 sobre el Esquema Nacional de Seguridad (ENS).

Precisamente el Real Decreto 3/2010 de 8 de Enero de desarrollo del Esquema Nacional de Seguridad fija los principios básicos y requisitos mínimos así como las medidas de protección a implantar en los sistemas de la Administración, y promueve la elaboración y difusión de guías de seguridad de las tecnologías de la información y las comunicaciones por parte de CCN para facilitar un mejor cumplimiento de dichos requisitos mínimos.

En definitiva, la serie de documentos CCN-STIC se elabora para dar cumplimiento a los cometidos del Centro Criptológico Nacional y a lo reflejado en el Esquema Nacional de Seguridad, conscientes de la importancia que tiene el establecimiento de un marco de referencia en esta materia que sirva de apoyo para que el personal de la Administración lleve a cabo su difícil, y en ocasiones, ingrata tarea de proporcionar seguridad a los sistemas de las TIC bajo su responsabilidad.

Agosto de 2012



Félix Sanz Roldán
Secretario de Estado
Director del Centro Criptológico Nacional

ÍNDICE

1.	INTRODUCCIÓN.....	2
1.1.	ANALIZADOR DE REDES NMAP.....	2
1.2.	OBJETIVOS DE LA GUÍA	4
1.3.	AUDIENCIA DE ESTE DOCUMENTO.....	5
1.4.	TECNOLOGÍAS PRESENTES	5
1.5.	TERMINOLOGÍA.....	6
1.6.	HISTORIAL DE VERSIONES DE NMAP	9
2.	INSTALACIÓN.....	10
2.1.	NMAP PARA LINUX	11
2.1.1.	INSTALACIÓN A PARTIR DE CÓDIGO FUENTE.....	11
2.1.2.	INSTALACIÓN BAJO LINUX A PARTIR DE PAQUETES (RPM Y DEB).....	12
2.2.	NMAP PARA WINDOWS	13
2.3.	INTERFAZ GRÁFICA DE USUARIO	14
2.4.	INTERACCIÓN CON APLICACIONES DE TERCEROS.....	16
3.	LÍNEA DE COMANDOS	18
3.1.	TÉCNICAS DE DESCUBRIMIENTO DE EQUIPOS	18
3.1.1.	NO PING (-Pn)	18
3.1.2.	LIST SCAN (-sL)	18
3.1.3.	NO PORT SCAN (-sn).....	20
3.1.4.	PING ARP (-PR).....	22
3.1.5.	PING TCP SYN (-PS<listado de puertos>).....	23
3.1.6.	PING TCP ACK (-PA<listado de puertos>).....	25
3.1.7.	PING UDP (-PU<lista de puertos>).....	26
3.1.8.	PINGS ICMP (-PE, -PP, -PM).....	27
3.1.9.	PING SCTP (-PY<listado de puertos>).....	28
3.1.10.	IP PROTOCOL PING (-PO<listado protocolos>).....	30
3.2.	TÉCNICAS BASADAS EN EL ESCANEÓ DE PUERTOS	31
3.2.1.	TCP SYN Scan (-sS)	31
3.2.2.	TCP CONNECT SCAN (-sT).....	32
3.2.3.	UDP SCAN (-sU)	33
3.2.4.	IDLE SCAN (-sI)	35
3.2.5.	TCP ACK SCAN (-sA).....	38
3.2.6.	TCP Null, FIN, Xmas scans (-sN, -sF, -sX).....	39
3.2.7.	TCP Maimon scan (-sM).....	41
3.2.8.	TCP Window scan (-sW).....	41
3.2.9.	SCTP INIT Scan (-sY)	42
3.2.10.	SCTP COOKIE-ECHO Scan (-sZ)	43
3.2.11.	IP protocol scan (-sO).....	44
3.2.12.	FTP bounce scan (-b <servidor ftp>)	45
3.3.	MEJORANDO EL RENDIMIENTO DEL ANÁLISIS	46
3.3.1.	CENTRAR EL ALCANCE DEL ANÁLISIS.....	46
3.3.2.	SEPARAR Y OPTIMIZAR LOS ANÁLISIS UDP.....	47
3.3.3.	TUNING TEMPORAL AVANZADO	47
3.3.4.	PLANTILLAS TEMPORALES.....	48
3.4.	ANÁLISIS DE REDES IPv6	49
3.5.	OTROS PARÁMETROS RELEVANTES	49
3.5.1.	OBJETIVOS A ANALIZAR (-iL, -iR).....	50
3.5.2.	SALIDA DE RESULTADOS (-oN, -oX, -oS, -oG, -oA)	50
3.5.3.	DETECCIÓN DE VERSIONES (-sV).....	50
3.5.4.	DETECCIÓN DE SISTEMA OPERATIVO (-O)	51
3.5.5.	USO DE SCRIPTS (-sC)	51
3.5.6.	ANÁLISIS AGRESIVO (-A).....	51
3.5.7.	INTERACTUANDO CON NMAP DURANTE UN ANÁLISIS.....	51
4.	PROCEDIMIENTOS.....	52

4.1.	TÉCNICAS DE DESCUBRIMIENTO DE EQUIPOS EN UNA SUBRED	52
4.1.1.	DESCRIPCIÓN DEL PROCEDIMIENTO	52
4.1.2.	CONFIGURACIÓN DE NMAP	53
4.1.3.	RESULTADOS	54
4.2.	TÉCNICAS DE ANÁLISIS DE PUERTOS	56
4.2.1.	DESCRIPCIÓN DEL PROCEDIMIENTO	57
4.2.2.	CONFIGURACIÓN DE NMAP	57
4.2.3.	RESULTADOS	60
4.3.	TÉCNICAS DE DETECCIÓN DE SERVICIOS Y DE SISTEMA OPERATIVO	64
4.3.1.	DESCRIPCIÓN DEL PROCEDIMIENTO	65
4.3.2.	CONFIGURACIÓN DE NMAP	66
4.3.3.	RESULTADOS	67
4.4.	TÉCNICAS DE EVASIÓN DE CORTAFUEGOS Y HERRAMIENTAS IDS/IPS	76
4.4.1.	DESCRIPCIÓN DEL PROCEDIMIENTO	77
4.4.2.	CONFIGURACIÓN DE NMAP	78
4.4.3.	RESULTADOS	79
4.5.	TÉCNICAS DE OPTIMIZACIÓN DEL ANÁLISIS	81
4.5.1.	DESCRIPCIÓN DEL PROCEDIMIENTO	81
4.5.2.	CONFIGURACIÓN DE NMAP	82
5.	NMAP SCRIPTING ENGINE	84
5.1.	INTRODUCCIÓN	84
5.2.	CATEGORÍAS	85
5.3.	FORMATO	86
5.3.1.	CAMPO <i>description</i>	86
5.3.2.	CAMPO <i>categories</i>	86
5.3.3.	CAMPO <i>author</i>	86
5.3.4.	CAMPO <i>license</i>	86
5.3.5.	CAMPO <i>dependencies</i>	86
5.3.6.	REGLAS	86
5.3.7.	FUNCIÓN <i>ACTION</i>	87
5.4.	LIBRERÍAS	87
5.5.	API	87
5.5.1.	ESTRUCTURAS <i>host Y port</i>	87
5.5.2.	API DE RED	89
5.5.3.	MANEJO DE EXCEPCIONES	89
5.6.	EJEMPLO	90
5.6.1.	CABECERA	90
5.6.2.	LAS REGLAS	90
5.6.3.	LA ACCIÓN	90
5.7.	EJEMPLO DE USO	91
5.8.	DEFENSAS CONTRA NMAP	92
5.8.1.	ESCANEOS PROACTIVOS	92
5.8.2.	CORTAFUEGOS CON POLÍTICA POR DEFECTO <i>DROP</i>	92
5.8.3.	SERVICIOS OCULTOS	92
ANEXOS		94
5.9.	ANEXO A.- ANÁLISIS DE RESULTADOS	94
5.9.1.	A1.- ANÁLISIS DE RESULTADOS DEL PROCEDIMIENTO 1	95
5.9.2.	A2.- ANÁLISIS DE RESULTADOS DEL PROCEDIMIENTO 2	96
5.9.3.	A3.- ANÁLISIS DE RESULTADOS DEL PROCEDIMIENTO 3	97
5.9.4.	A4.- ANÁLISIS DE RESULTADOS DEL PROCEDIMIENTO 4	98
5.9.5.	A5.- ANÁLISIS DE RESULTADOS DEL PROCEDIMIENTO 5	99
5.10.	ANEXO B.- LISTADO DE COMANDOS NMAP	100

Figuras

FIGURA 1: INSTALABLES EN LA VERSIÓN PARA WINDOWS DE NMAP	13
FIGURA 2: [ZENMAP] VENTANA PRINCIPAL CON RESULTADO DE ANÁLISIS	15
FIGURA 3: [ZENMAP] VISTA DE EQUIPOS Y PUERTOS	1
FIGURA 4: [ZENMAP] VISTA DE SERVICIOS Y EQUIPOS	1
FIGURA 5: [ZENMAP] VISTA DE TOPOLOGÍA DE RED.....	1
FIGURA 6: [ZENMAP] VISTA DE DETALLES DE EQUIPO	16
FIGURA 7: [ZENMAP] PANTALLA DE CREACIÓN DE PERFIL DE ANÁLISIS.....	16
FIGURA 8.- TÉCNICA BÁSICA DE DESCUBRIMIENTO LIST SCAN (-sL)	19
FIGURA 9.- TÉCNICA BÁSICA DE DESCUBRIMIENTO PING SCAN (-SN).....	21
FIGURA 10.- TÉCNICA BÁSICA DE DESCUBRIMIENTO PING SCAN (-SN) EN EQUIPOS SIN PRIVILEGIOS	21
FIGURA 11.- TÉCNICA BÁSICA DE DESCUBRIMIENTO PING ARP (-PR)	23
FIGURA 12.- TÉCNICA AVANZADA DE DESCUBRIMIENTO PING TCP SYN (-PS).....	24
FIGURA 13.- TÉCNICA AVANZADA DE DESCUBRIMIENTO PING TCP SYN (-PS) SOBRE VARIOS PUERTOS	25
FIGURA 14.- TÉCNICA AVANZADA DE DESCUBRIMIENTO PING TCP ACK (-PA)	26
FIGURA 15.- TÉCNICA AVANZADA DE DESCUBRIMIENTO PING UDP (-PU).....	27
FIGURA 16.- TÉCNICA AVANZADA DE DESCUBRIMIENTO PINGS ICMP (-PE, -PP, -PM)	28
FIGURA 17.- TÉCNICA AVANZADA DE IP PROTOCOL PING (-PO)	30
FIGURA 18.- TÉCNICA BÁSICA DE ESCANEO TCP SYN SCAN (-sS).....	32
FIGURA 19.- TÉCNICA BÁSICA DE ESCANEO TCP CONNECT SCAN (-sT).....	33
FIGURA 20.- TÉCNICA BÁSICA DE ESCANEO UDP SCAN (-sU).....	35
FIGURA 21.- TÉCNICA AVANZADA DE ESCANEO IDLE SCAN (-sI: ORIGEN Y OBJETIVO).....	37
FIGURA 22.- TÉCNICA AVANZADA DE ESCANEO IDLE SCAN (-SI): ZOMBIE Y OBJETIVO	38
FIGURA 23.- TÉCNICA AVANZADA DE ESCANEO TCP ACK SCAN (-sA)	39
FIGURA 24.- TÉCNICA AVANZADA DE ESCANEO TCP NULL, FIN Y XMAS SCANS (-sN, -sF, -sX, -sM)	41
FIGURA 25.- TÉCNICA AVANZADA DE ESCANEO IP PROTOCOL SCAN (-sO).....	45
FIGURA 26.- RESULTADOS PROCEDIMIENTO 1: LISTADO DE MÁQUINAS PARA WWW.UNIVERSIDAD.ES/24.	55
FIGURA 27.- RESULTADOS PROCEDIMIENTO 1: ESTADO DE MÁQUINAS PARA WWW.EMPRESA.ES/24.	56
FIGURA 28.- ENUMERACIÓN DE PUERTOS CON SYN STEALTH. PROCEDIMIENTO2 - CASO 1	58
FIGURA 29.- IDENTIFICACIÓN DE SERVIDORES EN UNA RED. PROCEDIMIENTO2-CASO5	60
FIGURA 30.- SALIDA PROCEDIMIENTO 2 - CASO 1.....	60
FIGURA 31.- SALIDA PROCEDIMIENTO 2 - CASO 2.....	61
FIGURA 32.- SALIDA PROCEDIMIENTO 2 - CASO 3.....	61
FIGURA 33.- SALIDA PROCEDIMIENTO 2 - CASO 4.....	61
FIGURA 34.- SALIDA PROCEDIMIENTO 2 - CASO 5.2.....	62
FIGURA 35.- SALIDA PROCEDIMIENTO 3 - CASO 1.....	67
FIGURA 36.- SALIDA PROCEDIMIENTO 3 - CASO 2.1.....	68
FIGURA 37.- SALIDA PROCEDIMIENTO 3 - CASO 2.2.....	70
FIGURA 38.- SALIDA PROCEDIMIENTO 3 - CASO 3.....	70
FIGURA 39.- SALIDA PROCEDIMIENTO 3 - CASO 4.....	75
FIGURA 40.- SALIDA PROCEDIMIENTO 4 - CASO 1.....	79
FIGURA 41.- SALIDA PROCEDIMIENTO 4 - CASO 2.....	80
FIGURA 42.- SALIDA PROCEDIMIENTO 4 - CASO 3.....	80

Tablas

TABLA 1. COMPARATIVA RESULTADOS PROCEDIMIENTO 1.....	56
TABLA 2. COMPARATIVA RESULTADOS PROCEDIMIENTO 2.....	64
TABLA 3. COMPARATIVA RESULTADOS PROCEDIMIENTO 3.....	76
TABLA 4. COMPARATIVA RESULTADOS PROCEDIMIENTO 4.....	80

1. INTRODUCCIÓN

1.1. ANALIZADOR DE REDES NMAP

1. *Nmap* (Network Mapper, mapeador de redes) es una sofisticada utilidad para la exploración y auditoría de seguridad de redes TCP/IP. Ha sido diseñado para escanear¹ de forma rápida, sigilosa y eficaz tanto equipos individuales como redes de gran tamaño. Es una herramienta gratuita, de código abierto bajo licencia GPL, bien documentada, multiplataforma, disponible para consola, y que ofrece también una interfaz gráfica para facilitar su uso. Está escrita por un *hacker* conocido como Fyodor², y se beneficia de las aportaciones de una nutrida comunidad de colaboradores.
2. *Nmap* es una popular herramienta de seguridad utilizada tanto por administradores de red y analistas de seguridad, como por atacantes. Esto es debido a la gran cantidad de información que es capaz de descubrir de una red utilizando una gran variedad de técnicas que la hacen notablemente efectiva y sigilosa. Para ello, *Nmap* explora equipos remotos mediante secuencias de paquetes TCP/IP tanto convencionales como no convencionales, es decir, paquetes en bruto convenientemente modificados que provocarán o no una respuesta en el objetivo de la cual poder extraer información. Entre esta información se encuentra, por ejemplo: el estado de los puertos y servicios, el sistema operativo, la presencia de cortafuegos, encaminadores u otros elementos de red, así como del direccionamiento IP de la subred. El tipo de respuestas recibidas ayudan a determinar la identidad de la pila TCP/IP implementada en el sistema operativo remoto.
3. La información extraída con *Nmap* puede ser utilizada para múltiples usos. Los más habituales son los siguientes:
 - Descubrimiento de subredes.
 - Análisis de penetración de redes y equipos.
 - Evaluación de la implantación de cortafuegos y de la eficacia de herramientas de detección y prevención de intrusiones.
 - Descubrimiento del estado de puertos de comunicaciones.
 - Descubrimiento de los servicios disponibles en un servidor, así como de sus versiones.
 - Descubrimiento del tipo y versión del sistema operativo instalado en el equipo remoto.

¹ Determinar qué equipos están activos y cuál es el estado de sus puertos y servicios de comunicaciones ofrecidos.

² Fyodor es el apodo de un conocido *hacker*, en honor al escritor ruso Fyodor Dostoyevsky, Ingeniero Informático y autor de *Nmap* así como de otras herramientas informáticas de seguridad. Actualmente sigue dirigiendo el proyecto *Nmap*, manteniendo la popular web de seguridad www.insecure.org, así como colaborando en diversos libros y publicaciones relacionados con *Nmap*.

- Obtención de información adicional acerca de servicios y equipos, a través de la ejecución de scripts convenientemente elaborados.
4. Cabe destacar, como ya se ha comentado anteriormente, que la información que puede proporcionar *NMap* es extensa y detallada, por lo que algunos de los usos indicados en esta lista pueden enfocarse desde una óptica de dudosa legalidad, siendo algunos de ellos previos a fines deshonestos como obtener acceso no autorizado a un sistema.
 5. Nmap nació con la idea de unificar en una sola herramienta multitud de escáneres de puertos de código abierto, tales como *Julian Assange Scanner*, *Reflscan SYN Scanner*, *SATAN UDP Scanner*, *Uriel Maimon FIN Scanner* y muchos más, tratando de superar las limitaciones que por separado tenía cada uno de ellos. Se desarrolló partiendo de cero, e implementaba versiones modificadas de múltiples tipos de análisis con el objetivo de hacerlo rápido y eficiente contra redes de gran tamaño.
 6. Con el tiempo, tanto la herramienta NMap como la comunidad a su alrededor han ido creciendo, lo que ha dotado a la herramienta de nuevas características y funcionalidad, convirtiéndola hoy por hoy en una suite que, además del analizador de red NMap, incluye las siguientes herramientas:
 - *Nping*: generador de paquetes, analizador de respuestas y medidor de tiempos de respuesta. Permite generar paquetes de un gran rango de protocolos, permitiendo a los usuarios manipular virtualmente cualquier campo de las cabeceras de los protocolos. Además de ser usado como una simple utilidad de ping, *Nping* puede ayudar, como generador de paquetes en bruto, a tareas como pruebas de estrés, envenenamiento ARP, rastreo de rutas, ataques de denegación de servicio, etcétera.
 - *Ncat*: reimplementación de la conocida herramienta *Netcat*. Es una herramienta de red que transporta paquetes entre distintas redes. Se ha diseñado para ser una herramienta robusta que pueda proveer de conectividad a otras aplicaciones y usuarios. Permite, por ejemplo, redireccionar tráfico de puertos TCP y UDP a otros destinos, ser utilizado como Proxy HTTP, incluso con autenticación, etcétera.
 - *Ndiff*: herramienta para la comparación de diferentes análisis realizados por *Nmap*. A partir de los ficheros de salida de dos análisis diferentes sobre la misma red, muestra las diferencias existentes entre ellos. Es de utilidad para mostrar cambios recientes en sus redes a administradores de red que realizan análisis periódicamente.
 - *Zenmap*: interfaz gráfica multiplataforma y libre, soportada oficialmente por los desarrolladores de *NMap*. Su objetivo es facilitar a los principiantes el uso de la

aplicación, mientras provee funcionalidades avanzadas para usuarios más experimentados.

7. Las herramientas *Nping*, *Ncat* y *Ndiff* quedan fuera del alcance de esta guía, por lo que su funcionamiento y características detalladas no se van a tratar aquí. Por otra parte, el funcionamiento de la interfaz gráfica Zenmap, al ser una capa sobre Nmap, si que va a ser tratado en el apartado correspondiente con las interfaces de usuario (ver apartado 2.3).

1.2. OBJETIVOS DE LA GUÍA

8. El presente trabajo pretende desarrollar una guía útil a la vez que tecnológicamente rigurosa. Con este propósito, el enfoque elegido ha sido notablemente práctico. Tras una pequeña introducción teórica sobre algunas de las técnicas que se pueden encontrar en NMap, se pasa a mostrar ejemplos representativos de situaciones reales a las que podría llegar a enfrentarse un administrador del sistema, de forma que se pueda observar el potencial de que dispone NMap. De este modo, para la obtención de resultados y conclusiones satisfactorias, se han definido los siguientes objetivos:
 - Elaboración de una Guía Avanzada de Nmap que permita a analistas de seguridad y administradores de sistemas evaluar con rigor el grado de penetración de un equipo, o un conjunto de éstos, cortafuegos o sistemas de detección y prevención de intrusos mediante esta herramienta, haciendo uso de las mismas técnicas que utilizarían sus potenciales atacantes.
 - La Guía Avanzada de *Nmap* se completa con una robusta documentación, entre la que se incluirán aquellas funcionalidades provistas que permitan conocer a usuarios avanzados los detalles de las comunicaciones a bajo nivel, producidas entre el motor de Nmap y los objetivos escaneados.
 - Más allá del empleo aislado de comandos, esta Guía Avanzada propone, a modo de ejemplos, unos completos procedimientos de análisis, que faciliten el uso y explotación de las utilidades que proporciona *Nmap*, por parte de usuarios con distintos niveles de conocimientos en seguridad y comunicaciones de sistemas.
 - Por último, esta Guía Avanzada aspira a realizar un adecuado tratamiento de los resultados obtenidos en los análisis para mejor aprovechamiento de la herramienta. Dichos resultados incluirán suficiente información estadística que permita comprender el impacto de la elección de las diferentes técnicas con respecto al rendimiento en tiempo de realización del análisis y a la cantidad de información extraída en el proceso.

1.3. AUDIENCIA DE ESTE DOCUMENTO

9. La presente guía tiene como motivación inicial la de servir a cualquier empresa, organismo, institución, administrador o usuario con preocupaciones o inquietudes en la evaluación remota del nivel de seguridad de sistemas conectados a una red TCP/IP, a través del análisis del estado de sus puertos de comunicaciones y del comportamiento de la pila de protocolos que implementan.
10. Naturalmente, esta audiencia debe ser concedora de que Nmap forma parte del conjunto de herramientas utilizadas tanto por administradores de sistemas como por atacantes y usuarios maliciosos, por ser la referencia en software de descubrimiento de equipos y escaneo de puertos. Por ello, se espera que esta guía permita al lector aumentar sus habilidades y grado de conocimiento en la protección de los sistemas y, por tanto, situar a su audiencia en una posición idónea, o incluso aventajada, para el descubrimiento de fallos y la posterior protección de los sistemas que administren frente a los posibles atacantes.

1.4. TECNOLOGÍAS PRESENTES

11. La mayoría de las tecnologías aplicadas a la seguridad en sistemas de información pueden ser utilizadas tanto para la protección de éstos como para propósitos menos honestos, incluso ilegales. La presente guía afronta el reto de mostrar a usuarios bien intencionados un conjunto de aplicaciones reales de algunas de estas tecnologías, pero no se abstrae de los usos controvertidos que pudiera ocasionar, advirtiéndolo en los casos que así pudiera ser.
12. Las tecnologías más importantes a las que se hace referencia en esta Guía Avanzada de Nmap son las siguientes:
 - **Escáner de Puertos:** un escáner de puertos es una herramienta de seguridad destinada principalmente a la búsqueda de puertos³ abiertos en una máquina remota. Habitualmente es utilizado tanto por administradores, como ayuda en la tarea de análisis de la seguridad de sus redes, como por usuarios malintencionados, para intentar comprometer las redes y acceder a recursos y servicios no autorizados. *Nmap* es un ejemplo escáner de puertos avanzado y junto con esta funcionalidad permite otras muchas, como podrá verse a lo largo de este documento.
 - **Capturador de tráfico de red/Analizador de protocolos:** un capturador de tráfico de red es una herramienta de seguridad capaz de interceptar y registrar el tráfico que pasa a través de una red de datos. Es habitual que sean utilizadas en combinación con un analizador de protocolos, capaz de decodificar y analizar si el contenido de las capturas cumple con alguna especificación o estándar particular.

³ Un puerto constituye cada uno de los huecos en que se puede alojar un servicio que desee ofrecer sus servicios al exterior. En un mismo equipo existen 65535 (64Ki) puertos TCP y otros tantos UDP. Dependiendo de su estado, se puede conocer si existe o no un servicio alojado en él.

Las aplicaciones más comunes de estas herramientas son: análisis de problemas de red, detección de intentos de intrusión, obtención de información para realizar una intrusión, monitorización del uso de la red, extracción de estadísticas de red, depuración de aplicaciones de comunicaciones, e incluso la obtención de información sensible no protegida. *WinPcap* y *tcpdump* son ejemplos de capturadores de tráfico, mientras que *WireShark*, que utiliza las anteriores para capturar tráfico, es el analizador de protocolos más utilizado.

- **Cortafuegos:** un cortafuegos es un sistema informático, simple o compuesto, residente en una máquina, o como elemento de interconexión de redes, que actúa de punto de conexión segura entre otros sistemas informáticos. Un cortafuegos se sitúa, a modo de frontera, entre dos o más redes con la finalidad de hacer cumplir unas determinadas directivas de seguridad sobre la comunicación entre ellas, constituyéndose como el mecanismo básico para la prevención y detección de amenazas de seguridad. *iptables* es un ejemplo de cortafuegos.
- **IDS/IPS (Sistema de detección/prevenición de intrusiones):** un sistema de detección de intrusiones (IDS) es una herramienta que analiza el tráfico de red y, en base a unas reglas predefinidas, identifica comportamientos maliciosos e intentos de explotación no autorizada de recursos. Si además de identificar, el dispositivo tiene la capacidad de bloquear el tráfico que constituye el ataque, se trata entonces de un sistema de prevención de intrusiones (IPS). *Snort* es un ejemplo de este tipo de herramienta. Generalmente estos sistemas son independientes, pero actualmente están evolucionando e introduciéndose como parte de las propias aplicaciones a proteger, en lo que se viene a llamar cortafuegos de nivel de aplicación.

1.5. TERMINOLOGÍA

13. A lo largo de esta guía se utilizan repetidamente una serie de términos que hacen referencia a conceptos técnicos, naturales al software de administración de redes, y en concreto a Nmap. En virtud del rigor que se desea imprimir a la Guía, se definen a continuación los siguientes términos:

- **Comando:** instrucción tipada en Nmap que contiene los argumentos necesarios para acotar las características de cada uno de los escaneos ejecutados con esta herramienta. El formato de un comando en Nmap responde a la siguiente sintaxis.

```
nmap -<Técnicas> -<Opciones> --<Modificadores> <Objetivos>
```

Obsérvese la distinta utilización normalizada del guión “-” para cada uno de los argumentos del comando en Nmap.

Las versiones de Nmap con interfaz gráfica de usuario permiten la composición de estos comandos de forma amigable, facilitando la labor del usuario menos acostumbrado a trabajar en línea de comandos.

- **Técnica:** cada una de las funcionalidades, entendiendo funcionalidades como distintos tipos de escaneo, existentes en *Nmap*. Una técnica es susceptible de ser configurada según los criterios del usuario mediante un conjunto de *Opciones* y *Modificadores* disponibles en la propia herramienta. En *Nmap* se definen las siguientes *Técnicas*:
 - **Técnicas de Descubrimiento de Equipos:** determinación del estado de una o un conjunto de máquinas.
 - **Técnicas de Escaneo de Puertos:** determinación del estado de uno o un conjunto de puertos.

Cada una de estas técnicas puede distinguirse por su especificidad o sofisticación tecnológica, para lo que la calificación de básica o avanzada será aplicada a lo largo de este documento.

- **Opción:** una *Opción* es cualquier argumento pasado a *Nmap* que no sea reconocida como una *Técnica* o un *Objetivo*.
- **Modificador *Nmap*:** los *Modificadores* permiten alterar tanto el comportamiento como las salidas de las *Técnicas* sobre las que actúan, dotándolas de mayor potencia y flexibilidad. Al igual que sucede con las *Técnicas* y por los mismos motivos, los *Modificadores* pueden ser calificados como básicos o avanzados. En última instancia, un Modificador puede verse como una *Opción* adicional, con el objeto de aumentar la granularidad y el espectro de funcionalidades que *Nmap* ofrece. *Toda funcionalidad de Nmap es una Opción, o un conjunto de Opciones, cuando intervienen Modificadores.*
- **Objetivo:** es aquella máquina o conjunto de máquinas (subred) sobre la que *Nmap* llevará a cabo un análisis, utilizando para ello una o más *Técnicas*. Los *Objetivos* se especifican mediante una URL, dirección IP, bloque de direcciones en el formato CIDR⁴ o intervalo de direcciones separado por un guión⁵. Cualquier argumento pasado en línea de comandos que no sea reconocido como una *Técnica*, *Opción* o *Modificador*, será tratado como un *Objetivo*.
- **Sonda:** cada uno de los paquetes de prueba enviados por *Nmap* a los *Objetivos* como parte de un escaneo. En general son paquetes especialmente contruidos que para obtener su propósito, provocar o no una respuesta en el destino, explotan alguna particularidad de la implementación de la pila TCP/IP del *Objetivo*.

⁴ CIDR, del término inglés *Classless Inter-Domain Routing*, es una notación en la que se define un conjunto de equipos utilizando la IP más baja de las que forman el grupo, seguido del número de bits comunes que tienen el conjunto de direcciones del rango. Por ejemplo: 192.168.1.0/24 agrupa los equipos cuya dirección tiene los primeros 24 bits iguales, es decir, desde el equipo 192.168.1.0 hasta el equipo 192.168.1.255. Más información en http://es.wikipedia.org/wiki/Classless_Inter-Domain_Routing

⁵ Este formato para fijar objetivos permite, para cada número de los que conforman una dirección IP, introducir un intervalo separado por un guión. Por ejemplo, el intervalo 172.16.0.1-20 analizará las direcciones IP desde la 172.16.0.1 hasta la 172.16.0.20, mientras que el intervalo 172.16.0-2.20-25 analizará las IP 172.16.0.20 hasta la 25, así como las 172.16.1.20 hasta la 25 y las 172.16.2.20 hasta la 25.

- **Estado de una máquina:** Las *Técnicas* de descubrimiento de equipos están encaminadas a averiguar el estado de una o varias máquinas, es decir, a realizar un proceso de determinación de cuales de los equipos indicados como *Objetivos* están activos, que equivale a decir “en línea”, o remotamente alcanzables. La obtención de esta información es de gran ayuda para la realización de escaneos de puertos a gran escala, puesto que los equipos no activos no serán considerados a tal efecto, reduciendo considerablemente el tiempo del proceso.
- **Estado de un puerto:** Aunque *Nmap* distingue más *Estados*, en general desde el punto de vista de una máquina remota, los puertos de comunicaciones tienen dos *Estados*: alcanzable e inalcanzable. Un puerto es “alcanzable” si no existe ninguna causa externa (p.ej. filtros intermedios) que evite el contacto entre los extremos. De este modo el origen tendrá información de si dicho puerto está a la escucha o está cerrado. Será “inalcanzable” en cualquier otro caso. Los puertos UDP abiertos, al no negociar una conexión de manera implícita, pueden dar la apariencia de que son inalcanzables.
- **Estado *Nmap* de un puerto:** La salida de *Nmap* es, dependiendo de las *Opciones* elegidas, una lista de equipos escaneados a los que se acompaña de información adicional como el *Estado* de sus puertos. *Nmap* define seis *Estados* distintos para recoger los distintos grados de incertidumbre en la determinación de si un puerto está abierto o cerrado, es decir, a la escucha o no de nuevas conexiones o paquetes. A diferencia del punto anterior, estos *Estados* no son propiedades intrínsecas de los puertos, sino que definen cómo son vistos desde el exterior. La razón de que *Nmap* sea considerado un escáner de puertos avanzado es, entre otros motivos, debido a esta granularidad en el *Estado* de dichos puertos. Los *Estados* son los siguientes:
 - **Abierto:** existe una aplicación en la máquina objetivo que está a la escucha de nuevas conexiones o paquetes TCP o UDP. Muestra un servicio disponible para ser usado en la red.
 - **Cerrado:** es un puerto alcanzable, pero no existe una aplicación a la escucha en él.
 - **Filtrado:** *Nmap* no ha recibido respuestas a las sondas enviadas hacia un puerto. Suele significar que una herramienta intermedia (generalmente cortafuegos, sondas IDS/IPS, otros elementos de la red, o cualquier otro tipo de filtro) está bloqueando dicho puerto, respondiendo con poca o ninguna información (en ocasiones con un ICMP de tipo “destino inalcanzable”). Esta circunstancia ralentiza notablemente el escaneo para descartar que se trata de un problema de congestión de la red.
 - **No filtrado:** Solo aparece tras un análisis de tipo ACK (ver 3.2.5). Es un puerto alcanzable, pero no es posible determinar si está abierto o cerrado.

- **Abierto | filtrado:** En este caso, *Nmap* no ha sido capaz de determinar si el puerto está abierto o filtrado debido a falta de respuestas, bien porque ésta o la *Sonda* están siendo eliminadas por algún tipo de filtro de paquetes. Se puede obtener en las técnicas UDP (ver 3.2.3), TCP Null, Fin y Xmas (ver 3.2.6), TCP Maimon (ver 3.2.7) e IP Protocol Scan (ver 3.2.8).
- **Cerrado | filtrado:** Solo se obtiene tras un escaneo de tipo *Idle* (ver 3.2.4). En este caso *Nmap* no ha sido capaz de determinar si el puerto está cerrado o filtrado.
- **Señuelo:** es un equipo activo cuya dirección es utilizada como origen de las *Sondas* con el fin de enmascarar la propia IP en el escaneo, dificultando la traza del origen verdadero. El *Objetivo* verá que recibe paquetes de varios orígenes, por lo que probablemente responda o investigue a todos ellos, haciendo del uso de señuelos una técnica moralmente controvertida.

1.6. HISTORIAL DE VERSIONES DE NMAP

14. El código fuente de Nmap (en lenguaje C) fue publicado por primera vez en septiembre de 1997, en un artículo de la revista Phrack Magazine. Desarrollos posteriores (ya en lenguaje C++) incluyeron mejores algoritmos para el descubrimiento de los servicios disponibles, nuevos tipos de escaneo así como soporte a nuevos protocolos de comunicaciones (p.e. IPv6). En febrero de 2004 alcanzó la versión 3.5 y dos años más tarde, en enero de 2006, la versión 4.0, que implementa numerosas mejoras.
15. En diciembre de 2006 se incluyó NSE, el motor de Scripting de NMap, que ha permitido a los usuarios ampliar la funcionalidad de NMap hasta límites insospechados, ya que permite utilizar los resultados del mapeo de puertos para extraer información adicional. Este ha sido uno de los campos en que más ha evolucionado NMap en los últimos años, apoyado por su comunidad. Así, ha pasando de disponer de 22 scripts en su primera versión, a tener disponibles alrededor de 300 scripts⁶, estando incluidos en la distribución oficial actual 178 de ellos.
16. Durante los años 2006 y 2007 se abogó por la sustitución de la anterior interfaz gráfica que estaba incluida por defecto en la distribución de NMap, llamada NmapFE, por otra más funcional, escalable y que presentaba mejoras sustanciales, sobre todo en el campo de visualización de resultados. La interfaz seleccionada fue UMIT, basada en Python y GTK, e incluida por defecto desde la versión 4.22SOC1. Posteriormente, el equipo de desarrollo de NMap decidió seguir el desarrollo de esta interfaz en paralelo al de sus creadores, cambiando su nombre por Zenmap.

⁶ Un listado actualizado de los scripts disponibles, así como enlaces para su descarga, se puede encontrar en: <http://nmap.org/nsedoc/index.html>

17. En enero de 2009 se añadieron en la distribución oficial de NMap las herramientas Ncat y Ndiff, mientras que en agosto de 2009 se añadió la herramienta Nping, completando la estructura actual de la suite de herramientas incluidas junto con NMap.
18. En mayo de 2012 se publicó la versión 6, que incluía como principal novedad el soporte completo para IPv6, de forma que las pruebas realizadas sobre estas redes ya no dependen de la implementación que realice el sistema operativo, con las mejoras que ello supone. Entre las principales mejoras de esta versión, además del soporte completo a IPv6, se mejora notablemente el motor http, se añaden una gran cantidad de scripts NSE, se mejora la herramienta Nping y el interfaz gráfico Zenmap, y se mejora la velocidad global de la aplicación.
19. La última versión estable, utilizada en la presente Guía, es la v6.01, disponible desde junio de 2012.

2. INSTALACIÓN

20. Nmap es una herramienta multiplataforma disponible en una gran variedad de diferentes sistemas, en especial sistemas basados en UNIX. El manual de instalación de Nmap ofrece guías de instalación para los siguientes sistemas operativos:
 1. Distribuciones Linux, tanto basadas en paquetes (RPM o deb), como a partir de código fuente.
 2. Sistemas Windows actuales: XP SP1 y posteriores, Vista, 7 y Windows Server 2003 y 2008. Existe una guía de instalación para equipos con Windows 2000⁷.
 3. Sistemas Oracle/Sun Solaris
 4. Sistemas Apple Mac OS X
 5. Sistemas BSD: FreeBSD, OpenBSD y NetBSD.
 6. Sistemas Amiga.
 7. Sistemas IBM AIX.
21. Esta Guía se centra en los aspectos de instalación para distribuciones Linux basadas en paquetes RPM (tales como Red Hat, Fedora, Suse y Mandrake), paquetes deb (como Debian y Ubuntu) así como para instalaciones a partir de la compilación del código fuente. También se ofrecen detalles para la instalación en sistemas Windows.

⁷ Esta guía se puede encontrar en https://secwiki.org/w/Nmap/Old_Windows_Releases.

2.1. NMAP PARA LINUX

2.1.1. INSTALACIÓN A PARTIR DE CÓDIGO FUENTE

22. La forma tradicional y más potente de instalación de Nmap es a partir de la compilación del código fuente. Esto permite asegurar que el usuario obtendrá la última versión disponible, al mismo tiempo que habilita al instalador a adaptarse a la estructura concreta directorios y librerías del sistema donde se ejecutará la herramienta.
23. Un ejemplo que justifica la elección de esta forma de instalación sobre las alternativas basadas en paquetes binarios es cuando se desea que Nmap haga uso de las librerías criptográficas OpenSSL para la detección de versiones, ya que los paquetes binarios publicados en la web oficial no incluyen dicha funcionalidad.
24. Con el script de compilación e instalación actual no es necesario activar expresamente el uso de OpenSSL durante el proceso de compilación, ya que el propio script busca las librerías en las ubicaciones más comunes. Sólo será necesario definir el uso de OpenSSL si las librerías están en un directorio no estándar⁸.
25. La instalación a partir de las fuentes es un proceso que en general no presenta dificultades, siempre que se disponga de privilegios de administrador. A continuación se detallan los pasos a seguir:
 - 1) Descargar la última versión de *Nmap* in formato .tar.bz2 o .tgz desde la página oficial de descargas de NMap⁹.
 - 2) Descomprimir el archivo descargado:
 - a) tar.bz2: **tar xvjf nmap-VERSION.tar.bz2**
 - b) tgz: **tar xvzf nmap-VERSION.tgz**
 - 3) Cambiar al directorio recién creado: **cd nmap-VERSION**
 - 4) Ejecutar el script configurador: **./configure**
 - a) Para conocer las directivas de configuración: **./configure --help**
 - 5) Construir *Nmap*: **make**
 - 6) Escalar a súper-usuario: **su**
 - 7) Instalar *Nmap*, documentación y archivos asociados: **make install**

⁸ La directiva que controla la ubicación de las librerías de OpenSSL es `--with-openssl=<ruta>`. Se puede encontrar un listado de las directivas permitidas en: <http://nmap.org/book/inst-source.html#inst-configure>

⁹ <http://nmap.org/download.html>

26. Siguiendo estos pasos, y si no ha surgido ningún problema, Nmap quedará instalado en /usr/local/bin/nmap. Se puede ejecutar la orden sin parámetros para comprobar que está correctamente instalado.

2.1.2. INSTALACIÓN BAJO LINUX A PARTIR DE PAQUETES (RPM Y DEB)

27. La instalación de Nmap a partir de paquetes (binarios y de fuentes) es habitualmente la opción más rápida y sencilla, ya que existen paquetes de Nmap en la mayoría de distribuciones actuales. Sin embargo, existen dos inconvenientes.
28. Por un lado, los paquetes generados por las distribuciones, al margen de las versiones oficiales de Nmap, suelen ser más antiguos, y a veces se tarda bastante tiempo en poner a disposición del usuario paquetes actualizados, por lo que el nivel de obsolescencia de estos paquetes dependerá de la política de actualizaciones seguida por cada distribución.
29. Como contrapartida, los paquetes proporcionados por las distribuciones si que incluyen generalmente soporte para OpenSSL por defecto, mientras que los paquetes proporcionados por Nmap carecen de este soporte.
30. Los desarrolladores de Nmap han puesto a disposición de la comunidad paquetes RPM que pueden ser utilizados en cualquiera de las distribuciones que permiten el uso de este sistema de empaquetado. Estos paquetes se pueden descargar en el siguiente enlace:

<http://nmap.org/download.html#linux-rpm>

31. La instalación completa de Nmap se compone de dos paquetes binarios: el primero, que incluye el ejecutable y los archivos de datos necesarios y el segundo, que es opcional e incluye la interfaz gráfica, llamada Zenmap.
32. Además, también se puede descargar un paquete con las fuentes. En este caso se obtiene un paquete con las fuentes, especialmente preparadas para ser posteriormente compiladas en la distribución elegida.
33. Para instalar los paquetes binarios, se debe abrir una consola con privilegios de administrador en la ruta donde se han alojado los ficheros descargados, y ejecutar los siguientes comandos:

```
# rpm -vhU nmap-<VERSION>.<ARQUITECTURA>.rpm
```

```
# rpm -vhU zenmap-<VERSION>.noarch.rpm
```

34. Para utilizar los paquetes que proporcionan las distribuciones más comunes, es suficiente con instalarlos a través de las herramientas de gestión que proporcionan las propias distribuciones. Las ordenes para las distribuciones más comunes son:

```
(Open)Suse: yast2 -i nmap zenmap
```

```
Debian/Ubuntu: apt-get install nmap zenmap
```

```
Fedora/RedHat: yum install nmap zenmap
```

2.2. NMAP PARA WINDOWS

35. En versiones anteriores, la versión Windows de Nmap venía sin interfaz gráfica, y carecía de asistente de instalación y librerías de acceso a la tarjeta de red y captura de paquetes. Afortunadamente, la versión autoinstalable actual incluye, además del propio Nmap, el resto de aplicaciones de su suite (Ndiff, Ncat y Nping), la interfaz gráfica Zenmap, las librerías de captura de paquetes WinPcap, y un parche para aumentar el rendimiento de la aplicación, ya que Windows adolece de ciertas deficiencias en su API que lo hacen por defecto más lento.

36. Para descargar esta versión, se debe acceder al siguiente enlace y seleccionar el paquete que se desee descargar (versión estable o de test):

<http://nmap.org/download.html#windows>

37. Una vez descargado, es suficiente con ejecutar el instalador, que nos guiará a través de todos los pasos necesarios en la instalación. A continuación se muestra una captura de los componentes incluidos en el instalador para Windows:

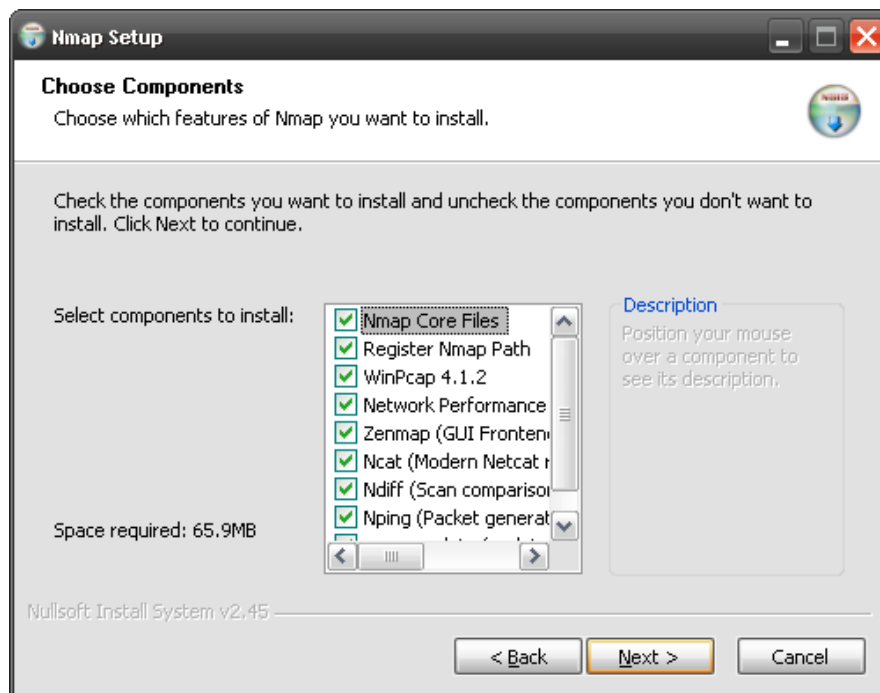


Figura 1. Instalables en la versión para Windows de Nmap

38. Además de este paquete autoinstalable, en la dirección anterior se puede descargar también una versión sin instalador que incluye únicamente los ficheros necesarios para ejecutar Nmap desde consola. En este caso, para que Nmap funcione correctamente será necesario, además de descomprimir el contenido del paquete en una carpeta de nuestra elección, realizar los siguientes pasos:

- Ejecutar el fichero *winpcap-<version>.exe* incluido, para instalar las librerías de acceso a red y captura de paquetes, en caso que no se disponga ya de las librerías instaladas.

- Ejecutar el fichero *nmap_performance.reg*, para evitar los problemas con la API de Windows mencionados anteriormente.
- Si utilizamos frecuentemente Nmap, añadirlo al path del sistema para poder ejecutarlo desde cualquier ubicación¹⁰.

2.3. INTERFAZ GRÁFICA DE USUARIO

39. Nmap es una herramienta de línea de comandos, aunque, a medida que ha ido popularizándose, han aparecido numerosas interfaces gráficas que hacían que la interacción con el usuario fuera más sencilla.
40. Anteriormente, el interfaz gráfico “oficial” de Nmap se llamaba NmapFE, pero en el año 2006 se inició un proceso para cambiar esa interfaz por otra desarrollada con tecnologías más actuales, libres, con nuevas opciones y multiplataforma.
41. La interfaz seleccionada fue UMIT, un proyecto basado en Python y GTK, que se incluyó por defecto en el fichero distribuible a partir de la versión 4.22SOC1. Posteriormente, el equipo de desarrollo de Nmap tomó el código de UMIT, renombrándolo a Zenmap, para seguir desarrollando por su parte la interfaz, al mismo tiempo que se desarrollaba Nmap.
42. El hecho de que esta interfaz esté programada en Python y GTK la hace fácilmente transportable entre distintas plataformas, por lo que se ha convertido en la interfaz más utilizada, cayendo muchos de los proyectos existentes anteriormente en desuso.
43. Para esta Guía se ha elegido Zenmap, por ser la interfaz oficial incluida por defecto en todas las versiones distribuibles de Nmap.

¹⁰ Más información sobre como realizar este cambio en: <http://nmap.org/book/inst-windows.html#inst-win-exec>

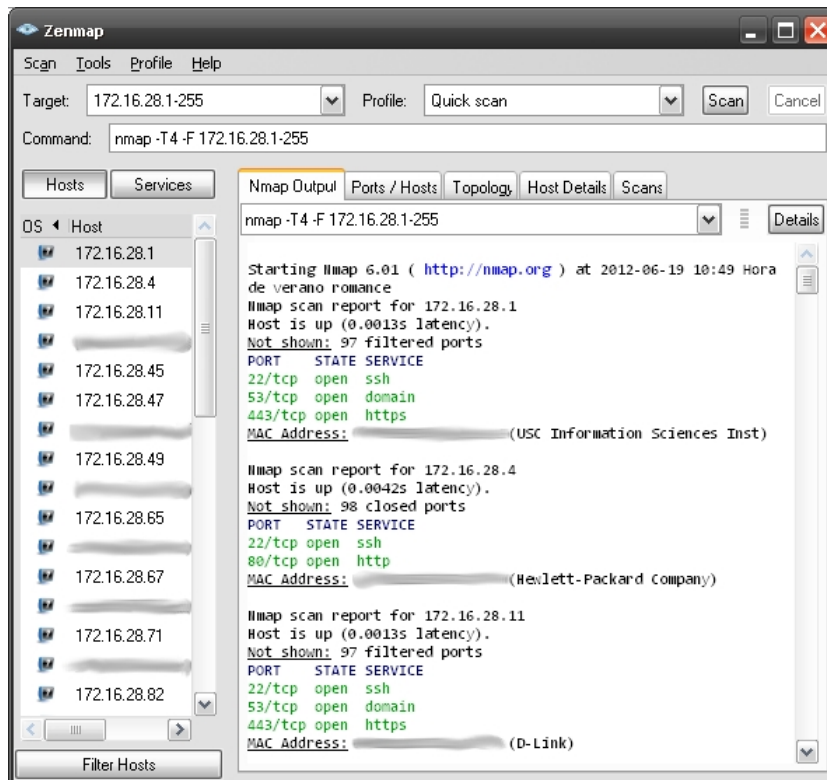


Figura 2. [Zenmap] Ventana principal con resultado de análisis

44. En la pantalla principal de la aplicación se puede ver, en la parte superior, un cuadro de texto llamado *Target* donde insertar los objetivos a analizar, seguida de un desplegable llamado *Profile* donde seleccionar el perfil que queremos aplicar al análisis. Justo debajo, el cuadro de texto *Command* nos permite personalizar los parámetros con que se ejecutará el análisis.
45. En la parte inferior derecha se encuentran un conjunto de pestañas, donde se muestra la información relevante. En la primera de ellas, llamada *Nmap Output*, se puede ver la salida de Nmap, a medida que este la va produciendo.
46. En la segunda pestaña, llamada *Ports/Hosts*, se puede ver un extracto de los puertos o equipos a los que afecta la selección hecha en el listado de la parte izquierda de la ventana, donde podemos ver un listado de equipos (Hosts) o servicios (Services), según el botón que pulsemos.

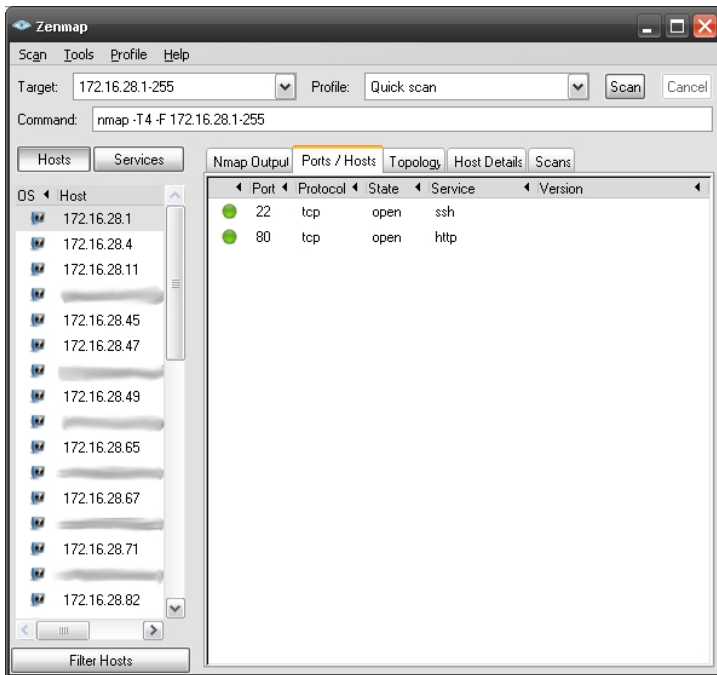


Figura 3. [Zenmap] Vista de equipos y puertos

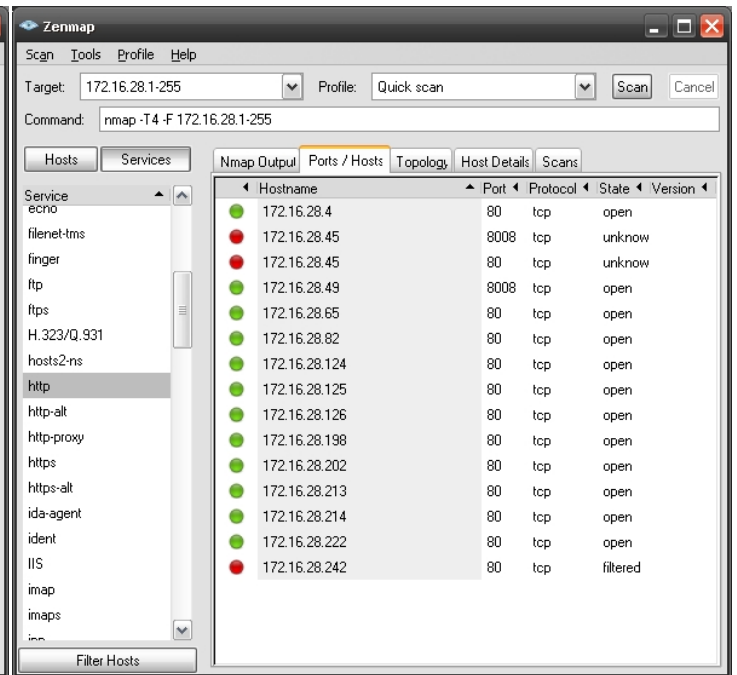


Figura 4. [Zenmap] Vista de servicios y equipos

47. En la siguiente pestaña, llamada *Topology*, podemos ver un gráfico con el esquema de la red analizada, según ha detectado Nmap.

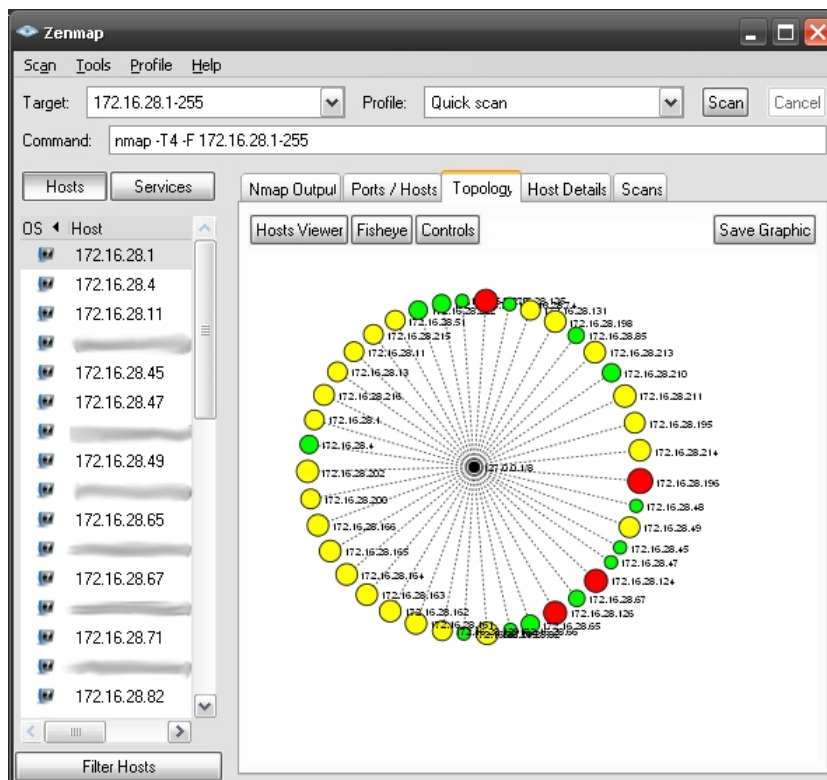


Figura 5. [Zenmap] Vista de topología de red

48. La penúltima pestaña permite visualizar detalles del equipo seleccionado en el listado de la parte izquierda, mientras que la última pestaña sirve de recopilación de los análisis lanzados en la sesión actual de la aplicación, y permite añadir análisis externos desde fichero o eliminar análisis que no sean necesarios.

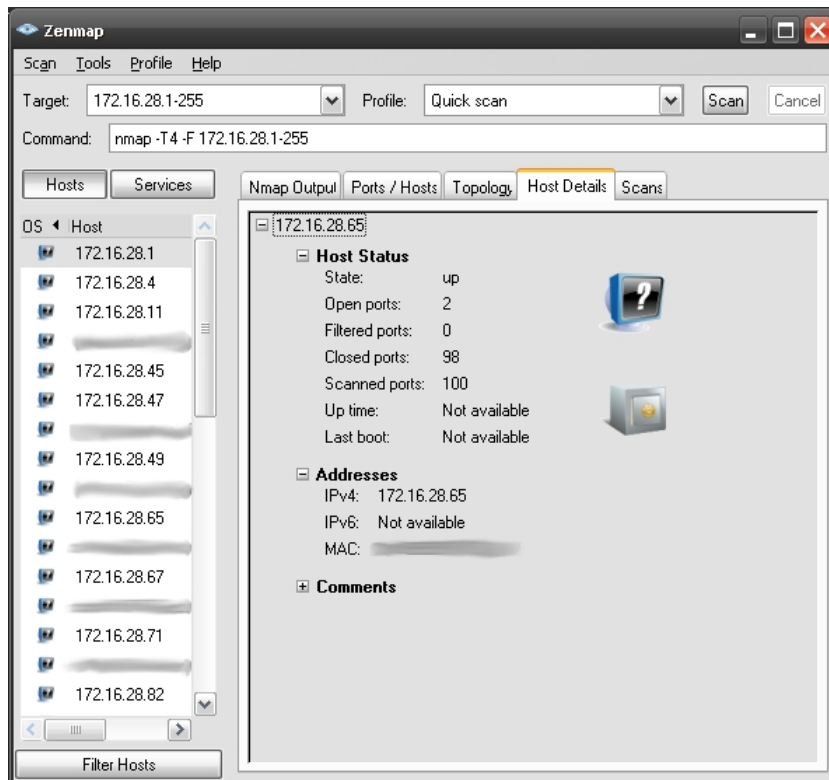


Figura 6. [Zenmap] Vista de detalles de equipo

49. Finalmente, cabe destacar la opción que permite añadir perfiles al desplegable de la pantalla principal. Esta opción se encuentra en el menú *Profile > New Profile or Command*, y nos permite, mediante una ventana con pestañas, seleccionar las opciones a aplicar al análisis.

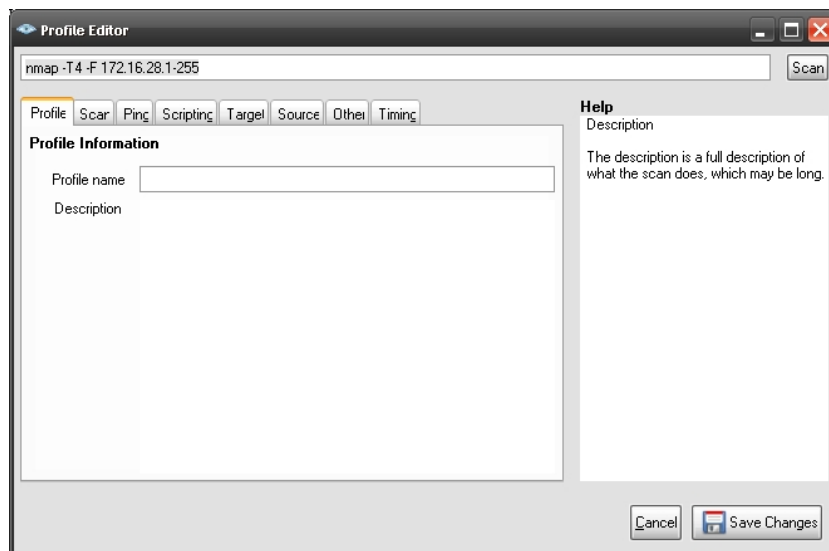


Figura 7. [Zenmap] Pantalla de creación de perfil de análisis

2.4. INTERACCIÓN CON APLICACIONES DE TERCEROS

50. A pesar de ser Zenmap la interfaz gráfica que ha adoptado el equipo de Nmap como propia, incluyéndola en el paquete oficial de la herramienta, todavía existen varias interfaces alternativas a la oficial, que pasamos a nombrar a continuación.

- **UMIT v1.0:** versión actual del desarrollo que sirvió como base para Zenmap, la interfaz gráfica distribuida actualmente con Nmap. Su aspecto es muy similar, pero posee algunas características diferenciadas, como un asistente para generar comandos de Nmap, la posibilidad de añadir extensiones, y la posibilidad de planificar la realización de escaneos. Está disponible para sistemas Windows, Linux/Unix y Mac.

Desde que se desligó de Nmap este proyecto ha crecido, creándose una versión web de este frontend (UMITWeb, o el nuevo UMITWeb-NG¹¹) también disponible para sistemas Windows y Linux, así como un generador de paquetes, e incluso una versión para analizar redes Bluetooth (UmitBT¹²).

Está disponible como paquete para la mayoría de distribuciones Linux estándar. Las versiones para otros sistemas operativos, así como paquetes fuente, se puede obtener desde su web oficial: <http://www.umatproject.org/>.

- **NmapSI4:** interfaz escrita en Qt, y disponible para Windows, Unix/Linux y Mac. Además de ofrecer toda la potencia de Nmap, permite realizar tareas adicionales como descubrir la ruta que siguen los paquetes (traceroute), encontrar vulnerabilidades existentes en puertos y versiones de servicios obtenidas por Nmap y tener un control pormenorizado de los script existentes y sus opciones de ejecución, entre otros. Como el anterior, existe paquete para la mayoría de distribuciones Linux actuales, y se puede obtener la versión actualizada desde su web oficial: <http://nmapsi4.org/>

51. Además de estas interfaces, que permiten al usuario lanzar de forma más sencilla sus escaneos, también existen otras aplicaciones que permiten al usuario manejar de forma fácil los ficheros que produce Nmap tras su ejecución. Algunas de ellas son:

- **Nmap Onepage:** Esta aplicación permite, a través de una cómoda interfaz web, interactuar con los ficheros XML que puede generar Nmap para los escaneos que realiza. Esta interfaz no necesita instalación, y ni siquiera necesita tener un servidor web dedicado para ella, ya que toda la funcionalidad está implementada con Javascript. Se puede utilizar en cualquier dispositivo que disponga de navegador web que soporte javascript, y se puede descargar desde el siguiente enlace: <https://bitbucket.org/holiman/nmap-onepage>
- **NetworkScanViewer:** Esta aplicación, nativa para Windows y que requiere .NET 4.0 para funcionar. Analiza un conjunto de ficheros de salida de Nessus y Nmap, y permite la adición de filtros y selecciones, para facilitar el tratamiento de la información recogida en los escaneos. Se puede encontrar más información en su Web: http://www.woanware.co.uk/?page_id=143

¹¹ <http://trac.umatproject.org/wiki/UmitWebNG>

¹² <http://trac.umatproject.org/wiki/UmitBluetooth>

3. LÍNEA DE COMANDOS

52. En esta sección se muestran los métodos y técnicas de análisis que tiene Nmap, su descripción y funcionamiento, y ejemplos de uso y los paquetes intercambiados. Tras esto, se muestran algunos detalles a tener en cuenta a la hora de planificar y realizar un análisis rápido y efectivo a una red, y el modo en que estos detalles se pueden definir en Nmap. En el Anexo B (ver 5.10) de esta guía se encuentra una tabla en que se muestran todas las opciones que acepta Nmap por línea de comandos.

53. La sintaxis de Nmap es la siguiente:

```
nmap [Tipo(s) de análisis] [Opciones] [Objetivos]
```

54. Los tipos de análisis y las opciones generalmente comienzan con un guión (“-“), a diferencia de los objetivos del análisis. Estos objetivos se pueden definir como direcciones IP, intervalos de direcciones, rangos CIDR o nombres de dominio (estos últimos también aceptan notación CIDR). De este modo, son válidos los objetivos 192.168.10.10, 172.16.128-130.0-255, 10.0.0.0/16, www.csirtcv.es y scanme.nmap.org/28.

3.1. TÉCNICAS DE DESCUBRIMIENTO DE EQUIPOS

55. En esta fase, la primera que realiza Nmap, se examina el conjunto de equipos que se ha pasado a Nmap para evaluar aquellos que están activos, y por tanto van a pasar a ser analizados. Existen varios métodos, descritos a continuación, que permiten realizar esta función.

3.1.1. NO PING (-Pn)

56. Esta opción evita completamente que Nmap realice la fase de Descubrimiento de Equipos. Es útil si se desea que todos los objetivos especificados sean considerados como activos, y de este modo se realice un escaneo de puertos en todos ellos, sin excepción.

57. El uso de esta opción puede tener una notable incidencia negativa en el rendimiento de un escaneo a gran escala, puesto que un escaneo de puertos contra una máquina inalcanzable consumirá mucho más tiempo, debido que vencerán todos los temporizadores de las respuestas esperadas a cada sonda enviada. Por otra parte, esta opción puede ser de utilidad si en la red analizada se bloquea, al menos en parte, el tráfico ICMP, que es el utilizado por defecto en la fase de descubrimiento de equipos.

58. En versiones anteriores de Nmap, esta opción se activaba con los modificadores -P0 o -PN.

3.1.2. LIST SCAN (-sL)

59. Esta opción únicamente lista los objetivos dados como argumentos, sin enviar paquete alguno a éstos. Por defecto, Nmap realiza una resolución DNS inversa de

los equipos a analizar, así que si se selecciona este método, los paquetes relacionados con esta resolución serán los únicos que se enviarán. Esta es una opción especialmente sigilosa (no intrusiva) con la cual obtener información potencialmente valiosa, y que puede servir también para comprobar que no se va a analizar ningún activo fuera de nuestro alcance.

- 60. La Figura 8 muestra las consulta DNS inversas (PTR) hechas para obtener los nombres de dominio de los objetivos indicados (www.ejemplo.es/29). En este caso todas obtienen respuesta. No se envía por tanto ningún paquete a los objetivos.
- 61. En esta primera parte de la figura se observa el resultado textual de la invocación del comando, mientras que en la siguiente se muestra el flujo gráfico de mensajes intercambiados.

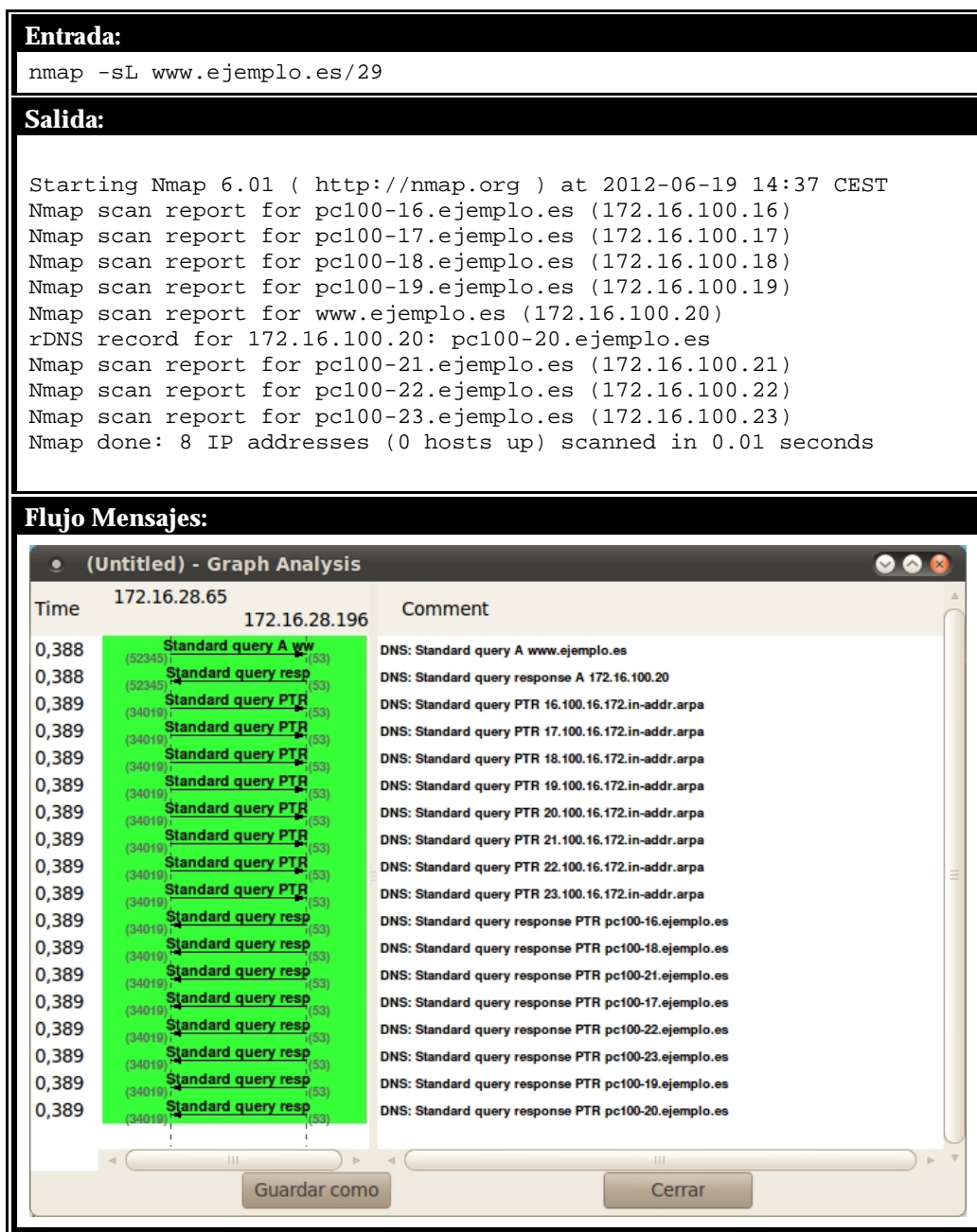


FIGURA 8. TÉCNICA BÁSICA DE DESCUBRIMIENTO LIST SCAN (-sL)

3.1.3. NO PORT SCAN (-sn)

62. También conocida como Ping Scan o Ping Sweep. Esta opción, cuando se indica de forma explícita, instruye a Nmap para que no realice un análisis de los puertos tras completar una fase estándar de descubrimiento de los equipos activos dentro del listado de equipos a analizar, a excepción de los análisis de scripts (--script) o de rutas (--traceroute), si se indican de forma explícita.
63. Si no se indica ninguna opción de descubrimiento de equipos, Nmap realiza por defecto las comprobaciones que se indican a continuación para tratar de descubrir los equipos que están activos y los que no, antes de iniciar la siguiente fase de análisis de puertos, para que esta sea más rápida.
64. Esta técnica es sensiblemente intrusiva ya que envía varias sondas a los objetivos, a diferencia del List Scan donde no se envía ninguna sonda a los objetivos, y suele utilizarse con el mismo fin. Permite realizar un reconocimiento ligero de la red objetivo de forma notablemente sigilosa. Esta técnica es igualmente útil para administradores que deseen de manera sencilla contabilizar el número de máquinas activas en su red, o monitorizar la disponibilidad de sus servidores.
65. Si se ejecuta sin privilegios de administración, Nmap utiliza las llamadas al sistema connect para conectarse a los puertos 80 y 443 de los objetivos. En cambio, si el usuario tiene permisos administrativos se envía por defecto un paquete TCP ACK al puerto 80 (-PA), un TCP SYN (-PS) al 443 además de un paquete ICMP Echo Request y un ICMP Timestamp Request, salvo cuando el usuario especifica otros parámetros, en cuyo caso este sirve únicamente para indicar que no se debe continuar con la fase de análisis de puertos.
66. Si el objetivo es local a la propia subred, se utilizará únicamente paquetes ARP (-PR).
67. En versiones anteriores de Nmap, esta técnica se activaba con el modificador -sP.
68. En la Figura 9 se muestra como Nmap descubre dos objetivos de forma totalmente distinta. El primero, al encontrarse en la misma subred que el origen es descubierto mediante una Ping ARP. Al segundo objetivo se le envían dos sondas, un paquete ICMP Echo Request y un Ping ACK, respondiendo solamente a la primera.

Entrada:
<code>nmap -sn 191.168.1.70, www.google.com</code>
Salida:
<pre>Starting Nmap 6.01 (http://nmap.org) at 2012-06-02 17:24 CEST Nmap scan report for 192.168.1.70 Host is up (0.00014s latency). MAC Address: AA:BB:CC:DD:EE:FF (Digital Equipment) Nmap scan report for www.google.com (173.194.78.105) Host is up (0.031s latency). Other addresses for www.google.com (not scanned): 173.194.78.106 173.194.78.147 173.194.78.99 173.194.78.103 173.194.78.104 rDNS record for 173.194.78.105: wg-in-f105.1e100.net Nmap done: 3 IP addresses (2 hosts up) scanned in 0.38 seconds</pre>
Flujo Mensajes:

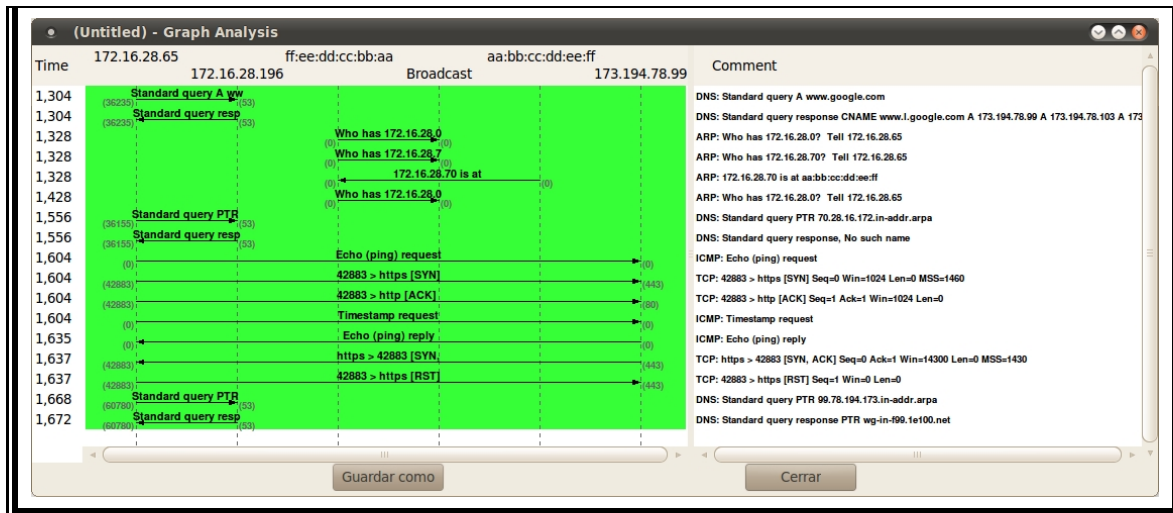


FIGURA 9. TÉCNICA BÁSICA DE DESCUBRIMIENTO PING SCAN (-SN)

Entrada:

```
nmap -sn 191.168.1.70, www.google.com
```

Salida:

```
Starting Nmap 6.01 ( http://nmap.org ) at 2012-06-02 17:24 CEST
Nmap scan report for 192.168.1.70
Host is up (0.00014s latency).
MAC Address: AA:BB:CC:DD:EE:FF (Digital Equipment)
Nmap scan report for www.google.com (173.194.78.105)
Host is up (0.031s latency).
Other addresses for www.google.com (not scanned): 173.194.78.106 173.194.78.147
173.194.78.99 173.194.78.103 173.194.78.104
rDNS record for 173.194.78.105: wg-in-f105.1e100.net
Nmap done: 3 IP addresses (2 hosts up) scanned in 0.38 seconds
```

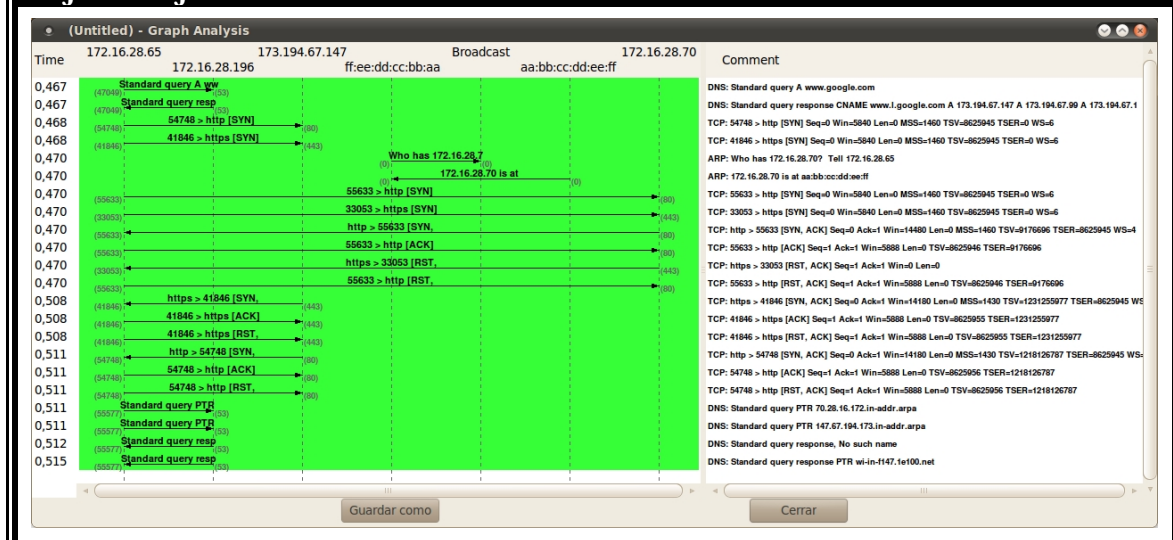
Flujo Mensajes:

FIGURA 10. TÉCNICA BÁSICA DE DESCUBRIMIENTO PING SCAN (-SN) EN EQUIPOS SIN PRIVILEGIOS

69. En las figuras anteriores se puede comprobar la diferencia entre ejecutar el mismo comando en un equipo sobre el que se tienen privilegios de administración y en otro sobre el que no se tienen permisos. En el segundo caso, se utilizan únicamente llamadas al sistema, por lo que siempre se realizan conexiones completas, a diferencia de en el equipo donde si se tienen privilegios.

3.1.4. PING ARP (-PR)

70. Uno de los escenarios en los que con más frecuencia se usa Nmap es para escanear redes locales Ethernet. Estas redes suelen tener muchos host inactivos, por lo que el proceso de escaneo basado en paquetes IP (ICMP echo request), que implica una resolución ARP anterior en cada caso, es notablemente lento, debido a los retardos introducidos por el sistema operativo en el envío de paquetes y al tamaño limitado de la cache ARP.
71. Esta técnica utiliza un algoritmo optimizado para realizar peticiones ARP, superando así las limitaciones de los sistemas operativos, que no están diseñados para hacer peticiones masivas. Debido a su especial fiabilidad y rapidez, no es necesario realizar pings IP si se recibe respuesta ARP para conocer si un equipo es alcanzable.
72. Esta técnica se utiliza por defecto cuando Nmap detecta que los equipos a analizar pertenecen a su misma red local, aunque no se especifique de forma explícita. Si se quiere evitar que se realice esta comprobación, se debe añadir el parámetro `-send-ip`, que evita que Nmap tome control del análisis ARP y únicamente envíe paquetes de tipo IP.
73. La Figura 11 muestra cómo el origen envía un ARP Request a la dirección de broadcast y el objetivo se descubre a sí mismo respondiendo con un ARP Response. Finalmente obtiene información adicional con una consulta al DNS inverso.

Entrada:
<pre>nmap -sn -PR -v 172.16.28.70</pre>
Salida:
<pre>Starting Nmap 6.01 (http://nmap.org) at 2012-06-02 17:50 CEST Nmap scan report for 172.16.28.70 Host is up (0.00017s latency). MAC Address: AA:BB:CC:DD:EE:FF (Digital Equipment) Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds</pre>
Flujo Mensajes:

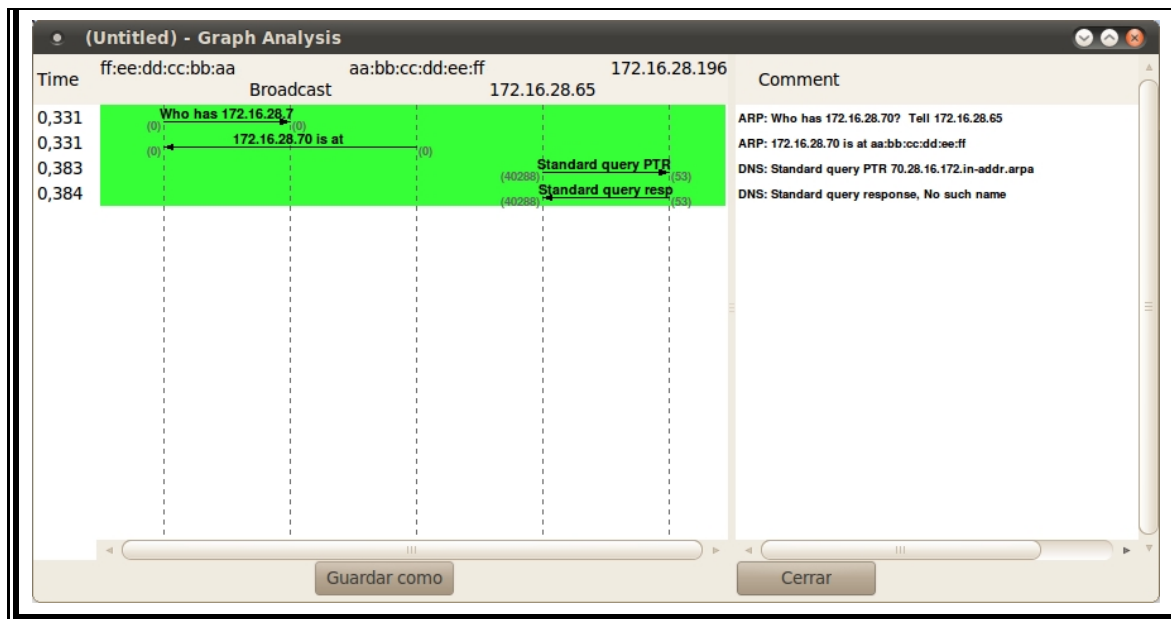


FIGURA 11. TÉCNICA BÁSICA DE DESCUBRIMIENTO PING ARP (-PR)

3.1.5. PING TCP SYN (-PS<listado de puertos>)

74. Esta técnica, si no se añade ningún puerto, envía al objetivo un paquete TCP vacío con el flag SYN activado al puerto 80. Se puede indicar un listado de puertos sobre los que realizar este análisis, separándolos por comas o introduciendo intervalos separados por un guión (p.ej. -PS22-25,80,443,8080).
75. El flag SYN indica al destino el deseo de establecer una conexión TCP por el origen. En este punto no interesa saber si el puerto analizado está abierto o cerrado, por lo que si Nmap recibe una respuesta de cualquier tipo del destino (ya sea un paquete RST indicando que el puerto está cerrado, o un paquete SYN/ACK indicando que se continua con el inicio de sesión TCP), NMap sabrá que el objetivo es alcanzable. Si por el contrario expira el temporizador, el destino se marcará como inalcanzable.
76. Si el usuario no tiene privilegios, se utiliza de forma automática una alternativa que consiste en enviar una solicitud de conexión TCP a través del sistema operativo (se utiliza la llamada del sistema connect). Del mismo modo que para un usuario privilegiado, si se recibe respuesta del objetivo, se sabe que este es alcanzable.
77. Este comportamiento, el uso de la llamada del sistema connect, también se utilizaba en versiones anteriores de Nmap para el análisis de redes IPv6, donde se utilizaban las funcionalidades proporcionadas por el sistema operativo. En la versión 6 se ha añadido la funcionalidad de generación de paquetes IPv6 en Nmap, por lo que se eliminan las restricciones de generación de paquetes impuestas por el sistema operativo para realizar análisis.
78. La eficacia de esta técnica es limitada, dado que muchos cortafuegos bloquean los paquetes SYN como medida preventiva para evitar el establecimiento de una conexión. En este caso la efectividad puede aumentar si se usa la técnica de Ping TCP ACK, descrita a continuación.

79. En la Figura 12 se puede observar cómo esta técnica envía un TCP SYN al puerto 80 del objetivo como si tratara de establecer una conexión. En este caso el objetivo responde afirmativamente al establecimiento, desvelando con ello su presencia.

Entrada:

```
nmap -sn -PS www.ejemplo.es
```

Salida:

```
Starting Nmap 6.01 ( http://nmap.org ) at 2012-06-02 17:56 CEST
Nmap scan report for www.ejemplo.es (172.16.100.20)
Host is up (0.0025s latency).
rDNS record for 172.16.100.20: pc100-20.ejemplo.es
Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds
```

Flujo Mensajes:

Time	172.16.28.65	172.16.100.20	Comment
0,138	Standard query A ww (52506) (53)		DNS: Standard query A www.ejemplo.es
0,139	Standard query resp (52506) (53)		DNS: Standard query response CNAME www.ejemplo.es A 172.16.100.20
0,168	61992 > http [SYN] (61992) (80)		TCP: 61992 > http [SYN] Seq=0 Win=1024 Len=0 MSS=1460
0,170	http > 61992 [SYN, ACK] (61992) (80)		TCP: http > 61992 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
0,170	61992 > http [RST] (61992) (80)		TCP: 61992 > http [RST] Seq=1 Win=0 Len=0
0,208	Standard query PTR (44967) (53)		DNS: Standard query PTR 20.100.16.172.in-addr.arpa
0,209	Standard query resp (44967) (53)		DNS: Standard query response PTR pc100-20.ejemplo.es

FIGURA 12. TÉCNICA AVANZADA DE DESCUBRIMIENTO PING TCP SYN (-PS)

Entrada:

```
nmap -sn -PS21,80,443 www.ejemplo.es
```

Salida:

```
Starting Nmap 6.01 ( http://nmap.org ) at 2012-06-02 18:00 CEST
Nmap scan report for www.ejemplo.es (172.16.100.20)
Host is up (0.0021s latency).
rDNS record for 172.16.100.20: pc100-20.ejemplo.es
Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
```

Flujo Mensajes:

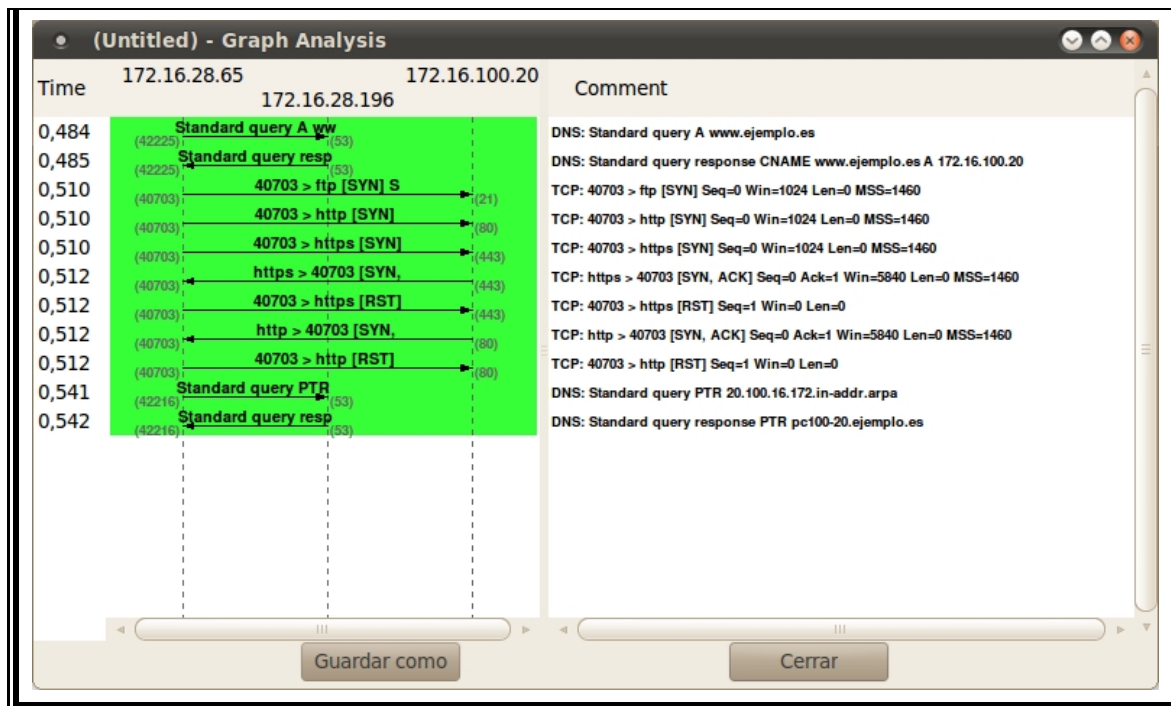


FIGURA 13. TÉCNICA AVANZADA DE DESCUBRIMIENTO PING TCP SYN (-PS) SOBRE VARIOS PUERTOS

3.1.6. PING TCP ACK (-PA<listado de puertos>)

80. El funcionamiento de esta técnica es idéntica a la de Ping TCP SYN, con la excepción de que ésta envía un paquete TCP al puerto 80 con el flag ACK activado. También se pueden añadir más puertos, indicando a continuación el listado de puertos sobre los que realizar este análisis, con el mismo formato que en casos anteriores.
81. Cuando un equipo recibe este tipo de paquetes sin existir previamente una conexión establecida, en principio deben responder con un paquete RST revelando de este modo su presencia. Esta técnica complementa la anterior al aumentar las posibilidades de traspasar filtros intermedios, como herramientas cortafuegos sin estado, debido a que muchos administradores configuran solamente reglas para interceptar paquetes entrantes SYN y no ACK. Los cortafuegos con estado suelen interceptar con éxito los paquetes inesperados, como el enviado con esta técnica, debido a que no se corresponde con ninguna sesión registrada previamente por un paquete SYN entrante. Una solución eficaz es combinar esta técnica con la anterior para que ambas sondas, SYN y ACK, sean enviadas.
82. En la Figura 14 se puede comprobar cómo esta técnica envía un TCP ACK al puerto 80 del objetivo como si tratara de asentar la recepción de un hipotético paquete enviado durante una conexión establecida. El objetivo niega que haya habido conexión alguna enviando un paquete RST, revelando con ello su presencia.

Entrada:

```
nmap -sn -PA scanme.nmap.org
```

Salida:

```
Starting Nmap 6.01 ( http://nmap.org ) at 2012-06-11 10:14 CEST
Nmap scan report for scanme.nmap.org (74.207.244.221)
Host is up (0.18s latency).
Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
```

Flujo Mensajes:

Time	172.16.28.65	74.207.244.221	172.16.28.195	Comment
0,420	(48055)	48055 > http [SYN]	(80)	TCP: 48055 > http [SYN] Seq=0 Win=5840 Len=0 MSS=1460 TSV=2029469 TSER=0 WS=
0,604	(48055)	http > 48055 [SYN]	(80)	TCP: http > 48055 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 TSV=5406525
0,604	(48055)	48055 > http [ACK]	(80)	TCP: 48055 > http [ACK] Seq=1 Ack=1 Win=5888 Len=0 TSV=2029515 TSER=540652555
0,604	(48055)	48055 > http [RST]	(80)	TCP: 48055 > http [RST, ACK] Seq=1 Ack=1 Win=5888 Len=0 TSV=2029515 TSER=5406
0,605	(57314)	Standard query PTR	(53)	DNS: Standard query PTR 221.244.207.74.in-addr.arpa
0,605	(57314)	Standard query resp	(53)	DNS: Standard query response PTR scanme.nmap.org

FIGURA 14. TÉCNICA AVANZADA DE DESCUBRIMIENTO PING TCP ACK (-PA)

3.1.7. PING UDP (-PU<lista de puertos>)

83. El enfoque de esta técnica es opuesto a las anteriores, por el hecho de que se envía paquetes a puertos que se considera estarán cerrados en el objetivo (por defecto se utiliza el puerto 31338). Esto es así porque, al ser el protocolo UDP sin conexión, un paquete enviado a un puerto abierto puede no recibir respuesta, aunque haya algún servicio escuchando en el puerto al que se ha enviado la sonda. Por el contrario, si se utiliza un puerto cerrado, el objetivo debería devolver un paquete ICMP del tipo Puerto Inalcanzable, dejando constancia de su existencia.

84. Tanto la falta de respuesta como la recepción de otro tipo de ICMPs será indicativo de destino inalcanzable. La principal ventaja de este tipo de escaneo es su capacidad de traspasar herramientas cortafuegos que sólo filtren paquetes TCP.

85. La Figura 15 muestra cómo esta técnica envía un paquete UDP al puerto 31338, que se considera con altas posibilidades de estar cerrado. En efecto lo está y responde con un paquete RST descubriendo con ello al objetivo.

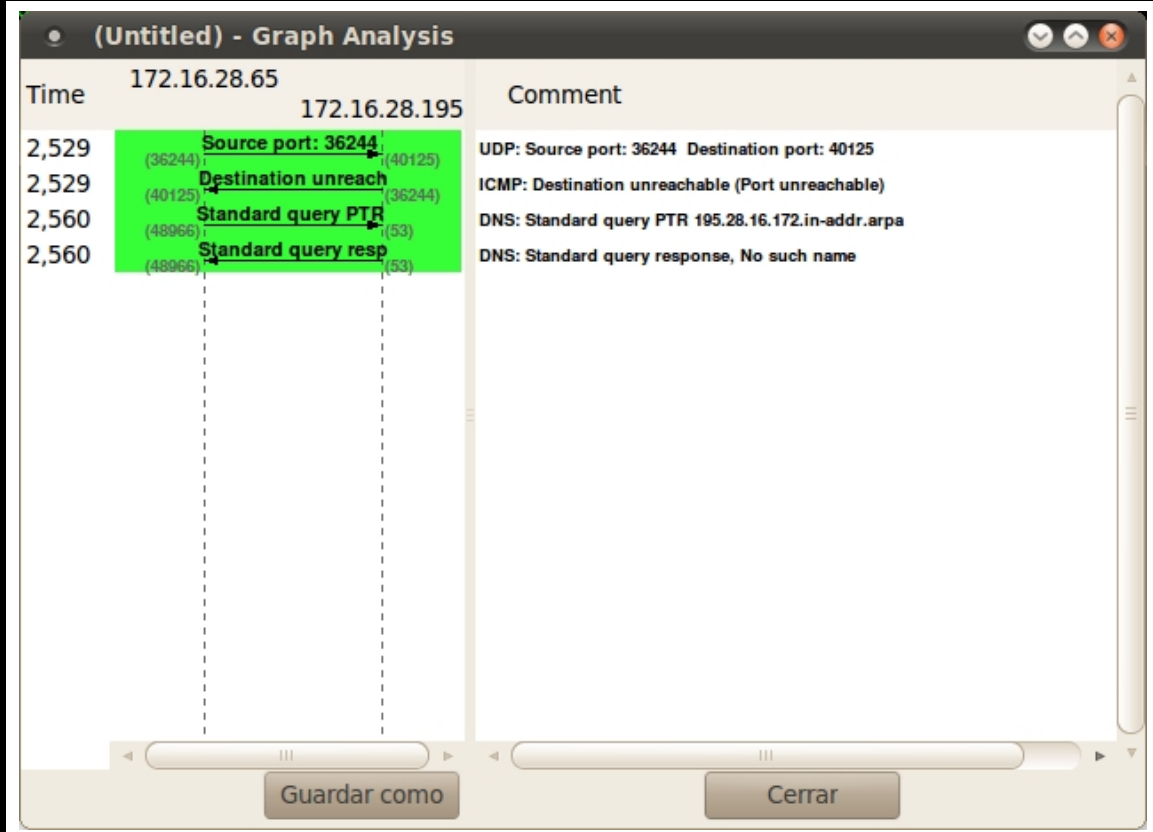
Entrada:																				
<code>nmap -PU -sn 172.16.28.195</code>																				
Salida:																				
<pre>Starting Nmap 6.01 (http://nmap.org) at 2012-06-11 10:26 CEST Nmap scan report for 172.16.28.195 Host is up (0.00061s latency). Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds</pre>																				
Flujo Mensajes:																				
 <table border="1"> <thead> <tr> <th>Time</th> <th>172.16.28.65</th> <th>172.16.28.195</th> <th>Comment</th> </tr> </thead> <tbody> <tr> <td>2,529</td> <td>(36244)</td> <td>(40125)</td> <td>UDP: Source port: 36244 Destination port: 40125</td> </tr> <tr> <td>2,529</td> <td>(40125)</td> <td>(36244)</td> <td>ICMP: Destination unreachable (Port unreachable)</td> </tr> <tr> <td>2,560</td> <td>(48966)</td> <td>(53)</td> <td>DNS: Standard query PTR 195.28.16.172.in-addr.arpa</td> </tr> <tr> <td>2,560</td> <td>(48966)</td> <td>(53)</td> <td>DNS: Standard query response, No such name</td> </tr> </tbody> </table>	Time	172.16.28.65	172.16.28.195	Comment	2,529	(36244)	(40125)	UDP: Source port: 36244 Destination port: 40125	2,529	(40125)	(36244)	ICMP: Destination unreachable (Port unreachable)	2,560	(48966)	(53)	DNS: Standard query PTR 195.28.16.172.in-addr.arpa	2,560	(48966)	(53)	DNS: Standard query response, No such name
Time	172.16.28.65	172.16.28.195	Comment																	
2,529	(36244)	(40125)	UDP: Source port: 36244 Destination port: 40125																	
2,529	(40125)	(36244)	ICMP: Destination unreachable (Port unreachable)																	
2,560	(48966)	(53)	DNS: Standard query PTR 195.28.16.172.in-addr.arpa																	
2,560	(48966)	(53)	DNS: Standard query response, No such name																	

FIGURA 15. TÉCNICA AVANZADA DE DESCUBRIMIENTO PING UDP (-PU)

3.1.8. PINGS ICMP (-PE, -PP, -PM)

86. La forma más extendida de realizar un descubrimiento de equipos es mediante la utilidad ping del sistema operativo, la cual envía paquetes ICMP Echo Request al destino y espera una respuesta ICMP Echo Reply. Nmap es capaz de imitar esta técnica mediante la opción -PE (Ping ICMP Echo). Dado que la mayoría de los filtros bloquean este tipo de paquetes, Nmap implementa dos técnicas más basadas en paquetes ICMP, que consisten en enviar paquetes de tipo ICMP Timestamp (-PP) y ICMP Addressmask (-PM), que deberían estar activos en todos los equipos que implementen el estándar RFC792 (la mayoría de los dispositivos existentes actualmente). Estas técnicas persiguen conseguir el descubrimiento de equipos en caso que únicamente se encuentren filtrados los paquetes ICMP Echo, y no se hayan tenido en cuenta otros tipos de paquetes ICMP en las reglas de bloqueo.

Entrada:

```
nmap -sn -PE -PP -PM www.google.es
```

Salida:

```
Starting Nmap 6.01 ( http://nmap.org ) at 2012-06-02 18:21 CEST
Nmap scan report for www.google.es (173.194.67.94)
Host is up (0.035s latency).
rDNS record for 173.194.67.94: wi-in-f94.1e100.net
Nmap done: 1 IP address (1 host up) scanned in 0.10 seconds
```

Flujo Mensajes:

Time	172.16.28.65	173.194.67.94	Comment
0,466	(52745)	(53)	DNS: Standard query A www.google.es
0,467	(52745)	(53)	DNS: Standard query response CNAME www-cctld.l.google.com A 173.194.67.94
0,499	(0)	(0)	ICMP: Echo (ping) request
0,499	(0)	(0)	ICMP: Timestamp request
0,499	(0)	(0)	ICMP: Address mask request
0,534	(0)	(0)	ICMP: Echo (ping) reply
0,564	(57760)	(53)	DNS: Standard query PTR 94.67.194.173.in-addr.arpa
0,564	(57760)	(53)	DNS: Standard query response PTR wi-in-f94.1e100.net

FIGURA 16. Técnica avanzada de descubrimiento Pings ICMP (-PE, -PP, -PM)

87. En la Figura 16 se ha mostrado la operatoria de esta técnica: se envían seguidos los tres tipos de paquetes ICMP request al objetivo, recibiendo casualmente respuestas a las tres peticiones, anunciando con ello que la máquina es alcanzable.

3.1.9. PING SCTP (-PY<listado de puertos>)

88. El protocolo SCTP¹³ pertenece a la capa de transporte, como TCP y UDP, inicialmente definido por el grupo SIGTRAN de IETF en el año 2000 para transportar señalización telefónica SS7 sobre IP, con la idea de dotar el protocolo IP de algunas de las características de confiabilidad de SS7, aunque su versatilidad le ha permitido expandirse en otras áreas.

89. Proporciona, como TCP, confiabilidad, control de flujo y secuenciación, aunque permite además el envío de mensajes fuera de orden, y es un protocolo orientado al mensaje. Otras características importantes son: capacidad de que los extremos de la conexión dispongan de más de una dirección IP (multihoming); capacidad para monitorizar y seleccionar caminos según las necesidades de la red; mecanismos de validación para evitar ataques y de notificación para evitar pérdidas o duplicados; así como multistream, o fragmentos independientes, que eliminan el Head-of-line

¹³ http://en.wikipedia.org/wiki/Stream_Control_Transmission_Protocol

blocking de TCP. Utiliza un handshake en cuatro fases (INIT, INIT-ACK, COOKIE-ECHO, COOKIE-ACK).

90. Este tipo de análisis envía sondas SCTP INIT al puerto 80 (se pueden definir otros puertos, pasándolos como parámetro), indicando que se quiere realizar una conexión SCTP con el objetivo. Si el equipo está levantado, responderá o bien con un paquete INIT-ACK (puerto abierto) o bien con un paquete ABORT (puerto cerrado). En cualquier otro caso se considerará el equipo como inactivo.

Entrada:

```
nmap -sn -PY -v 172.16.28.195
```

Salida:

```
Starting Nmap 6.01 ( http://nmap.org ) at 2012-07-12 09:35 CEST
Initiating Ping Scan at 09:35
Scanning 172.16.28.195 [1 port]
Completed Ping Scan at 09:35, 0.03s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 09:35
Completed Parallel DNS resolution of 1 host. at 09:35, 0.00s elapsed
Nmap scan report for 172.16.28.195
Host is up (0.00062s latency).
Read data files from: /usr/local/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds
Raw packets sent: 1 (52B) | Rcvd: 1 (80B)
```

Flujo Mensajes:

Time	172.16.28.65	172.16.28.195	Comment
2,231	(63547)	INIT (80)	SCTP: INIT
2,232	(80)	Destination unreachable (63547)	ICMP: Destination unreachable (Protocol unreachable)
2,263	(55037)	Standard query PTR (53)	DNS: Standard query PTR 195.28.16.172.in-addr.arpa
2,263	(55037)	Standard query resp (53)	DNS: Standard query response, No such name

Guardar como Cerrar

3.1.10. IP PROTOCOL PING (-PO<listado protocolos>)

91. Una de las técnicas más novedosas para el descubrimiento de equipos consiste en el envío de paquetes con un protocolo concreto especificado en sus cabeceras. Por defecto, se envían sondas con los protocolos 1 (ICMP), 2 (IGMP) y 4 (Encapsulado IP), aunque se puede, del mismo modo que en otros casos, introducir un listado de protocolos a utilizar.
92. Para los protocolos ICMP, IGMP, TCP (protocolo 6) y UDP (protocolo 17), se envían paquetes con las cabeceras propias del protocolo, mientras que para el resto de protocolos, se envía un paquete IP sin contenido tras la cabecera IP.
93. Este método espera respuestas utilizando el mismo protocolo, o paquetes ICMP del tipo Protocolo Inalcanzable, que indiquen que el equipo objetivo está vivo.

Entrada:

```
nmap -sn -PO www.google.es
```

Salida:

```
Starting Nmap 6.01 ( http://nmap.org ) at 2012-06-02 18:23 CEST
Nmap scan report for www.google.es (173.194.67.94)
Host is up (0.034s latency).
rDNS record for 173.194.67.94: wi-in-f94.1e100.net
Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds
```

Flujo Mensajes:

Time	172.16.28.65	173.194.67.94	Comment
0,480	Standard query A www		DNS: Standard query A www.google.es
0,481	Standard query resp		DNS: Standard query response CNAME www-cctld.l.google.com A 173.194.67.94
0,505	Echo (ping) request		ICMP: Echo (ping) request
0,505	V1 Membership Query		IGMP: V1 Membership Query
0,505	[Malformed Packet]		IP: [Malformed Packet]
0,539	Echo (ping) reply		ICMP: Echo (ping) reply
0,586	Standard query PTR		DNS: Standard query PTR 94.67.194.173.in-addr.arpa
0,586	Standard query resp		DNS: Standard query response PTR wi-in-f94.1e100.net

FIGURA 17.- Técnica avanzada de IP Protocol Ping (-PO)

94. En el ejemplo de la Figura 17 se puede ver, como se indicaba anteriormente, que algunos de los paquetes enviados no están correctamente formados, ya que Nmap únicamente rellena las cabeceras de los mismos, sin introducir datos válidos en ellos.

3.2. TÉCNICAS BASADAS EN EL ESCANEADO DE PUERTOS

3.2.1. TCP SYN Scan (-sS)

95. Esta técnica también es conocida como SYN Stealth o Half-Open Scan. Es la técnica más popular y la utilizada por defecto, permitiendo el escaneo de miles de puertos por segundo en redes rápidas sin presencia de herramientas cortafuegos. Esta opción es relativamente sigilosa y rápida, ya que no finaliza las conexiones que abre, así como fiable al no depender, como otras técnicas avanzadas, de las particularidades de las diferentes implementaciones de la pila TCP/IP presentes en plataformas específicas. Permite por tanto una diferenciación clara entre el estado abierto, cerrado o filtrado de un puerto.
96. Este análisis es el utilizado por defecto, por lo que si lanzamos Nmap sin parámetros, y tenemos los permisos de administración necesarios para poder ejecutarlo, este será el análisis realizado.
97. Una adición muy útil es agregar la revisión de versiones para cada puerto abierto encontrado, combinándola con la opción -sV, para tratar de identificar el tipo y la versión de los servicios descubiertos.
98. Tras realizar un análisis con esta técnica, los puertos pueden encontrarse en tres estados: abiertos (si se ha recibido un paquete SYN/ACK como respuesta a la sonda enviada), cerrados (si se ha recibido un paquete RST como respuesta) o filtrados (si no se ha recibido nada o se recibe algún paquete de tipo ICMP Inalcanzable).
99. En la Figura 18 se muestra cómo esta técnica no negocia una conexión de forma completa como lo hacía la anterior. Por ejemplo, para el puerto abierto de ftp del objetivo, no se finaliza completamente el proceso habitual tree-way handshake (sólo SYN y SYN-ACK).

```

Entrada:
nmap -Pn -sS -p 21,80,8080-8082 -v 172.16.28.214

Salida:
Starting Nmap 6.01 ( http://nmap.org ) at 2012-06-02 18:42 CEST
Initiating Parallel DNS resolution of 1 host. at 18:42
Completed Parallel DNS resolution of 1 host. at 18:42, 0.00s elapsed
Initiating SYN Stealth Scan at 18:42
Scanning 172.16.28.214 [5 ports]
Discovered open port 80/tcp on 172.16.28.214
Completed SYN Stealth Scan at 18:42, 1.23s elapsed (5 total ports)
Nmap scan report for 172.16.28.214
Host is up (0.00056s latency).
PORT      STATE      SERVICE
21/tcp    filtered  ftp
80/tcp    open       http
8080/tcp  closed    http-proxy
8081/tcp  closed    blackice-icecap
8082/tcp  closed    blackice-alerts

Read data files from: /usr/local/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.33 seconds
Raw packets sent: 6 (264B) | Rcvd: 4 (164B)

```

Flujo Mensajes:

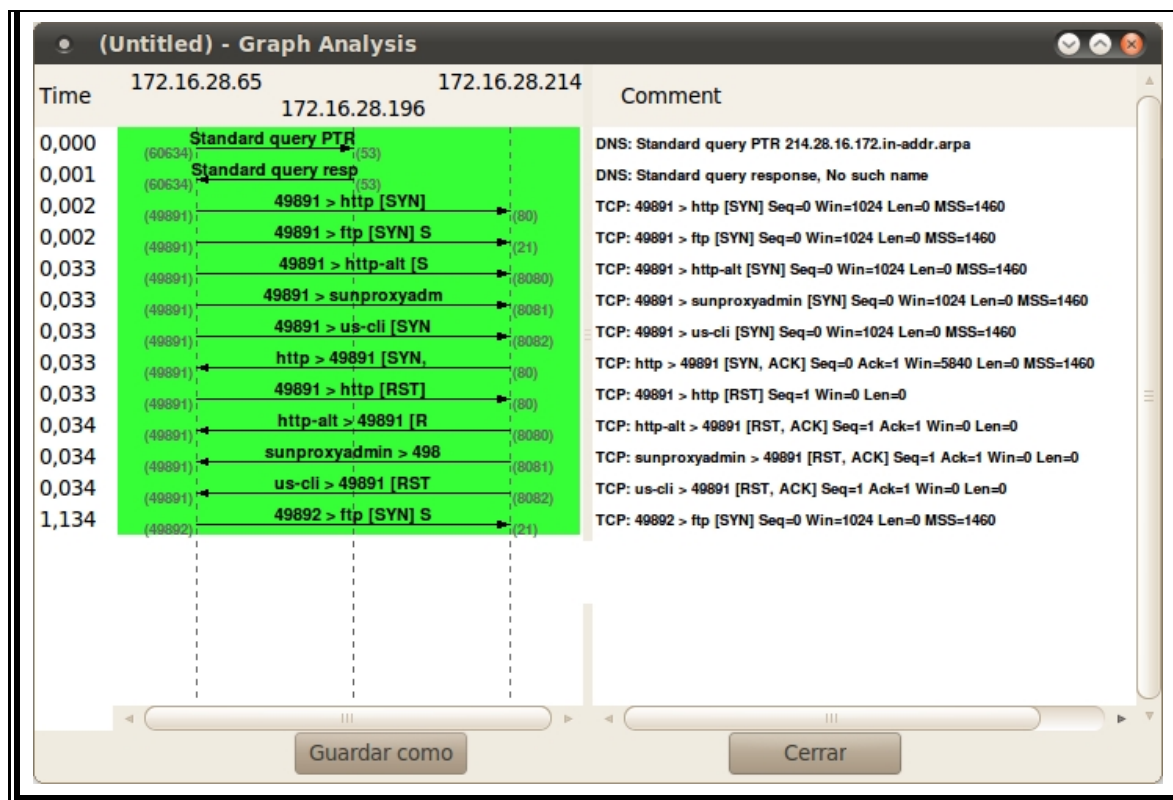


FIGURA 18. TÉCNICA BÁSICA DE ESCANEO TCP SYN SCAN (-sS)

3.2.2. TCP CONNECT SCAN (-sT)

100. Esta técnica se utiliza por defecto cuando no es posible la utilización de SYN Scan (-sS), cuando el usuario no tiene suficientes privilegios. Para su funcionamiento, usa las llamadas de alto nivel del sistema operativo para crear los paquetes (concretamente la llamada *connect()*) y para obtener la información de los intentos de conexión, al igual que cualquier otra aplicación.
101. Esta técnica es menos eficiente que SYN Scan porque Nmap no toma el control de los paquetes enviados, como hace en la mayoría de las otras técnicas, y en segundo lugar porque termina todas las conexiones, en lugar de hacer un *half-open reset*. Por este motivo, es menos sigilosa, siendo probable que un IDS/IPS registre los intentos de conexión.
102. Del mismo modo que en la técnica SYN Scan, según esta técnica los puertos pueden estar en tres estados: abierto, cerrado y filtrado.
103. La Figura 19 muestra cómo esta técnica negocia una conexión de forma completa con aquellos puertos que han respondido. Por ejemplo, para el puerto abierto de telnet del objetivo, se muestra el proceso de establecimiento de conexión *tree-way handshake* (SYN, SYN-ACK, ACK).

Entrada:

```
sudo nmap -Pn -sT -p 21,80,8080-8082 -v 172.16.28.214
```

Salida:

```

Starting Nmap 6.01 ( http://nmap.org ) at 2012-06-02 18:43 CEST
Initiating Parallel DNS resolution of 1 host. at 18:43
Completed Parallel DNS resolution of 1 host. at 18:43, 0.00s elapsed
Initiating Connect Scan at 18:43
Scanning 172.16.28.214 [5 ports]
Discovered open port 80/tcp on 172.16.28.214
Completed Connect Scan at 18:43, 1.20s elapsed (5 total ports)
Nmap scan report for 172.16.28.214
Host is up (0.00064s latency).
PORT      STATE      SERVICE
21/tcp    filtered  ftp
80/tcp    open      http
8080/tcp   closed    http-proxy
8081/tcp   closed    blackice-icecap
8082/tcp   closed    blackice-alerts

Read data files from: /usr/local/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.27 seconds

```

Flujo Mensajes:

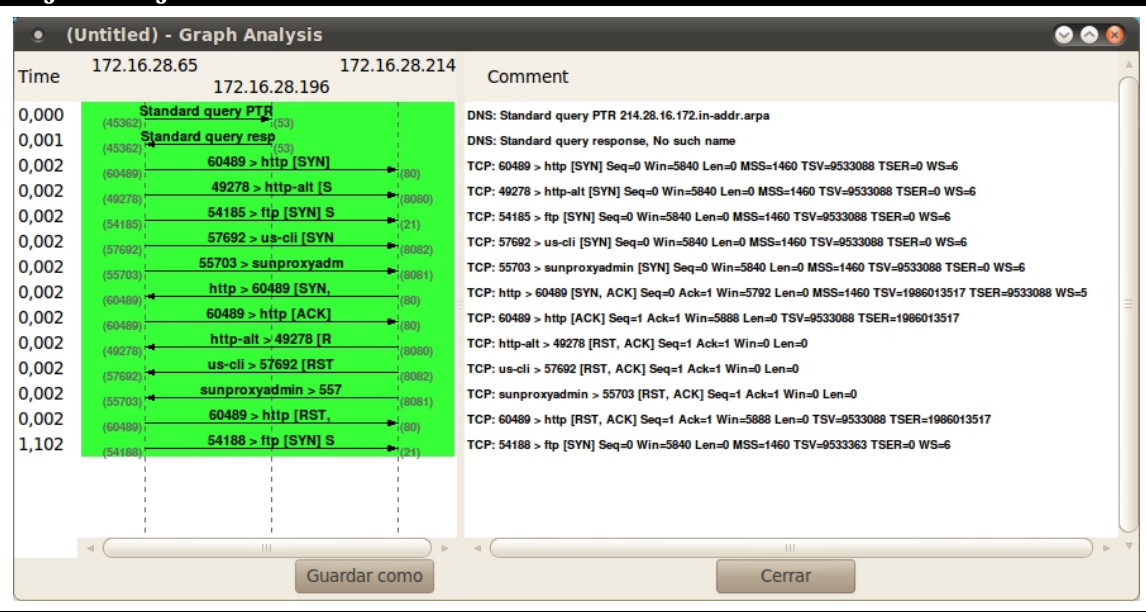


FIGURA 19. TÉCNICA BÁSICA DE ESCANEO TCP CONNECT SCAN (-sT)

3.2.3. UDP SCAN (-sU)

104. No todos los servicios populares corren sobre TCP. Los ejemplos más comunes de servicios UDP son los protocolos DNS (puerto 53), SNMP (puertos 161 y 162) y DHCP (puertos 67 y 68). En ocasiones estos puertos son ignorados en los análisis y auditorías debido a que el escaneo UDP es en general más complejo y lento. Esto es un error, debido a que los servicios UDP pueden ser explotados por atacantes del mismo modo que los servicios TCP. Esta técnica puede ayudar al inventario de estos puertos, pudiéndose combinar al mismo tiempo con otras técnicas de escaneo TCP, como SYN scan (-sS).

105. Con esta técnica es posible obtener puertos en cuatro estados distintos: abierto (significa que se ha obtenido una respuesta del puerto analizado), cerrado (si se obtiene una respuesta de tipo ICMP Puerto Inalcanzable), filtrado (si se obtiene cualquier otro tipo de paquete ICMP inalcanzable) y abierto/filtrado, que indica que no se ha recibido ningún tipo de respuesta desde el puerto analizado, a pesar de haber lanzado varias retransmisiones.

106. Como se ha indicado anteriormente, el análisis del protocolo UDP es más complejo. El primer problema que encontramos es que es normal que los puertos UDP no respondan a paquetes que les llegan, ya que el protocolo no está orientado a conexión, y los programadores suelen utilizar en este caso estructuras personalizadas para comunicarse con los puertos UDP, que Nmap desconoce en muchos casos, no pudiendo generar paquetes que permitan obtener una respuesta.
107. Otro problema consiste en que hay máquinas (entre ellas algunas máquinas Linux y Solaris) que tienen definida una tasa máxima de respuestas ICMP Puerto Inalcanzable a generar (e.g. no generar más de 1 paquete de este tipo por segundo). Para evitar que el objetivo deseche paquetes que no pueda procesar, Nmap es capaz de detectar la tasa máxima de envío de este tipo de paquetes y adaptar su velocidad para que los resultados del análisis sean exactos. Esto puede hacer que sea necesario invertir demasiado tiempo en esta técnica si se realiza un análisis de todos los puertos disponibles.
108. Otro problema es el estado abierto/filtrado, que no asegura que un puerto esté abierto o cerrado. La única forma de identificar con relativa seguridad si está abierto o no un puerto UDP es utilizando `-sUV` para determinar la versión de lo que en él escucha. Al identificar un puerto con el estado abierto/filtrado, la opción `-sV` envía diferentes tipos de solicitudes UDP conocidas en busca de obtener una respuesta que permita identificar lo que en él escucha. Si ninguna de las solicitudes enviadas genera una respuesta se mantendrá el estado abierto/filtrado del puerto, en caso contrario, se tendrá un buen indicativo del tipo de aplicación y su versión. Cabe destacar que la utilización de este modificador en los análisis aumenta considerablemente su tiempo de ejecución, al tener que enviar a cada puerto analizado sondas adicionales para cada uno de los tipos de servicio reconocidos por Nmap.
109. La Figura 20 muestra cómo al enviar sondas a cinco puertos UDP, sólo tres responden que no son alcanzables. Nmap considera que los dos que no han respondido o bien han sido filtrados o bien están abiertos.

Entrada:

```
nmap -Pn -sU -p 66-70 -v 172.16.28.214
```

Salida:

```
Starting Nmap 6.01 ( http://nmap.org ) at 2012-06-02 18:48 CEST
Initiating Parallel DNS resolution of 1 host. at 18:48
Completed Parallel DNS resolution of 1 host. at 18:48, 0.00s elapsed
Initiating UDP Scan at 18:48
Scanning 172.16.28.214 [5 ports]
Completed UDP Scan at 18:48, 1.23s elapsed (5 total ports)
Nmap scan report for 172.16.28.214
Host is up (0.00055s latency).
PORT      STATE      SERVICE
66/udp    closed    sqlnet
67/udp    closed    dhcpc
68/udp    open|filtered dhcpc
69/udp    open|filtered tftp
70/udp    open|filtered gopher

Read data files from: /usr/local/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.34 seconds
Raw packets sent: 8 (224B) | Rcvd: 2 (112B)
```

Flujo Mensajes:

Time	172.16.28.65	172.16.28.214	Comment
1,299	(34539) Standard query PTR	(53)	DNS: Standard query PTR 214.28.16.172.in-addr.arpa
1,300	(34539) Standard query resp	(53)	DNS: Standard query response, No such name
1,326	(34659) Source port: 34659	(69)	TFTP: Source port: 34659 Destination port: tftp[Malformed Packet]
1,326	(34659) Source port: 34659	(68)	TFTP: Source port: 34659 Destination port: bootpc[Malformed Packet]
1,326	(34659) Source port: 34659	(66)	TFTP: Source port: 34659 Destination port: sql*net[Malformed Packet]
1,326	(34659) Source port: 34659	(70)	TFTP: Source port: 34659 Destination port: gopher[Malformed Packet]
1,326	(34659) Source port: 34659	(67)	TFTP: Source port: 34659 Destination port: bootps[Malformed Packet]
1,326	(66) Destination unreachable	(34659)	ICMP: Destination unreachable (Port unreachable)
1,326	(67) Destination unreachable	(34659)	ICMP: Destination unreachable (Port unreachable)
2,427	(34660) Source port: 34660	(70)	UDP: Source port: 34660 Destination port: gopher
2,427	(34660) [Malformed Packet]	(68)	BOOTP: [Malformed Packet]
2,427	(34660) Source port: 34660	(69)	TFTP: Source port: 34660 Destination port: tftp[Malformed Packet]

FIGURA 20. TÉCNICA BÁSICA DE ESCANEO UDP SCAN (-sU)

3.2.4. IDLE SCAN (-sI)

110. Esta técnica avanzada explota una “vulnerabilidad” de muchas implementaciones de la pila TCP/IP, consistente en la posibilidad de predecir (simple incremento en peticiones consecutivas) el identificador de fragmento de los paquetes IP (IP ID). Con ello se permite extraer información mediante el análisis de las secuencias predecibles del identificador (IPID) de los paquetes IP. Ni su fundamento ni su funcionamiento son triviales, por lo que una descripción completa escapa al propósito de esta sección.

111. Básicamente esta técnica permite escanear un objetivo sin enviarle un solo paquete utilizando la propia dirección IP origen, por lo que se considera la técnica más avanzada y sigilosa de todas las presentes en Nmap. Para ello es necesario utilizar un tercer equipo, denominado zombie, cuya IP tomaremos para que aparezca como fuente de las sondas desde el punto de vista de la máquina objetivo.
112. En este caso, se pueden obtener dos estados de puertos distintos: abierto (si se detecta a través del zombie que ha existido respuesta), o cerrado/filtrado (si no se detecta respuesta a través del zombie, bien porque el objetivo no responde al zombie, o bien porque el objetivo responde con un paquete RST, que es ignorado por el zombie).
113. El uso de esta técnica es altamente controvertido, desde el momento en que ésta usa sin autorización los recursos de una máquina (zombie), que además se registraría falsamente por herramientas IDS/IPS como el origen de un proceso de escaneo de puertos. Para su correcto funcionamiento es necesario que la máquina zombie sea alcanzable y su implementación TCP/IP genere una secuencia de identificadores IP predecible. Debido a que esta técnica escanea un objetivo desde el punto de vista de una tercera máquina, es posible determinar de este modo las relaciones de confianza entre distintas máquinas. Para ello es posible indicar una lista de equipos zombie que pudieran ser de confianza del objetivo.
114. Para tratar de encontrar un zombie que cumpla con los requisitos, se puede escanear una subred utilizando los parámetros siguientes:

```
nmap -P0 -sN -n -v -p 80 --scanflags SYN,ACK <subred_objetivo>
```

115. Esta comando envía los mismos paquetes (TCP con SYN+ACK) que usa Idle Scan al inicio del proceso. Cualquier objetivo válido debería responder con un paquete RST y por tanto deberá aparecer el puerto 80 como no filtrado.
116. El flujo de mensajes de la Figura 21 se presenta la traza de mensajes observable desde el origen (172.16.28.65). Inicialmente el origen recoge información del IPID de la máquina zombie (172.16.28.124). Inmediatamente después inicia la fase de descubrimiento del objetivo, enviando paquetes al zombie con la dirección origen falsificada (la del objetivo: 172.16.28.51). Seguidamente con una sonda al zombie determina si el objetivo es o no alcanzable (analiza el IPID). Ahora comprueba nuevamente el IPID del zombie e inicia el escaneo del puerto TCP445 del objetivo enviando un paquete SYN con la dirección origen falsificada. Finalmente busca si se ha incrementado el IPID para determinar el estado del puerto 445, en este caso abierto.

Entrada:

```
nmap -Pn -p 445 -v -sI 172.16.28.124 172.16.28.51
```

Salida:

```

Starting Nmap 6.01 ( http://nmap.org ) at 2012-07-11 10:45 CEST
Initiating ARP Ping Scan at 10:45
Scanning 172.16.28.51 [1 port]
Completed ARP Ping Scan at 10:45, 0.03s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 10:45
Completed Parallel DNS resolution of 1 host. at 10:45, 0.00s elapsed
Initiating idle scan against 172.16.28.51 at 10:45
Idle scan using zombie 172.16.28.124 (172.16.28.124:80); Class: Incremental
Discovered open port 445/tcp on 172.16.28.51
Completed idle scan against 172.16.28.51 at 10:45, 0.83s elapsed (1 ports)
Nmap scan report for 172.16.28.51
Host is up (0.0067s latency).
PORT      STATE SERVICE
445/tcp   open  microsoft-ds
MAC Address: AA:BB:CC:DD:EE:FF (Cadmus Computer Systems)

Read data files from: /usr/local/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.97 seconds
Raw packets sent: 18 (776B) | Rcvd: 12 (468B)

```

Flujo Mensajes:

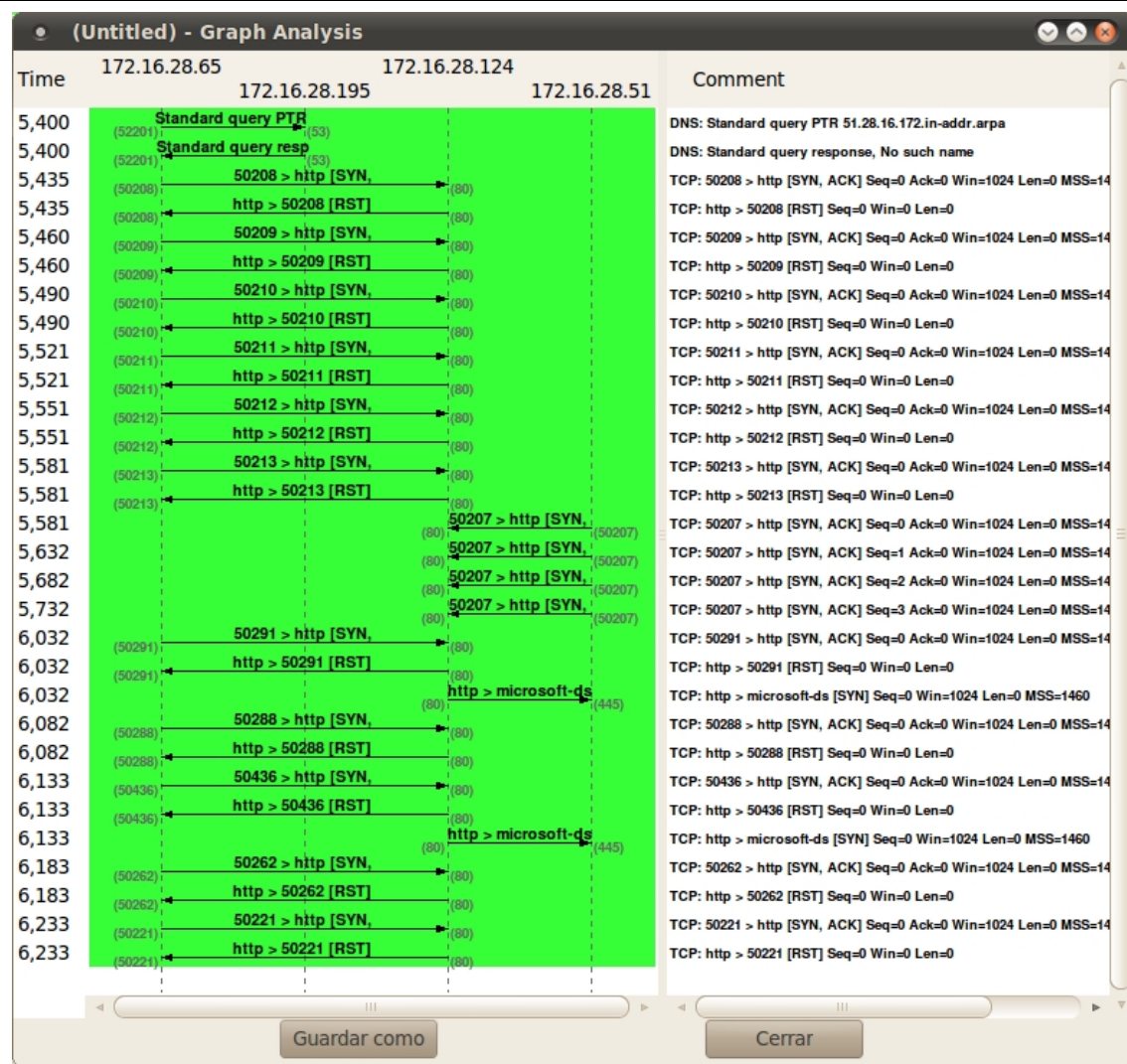


FIGURA 21. Técnica avanzada de escaneo Idle Scan (-sI: origen y objetivo)

117. El siguiente flujo de mensajes (Figura 22) muestra la traza de mensajes observable desde el objetivo (172.16.28.51). Como se puede ver, todos los mensajes que recibe el objetivo aparentemente vienen de la máquina zombie (172.16.28.124). Para el objetivo es como si el zombie tratara de establecer una conexión por dos

veces al puerto TCP445. A la vista de esta traza es imposible averiguar que es otra máquina la que está causando este flujo de mensajes.

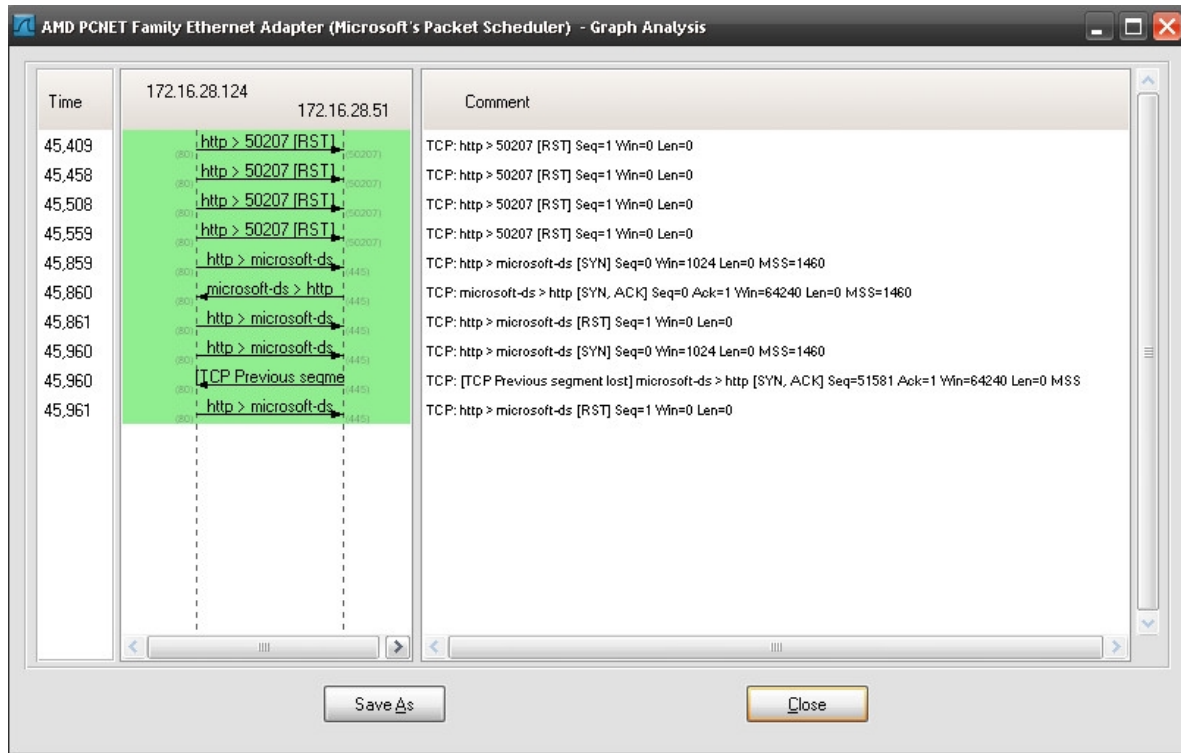


FIGURA 22. Técnica avanzada de escaneo Idle Scan (-sI): zombie y objetivo

3.2.5. TCP ACK SCAN (-sA)

118. También conocida como ACK Stealth, esta técnica se diferencia del resto porque sólo determina si los puertos del objetivo se encuentran filtrados o no, en lugar de definir si están abiertos o cerrados. Por ello se utiliza para averiguar el conjunto de reglas que aplica una herramienta cortafuegos sobre el objetivo, determinando igualmente cuándo estas reglas son con estado y a qué puertos afectan.

119. Esto se realiza enviando una sonda TCP con el flag ACK activo, por lo que un puerto no filtrado debería responder con un paquete RST. Según las respuestas recibidas, un puerto puede estar en estado no filtrado (si se recibe una respuesta RST), o filtrado (si se recibe un error ICMP inalcanzable o no se recibe ninguna respuesta, incluso tras varias retransmisiones).

120. La Figura 23 muestra cómo el origen envía ACKs a seis puertos del objetivo, recibiendo sólo dos paquetes RST (del puerto 25 y 80). De este modo Nmap comprueba que al menos estos dos puertos no están siendo filtrados.

Entrada:

```
nmap -Pn -sA -p 443-445,3389 -v 172.16.28.51
```

Salida:


```

Starting Nmap 6.01 ( http://nmap.org ) at 2012-07-12 10:05 CEST
Initiating ARP Ping Scan at 10:05
Scanning 172.16.28.51 [1 port]
Completed ARP Ping Scan at 10:05, 0.03s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 10:05
Completed Parallel DNS resolution of 1 host. at 10:05, 0.00s elapsed
Initiating ACK Scan at 10:05
Scanning 172.16.28.51 [4 ports]
Completed ACK Scan at 10:05, 1.23s elapsed (4 total ports)
Nmap scan report for 172.16.28.51
Host is up (0.00049s latency).
PORT      STATE      SERVICE
443/tcp   filtered   https
444/tcp   filtered   snpp
445/tcp   unfiltered microsoft-ds
3389/tcp   filtered   ms-wbt-server
MAC Address: AA:BB:CC:DD:EE:FF (Cadmus Computer Systems)

Read data files from: /usr/local/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.41 seconds
Raw packets sent: 8 (308B) | Rcvd: 4 (236B)

```

Flujo Mensajes:

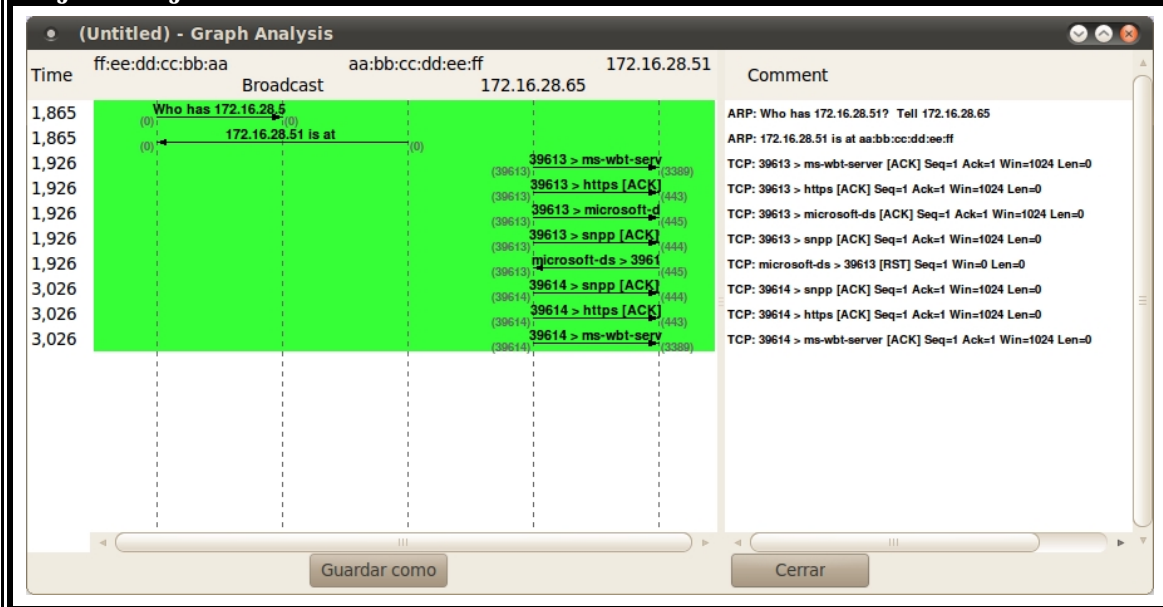


FIGURA 23. Técnica avanzada de escaneo TCP ACK scan (-sA)

3.2.6. TCP Null, FIN, Xmas scans (-sN, -sF, -sX)

121. De manera similar a la anterior técnica, éstas se basan en enviar sondas TCP con distintos flags activados, pero se aprovecha de una indefinición¹⁴ en el estándar RFC 793 para provocar una respuesta en el objetivo que determine si un puerto está abierto o cerrado. El fundamento de estas técnicas reside en que los puertos cerrados de equipos compatibles con esta RFC responderán con un RST a cualquier paquete que no contenga un flag SYN, RST o ACK, mientras que no emitirán respuesta alguna si el puerto está abierto.

¹⁴ RFC 793, página 65: “Si el estado del puerto destino es CERRADO [...] un segmento entrante que no contenga un RST causará el envío de un RST como respuesta.” La siguiente página trata el caso del envío de paquetes a puertos ABIERTOS sin alguno de los bits SYN, RST o ACK activados, estableciendo que: “[...] no es probable que se reciba uno, pero si es el caso, se debe desechar el segmento y volver.”

122. La técnica Null Scan (-sN) envía una sonda sin ningún flag activado, la técnica FIN Scan (-sF) activa únicamente el flag FIN y, finalmente, la Xmas Scan (-sX) activa los flags FIN, PSH y URG. Estas técnicas son eficaces traspasando algunas herramientas cortafuegos sin estado y encaminadores con filtro de paquetes. Estas cuatro técnicas utilizan conjuntos de flags determinados, aunque cualquier combinación de flags TCP es posible si se utiliza el modificador --scanflags.
123. Según las respuestas obtenidas, Nmap clasifica los puertos en: abiertos/filtrados (si no se recibe ninguna respuesta), cerrados (si se recibe un paquete RST) o filtrados (si se recibe algún tipo de error ICMP inalcanzable).
124. Además, son ligeramente más sigilosas que la técnica básica SYN Scan, aunque actualmente la mayor parte de las herramientas IDS/IPS sea posible configurarlas para detectar estos tipos de escaneo. Sin embargo, los resultados no serán fiables en sistemas Windows, Cisco, BSDI y OS/400, debido a que no siguen al pie de la letra el RFC 793 y envían paquetes RST esté el puerto abierto o no. Los resultados en sistemas Unix si que suelen ser fiables y correctos.
125. Los resultados obtenidos para las cuatro técnicas son prácticamente idénticos. A modo de ejemplo sólo se representa una de ellas, -sX.
126. En la Figura 24 se puede observar cómo el origen envía 5 paquetes TCP con los flags FIN, PSH y URG activados a los 5 puertos indicados del objetivo (del 21 al 25). De todos menos uno recibe un RST, por lo que Nmap sabe que están cerrados. En cambio del puerto 22 no recibe respuesta alguna, por lo que o está filtrado o abierto.

```

Entrada:
nmap -sX -p 21-25 -v 172.16.28.82

Salida:
Starting Nmap 6.01 ( http://nmap.org ) at 2012-07-12 10:27 CEST
Initiating ARP Ping Scan at 10:27
Scanning 172.16.28.82 [1 port]
Completed ARP Ping Scan at 10:27, 0.03s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 10:27
Completed Parallel DNS resolution of 1 host. at 10:27, 0.00s elapsed
Initiating XMAS Scan at 10:27
Scanning 172.16.28.82 [5 ports]
Completed XMAS Scan at 10:27, 1.23s elapsed (5 total ports)
Nmap scan report for 172.16.28.82
Host is up (0.00018s latency).
PORT      STATE      SERVICE
21/tcp    closed    ftp
22/tcp    open|filtered ssh
23/tcp    closed    telnet
24/tcp    closed    priv-mail
25/tcp    closed    smtp
MAC Address: AA:BB:CC:DD:EE:FF (Hewlett-Packard Company)

Read data files from: /usr/local/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.40 seconds
Raw packets sent: 7 (268B) | Rcvd: 5 (188B)

```

Flujo Mensajes:

132. Los resultados obtenidos con esta técnica no son siempre fiables, ya que dependen de detalles muy particulares no implementados en la totalidad de los sistemas existentes. Así, si todos los puertos aparecen como cerrados, muy probablemente el sistema no tendrá este comportamiento. Si obtenemos la mayoría de puertos cerrados, y unos pocos abiertos, es probable que el objetivo tenga este comportamiento y los resultados sean fiables. En algunos casos también se ha comprobado el efecto contrario, detectando Nmap la mayoría de puertos abiertos y unos pocos cerrados. En este caso, esos puertos que se detectan como cerrados son los que realmente están abiertos.
133. Dada la relativa fiabilidad de esta técnica se ha decidido no incluir ningún ejemplo en esta guía.

3.2.9. Sctp INIT Scan (-sY)

134. El protocolo Sctp es una alternativa relativamente nueva a TCP y UDP, que se ha descrito brevemente en el apartado 3.1.9.
135. La técnica Sctp INIT Scan es el equivalente para el protocolo Sctp de la técnica TCP SYN Scan, ya que lanza sondas de inicio de conexión a los objetivos. Es poco intrusiva y sigilosa, ya que, al igual que la técnica TCP SYN Scan, nunca acaba de iniciar las conexiones que solicita. También es bastante rápida, permitiendo el análisis de miles de puertos por segundo en redes rápidas y que no estén limitadas por cortafuegos restrictivos.
136. Esta técnica envía una sonda Sctp con un fragmento de tipo INIT, como en el inicio de una comunicación Sctp normal, y espera una respuesta. Una respuesta con un fragmento de tipo INIT-ACK indica que el puerto está abierto, mientras que un fragmento de tipo ABORT en la respuesta indica que el puerto está cerrado. Si no se recibe respuesta tras varias retransmisiones, o se recibe un código de error ICMP Inalcanzable, el puerto se considera filtrado.

Entrada:

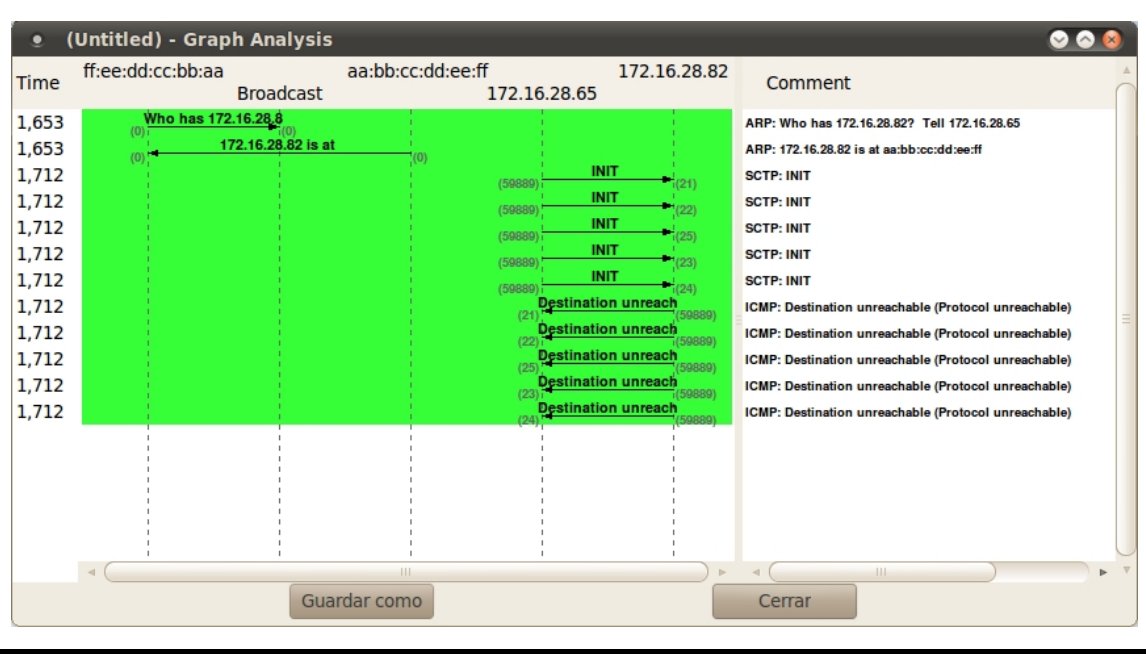
```
nmap -Pn -sY -p21-25 -v 172.16.28.82
```

Salida:

```
Starting Nmap 6.01 ( http://nmap.org ) at 2012-07-12 10:38 CEST
Initiating ARP Ping Scan at 10:38
Scanning 172.16.28.82 [1 port]
Completed ARP Ping Scan at 10:38, 0.03s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 10:38
Completed Parallel DNS resolution of 1 host. at 10:38, 0.00s elapsed
Initiating SCTP INIT Scan at 10:38
Scanning 172.16.28.82 [5 ports]
Completed SCTP INIT Scan at 10:38, 0.03s elapsed (5 total ports)
Nmap scan report for 172.16.28.82
Host is up (0.00017s latency).
PORT      STATE SERVICE
21/sctp   filtered ftp
22/sctp   filtered ssh
23/sctp   filtered unknown
24/sctp   filtered unknown
25/sctp   filtered unknown
MAC Address: AA:BB:CC:DD:EE:FF (Hewlett-Packard Company)

Read data files from: /usr/local/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
Raw packets sent: 6 (288B) | Rcvd: 6 (428B)
```

Flujo Mensajes:



3.2.10. SCTP COOKIE-ECHO Scan (-sZ)

- 137. Esta técnica es más avanzada que la SCTP INIT Scan (3.2.9), ya que toma ventaja del hecho de que las implementaciones de SCTP deben descartar de forma silenciosa sondas con fragmentos COOKIE ECHO si el puerto está abierto, pero deben responder con un fragmento tipo ABORT si el puerto está cerrado.
- 138. Una ventaja de esta técnica es que el uso de este tipo de fragmentos para realizar un análisis no es tan obvio como el uso de fragmentos INIT. Además, cabe la posibilidad de que existan cortafuegos sin estado que bloquean únicamente fragmentos INIT, que son los que generalmente inician la conexión.
- 139. Por otra parte, el problema es que esta técnica no puede diferenciar entre puertos abiertos y filtrados, dejando como resultado el puerto como abierto|filtrado en ambos casos.

3.2.11. IP protocol scan (-sO)

140. Esta opción no es técnicamente un escaneo de puertos ya que envía sondas IP iterando sobre el campo tipo de protocolo IP, de 8 bits de tamaño, en lugar de hacerlo sobre el número de puerto TCP o UDP.
141. Las sondas que especifiquen protocolos no soportados en el objetivo provocarán una respuesta ICMP protocolo inalcanzable, que Nmap aprovecha para enumerar el conjunto de protocolos de transporte soportados. En general las cabeceras IP enviadas no contendrán datos, a excepción de los protocolos TCP, UDP e ICMP para los cuales Nmap dispone de funciones que generan cabeceras bien formadas.
142. Según la respuesta recibida, los protocolos se clasifican en: abiertos (se recibe alguna respuesta del tipo de protocolo indicado), cerrados (si se recibe una respuesta de tipo ICMP protocolo inalcanzable), filtrados (si se recibe cualquier otro tipo de error ICMP inalcanzable) o abierto/filtrado (si no se recibe ninguna respuesta, incluso después de varias retransmisiones).
143. En la Figura 25 se traza la técnica de escaneo de protocolos. Se envían sondas al objetivo para averiguar si éste soporta los protocolos IP siguientes: ICMP, IGMP, TCP, UDP e IPv6 (1, 2, 6, 17 y 41 respectivamente), recibiendo confirmación de ICMP y TCP. Del resto no se obtiene mensaje de ICMP protocol unreachable, por lo que en principio se soportan aunque pudieran estar filtrados.

```

Entrada:
nmap -Pn -v -sO -p1,2,6,17,41 172.16.28.51

Salida:
Starting Nmap 6.01 ( http://nmap.org ) at 2012-07-12 11:02 CEST
Initiating ARP Ping Scan at 11:02
Scanning 172.16.28.51 [1 port]
Completed ARP Ping Scan at 11:02, 0.03s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 11:02
Completed Parallel DNS resolution of 1 host. at 11:02, 0.00s elapsed
Initiating IPProto Scan at 11:02
Scanning 172.16.28.51 [5 ports]
Discovered open port 6/ip on 172.16.28.51
Discovered open port 1/ip on 172.16.28.51
Discovered open port 17/ip on 172.16.28.51
Completed IPProto Scan at 11:02, 1.23s elapsed (5 total ports)
Nmap scan report for 172.16.28.51
Host is up (0.00078s latency).
PROTOCOL STATE      SERVICE
1      open      icmp
2      open|filtered igmp
6      open      tcp
17     open      udp
41     open|filtered ipv6
MAC Address: AA:BB:CC:DD:EE:FF (Cadmus Computer Systems)

Read data files from: /usr/local/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.36 seconds
Raw packets sent: 8 (220B) | Rcvd: 4 (152B)

```

Flujo Mensajes:

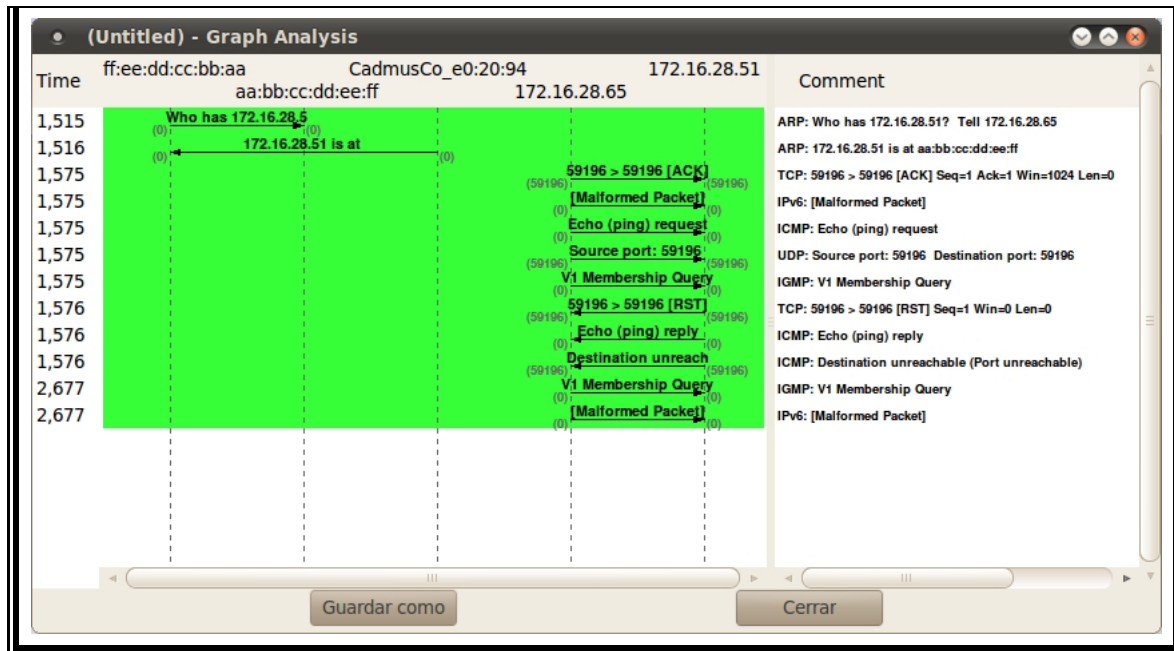


FIGURA 25. Técnica avanzada de escaneo IP protocol scan (-sO)

3.2.12. FTP bounce scan (-b <servidor ftp>)

144. Esta técnica explota una interesante característica del protocolo FTP (RFC 959), que permite establecer las denominadas conexiones Proxy-FTP, consistentes en que un cliente pueda solicitar a un servidor FTP que envíe archivos a una tercera máquina.
145. El abuso de esta característica puede ser aprovechado para realizar muchos y diversos tipos de ataques, por lo que ha dejado de ser implementada en la mayoría de los servidores en la actualidad. Uno de los usos controvertidos para lo que se podría utilizar, y que Nmap implementa, es para hacer un análisis de puertos desde el servidor FTP simplemente indicando que envíe un archivo a un conjunto de puertos del objetivo.
146. Esta técnica tiene un parámetro que indica el servidor FTP que se utilizará como proxy. El parámetro se puede indicar como [usuario]:<contraseña>@<servidor>[:<puerto>], siendo los campos entre corchetes opcionales.
147. Los mensajes de error que proporcione el servidor FTP indicarán a Nmap si los puertos están abiertos o cerrados. Esta técnica puede ser muy eficaz traspasando herramientas cortafuegos, puesto que los servidores FTP suelen establecer relaciones de confianza con equipos de redes internas, que otros equipos desde internet no pueden alcanzar. Sin embargo su uso naturalmente es controvertido al utilizar recursos de una máquina sin autorización de sus administradores.
148. Dada la antigüedad del ataque, y la poca utilidad práctica que pueda tener el mismo hoy en día, se decide no incluir un ejemplo práctico de ejecución de esta técnica.

3.3. MEJORANDO EL RENDIMIENTO DEL ANÁLISIS

149. Una vez conocidas las técnicas utilizadas por Nmap para el análisis de redes, se procede a detallar otros parámetros de configuración que pueden ser útiles a la hora de reducir el tiempo que consumen estos análisis.

3.3.1. CENTRAR EL ALCANCE DEL ANÁLISIS

150. Para ajustar al máximo la duración de un análisis, en primer lugar se debe tener claro qué se quiere analizar. Por ejemplo, si únicamente queremos un listado de los equipos disponibles en una determinada subred, no debemos lanzar el análisis con las opciones por defecto, sino que nos basta con realizar un Ping Scan (véase 3.1.3).

151. Por otro lado, aunque existen 65335 puertos TCP y otros tantos UDP, la mayoría de puertos están casi siempre cerrados, por lo que analizarlos todos (con la opción -p- o -p all) puede tomar gran cantidad de tiempo, ya que se debe esperar a que venzan los temporizadores de cada sonda que se envía. Nmap tiene una clasificación interna de puertos en función de su popularidad, y por defecto analiza los 1000 puertos más populares. Este valor se puede modificar a los 100 puertos más populares utilizando la opción -F (Fast Scan), o a cualquier otro valor con el parámetro --top-ports <número>. Por defecto, Nmap comprueba los puertos de manera aleatoria, pero se puede elegir una comprobación de puertos secuencial utilizando el parámetro -r.

152. Si se tiene una lista personalizada de puertos a analizar, se puede utilizar el parámetro -p <lista> para indicarlo. Los elementos de esta lista están separados por comas, y se pueden introducir rangos separados por guiones (p.ej. -p 22,80,443,6666-7000,8080,8443). Los rangos también pueden ser abiertos, indicando en este caso los puertos mayores o menores que el indicado (p.ej. -p 1024- indica los puertos a partir del 1024 hasta el 65535).

153. Los puertos de esta lista personalizada se pueden agrupar por protocolo, añadiendo el identificador U para puertos UDP, T para TCP y S para SCTP (p.ej. -p U:53,111,T:21-25,80,139,S:9).

154. También existe una opción que permite analizar sólo puertos que estén incluidos en el fichero nmap-services. En este caso se debe introducir el rango de puertos entre corchetes (p.ej. -p [-1024] analiza los puertos hasta el 1024 incluidos en el fichero nmap-services).

155. Otra opción que se utiliza con mucha frecuencia es la opción -A, que, además del análisis de puertos, realiza detección del sistema operativo y de versiones de los servicios, lanza la ejecución de scripts y traza la ruta de los paquetes hasta su destino. Todas estas opciones pueden no ser necesarias en todos los análisis, por lo que es conveniente definir por separado únicamente las que sean necesarias para agilizar el proceso de análisis.

3.3.2. SEPARAR Y OPTIMIZAR LOS ANÁLISIS UDP

156. Como se ha indicado en la sección dedicada a la técnica UDP Scan (véase 3.2.3), este tipo de análisis son generalmente complejos y costosos, ya que, al no ser un protocolo orientado a conexión, los servicios que utilizan estos puertos no suelen devolver ninguna información. Esto hace que, habitualmente, se queden fuera de los análisis de seguridad.
157. Uno de los problemas, ya discutidos en la sección relativa a la técnica UDP, consiste en el límite de paquetes ICMP que el objetivo envía como respuesta a sondas enviadas por Nmap (véase párrafo 107). Aunque Nmap detecta este comportamiento, hacer un análisis de todos los puertos UDP de un equipo puede llevar un tiempo considerable. Además de restringiendo el número de puertos a analizar, como se indica en la sección anterior, este tiempo se puede reducir indicando a Nmap que analice más equipos en paralelo mediante el parámetro `--min-hostgroup <número>`.
158. Otro de los problemas principales de los análisis UDP radica en la diferenciación de los estados abierto y filtrado. Para tratar de diferenciar entre ambos estados, se puede utilizar el análisis de versiones (parámetro `-sV`), como se ha comentado anteriormente (véase párrafo 108), a costa de que el tiempo necesario para completar el análisis crezca significativamente. Para tratar de mitigar el incremento del tiempo de ejecución se puede utilizar el modificador `--version-intensity <número>` con valor cero, de forma que no se envíen todas las posibles sondas de que Nmap dispone, y se envíen únicamente aquellas con mayor probabilidad de éxito.

3.3.3. TUNING TEMPORAL AVANZADO

159. Nmap dispone de modificadores que permiten fijar de forma personalizada distintos parámetros que afectan a la duración de los análisis. Estos son:
160. `--min-parallelism <valor>`, `--max-parallelism <valor>`: Indica el nivel mínimo o máximo de sondas pendientes de respuesta a manejar. Por defecto, Nmap calcula este valor de forma dinámica, basándose en el rendimiento de la red. Si detecta que se están perdiendo paquetes, ralentiza el envío de sondas y reduce el número de respuestas pendientes, para no perder precisión
161. `--min-hostgroup <valor>`, `--max-hostgroup <valor>`: Nmap tiene la habilidad de analizar varios equipos en paralelo, haciendo grupos de direcciones IP. Por defecto, ajusta automáticamente este valor, dando valores más altos a análisis UDP que a los TCP, por razones de eficiencia. También es recomendable ajustar un valor alto si el número de puertos a analizar por equipo es pequeño.
162. `--min-rtt-timeout <msec>`, `--max-rtt-timeout <msec>`, `--initial-rtt-timeout <msec>`: Cuando Nmap envía una sonda, mantiene el canal abierto a la espera de una respuesta. Si no la obtiene en un tiempo determinado, la da por perdida y pasa a la siguiente tarea, que puede ser retransmitir la sonda o pasar a otro puerto. Estos

modificadores determinan el tiempo que Nmap debe esperar hasta que la sonda se descarte. Elegir el valor correcto puede acelerar el escaneo, por ejemplo, en situaciones en las que hay cortafuegos presentes o cuando estamos ante una red especialmente rápida y poco congestionada.

163. `--max-retries <valor>`: Especifica el número máximo de veces que Nmap debe retransmitir una sonda antes de descartarla definitivamente. Si no se recibe respuesta, puede ser que la sonda se haya filtrado en un cortafuegos, que un objetivo tenga el ratio de respuesta limitada, o que simplemente se haya perdido por la red. Este valor determina en gran medida la eficiencia y eficacia del escaneo. Un valor cercano a 0 hará que el escaneo sea muy rápido, a costa de perder precisión. Un valor alto lo ralentizaría, pero obtendríamos unos valores precisos. Se deberá fijar el valor en función de la fiabilidad de la red.
164. `--host-timeout <msec>`: En algunas ocasiones Nmap necesita mucho tiempo para hacer un análisis completo a un equipo, ya sea por cortafuegos muy restrictivos, hardware o software pobre, limitación de envío de paquetes, o red saturada. Debido a esto, un análisis global estaría retrasándose demasiado por un único equipo. Se podría entonces especificar con este modificador que descarte el escaneo del equipo si se sobrepasa un límite de tiempo concreto.
165. `--scan-delay <msec>`, `--max-scan-delay <msec>`: Controlan el tiempo de espera entre cada sonda enviada a un mismo objetivo. Útil para tratar de evadir sistemas IDS o IPS.
166. `--min-rate <paquetes/segundo>`, `--max-rate <paquetes/segundo>`: Tasa mínima y máxima de paquetes por segundo que Nmap envía. Por defecto no se establecen tasas mínimas ni máximas.
167. `--defeat-rst-ratelimit`: Además de la limitación de errores ICMP expuesta en secciones anteriores, algunos sistemas comienzan a aplicar limitaciones similares al número de paquetes RST que envían. Por defecto Nmap intenta adaptarse a estas limitaciones para lograr unos resultados tan fiables como sea posible. Este parámetro instruye a Nmap para que no tenga en cuenta esta limitación. En algunos tipos de análisis puede ser útil (por ejemplo utilizando la técnica SYN Scan, si únicamente nos importan los puertos abiertos y no nos importa no distinguir entre puertos filtrados o cerrados), pero generalmente reduce la fiabilidad del análisis. Se encuentra deshabilitado por defecto.

3.3.4. PLANTILLAS TEMPORALES

168. Para facilitar el uso de los parámetros indicados en la sección anterior, Nmap incluye varias plantillas con las que intenta dar cobertura al mayor número posible de situaciones en lo que a análisis de equipos se refiere. Estas plantillas se pueden fijar mediante el parámetro `-T`, seguidas del número de plantilla sin espacios (p.ej. `-T4`), o seguidas de su nombre (p.ej. `-T polite`).

	T0	T1	T2	T3	T4	T5
Nombre	paranoid	sneaky	polite	normal	aggressive	insane
min-rtt-timeout	100	100	100	100	100	50
max-rtt-timeout	300,000	15,000	10,000	10,000	1,250	300
initial-rtt-timeout	300,000	15,000	1,000	1,000	500	250
max-retries	10	10	10	10	6	2
host-timeout	0	0	0	0	0	900,000
min-parallelism	Dinámico					
max-parallelism	1	1	1	Dinámico		
min-hostgroup	Dinámico					
max-hostgroup	Dinámico					

169. Por defecto, y si no se indica lo contrario, se utiliza la plantilla normal, por lo que introducir el parámetro -T3 no tiene ningún efecto sobre el análisis.
170. En redes de banda ancha o redes locales, se recomienda utilizar -T4, siendo -T5 solo recomendable en redes extremadamente rápidas y poco congestionadas. Por otra parte, las plantillas -T0 y -T1 pueden ser útiles para evitar alertas en sistemas IDS, a costa de aumentar extraordinariamente el tiempo de análisis.

3.4. ANÁLISIS DE REDES IPv6

171. Nmap ofrece soporte para analizar redes IPv6 desde 2002. En versiones anteriores, Nmap utilizaba la implementación de IPv6 proporcionada por el sistema operativo para funcionar, lo que hacía que los resultados bajo redes de este tipo pudieran no ser completos. A partir de la versión 6 este problema se ha solucionado, implementándose todo el motor de análisis para IPv6, lo que hace que Nmap deje de depender de las llamadas al sistema operativo, y aumenta su potencia y flexibilidad a la hora de analizar también redes IPv6. La sintaxis es la misma que en un análisis normal, siendo necesario añadir el parámetro -6 para que el análisis se realice a través de IPv6.
172. Para que este tipo de análisis tenga éxito, tanto el equipo donde se ejecuta Nmap como los objetivos a analizar deben tener una dirección IPv6 válida. Dada la desigual penetración de IPv6 en los ISP, es posible que el equipo desde el que se lanza el análisis no tenga todavía una dirección Ipv6, en cuyo caso es posible utilizar algún proveedor de túneles IPv6¹⁵ para poder llevar a cabo el análisis.

3.5. OTROS PARÁMETROS RELEVANTES

173. A continuación se muestran algunos parámetros que no han sido discutidos a lo largo de esta sección al no corresponder con técnicas de análisis ni de optimización, pero pueden ser útiles a la hora de realizar un análisis. La definición

¹⁵ Se puede encontrar un listado de proveedores de túneles IPv6 en:
http://en.wikipedia.org/wiki/List_of_IPv6_tunnel_brokers

de parámetros hecha aquí no es completa. Para un listado completo de todos los parámetros y opciones disponibles, se recomienda revisar el Anexo B (ver 5.10).

3.5.1. OBJETIVOS A ANALIZAR (-iL, -iR)

174. Por defecto, Nmap toma como objetivos a analizar los equipos o redes que se le pasan por parámetro. Se puede cambiar este comportamiento de varias formas.
175. Si se dispone de un listado de direcciones IP o nombres de dominio en un fichero, se puede indicar a Nmap que lea los objetivos de dicho fichero con la opción -iL <fichero>. Para que sea correctamente reconocido, cada elemento del fichero debe estar en una línea distinta.
176. Por otra parte, si no se tiene un objetivo definido y simplemente se están probando las características de una red, se puede utilizar la función -iR <número> para realizar un análisis a <número> equipos al azar.
177. Se pueden introducir también exclusiones, utilizando el parámetro --exclude, seguido de una lista de equipos o subredes a excluir, separados por comas. Si se dispone de un fichero con todas las exclusiones, se puede añadir a Nmap con el parámetro --excludefile <fichero>.

3.5.2. SALIDA DE RESULTADOS (-oN, -oX, -oS, -oG, -oA)

178. Estas opciones hacen que Nmap, además de mostrar la salida por pantalla, guarde un fichero con los resultados del análisis. Existen 4 formatos de salida, que son:
179. Normal (-oN <fichero>) muy similar a la salida interactiva, sin trazas y problemas relacionados con la ejecución del análisis.
180. XML (-oX <fichero>) formato XML estándar que puede ser convertido a HTML o utilizado por otras aplicaciones, como las interfaces gráficas.
181. Grepable (-oG <fichero>): formato que muestra la mayoría de información relevante en una línea, para facilitar su tratamiento con la herramienta de consola grep.
182. Script Kiddie (-oS <fichero>): es idéntico a la salida normal, pero modificando algunas letras para imitar el modo de escritura utilizado históricamente por los "Script Kiddies".
183. Existe además una quinta opción (-oA <nombre>) que genera la salida en los formatos normal, grepable y XML en tres ficheros del mismo nombre y con extensión nmap, xml y gnmap, respectivamente.

3.5.3. DETECCIÓN DE VERSIONES (-sV)

184. Esta opción hace que, además de determinar qué puertos están abiertos, se envíen sondas adicionales para tratar de descubrir qué servicios y versiones de los mismos se están ejecutando.

185. El número de pruebas que se envía se puede controlar mediante la opción `--version-intensity <número>`. Esta opción acepta un parámetro, que puede variar de 0 a 9, indicando la cantidad de pruebas que se ejecutarán, empezando en 0 con las más comunes y añadiendo pruebas menos comunes hasta el valor 9, donde se ejecutan todas las pruebas de que dispone Nmap. El valor por defecto cuando no se indica esta opción es 7.

3.5.4. DETECCIÓN DE SISTEMA OPERATIVO (-O)

186. Nmap dispone de una función que permite el reconocimiento del sistema operativo del objetivo a analizar basándose en la comparación de huellas de la pila TCP/IP de los objetivos con una base de firmas recopilada por Nmap.
187. Esta acción se habilita con el parámetro `-O` (letra o mayúscula).

3.5.5. USO DE SCRIPTS (-sC)

188. Además de permitir el descubrimiento de puertos y servicios abiertos en las máquinas objetivo, Nmap puede lanzar comprobaciones extra a los equipos y puertos encontrados para obtener más información. Estas comprobaciones se ejecutan en forma de scripts, la mayoría de los cuales vienen integrados en el paquete distribuido de Nmap. Para lanzar un conjunto de scripts por defecto se utiliza la opción `-sC`. Esta funcionalidad, así como directrices para gestionar y crear nuevos scripts, se tratan en la sección 5 de la presente guía.

3.5.6. ANÁLISIS AGRESIVO (-A)

189. Esta opción es un atajo que combina los siguientes parámetros: detección de sistema operativo (`-sO`), detección de versiones (`-sV`), ejecución de scripts (`-sC`) y traza de la ruta seguida por los paquetes (`--traceroute`). Puede ser útil para realizar un análisis exhaustivo de los objetivos, pero hay que tener en cuenta que envía muchas sondas a cada puerto a analizar, por lo que el tiempo de ejecución del análisis puede aumentar considerablemente en redes lentas o congestionadas, o si existen elementos que filtren el tráfico.

3.5.7. INTERACTUANDO CON NMAP DURANTE UN ANÁLISIS

190. Durante la ejecución de un análisis se puede instruir a Nmap para que nos ofrezca información relevante acerca del proceso de análisis o de su propio funcionamiento.
191. Al pulsar las teclas `v` o `V` se aumenta o disminuye, respectivamente, la cantidad de información que Nmap muestra sobre su funcionamiento.
192. Las teclas `d` y `D` aumentan y disminuyen, respectivamente, la cantidad de información de depuración que Nmap recoge.
193. Con las teclas `p` y `P` se activa y desactiva, respectivamente, el rastreo de paquetes.
194. Se puede consultar una ayuda con los parámetros anteriores pulsando la tecla `?`.

195. Finalmente, cualquier otra tecla pulsada imprime un mensaje de estado indicando el porcentaje de análisis que se ha realizado y el tiempo estimado para finalizar el análisis. Un ejemplo de mensaje podría ser el siguiente:

```
Stats: 0:00:07 elapsed; 20 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 33.33% done; ETC: 20:57 (0:00:12 remaining)
```

4. PROCEDIMIENTOS

196. Una vez se han asentado las bases teóricas necesarias y se han puesto a prueba las opciones principales, este capítulo las combina con la finalidad de ayudar a administradores de red y sistemas a la realización de múltiples tareas, todo desde un punto de vista práctico. Estas tareas se han agrupado en una serie de Procedimientos, cada uno de los cuales representa un marco de trabajo común.
197. Estos procedimientos se han estructurado de forma que se presenta, en primer lugar, los tipos de análisis cubiertos por el procedimiento (descubrimiento de equipos, escaneo de puertos, detección de servicios, etc.).
198. Seguidamente, se describe el procedimiento, dando ejemplos prácticos en forma de situaciones en las que un análisis como los expresados en el procedimiento podría ser válido, y se muestran una serie de objetivos de ejemplo sobre los que se lanzarán análisis.
199. Finalmente, se muestran las líneas de configuración de Nmap necesarias para llevar a cabo los análisis según se ha indicado anteriormente, así como resultados de varias ejecuciones de dichas líneas de configuración contra los objetivos de ejemplo que se han propuesto.

4.1. TÉCNICAS DE DESCUBRIMIENTO DE EQUIPOS EN UNA SUBRED

200. Una de las responsabilidades más importantes de cualquier administrador de red es maximizar la disponibilidad de los recursos conectados a su red, aspecto clave de la seguridad de cualquier sistema. Esta preocupación le debe obligar a tener conocimiento actualizado de qué recursos mantiene y de cuál es su estado, activo o inactivo. Naturalmente esta información es también deseada, desde el punto de vista de un atacante, y su obtención con frecuencia es una actividad precursora de un acto de hacking o asalto a un sistema sin consentimiento de su administrador.
201. Este procedimiento revela cómo recabar información útil del estado de máquinas conectadas a una misma subred con ayuda de Nmap, conociendo si son o no alcanzables remotamente, así como la información que es posible obtener de ellas mediante consultas a un servidor DNS.

4.1.1. DESCRIPCIÓN DEL PROCEDIMIENTO

202. Este procedimiento ilustra el potencial de Nmap en uno de sus usos principales, el descubrimiento de equipos, y revela cómo obtener información que pueda ser de gran ayuda a un administrador de red en la tarea rutinaria de velar por la seguridad y buen funcionamiento de la red que gestiona.

Aplicaciones para administradores

El caso práctico aquí expuesto puede ser extrapolado para servir de ayuda en las siguientes tareas administrativas para el mantenimiento rutinario de una red:

- Inventario de nombres de máquina presentes en una red.
- Localización de máquinas activas y no activas (es decir, alcanzables o no).
- Auditoría de una red: Evaluación del nivel de seguridad de una red.
- Obtener un conjunto de objetivos potenciales para un escaneo de puertos eficiente.

Objetivos analizados

Para este procedimiento se han elegido varios conjuntos de objetivos, teniendo en cuenta distintos criterios como su proximidad al equipo desde que se ejecuta el análisis, la cantidad de equipos levantados, y las características de las redes en que se sitúan.

Las distintas ejecuciones de Nmap se han lanzado desde un equipo cliente conectado a una red dependiente de RedIRIS, por lo que los análisis a redes institucionales y educacionales, también conectadas con RedIRIS, serán más rápidas.

Así, los objetivos marcados para este procedimiento son los siguientes:

- **www.empresa.es/24**: máquinas pertenecientes a una red de una empresa española.
- **www.universidad.es/24**: conjunto de máquinas pertenecientes una universidad española.
- **192.168.1.0/24**: subred local a la máquina que realiza el análisis.
- **www.organiza.org/24**: subred de máquinas establecidas fuera del territorio español (en concreto en Estados Unidos).
- **www.entidad.com/24**: máquinas pertenecientes a una entidad extranjera situada en un país lejano con muchos elementos intermedios. Para el ejemplo se ha utilizado una entidad japonesa.

4.1.2. CONFIGURACIÓN DE NMAP

203. La tarea de descubrimiento de equipos en una subred es, gracias a las técnicas que proporciona Nmap, especialmente sencilla de ejecutar, siendo sus resultados también de fácil interpretación. Para completar este procedimiento se van a utilizar las siguientes técnicas descritas en el anterior capítulo, divididas en dos grupos:

- **Obtención de nombres de máquina de los objetivos**

- List Scan `-sL`:

```
nmap -sL -v www.universidad.es/24
```

- **Determinación del estado de las máquinas**

- Ping Sweep `-sn` (sin más opciones, realiza `-PA80`, `-PS443`, `-PE` y `-PP`):

```
nmap -sn -v www.universidad.es/24
```

- o Ping TCP ACK **-PA** (puerto 80 por defecto)

```
nmap -sn -PA -v www.universidad.es/24
```

- o Ping TCP SYN **-PS** (puerto 80 por defecto)

```
nmap -sn -PS -v www.universidad.es/24
```

- o Ping ICMP Echo **-PE**

```
nmap -sn -PE -v www.universidad.es/24
```

204. Para determinar el estado de las máquinas se han utilizado varias técnicas ya que, como se indicó en la definición de cada una de las técnicas, los resultados pueden diferir, dependiendo de los servicios que estén disponibles en dichas máquinas o la configuración que se haya hecho de los cortafuegos.

4.1.3. RESULTADOS

205. A continuación se muestran dos salidas reducidas de Nmap para las dos partes de este Procedimiento. Para la parte de obtención de nombres de máquina de los objetivos, realizada con la técnica List Scan, se han suprimido la mayoría de resultados que no obtuvieron respuesta al hacer la resolución inversa DNS.

206. Para la fase de determinación del estado de las máquinas, realizada con las técnicas Ping Sweep, Ping TCP ACK y Ping ICMP Echo, se han suprimido algunas de las entradas, a pesar de tratarse de máquinas alcanzables, a fin de reducir el tamaño de la salida mostrada.


```
Starting Nmap 6.01 ( http://nmap.org ) at 2012-06-26 14:46 CEST
Nmap scan report for 169.254.0.0
Nmap scan report for internet.universidad.es (169.254.0.1)
Nmap scan report for 169.254.0.2
[ ... ]
Nmap scan report for 169.254.0.6
Nmap scan report for gw.universidad.es (169.254.0.7)
Nmap scan report for teclado.universidad.es (169.254.0.8)
Nmap scan report for wisdom.universidad.es (169.254.0.9)
Nmap scan report for artemisa.universidad.es (169.254.0.10)
Nmap scan report for 169.254.0.11
Nmap scan report for 169.254.0.12
Nmap scan report for pluton.universidad.es (169.254.0.13)
Nmap scan report for heretic.universidad.es (169.254.0.14)
Nmap scan report for alcas.universidad.es (169.254.0.15)
Nmap scan report for 169.254.0.16
[ ... ]
Nmap scan report for 169.254.0.21
Nmap scan report for omega.universidad.es (169.254.0.22)
Nmap scan report for gamma.universidad.es (169.254.0.23)
Nmap scan report for 169.254.0.24
Nmap scan report for kabuto.universidad.es (169.254.0.25)
Nmap scan report for 169.254.0.26
Nmap scan report for 169.254.0.27
Nmap scan report for 169.254.0.28
Nmap scan report for www.universidad.es (169.254.0.29)
Nmap scan report for 169.254.0.30
[ ... ]
Nmap scan report for 169.254.0.34
Nmap scan report for homero.universidad.es (169.254.0.35)
Nmap scan report for 169.254.0.36
[ ... ]
Nmap scan report for 169.254.0.56
Nmap scan report for caronte.universidad.es (169.254.0.57)
Nmap scan report for 169.254.0.58
Nmap scan report for 169.254.0.59
Nmap scan report for rvald.universidad.es (169.254.0.60)
Nmap scan report for 169.254.0.61
[ ... ]
Nmap scan report for 169.254.0.129
Nmap scan report for quijote.universidad.es (169.254.0.130)
Nmap scan report for 169.254.0.131
[ ... ]
Nmap scan report for 169.254.0.255
Nmap done: 256 IP addresses (0 hosts up) scanned in 0.05 seconds
```

FIGURA 26. RESULTADOS PROCEDIMIENTO 1: LISTADO DE MÁQUINAS PARA WWW.UNIVERSIDAD.ES/24.

```
Starting Nmap 6.01 ( http://nmap.org ) at 2012-06-26 14:41 CEST
Nmap scan report for 169.254.128.0
Host is up (0.025s latency).
Nmap scan report for 169.254.128.1.es.isp.net (169.254.128.1)
Host is up (0.052s latency).
[ ... ]
Nmap scan report for www.empresa.es (169.254.128.24)
Host is up (0.051s latency).
rDNS record for 169.254.128.24: correo.empresa.es
Nmap scan report for xuquer.empresa.es (169.254.128.27)
Host is up (0.035s latency).
Nmap scan report for 169.254.128.28.es.isp.net (169.254.128.28)
Host is up (0.035s latency).
[ ... ]
Nmap scan report for 169.254.128.79.es.isp.net (169.254.128.79)
Host is up (0.028s latency).
Nmap scan report for mail01.empresa2.es (169.254.128.105)
Host is up (0.030s latency).
Nmap scan report for 169.254.128.106.es.isp.net (169.254.128.106)
Host is up (0.028s latency).
[ ... ]
Nmap scan report for 169.254.128.230.es.isp.net (169.254.128.230)
Host is up (0.031s latency).
Nmap done: 256 IP addresses (57 hosts up) scanned in 3.17 seconds
```

FIGURA 27. RESULTADOS PROCEDIMIENTO 1: ESTADO DE MÁQUINAS PARA WWW.EMPRESA.ES/24.

207. La siguiente tabla (Tabla 1) recoge una comparativa cuantitativa de los resultados anteriores hechos sobre cada uno de los objetivos definidos anteriormente. Para obtener un resultado estable, se ha lanzado cada análisis 5 veces, mostrándose en la tabla el tiempo medio de ejecución del análisis. De este modo es posible relativizar el análisis y extraer conclusiones sobre el impacto en el rendimiento de la técnica según las opciones elegidas. La interpretación de estos resultados se realiza en el siguiente capítulo.

COMPARATIVA Descubrimiento	Salto ¹⁶	RTT Medio ¹⁷	List Scan -sL	Ping Sweep -sn	Ping TCP ACK -sn -PA	Ping TCP SYN -sn -PS	Ping ICMP Echo -sn -PE
www.empresa.es/24	14	30 ms	0,22s	57 en 3.17s	0 maq en 52,15s	50 maq en 3,28s	22 maq en 1,86s
www.universidad.es/24	12	10.6 ms	0,09s	32 máq. en 38,25s	0 máq. en 52,11s	35 maq en 0,95s	0 máq. en 52,11s
192.168.1.0/24	0	<0.5 ms	2,52s	40 máq. en 3,41s	19 maq en 27,35s	34 maq. en 9,44s	37 maq en 1,85s
www.organiza.org/24	17	186.4 ms	2,41s	220 máq. en 5,21s	0 máq. en 52,18s	203 maq en 4,5s	210 máq. en 4,68s
www.entidad.com/24	25	306 ms	2,45s	14 en 36,33s	0 maq. en 52.11s	10 maq en 6,5s	9 maq en 27,11s

TABLA 1. COMPARATIVA RESULTADOS PROCEDIMIENTO 1.

4.2. TÉCNICAS DE ANÁLISIS DE PUERTOS

208. Otra de las funcionalidades principales de Nmap consiste en analizar los puertos de los equipos fijados como objetivos, para descubrir aquellos que son accesibles y, por tanto, tienen servicios asociados. Esta información, como complemento a la obtenida en el procedimiento anterior, es útil a un administrador de red para fijar su ámbito de actuación en lo que al control de sus redes se refiere.

209. Además, se puede dar el caso de que algunas de las máquinas pertenecientes al conjunto de objetivos no se hayan mostrado como activas en el análisis inicial, si se encuentran desconectadas, han sido adecuadamente aisladas del resto de la red, o si se controlan los tipos de comunicaciones utilizados en la fase de descubrimiento de equipos. En este último caso será necesario realizar un análisis incluyendo dichas máquinas, para cubrir todos los servicios disponibles en la red.

210. No es recomendable, por otra parte, lanzar este tipo de análisis sobre un conjunto grande de equipos en solitario, sin haber reducido antes el conjunto de objetivos a analizar utilizando técnicas del procedimiento anterior.

211. Para cada uno de los puertos seleccionados en el análisis (1000 por defecto, hasta un máximo de 65535), se deben enviar sondas y esperar a una respuesta que, en la

¹⁶ El número de saltos se ha obtenido añadiendo la opción --traceroute en la ejecución de Nmap (nmap -sn --traceroute www.universidad.es).

¹⁷ El RTT medio se ha obtenido mediante la orden hping3 al puerto 80 del equipo objetivo (hping3 -S -p 80 -c 100 www.universidad.es)

inmensa mayoría de casos, no llega, lo que implica el vencimiento de los temporizadores en cada prueba. Además, se envían varias retransmisiones de cada paquete, aumentando el tiempo que se emplea en comprobar cada puerto. Si este tiempo se multiplica por el número de puertos y el número de equipos a analizar, podemos convertir un análisis relativamente rápido en un análisis muy costoso.

212. Un administrador de red también debe ser consciente de que Nmap es una herramienta que es utilizada frecuentemente por atacantes para recabar información antes de consumar el asalto a un sistema remoto. Por esta razón, los administradores han de conocer las técnicas de análisis de puertos existentes, para evaluar y ayudar a incrementar la seguridad de los sistemas que gestionan.

4.2.1. DESCRIPCIÓN DEL PROCEDIMIENTO

213. Este Procedimiento plantea diversos casos en los que es posible recabar información útil mediante técnicas de escaneo de puertos. Debido a que estas técnicas requieren de bastante tiempo para su finalización, los casos planteados ilustran el potencial de Nmap en tareas administrativas que afectan a conjuntos de objetivos de distintos tamaños.

Aplicaciones para Administradores

Los casos prácticos aquí expuestos puede ser extrapolados para servir de ayuda en las siguientes tareas administrativas para el mantenimiento rutinario de una red:

- Enumeración de los puertos abiertos y cerrados en una máquina local o remota.
- Identificación de servidores y servicios en una red.
- Descubrimiento de las reglas de una herramienta cortafuegos.
- Auditoría de una red: Evaluación del nivel de seguridad de una red.

Objetivos analizados

Se han elegido varios conjuntos de objetivos para proporcionar suficientes casos representativos a este Procedimiento. Los objetivos son los siguientes:

- **localhost**: máquina con sistema operativo Linux con *kernel* v2.6.32 desde donde se realizan los análisis de este procedimiento.
- **dejavu**: máquina Windows XP SP2 perteneciente a la red local (*FastEthernet*) del origen.
- **www.empresa.es/24**: máquinas pertenecientes a una red de una empresa española.
- **www.organiza.org/24**: subred de máquinas establecidas fuera del territorio español (en concreto en Estados Unidos).

4.2.2. CONFIGURACIÓN DE NMAP

214. Los casos a continuación propuestos están específicamente elegidos para cubrir las aplicaciones para administradores expuestas en la anterior sección. Dado que éste es un procedimiento administrativo básico, se ha elegido la técnica de escaneo de puertos por defecto de Nmap, TCP SYN Scan (ver 3.2.1), debido a su sencillez y a que arroja resultados muy fiables. Por este mismo motivo, se han restringido los casos al escaneo de rangos de puertos TCP, excluyéndose los UDP. Existen otras técnicas más avanzadas que serán tratadas en secciones posteriores.
215. A continuación se incluyen los comandos de ejecución así como algunas capturas de pantalla de la interfaz de usuario.
- **CASO 1: enumeración de los puertos TCP abiertos y cerrados en una máquina local.** Este caso no es aplicable si se realiza desde una máquina con sistema operativo Windows. Se escanean todos los puertos (del 1 al 65.535)

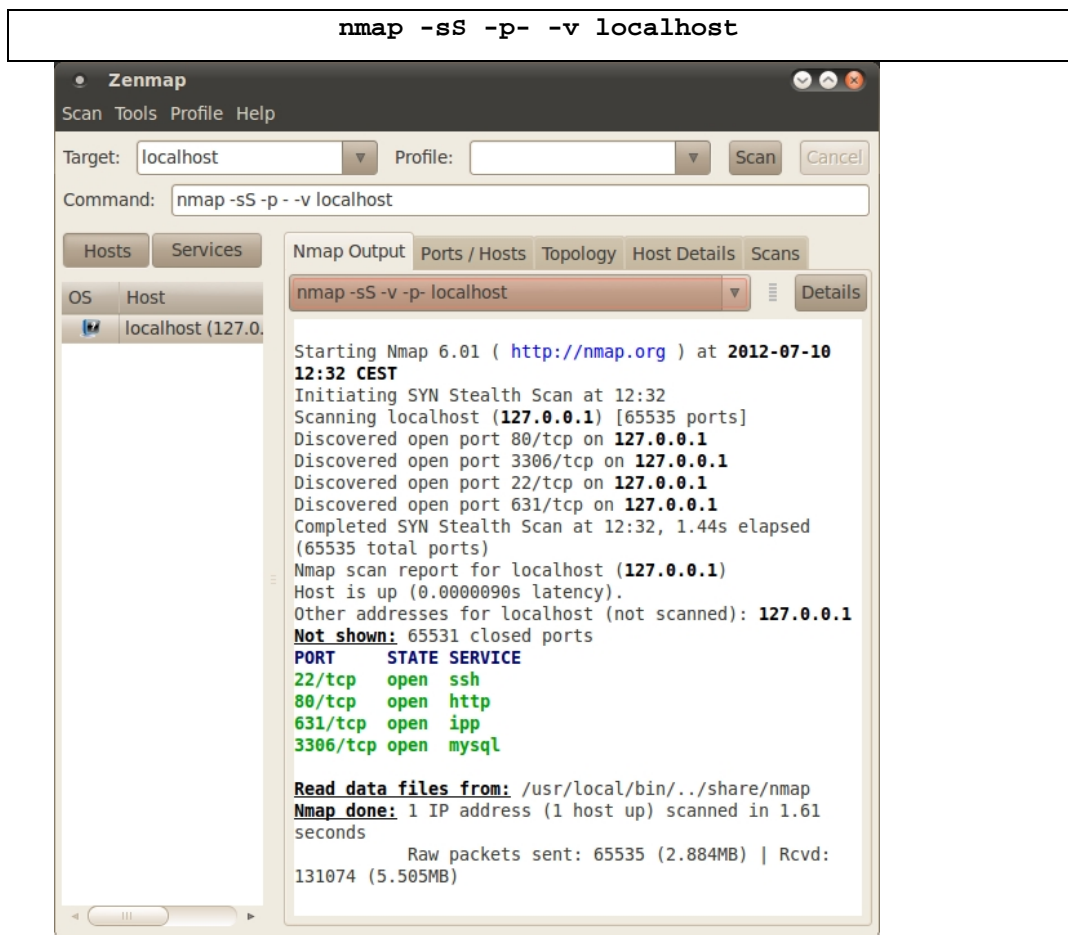


FIGURA 28. ENUMERACIÓN DE PUERTOS CON SYN STEALTH. PROCEDIMIENTO2 - CASO 1

- **CASO 2: enumeración de los puertos TCP abiertos y cerrados en una máquina perteneciente a la misma subred local.** Para este caso se han instalado varios servicios y se ha desactivado la herramienta de cortafuegos personal. Se escanean todos los puertos TCP, en este caso de modo secuencial (añadiendo el modificador -r).

nmap -sS -p- -r -v dejavu

- **CASO 3: enumeración de los puertos TCP abiertos y cerrados en una máquina externa.** Se entiende por externa a una no perteneciente a la red local por lo que pueden transcurrir varias decenas de milisegundos por cada mensaje de respuesta esperado. En este caso, con el fin de acelerar el proceso, se escanea un conjunto reducido de puertos en lugar de todos. Se analizan los 1300 más comunes (con la opción `--top-ports 1300`).

```
nmap -sS --top-ports 1300 -v www.empresa.es
```

- **CASO 4: descubrimiento de las reglas de una herramienta cortafuegos.** Este caso es idéntico al CASO2, con la salvedad de que se ha activado el cortafuegos personal que viene instalado en Windows por defecto. Esto permitirá evaluar las reglas que se están aplicando.

```
nmap -sS -p- -r -v dejavu
```

- **CASO 5: identificación de servidores en una red.** Este caso ilustra como poder conocer la presencia de servidores comunes en una red, entendiendo por comunes aquellos que tienen FTP, SSH, TELNET, SMTP, HTTP y HTTPS, que en general se corresponden respectivamente con los puertos TCP [21, 22, 23, 25, 80, 443]. Para mejorar la eficiencia del proceso, y si no se indica de forma explícita, Nmap ejecuta una fase de descubrimiento por defecto con la técnica Ping Scan (ver 3.1.3), que incluye las siguientes pruebas `-PA80 -PS443 -PE -PP`. Se incluye también una prueba que no realiza la fase previa de descubrimiento, para mostrar las diferencias temporales existentes.

```
(Caso5.1): nmap -sS -p 21-23,25,80,443 -v www.empresa.es/24
```

```
(Caso5.2): nmap -sS -p 21-23,25,80,443 -v www.organiza.org/24
```

```
(Caso5.3): nmap -Pn -p 21-23,25,80,443 -v www.organiza.org/24
```

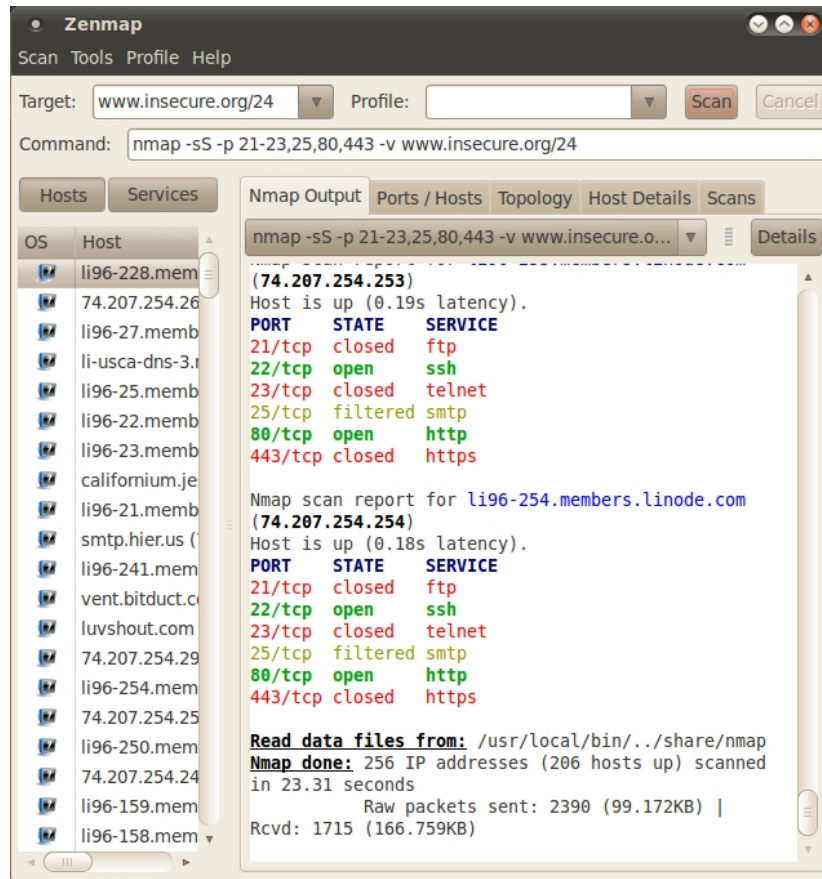


FIGURA 29. IDENTIFICACIÓN DE SERVIDORES EN UNA RED. PROCEDIMIENTO2-CASO5

4.2.3. RESULTADOS

```
Starting Nmap 6.01 ( http://nmap.org ) at 2012-06-04 12:53 CEST
Initiating SYN Stealth Scan at 12:53
Scanning localhost (127.0.0.1) [65535 ports]
Discovered open port 80/tcp on 127.0.0.1
Discovered open port 22/tcp on 127.0.0.1
Discovered open port 3306/tcp on 127.0.0.1
Discovered open port 631/tcp on 127.0.0.1
Completed SYN Stealth Scan at 12:53, 1.45s elapsed (65535 total ports)
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000090s latency).
Other addresses for localhost (not scanned): 127.0.0.1
Not shown: 65531 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
631/tcp   open  ipp
3306/tcp  open  mysql

Read data files from: /usr/local/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.63 seconds
Raw packets sent: 65535 (2.884MB) | Rcvd: 131074 (5.505MB)
```

FIGURA 30.- SALIDA PROCEDIMIENTO 2 - CASO 1

```
Starting Nmap 6.01 ( http://nmap.org ) at 2012-06-04 12:56 CEST
Initiating ARP Ping Scan at 12:56
Scanning dejavu (172.16.28.45) [1 port]
Completed ARP Ping Scan at 12:56, 0.03s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 12:56
Completed Parallel DNS resolution of 1 host. at 12:56, 0.00s elapsed
Initiating SYN Stealth Scan at 12:56
Scanning dejavu (172.16.28.45) [65535 ports]
Discovered open port 135/tcp on 172.16.28.45
Discovered open port 139/tcp on 172.16.28.45
Discovered open port 445/tcp on 172.16.28.45
```

```

Discovered open port 3389/tcp on 172.16.28.45
Completed SYN Stealth Scan at 12:57, 17.72s elapsed (65535 total ports)
Nmap scan report for dejavu (172.16.28.45)
Host is up (0.0014s latency).
Not shown: 65531 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
MAC Address: AA:BB:CC:DD:EE:FF (Cadmus Computer Systems)

Read data files from: /usr/local/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 17.96 seconds
Raw packets sent: 68813 (3.028MB) | Rcvd: 65536 (2.621MB)

```

FIGURA 31.- SALIDA PROCEDIMIENTO 2 - CASO 2

```

Starting Nmap 6.01 ( http://nmap.org ) at 2012-06-04 13:00 CEST
Initiating Ping Scan at 13:00
Scanning www.empresa.es (169.254.100.20) [4 ports]
Completed Ping Scan at 13:00, 0.04s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 13:00
Completed Parallel DNS resolution of 1 host. at 13:00, 0.00s elapsed
Initiating SYN Stealth Scan at 13:00
Scanning www.empresa.es (169.254.100.20) [1300 ports]
Discovered open port 22/tcp on 169.254.100.20
Discovered open port 80/tcp on 169.254.100.20
Discovered open port 443/tcp on 169.254.100.20
Completed SYN Stealth Scan at 13:00, 4.87s elapsed (1300 total ports)
Nmap scan report for www.empresa.es (169.254.100.20)
Host is up (0.0021s latency).
rDNS record for 169.254.100.20: pc100-20.empresa.es
Not shown: 1296 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
113/tcp   closed ident
443/tcp   open  https

Read data files from: /usr/local/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 5.04 seconds
Raw packets sent: 2602 (114.464KB) | Rcvd: 7 (296B)

```

FIGURA 32.- SALIDA PROCEDIMIENTO 2 - CASO 3

```

Starting Nmap 6.01 ( http://nmap.org ) at 2012-06-04 13:04 CEST
Initiating ARP Ping Scan at 13:04
Scanning dejavu (172.16.28.45) [1 port]
Completed ARP Ping Scan at 13:04, 0.03s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 13:04
Completed Parallel DNS resolution of 1 host. at 13:04, 0.00s elapsed
Initiating SYN Stealth Scan at 13:04
Scanning dejavu (172.16.28.45) [65535 ports]
SYN Stealth Scan Timing: About 2.22% done; ETC: 13:27 (0:22:45 remaining)
SYN Stealth Scan Timing: About 4.50% done; ETC: 13:27 (0:21:34 remaining)
Discovered open port 3389/tcp on 172.16.28.45
SYN Stealth Scan Timing: About 19.26% done; ETC: 13:12 (0:06:21 remaining)
SYN Stealth Scan Timing: About 45.97% done; ETC: 13:08 (0:02:22 remaining)
SYN Stealth Scan Timing: About 78.86% done; ETC: 13:07 (0:00:40 remaining)
Completed SYN Stealth Scan at 13:07, 167.38s elapsed (65535 total ports)
Nmap scan report for dejavu (172.16.28.45)
Host is up (0.00057s latency).
Not shown: 65534 filtered ports
PORT      STATE SERVICE
3389/tcp  open  ms-wbt-server
MAC Address: AA:BB:CC:DD:EE:FF (Cadmus Computer Systems)

Read data files from: /usr/local/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 167.59 seconds
Raw packets sent: 131148 (5.770MB) | Rcvd: 80 (3.504KB)

```

FIGURA 33.- SALIDA PROCEDIMIENTO 2 - CASO 4

```

Starting Nmap 6.01 ( http://nmap.org ) at 2012-06-04 13:43 CEST
Initiating Ping Scan at 13:43

```



```

Scanning 256 hosts [4 ports/host]
Completed Ping Scan at 13:43, 2.82s elapsed (256 total hosts)
Initiating Parallel DNS resolution of 256 hosts. at 13:43
Completed Parallel DNS resolution of 256 hosts. at 13:43, 0.16s elapsed
Nmap scan report for 169.254.254.0 [host down]
Nmap scan report for 169.254.254.4 [host down]
[ ... ]
Nmap scan report for 169.254.254.182 [host down]
Initiating SYN Stealth Scan at 13:43
Scanning 170 hosts [6 ports/host]
Discovered open port 21/tcp on 169.254.254.7
Discovered open port 21/tcp on 169.254.254.30
[ ... ]
Discovered open port 22/tcp on 169.254.254.53
Discovered open port 80/tcp on 169.254.254.14
Discovered open port 443/tcp on 169.254.254.157
Completed SYN Stealth Scan at 13:44, 8.99s elapsed (1020 total ports)
Nmap scan report for gateway.hosting.com (169.254.254.1)
Host is up (0.19s latency).
PORT      STATE      SERVICE
21/tcp    closed    ftp
22/tcp    closed    ssh
23/tcp    closed    telnet
25/tcp    filtered  smtp
80/tcp    closed    http
443/tcp   closed    https

[ ... ]

Nmap scan report for www.organiza.org (169.254.254.18)
Host is up (0.18s latency).
rDNS record for 169.254.254.18: web.organiza.org
PORT      STATE      SERVICE
21/tcp    filtered  ftp
22/tcp    open      ssh
23/tcp    filtered  telnet
25/tcp    filtered  smtp
80/tcp    open      http
443/tcp   open      https

Nmap scan report for 169.254.254.200 [host down]
Nmap scan report for 169.254.254.212 [host down]
Nmap scan report for 169.254.254.255 [host down]
Initiating SYN Stealth Scan at 13:44
Scanning 46 hosts [6 ports/host]
Discovered open port 21/tcp on 169.254.254.224
Discovered open port 21/tcp on 169.254.254.236
[ ... ]
Discovered open port 80/tcp on 169.254.254.221
Discovered open port 80/tcp on 169.254.254.254
Completed SYN Stealth Scan at 13:44, 4.64s elapsed (276 total ports)
Nmap scan report for 201.members.hosting.com (169.254.254.201)
Host is up (0.18s latency).
PORT      STATE      SERVICE
21/tcp    closed    ftp
22/tcp    open      ssh
[ ... ]
Nmap scan report for 254.members.hosting.com (169.254.254.254)
Host is up (0.18s latency).
PORT      STATE      SERVICE
21/tcp    closed    ftp
22/tcp    open      ssh
23/tcp    closed    telnet
25/tcp    filtered  smtp
80/tcp    open      http
443/tcp   closed    https

Read data files from: /usr/local/bin/./share/nmap
Nmap done: 256 IP addresses (216 hosts up) scanned in 16.82 seconds
Raw packets sent: 2434 (101.224KB) | Rcvd: 1398 (92.892KB)

```

FIGURA 34.- SALIDA PROCEDIMIENTO 2 - CASO 5.2

216. La Tabla 2 recoge una comparativa cuantitativa de los resultados de los casos anteriores. Cabe destacar las diferencias muy notables en el tiempo de ejecución de

las pruebas. Del mismo modo es muy destacable la valiosa información que puede dar el caso 5 en realmente pocos segundos. La interpretación de estos resultados se realiza en el siguiente capítulo (ver 5.9.2).

COMP. Escaneo	Direcciones analizadas	Puertos analizados	Estado puertos Abiertos/Cerrados/Filtrados	Paquetes enviados	Paquetes recibidos	Tiempo	puertos/s
CASO1	1	65.535	4/65.531/0	65.535	131.074	1,63s	40.205
CASO2	1	65.535	4/65.531/0	68.813	65.536	17,96s	3648,9
CASO3	1	1.300	3/1/1296	2.602	7	5,04s	257,9
CASO4	1	65.535	1/65.534/0	131.148	80	167,59s	413,2
CASO5.1	127 de 256	127x6 = 762	ftp: 34/48/45 ssh: 69/0/58 telnet: 4/57/66 smtp: 50/38/39 http: 81/16/30 https: 46/46/35 Total: 284/205/273	2.297	630	35,97s	21,18
CASO5.2 ¹⁸	206 de 256	206x6 = 1236	ftp: 41/119/46 ssh: 149/28/29 telnet: 0/150/56 smtp: 0/0/206 http: 167/28/11 https: 57/127/22 Total: 414/452/370	2.390	1.715	23,31s	53,02
CASO5.3 ¹⁸	256	256x6=1536	ftp: 41/119/96 ssh: 149/28/79 telnet: 0/150/106 smtp: 0/0/256 http: 167/28/61 https: 57/127/72 Total: 414/452/670	2.238	1.038	12,19s	126

Tabla 2. Comparativa resultados Procedimiento 2

4.3. TÉCNICAS DE DETECCIÓN DE SERVICIOS Y DE SISTEMA OPERATIVO

217. En el procedimiento anterior se pone a prueba gran parte del potencial de Nmap como herramienta de análisis de puertos. Nmap es capaz de ir más allá y proporcionar aún más información, aumentando con ello la fiabilidad y utilidad de sus resultados. Esto se consigue activando la opción de detección de servicios, que se explotará a lo largo de este Procedimiento.

218. La salida obtenida tras un análisis de puertos convencional es una lista de entradas formadas por los siguientes elementos:

[Número de puerto] - [Protocolo transporte] - [Estado] - [Servicio]

219. A continuación se puede ver un ejemplo, extraído del procedimiento anterior:

```
Starting Nmap 6.01 ( http://nmap.org ) at 2012-06-04 12:53 CEST
Initiating SYN Stealth Scan at 12:53
Scanning localhost (127.0.0.1) [65535 ports]
Discovered open port 80/tcp on 127.0.0.1
Discovered open port 22/tcp on 127.0.0.1
Discovered open port 3306/tcp on 127.0.0.1
Discovered open port 631/tcp on 127.0.0.1
Completed SYN Stealth Scan at 12:53, 1.45s elapsed (65535 total ports)
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000090s latency).
Other addresses for localhost (not scanned): 127.0.0.1
Not shown: 65531 closed ports
```

¹⁸ Esta parte del análisis se ha realizado desde un equipo que tenía filtrado el puerto 25 en la salida a Internet, por ello se detectan todos los servidores SMTP como filtrados.

```
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
631/tcp   open  ipp
3306/tcp  open  mysql

Read data files from: /usr/local/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.63 seconds
Raw packets sent: 65535 (2.884MB) | Rcvd: 131074 (5.505MB)
```

220. Sin embargo el campo “Servicio” es en este caso una mera suposición, obtenida de un fichero de Nmap que almacena correspondencias estándar entre puerto y servicio, o servicios conocidos (del inglés well-known services). Es el fichero nmap-services, que se encuentra en la carpeta de Nmap. La misma información también se puede obtener en plataformas UNIX en el fichero /etc/services. Aunque esta información es en la mayoría de las ocasiones fiable, es totalmente posible tener un servidor HTTP o SMTP en puertos distintos del 80/tcp o 25/tcp respectivamente. Esta circunstancia revela un campo de estudio del análisis de vulnerabilidades.
221. Nmap permite realizar pruebas adicionales en cada uno de los puertos detectados para asegurarse del tipo de servicios, e incluso versiones de los mismos, que existen tras dichos puertos. Esta información puede ser útil para un administrador para, por poner un ejemplo, saber si existen vulnerabilidades conocidas en alguno de los servicios disponible.
222. Igualmente podría ser de su interés hacer un inventario de plataformas instaladas en los equipos de la red mediante la opción de detección de sistema operativo. Con ella se podrá distinguir por ejemplo sistemas Windows de UNIX e incluso sistemas operativos embebidos, como los incorporados en dispositivos de interconexión de redes.

4.3.1. DESCRIPCIÓN DEL PROCEDIMIENTO

223. Este procedimiento presenta unos modificadores Nmap aplicados sobre las técnicas de análisis de puertos vistas en el anterior Procedimiento. Éstas son complementarias y ayudan a administradores de red y sistemas a resolver determinadas tareas avanzadas destinadas a hacer más fiable y ampliar la información obtenida en un escaneo de puertos.
224. Este aumento de información y mejora de fiabilidad conlleva inevitablemente un incremento notable en el tiempo del proceso, por lo que deberá ser tenido en cuenta a la hora de la elección del conjunto de puertos y objetivos.

Aplicaciones para Administradores

Los casos prácticos aquí expuestos pueden ser extrapolados para servir de ayuda en las siguientes tareas administrativas para el mantenimiento rutinario de una red:

- Identificación de versiones de servicios.
- Identificación precisa de servidores en una red.
- Determinación precisa del estado de puertos UDP.

- Inventario de plataformas instaladas en una red.

Objetivos analizados

Se han elegido varios conjuntos de objetivos para proporcionar suficientes casos representativos a este Procedimiento. Los objetivos son los siguientes:

- **dejavu:** máquina Windows XP SP2 perteneciente a la red local (*FastEthernet*) del origen.
- **router:** Conmutador/encaminador *FastEthernet-ADSL* doméstico.
- **www.organiza.org/24:** subred de máquinas establecidas fuera del territorio español (en concreto en Estados Unidos).
- **www.empresa.es/24:** máquinas pertenecientes a una red de una empresa española.
- **172.16.28.0/24:** subred local a la máquina que realiza el análisis.

4.3.2. CONFIGURACIÓN DE NMAP

225. Los casos a continuación propuestos están específicamente elegidos para cubrir las aplicaciones para administradores expuestas en la anterior sección. Para cada una de dichas aplicaciones se ha elaborado un caso de prueba representativo. Los modificadores de detección de versiones y sistema operativo incrementan notablemente el tiempo de la prueba, por lo que es necesario elegir un conjunto de puertos y objetivos lo más reducido que sea posible.

226. Tal y como se ha hecho en el anterior Procedimiento, se incluyen los comandos de ejecución así como algunas capturas de pantalla de la interfaz de usuario.

- **CASO 1: identificación de versiones de servicios.** Este caso repite el escaneo realizado en el Caso2 del Procedimiento 2 (sección 4.2), identificando las versiones pero sobre un conjunto menor de puertos, aplicando el modificador **-F** con el que únicamente se analizan los 100 puertos más comunes.

```
nmap -ss -sV -F -v dejavu
```

- **CASO 2: identificación precisa de servidores en una red.** Este caso complementa la información obtenida en el Caso5 del Procedimiento2 (sección 4.2). Nótese el incremento sustancial en el tiempo de ejecución del proceso.

```
nmap -ss -sV -p 22,25,80 -v www.empresa.es/24
```

```
nmap -ss -sV -p 22,25,80 -v www.organiza.org/24
```

- **CASO 3: determinación precisa del estado de puertos UDP.** La técnica de escaneo de puertos UDP (-sU), con frecuencia no es capaz de determinar si un puerto está abierto o filtrado. El detector de versiones puede ayudar en esta labor.

```
nmap -sU -sV -p 53,68,80 -v router
```

- **CASO 4: inventario de plataformas instaladas en una red.** Este caso se explota la funcionalidad de detección de sistema operativo, además de la de detección de servicios. Para que sea precisa, se requiere analizar un conjunto pequeño pero

variado de puertos TCP, de modo que aporten información sobre la plataforma instalada. No se utiliza en este caso la opción `-v` por generar demasiada información superflua.

```
nmap -ss -sV -O -v -F 172.16.28.0/25
```

4.3.3. RESULTADOS

```
Starting Nmap 6.01 ( http://nmap.org ) at 2012-07-13 12:40 CEST
NSE: Loaded 17 scripts for scanning.
Initiating Parallel DNS resolution of 1 host. at 12:40
Completed Parallel DNS resolution of 1 host. at 12:40, 0.00s elapsed
Initiating SYN Stealth Scan at 12:40
Scanning 172.16.28.65 [100 ports]
Discovered open port 23/tcp on 172.16.28.65
Discovered open port 80/tcp on 172.16.28.65
Completed SYN Stealth Scan at 12:40, 0.03s elapsed (100 total ports)
Initiating Service scan at 12:40
Scanning 2 services on 172.16.28.65
Completed Service scan at 12:40, 6.18s elapsed (2 services on 1 host)
NSE: Script scanning 172.16.28.65.
Nmap scan report for 172.16.28.65
Host is up (0.000011s latency).
Not shown: 98 closed ports
PORT      STATE SERVICE VERSION
23/tcp    open  ssh      OpenSSH 5.3p1 Debian 3ubuntu7 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.2.14 ((Ubuntu))
Service Info: OS: Linux; CPE: cpe:/o:linux:kernel

Read data files from: /usr/local/bin/./share/nmap
Service detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.92 seconds
Raw packets sent: 100 (4.400KB) | Rcvd: 202 (8.488KB)
```

FIGURA 35. SALIDA PROCEDIMIENTO 3 - CASO 1

```
Starting Nmap 6.01 ( http://nmap.org ) at 2012-07-10 14:09 CEST
Nmap scan report for pc100-11.empresa.es (169.254.100.11)
Host is up (0.0025s latency).
PORT      STATE SERVICE VERSION
22/tcp    filtered ssh
25/tcp    filtered smtp
80/tcp    open  http     Apache httpd

[ ... ]

Nmap scan report for pc100-20.empresa.es (169.254.100.20)
Host is up (0.0023s latency).
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      Dropbear sshd 0.52 (protocol 2.0)
25/tcp    filtered smtp
80/tcp    open  http     VMware ESXi 4.0 Server httpd
Service Info: Host: pango; OS: Linux; CPE: cpe:/o:linux:kernel

Nmap scan report for www.empresa.es (169.254.100.23)
Host is up (0.035s latency).
rDNS record for 169.254.100.23: pc100-23.empresa.es
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 5.5p1 Debian 6+squeeze2 (protocol 2.0)
25/tcp    filtered smtp
80/tcp    open  http     Apache httpd
Service Info: OS: Linux; CPE: cpe:/o:linux:kernel

[ ... ]

Nmap scan report for pc100-115.empresa.es (169.254.100.115)
Host is up (0.0027s latency).
PORT      STATE SERVICE VERSION
22/tcp    filtered ssh
25/tcp    filtered smtp
80/tcp    open  http     Apache Tomcat/Coyote JSP engine 1.1

Nmap scan report for pc100-116.empresa.es (169.254.100.116)
```

```

Host is up (0.0025s latency).
PORT      STATE      SERVICE VERSION
22/tcp    filtered  ssh
25/tcp    filtered  smtp
80/tcp    open      http      Apache httpd 2.2.3 ((Debian) mod_jk/1.2.18 PHP/5.2.0-8+etch13
mod_ssl/2.2.3 OpenSSL/0.9.8c)

Nmap scan report for pc100-122.empresa.es (169.254.100.122)
Host is up (0.0100s latency).
PORT      STATE      SERVICE VERSION
22/tcp    filtered  ssh
25/tcp    open      smtp      Postfix smtpd
80/tcp    open      http      Apache httpd
Service Info: Host:  dunas.empresa.es

[ ... ]

Nmap scan report for pc100-131.empresa.es (169.254.100.131)
Host is up (0.0033s latency).
PORT      STATE      SERVICE VERSION
22/tcp    filtered  ssh
25/tcp    filtered  smtp
80/tcp    filtered  http

Service detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 256 IP addresses (30 hosts up) scanned in 36.81 seconds

```

FIGURA 36.- SALIDA PROCEDIMIENTO 3 - CASO 2.1

```

Starting Nmap 6.01 ( http://nmap.org ) at 2012-07-10 14:12 CEST
NSE: Loaded 17 scripts for scanning.
Initiating Ping Scan at 14:12
Scanning 256 hosts [4 ports/host]
Completed Ping Scan at 14:12, 2.55s elapsed (256 total hosts)
Initiating Parallel DNS resolution of 256 hosts. at 14:12
Completed Parallel DNS resolution of 256 hosts. at 14:12, 2.50s elapsed
Nmap scan report for 169.254.254.0 [host down]
Nmap scan report for 169.254.254.4 [host down]

[ ... ]

Nmap scan report for 169.254.254.251 [host down]
Nmap scan report for 169.254.254.255 [host down]
Initiating SYN Stealth Scan at 14:12
Scanning 206 hosts [3 ports/host]
Discovered open port 22/tcp on 169.254.254.10
Discovered open port 22/tcp on 169.254.254.7

[ ... ]

Discovered open port 80/tcp on 169.254.254.238
Discovered open port 80/tcp on 169.254.254.247
Completed SYN Stealth Scan at 14:12, 7.07s elapsed (618 total ports)
Initiating Service scan at 14:12
Scanning 314 services on 206 hosts
Service scan Timing: About 49.36% done; ETC: 14:13 (0:00:32 remaining)
Completed Service scan at 14:13, 61.97s elapsed (314 services on 206 hosts)
NSE: Script scanning 206 hosts.
Initiating NSE at 14:13
Completed NSE at 14:13, 5.38s elapsed
Nmap scan report for gateway.hosting.com (169.254.254.1)
Host is up (0.19s latency).
PORT      STATE      SERVICE VERSION
22/tcp    closed    ssh
25/tcp    filtered  smtp
80/tcp    closed    http

[ ... ]

Nmap scan report for tank.company.com (169.254.254.6)
Host is up (0.18s latency).
PORT      STATE      SERVICE VERSION
22/tcp    open      ssh      OpenSSH 5.9p1 Debian 5ubuntu1 (protocol 2.0)
25/tcp    filtered  smtp
80/tcp    open      http     nginx 1.1.19
Service Info: OS: Linux; CPE: cpe:/o:linux:kernel

```



```

Nmap scan report for 7.members.hosting.com (169.254.254.7)
Host is up (0.18s latency).
PORT      STATE      SERVICE  VERSION
22/tcp    open      ssh      OpenSSH 4.3 (protocol 2.0)
25/tcp    filtered  smtp
80/tcp    open      http     Apache httpd 2.2.3 ((CentOS))

Nmap scan report for newspaper.net (169.254.254.8)
Host is up (0.18s latency).
PORT      STATE      SERVICE  VERSION
22/tcp    closed    ssh
25/tcp    filtered  smtp
80/tcp    open      http     Apache httpd 2.2.16 ((Debian))

Nmap scan report for 9.members.hosting.com (169.254.254.9)
Host is up (0.18s latency).
PORT      STATE      SERVICE  VERSION
22/tcp    open      ssh      OpenSSH 5.5p1 Debian 6+squeeze2 (protocol 2.0)
25/tcp    filtered  smtp
80/tcp    open      http     Apache httpd 2.2.16 ((Debian))
Service Info: OS: Linux; CPE: cpe:/o:linux:kernel

Nmap scan report for mail.newspaper.com (169.254.254.10)
Host is up (0.18s latency).
PORT      STATE      SERVICE  VERSION
22/tcp    open      ssh      OpenSSH 5.3p1 Debian 3ubuntu7 (protocol 2.0)
25/tcp    filtered  smtp
80/tcp    open      http     nginx 1.2.1
Service Info: OS: Linux; CPE: cpe:/o:linux:kernel

[ ... ]

Nmap scan report for 32.members.hosting.com (169.254.254.32)
Host is up (0.19s latency).
PORT      STATE      SERVICE  VERSION
22/tcp    open      ssh      OpenSSH 5.3p1 Debian 3ubuntu7 (protocol 2.0)
25/tcp    filtered  smtp
80/tcp    open      http     nginx 0.7.65
Service Info: OS: Linux; CPE: cpe:/o:linux:kernel

Nmap scan report for 34.members.hosting.com (169.254.254.34)
Host is up (0.18s latency).
PORT      STATE      SERVICE  VERSION
22/tcp    open      ssh      OpenSSH 4.3 (protocol 2.0)
25/tcp    filtered  smtp
80/tcp    open      http?
1 service unrecognized despite returning data. If you know the service/version, please
submit the following fingerprint at http://www.insecure.org/cgi-bin/servicefp-
submit.cgi :
SF-Port80-TCP:V=6.01%I=7%D=7/10%Time=4FFC1C2A%P=i686-pc-linux-gnu%r(GetReq
SF:uest,6A,"HTTP/1.1\x20404\x20Not\x20Found\r\nX-Powered-By:\x20Express\r
SF:\nContent-Type:\x20text/plain\r\nConnection:\x20close\r\n\r\nCannot\x20
SF:GET\x20/")%r(HTTPOptions,81,"HTTP/1.1\x20200\x20OK\r\nX-Powered-By:\x2
SF:0Express\r\nContent-Type:\x20text/html;\x20charset=utf-8\r\nContent-Len
SF:gth:\x200\r\nAllow:\x20\r\nConnection:\x20close\r\n\r\n")%r(FourOhFourR
SF:equest,8D,"HTTP/1.1\x20404\x20Not\x20Found\r\nX-Powered-By:\x20Express
SF:\r\nContent-Type:\x20text/plain\r\nConnection:\x20close\r\n\r\nCannot\x
SF:20GET\x20/nice%20ports%2C/Tri%6Eity\.txt%2ebak");

Nmap scan report for www.studio.com (169.254.254.36)
Host is up (0.18s latency).
PORT      STATE      SERVICE  VERSION
22/tcp    open      tcpwrapped
25/tcp    filtered  smtp
80/tcp    open      http     Apache httpd 2.2.14 ((Ubuntu))

[ ... ]

Nmap scan report for 254.members.hosting.com (169.254.254.254)
Host is up (0.18s latency).
PORT      STATE      SERVICE  VERSION
22/tcp    open      ssh      OpenSSH 5.5p1 Debian 6+squeezel (protocol 2.0)
25/tcp    filtered  smtp
80/tcp    open      http     Apache httpd 2.2.16 ((Debian))
Service Info: OS: Linux; CPE: cpe:/o:linux:kernel

```

```

Read data files from: /usr/local/bin/./share/nmap
Service detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 256 IP addresses (206 hosts up) scanned in 80.36 seconds
Raw packets sent: 1607 (64.732KB) | Rcvd: 754 (50.248KB)

```

FIGURA 37.- SALIDA PROCEDIMIENTO 3 - CASO 2.2

```

Starting Nmap 6.01 ( http://nmap.org ) at 2012-07-10 17:24 CEST
NSE: Loaded 17 scripts for scanning.
Initiating ARP Ping Scan at 17:24
Scanning 192.168.1.1 [1 port]
Completed ARP Ping Scan at 17:24, 0.13s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 17:24
Completed Parallel DNS resolution of 1 host. at 17:24, 0.05s elapsed
Initiating UDP Scan at 17:24
Scanning 192.168.1.1 [3 ports]
Discovered open port 53/udp on 192.168.1.1
Completed UDP Scan at 17:24, 1.15s elapsed (3 total ports)
Initiating Service scan at 17:24
Scanning 1 service on 192.168.1.1
Completed Service scan at 17:24, 1.95s elapsed (1 service on 1 host)
NSE: Script scanning 192.168.1.1.
Nmap scan report for 192.168.1.1
Host is up (0.0062s latency).
PORT      STATE SERVICE VERSION
53/udp    open  domain  ISC BIND (Fake version: Nominum Vantio 5.3.0.0)
68/udp    closed dhcpc
80/udp    closed http
MAC Address: AA:BB:CC:DD:EE:FF (Shenzhen Gongjin Electronics Co.)

Read data files from: /usr/local/bin/./share/nmap
Service detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.23 seconds
Raw packets sent: 5 (164B) | Rcvd: 4 (180B)

```

FIGURA 38.- SALIDA PROCEDIMIENTO 3 - CASO 3

```

Starting Nmap 6.01 ( http://nmap.org ) at 2012-07-13 14:22 CEST
NSE: Loaded 17 scripts for scanning.
Initiating ARP Ping Scan at 14:22
Scanning 65 hosts [1 port/host]
Completed ARP Ping Scan at 14:23, 0.44s elapsed (65 total hosts)
Initiating Parallel DNS resolution of 65 hosts. at 14:23
Completed Parallel DNS resolution of 65 hosts. at 14:23, 0.00s elapsed
Nmap scan report for 172.16.28.0 [host down]

[ ... ]

Nmap scan report for 172.16.28.64 [host down]
Initiating Parallel DNS resolution of 1 host. at 14:23
Completed Parallel DNS resolution of 1 host. at 14:23, 0.00s elapsed
Initiating SYN Stealth Scan at 14:23
Scanning 9 hosts [100 ports/host]
Discovered open port 443/tcp on 172.16.28.1

[ ... ]

Completed SYN Stealth Scan at 14:23, 2.27s elapsed (900 total ports)
Initiating Service scan at 14:23
Scanning 25 services on 9 hosts
Completed Service scan at 14:24, 73.56s elapsed (25 services on 9 hosts)
Initiating OS detection (try #1) against 9 hosts
NSE: Script scanning 9 hosts.
Nmap scan report for 172.16.28.1
Host is up (0.00020s latency).
Not shown: 97 filtered ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 5.1p1 (FreeBSD 20080901; protocol 2.0)
53/tcp    open  domain   dnsmasq 2.45
443/tcp   open  ssl/http lighttpd 1.4.23
MAC Address: AA:BB:CC:DD:EE:FF (USC Information Sciences Inst)
Warning: OSScan results may be unreliable because we could not find at least 1 open
and 1 closed port
Device type: specialized
Running: Comau embedded

```

```
OS details: Comau C4G robot control unit
Uptime guess: 0.000 days (since Fri Jul 13 14:24:15 2012)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=256 (Good luck!)
IP ID Sequence Generation: Randomized
Service Info: OS: FreeBSD; CPE: cpe:/o:freebsd:freebsd

Nmap scan report for 172.16.28.4
Host is up (0.00040s latency).
Not shown: 98 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      Mocana embedded SSH (protocol 2.0)
80/tcp    open  http     eHTTP 2.0 (HP 5406zl switch http config)
MAC Address: AA:BB:CC:DD:EE:FF (Hewlett-Packard Company)
Device type: switch
Running: HP embedded
OS details: HP ProCurve 5406zl switch
Uptime guess: 42.403 days (since Fri Jun  1 04:43:21 2012)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=130 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: Device: switch; CPE: cpe:/h:hp:5406zl

Nmap scan report for 172.16.28.11
Host is up (0.00014s latency).
Not shown: 97 filtered ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 5.1p1 (FreeBSD 20080901; protocol 2.0)
53/tcp    open  domain   dnsmasq 2.45
443/tcp   open  ssl/http lighttpd 1.4.23
MAC Address: AA:BB:CC:DD:EE:FF (D-Link)
Warning: OSScan results may be unreliable because we could not find at least 1 open
and 1 closed port
Device type: specialized
Running: Comau embedded
OS details: Comau C4G robot control unit
Uptime guess: 0.000 days (since Fri Jul 13 14:24:13 2012)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=264 (Good luck!)
IP ID Sequence Generation: Randomized
Service Info: OS: FreeBSD; CPE: cpe:/o:freebsd:freebsd

Nmap scan report for 172.16.28.13
Host is up (0.00018s latency).
Not shown: 97 filtered ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 5.1p1 (FreeBSD 20080901; protocol 2.0)
53/tcp    open  domain   dnsmasq 2.45
443/tcp   open  ssl/http lighttpd 1.4.23
MAC Address: AA:BB:CC:DD:EE:FF (D-Link)
Warning: OSScan results may be unreliable because we could not find at least 1 open
and 1 closed port
Device type: specialized
Running: Comau embedded
OS details: Comau C4G robot control unit
Uptime guess: 0.000 days (since Fri Jul 13 14:24:16 2012)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=258 (Good luck!)
IP ID Sequence Generation: Randomized
Service Info: OS: FreeBSD; CPE: cpe:/o:freebsd:freebsd

Nmap scan report for dejavu (172.16.28.45)
Host is up (0.00045s latency).
Not shown: 96 closed ports
PORT      STATE SERVICE VERSION
135/tcp   open  msrpc    Microsoft Windows RPC
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds Microsoft Windows XP microsoft-ds
3389/tcp  open  ms-wbt-server Microsoft Terminal Service
MAC Address: AA:BB:CC:DD:EE:FF (Cadmus Computer Systems)
Device type: general purpose
Running: Microsoft Windows XP
OS CPE: cpe:/o:microsoft:windows_xp::sp2 cpe:/o:microsoft:windows_xp::sp3
OS details: Microsoft Windows XP SP2 or SP3
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=261 (Good luck!)
IP ID Sequence Generation: Incremental
```

```

Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 172.16.28.47
Host is up (0.0012s latency).
Not shown: 98 filtered ports
PORT      STATE SERVICE VERSION
80/tcp    closed http
443/tcp   closed https
MAC Address: AA:BB:CC:DD:EE:FF (Cadmus Computer Systems)
Device type: general purpose|media device
Running: Microsoft Windows 2000|XP|2003|98|NT, Motorola Windows PocketPC/CE
OS CPE: cpe:/o:microsoft:windows_2000 cpe:/o:microsoft:windows_xp::sp2
cpe:/o:microsoft:windows_xp::sp3 cpe:/o:microsoft:windows_server_2003
cpe:/o:microsoft:windows_98 cpe:/o:microsoft:windows_nt::sp6
cpe:/o:motorola:windows_ce
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

Nmap scan report for 172.16.28.49
Host is up (0.00049s latency).
Not shown: 96 closed ports
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds Microsoft Windows XP microsoft-ds
8008/tcp  open  printer
MAC Address: AA:BB:CC:DD:EE:FF (Cadmus Computer Systems)
Device type: general purpose
Running: Microsoft Windows XP
OS CPE: cpe:/o:microsoft:windows_xp::sp2 cpe:/o:microsoft:windows_xp::sp3
OS details: Microsoft Windows XP SP2 or SP3
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=256 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 172.16.28.50
Host is up (0.00059s latency).
Not shown: 97 closed ports
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds Microsoft Windows XP microsoft-ds
MAC Address: AA:BB:CC:DD:EE:FF (Cadmus Computer Systems)
Device type: general purpose
Running: Microsoft Windows XP
OS CPE: cpe:/o:microsoft:windows_xp::sp2 cpe:/o:microsoft:windows_xp::sp3
OS details: Microsoft Windows XP SP2 or SP3
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=263 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 172.16.28.51
Host is up (0.00051s latency).
Not shown: 97 closed ports
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds Microsoft Windows XP microsoft-ds
MAC Address: AA:BB:CC:DD:EE:FF (Cadmus Computer Systems)
Device type: general purpose
Running: Microsoft Windows XP
OS CPE: cpe:/o:microsoft:windows_xp::sp2 cpe:/o:microsoft:windows_xp::sp3
OS details: Microsoft Windows XP SP2 or SP3
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=255 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Initiating ARP Ping Scan at 14:24
Scanning 62 hosts [1 port/host]
Completed ARP Ping Scan at 14:24, 0.64s elapsed (62 total hosts)
Initiating Parallel DNS resolution of 62 hosts. at 14:24
Completed Parallel DNS resolution of 62 hosts. at 14:24, 0.00s elapsed
Initiating SYN Stealth Scan at 14:24
Scanning 172.16.28.65 [100 ports]

```

```
Discovered open port 23/tcp on 172.16.28.65
Discovered open port 80/tcp on 172.16.28.65
Completed SYN Stealth Scan at 14:24, 0.03s elapsed (100 total ports)
Initiating Service scan at 14:24
Scanning 2 services on 172.16.28.65
Completed Service scan at 14:24, 6.00s elapsed (2 services on 1 host)
Initiating OS detection (try #1) against 172.16.28.65
NSE: Script scanning 172.16.28.65.
Nmap scan report for 172.16.28.65
Host is up (0.000038s latency).
Not shown: 98 closed ports
PORT      STATE SERVICE VERSION
23/tcp    open  ssh      OpenSSH 5.3p1 Debian 3ubuntu7 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.2.14 ((Ubuntu))
Device type: general purpose
Running: Linux 2.6.X|3.X
OS CPE: cpe:/o:linux:kernel:2.6 cpe:/o:linux:kernel:3
OS details: Linux 2.6.32 - 3.2
Uptime guess: 0.260 days (since Fri Jul 13 08:10:47 2012)
Network Distance: 0 hops
TCP Sequence Prediction: Difficulty=263 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OS: Linux; CPE: cpe:/o:linux:kernel

Nmap scan report for 172.16.28.68 [host down]

[ ... ]

Nmap scan report for 172.16.28.127 [host down]
Initiating SYN Stealth Scan at 14:24
Scanning 11 hosts [100 ports/host]
Discovered open port 443/tcp on 172.16.28.82

[ ... ]

Completed SYN Stealth Scan at 14:24, 2.04s elapsed (1100 total ports)
Initiating Service scan at 14:24
Scanning 30 services on 11 hosts
Stats: 0:01:50 elapsed; 117 hosts completed (21 up), 11 undergoing Service Scan
Service scan Timing: About 96.97% done; ETC: 14:24 (0:00:01 remaining)
Completed Service scan at 14:24, 18.32s elapsed (33 services on 11 hosts)
Initiating RPCGrind Scan against 172.16.28.71 at 14:24
Completed RPCGrind Scan against 172.16.28.71 at 14:24, 0.00s elapsed (1 port)
Initiating OS detection (try #1) against 11 hosts
Retrying OS detection (try #2) against 3 hosts
NSE: Script scanning 11 hosts.
Nmap scan report for 172.16.28.66
Host is up (0.00014s latency).
All 100 scanned ports on 172.16.28.66 are filtered
MAC Address: AA:BB:CC:DD:EE:FF (Hewlett-Packard Company)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

Nmap scan report for 172.16.28.67
Host is up (0.00023s latency).
Not shown: 99 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 5.8p1 Debian 7ubuntu1 (protocol 2.0)
MAC Address: AA:BB:CC:DD:EE:FF (Hewlett-Packard Company)
Device type: general purpose
Running: Linux 2.6.X|3.X
OS CPE: cpe:/o:linux:kernel:2.6 cpe:/o:linux:kernel:3
OS details: Linux 2.6.38 - 3.2
Uptime guess: 1.262 days (since Thu Jul 12 08:07:16 2012)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=264 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OS: Linux; CPE: cpe:/o:linux:kernel

Nmap scan report for 172.16.28.70
Host is up (0.00015s latency).
All 100 scanned ports on 172.16.28.70 are closed
MAC Address: AA:BB:CC:DD:EE:FF (Hewlett-Packard Company)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

Nmap scan report for 172.16.28.71
```

```
Host is up (0.00017s latency).
Not shown: 98 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 6.0p1 Debian 2 (protocol 2.0)
111/tcp   open  rpcbind
MAC Address: AA:BB:CC:DD:EE:FF (Hewlett-Packard Company)
Device type: general purpose
Running: Linux 2.6.X|3.X
OS CPE: cpe:/o:linux:kernel:2.6 cpe:/o:linux:kernel:3
OS details: Linux 2.6.38 - 3.2
Uptime guess: 0.273 days (since Fri Jul 13 07:51:53 2012)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=261 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OS: Linux; CPE: cpe:/o:linux:kernel

Nmap scan report for 172.16.28.74
Host is up (0.00013s latency).
All 100 scanned ports on 172.16.28.74 are closed
MAC Address: AA:BB:CC:DD:EE:FF (Hewlett-Packard Company)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

Nmap scan report for 172.16.28.81
Host is up (0.00039s latency).
Not shown: 99 filtered ports
PORT      STATE SERVICE VERSION
135/tcp   open  msrpc    Microsoft Windows RPC
MAC Address: AA:BB:CC:DD:EE:FF (Hewlett Packard)
Warning: OSScan results may be unreliable because we could not find at least 1 open
and 1 closed port
Device type: general purpose
Running: Microsoft Windows 7
OS CPE: cpe:/o:microsoft:windows_7
OS details: Microsoft Windows 7
Uptime guess: 0.262 days (since Fri Jul 13 08:07:59 2012)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=262 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 172.16.28.82
Host is up (0.00015s latency).
Not shown: 93 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 6.0p1 Debian 2 (protocol 2.0)
80/tcp    open  http         Apache httpd 2.2.22 ((Debian))
113/tcp   open  ident
139/tcp   open  netbios-ssn Samba smbd 3.X (workgroup: WORKGROUP)
443/tcp   open  ssl/http     Apache httpd 2.2.22 ((Debian))
445/tcp   open  netbios-ssn Samba smbd 3.X (workgroup: WORKGROUP)
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
MAC Address: AA:BB:CC:DD:EE:FF (Hewlett-Packard Company)
Device type: general purpose
Running: Linux 2.6.X|3.X
OS CPE: cpe:/o:linux:kernel:2.6 cpe:/o:linux:kernel:3
OS details: Linux 2.6.38 - 3.2
Uptime guess: 0.260 days (since Fri Jul 13 08:10:45 2012)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=261 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OS: Linux; CPE: cpe:/o:linux:kernel

Nmap scan report for 172.16.28.84
Host is up (0.00012s latency).
Not shown: 99 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 5.3p1 Debian 3ubuntu7 (protocol 2.0)
MAC Address: AA:BB:CC:DD:EE:FF (Hewlett-Packard Company)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:kernel:2.6
OS details: Linux 2.6.32 - 2.6.35
Uptime guess: 1.088 days (since Thu Jul 12 12:18:54 2012)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=253 (Good luck!)
IP ID Sequence Generation: All zeros
```

```

Service Info: OS: Linux; CPE: cpe:/o:linux:kernel

Nmap scan report for 172.16.28.124
Host is up (0.00023s latency).
Not shown: 93 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          HP LaserJet P4014 printer ftpd
23/tcp    open  telnet       HP JetDirect telnetd
80/tcp    open  http         HP-ChaiSOE 1.0 (HP LaserJet http config)
443/tcp   open  ssl/http     HP-ChaiSOE 1.0 (HP LaserJet http config)
515/tcp   open  printer
631/tcp   open  http         HP-ChaiSOE 1.0 (HP LaserJet http config)
9100/tcp  open  jetdirect?
MAC Address: AA:BB:CC:DD:EE:FF (Hewlett-Packard Company)
Device type: printer
Running: HP embedded
OS details: HP LaserJet 2055dn, 2420, P3005, CP4005, 4250, or P4014 printer
Uptime guess: 11.041 days (since Mon Jul  2 13:25:10 2012)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=263 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: Device: printer

Nmap scan report for 172.16.28.125
Host is up (0.00049s latency).
Not shown: 93 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          HP JetDirect ftpd
23/tcp    open  telnet       HP JetDirect printer telnetd
80/tcp    open  http         HP-ChaiSOE 1.0 (HP LaserJet http config)
443/tcp   open  ssl/http     HP-ChaiSOE 1.0 (HP LaserJet http config)
515/tcp   open  printer
631/tcp   open  http         HP-ChaiSOE 1.0 (HP LaserJet http config)
9100/tcp  open  jetdirect?
MAC Address: AA:BB:CC:DD:EE:FF (Hewlett-Packard Company)
Device type: printer
Running: HP embedded
OS details: HP LaserJet 3800, 4250, 4345, 9040 printer
Uptime guess: 10.798 days (since Mon Jul  2 19:15:40 2012)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=25 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: Devices: print server, printer

Nmap scan report for 172.16.28.126
Host is up (0.00031s latency).
Not shown: 93 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          HP FTP Print Server 3.0 (HP LaserJet 4250 printer)
23/tcp    open  telnet       HP JetDirect telnetd
80/tcp    open  http         HP-ChaiSOE 1.0 (HP LaserJet http config)
443/tcp   open  ssl/http     HP-ChaiSOE 1.0 (HP LaserJet http config)
515/tcp   open  printer
631/tcp   open  http         HP-ChaiSOE 1.0 (HP LaserJet http config)
9100/tcp  open  jetdirect?
MAC Address: AA:BB:CC:DD:EE:FF (Hewlett-Packard Company)
Device type: printer
Running: HP embedded
OS details: HP LaserJet 2055dn, 2420, P3005, CP4005, 4250, or P4014 printer
Uptime guess: 10.200 days (since Tue Jul  3 09:37:08 2012)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=255 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: Device: printer

Read data files from: /usr/local/bin/./share/nmap
OS and Service detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 128 IP addresses (21 hosts up) scanned in 116.84 seconds
Raw packets sent: 3396 (171.570KB) | Rcvd: 1922 (89.180KB)

```

FIGURA 39.- SALIDA PROCEDIMIENTO 3 - CASO 4

227. En la Tabla 3 se recoge una comparativa cuantitativa de los resultados de los casos anteriores. Las técnicas y modificadores utilizados en este Procedimiento afectan

muy notablemente al tiempo de ejecución de las pruebas, siendo elevados en general. A pesar de ello la información que es posible obtener gracias a Nmap es, como puede verse, especialmente detallada y valiosa. La interpretación de estos resultados se realiza en el Anexo A (ver 5.9).

CASO	Dirs/puertos analizados	Información de versión y sistema operativo	Paquetes env./rec./seg. ¹⁹	puertos/s
1	1/100	3389/tcp ms-wbt-server Microsoft Terminal Service	200/2 /8,11s	12,33
2.1	30 de 256/ 30x3 = 90	22/tcp ssh OpenSSH 5.5: 1 Dropbear: 1 25/tcp smtp Postfix smtpd: 1 80/tcp http Apache httpd: 10 Apache httpd 2.2.x: 2 Apache Tomcat: 1 Vmware ESXi Server httpd: 1	1100/88 /29,7s	3,03
2.2	206 de 256/ 206x3 = 618	22/tcp ssh OpenSSH · 4.x: 22 · 5.x: 122 · 6.x: 1 80/tcp http nginx: 47 Apache httpd: 14 Apache http 2.2.x CentOS: 12 Apache http 2.2.x Debian: 17 Apache http 2.2.x Ubuntu: 62 otros: 9	1.607/754 /80,36s	7,69
3	1/3(UDP)	53/udp domain ISC BIND (Fake version: Nominum Vantio 5.3.0.0)	5/4/12,23s	0,245
4	21 de 128/ 21x100 = 2100	3 Impresoras HP. 1 Switch HP. 5 Equipos Linux >= 2.6.32. 6 Equipos Windows 3 Equipo FreeBSD	3396/1922 /116,84s	17,97

Tabla 3. Comparativa resultados Procedimiento 3

4.4. TÉCNICAS DE EVASIÓN DE CORTAFUEGOS Y HERRAMIENTAS IDS/IPS

228. Este Procedimiento aborda un conjunto de modificadores avanzados, compatibles con las técnicas vistas en los anteriores, que pueden hacer que, en algunos casos, las sondas que envía Nmap traspasen cortafuegos, e incluso puedan no ser detectadas por herramientas de detección y prevención de intrusiones IDS/IPS.

229. A principio de los 90, se empezó a generalizar el uso de herramientas cortafuegos con el expreso propósito de reducir la conectividad. De este modo, las grandes redes corporativas fueron aisladas de Internet mediante Proxies, sistemas de traducción de direcciones de red (NAT) y la implantación de filtros de paquetes.

230. Este tipo de sistemas como las herramientas cortafuegos, pueden convertir la tarea de descubrimiento de una red en algo sumamente difícil. Nmap ofrece numerosas características destinadas a la comprensión de estas redes complejas y a la verificación de que estos filtros están funcionando como es de esperar. Nmap ofrece también mecanismos para traspasar defensas pobremente implementadas. Uno de los mejores métodos para conocer la seguridad de una red es tratar de

¹⁹ Esta contabilización no tiene en cuenta los paquetes enviados por la detección de versiones.

hacer fracasar sus controles, adoptando el rol de un atacante que utilice las funcionalidades descritas en este Procedimiento.

231. De forma adicional al establecimiento de restricciones a la actividad en una red, se ha ido paulatinamente extendiendo el uso de técnicas de monitorización de tráfico, mediante sistemas de detección de intrusiones. La mayor parte de los sistemas IDS incorporan reglas para la detección de análisis con Nmap, porque son en ocasiones una actividad precursora de un ataque. Muchas de estas herramientas han evolucionado en Sistemas de Prevención de Intrusiones (IPS), debido a que son capaces de bloquear activamente aquel tráfico que se considere malicioso. Desafortunadamente para administradores de redes y suministradores de estas herramientas, la detección fiable de actividades peligrosas mediante el análisis de tráfico de red es un problema muy complejo. Atacantes con la suficiente paciencia, habilidad y ayuda de ciertas funcionalidades de Nmap podrían ser capaces de traspasar herramientas IDS/IPS sin ser detectados. Mientras tanto, los administradores que utilicen estas herramientas deberán afrontar el coste de analizar un elevado número de falsos positivos provocados por actividades normales que hayan sido mal diagnosticadas.
232. A la vista de este tipo de funcionalidades implementadas en Nmap, la gente podría pensar que no es adecuado ofrecer mecanismos dirigidos a la evasión de cortafuegos y herramientas IDS/IPS. Se podría argumentar que son útiles tanto a administradores ocupados en aumentar la seguridad como a atacantes. Independientemente de la existencia de estas opciones en Nmap, los atacantes podrían elaborar fácilmente sus propios métodos para el mismo fin, mientras que los administradores tendrían que hacer su trabajo de una manera mucho más dura.

4.4.1. DESCRIPCIÓN DEL PROCEDIMIENTO

233. El anterior procedimiento mostraba cómo es posible obtener más cantidad y más precisa información de los objetivos escaneados utilizando unos modificadores especiales los cuales producían un envío elevado de sondas, convirtiendo los análisis en prácticas muy poco sigilosas. En este Procedimiento, la preocupación es cómo conseguir que las acciones de Nmap pasen inadvertidas a cortafuegos y herramientas IDS/IPS, probando de este modo su eficacia.
234. Llegar a ese nivel de sigilo puede ser un objetivo inalcanzable en muchos casos, puesto que las herramientas de filtrado y monitorización de tráfico son cada vez más sofisticadas, y sus administradores están más informados y preparados para la defensa de sus sistemas.
235. En cualquier caso, como se podrá ver en uno de los casos expuestos en este Procedimiento, si no se puede ser lo suficientemente sigiloso, siempre se podrá ser todo lo contrario, es decir, generar un volumen tal de falsos positivos que nuestra acción pase igualmente inadvertida.

Aplicaciones para Administradores

Las funcionalidades de *Nmap* expuestas en este Procedimiento, podrían servir a atacantes para múltiples fines. En cambio, desde el punto de vista de su uso administrativo se pueden identificar las siguientes:

- Conocer el funcionamiento de técnicas muy sofisticadas y sigilosas de escaneo de puertos, utilizadas frecuentemente por atacantes remotos de sistemas.
- Ayuda a la auditoría e implementación óptima de herramientas cortafuegos e IDS/IPS.
- Descubrimiento de relaciones de confianza IP entre máquinas.
- Escaneo de una red desde el punto de vista de una máquina distinta al origen. Identificación de máquinas susceptibles de ser utilizadas como *zombie* en un *Idle Scan*.

Objetivos analizados

Se han elegido varios conjuntos de objetivos para proporcionar suficientes casos representativos a este Procedimiento. Los objetivos son los siguientes:

- **dejavu:** máquina Windows XP SP3 perteneciente a la red local (*FastEthernet*) del origen.
- **Impresora:** Impresora de red que se utilizará como equipo Zombie desde el que lanzar el análisis de tipo Idle.
- **www.empresa.es/24:** máquinas pertenecientes a una red de una empresa española.

4.4.2. CONFIGURACIÓN DE NMAP

236. A continuación se describen dos casos específicamente configurados para cubrir las aplicaciones para administradores para este Procedimiento, desde enfoques bien distintos: el primero trata de ser muy sigiloso pasando inadvertido por cortafuegos y sistemas IDS/IPS. El segundo trata de generar muchos falsos positivos haciendo ineficaces las alertas IDS. El tercer caso podría generar positivos por actividad de análisis en sistemas IDS, pero el origen real no podrá ser identificado de este modo.

237. A continuación se incluyen los casos de prueba, así como comandos de ejecución de los mismos.

- **CASO 1: evasión sigilosa.** Este caso presenta un escaneo de puertos a un conjunto de objetivos, similar al CASO5 del Procedimiento2, pero de manera relativamente más sigilosa:

```
nmap -sS -Pn -p 22,80 -f --data-length 99 --randomize-host -g 22 -T 1
-v www.empresa.es/30
```

- **CASO 2: evasión mediante falsos positivos.** Este caso se realiza un pequeño escaneo de puertos contra un solo objetivo, pero realizado aparentemente por varias máquinas. El uso de señuelos genera muchos falsos positivos, y de este modo oculta al verdadero origen del análisis.

```
nmap -sS -p 135-139,445,3389 -Pn -D
172.16.28.100,172.16.28.101,ME,172.16.28.102 -v dejavu
```

- **CASO 3: evasión mediante ocultación del origen.** Este caso realiza un sofisticado escaneo mediante una técnica indirecta llamada Idle Scan (ver 3.2.4). Se vale de un intermediario, llamado *Zombie*, que aparecerá como presunto origen del análisis en los sistemas IDS/IPS. Ningún paquete es enviado al objetivo utilizando la dirección IP del origen real.

```
nmap -sI impresora -p 23-25 -P0 -v dejavu
```

4.4.3. RESULTADOS

```
Starting Nmap 6.01 ( http://nmap.org ) at 2012-07-10 18:06 CEST
Initiating Parallel DNS resolution of 4 hosts. at 18:06
Completed Parallel DNS resolution of 4 hosts. at 18:06, 0.00s elapsed
Initiating SYN Stealth Scan at 18:06
Scanning 4 hosts [2 ports/host]
SYN Stealth Scan Timing: About 12.50% done; ETC: 18:10 (0:03:37 remaining)
SYN Stealth Scan Timing: About 25.00% done; ETC: 18:10 (0:03:03 remaining)
SYN Stealth Scan Timing: About 37.50% done; ETC: 18:10 (0:02:32 remaining)
SYN Stealth Scan Timing: About 50.00% done; ETC: 18:10 (0:02:01 remaining)
SYN Stealth Scan Timing: About 62.50% done; ETC: 18:10 (0:01:31 remaining)
SYN Stealth Scan Timing: About 75.00% done; ETC: 18:10 (0:01:00 remaining)
SYN Stealth Scan Timing: About 87.50% done; ETC: 18:10 (0:00:30 remaining)
Completed SYN Stealth Scan at 18:10, 255.06s elapsed (8 total ports)
Nmap scan report for pc100-20.empresa.es (169.254.100.20)
Host is up.
PORT      STATE      SERVICE
22/tcp    filtered  ssh
80/tcp    filtered  http

Nmap scan report for pc100-22.empresa.es (169.254.100.22)
Host is up.
PORT      STATE      SERVICE
22/tcp    filtered  ssh
80/tcp    filtered  http

Nmap scan report for pc100-21.empresa.es (169.254.100.21)
Host is up.
PORT      STATE      SERVICE
22/tcp    filtered  ssh
80/tcp    filtered  http

Nmap scan report for www.empresa.es (169.254.100.23)
Host is up.
rDNS record for 169.254.100.23: pc100-23.empresa.es
PORT      STATE      SERVICE
22/tcp    filtered  ssh
80/tcp    filtered  http

Read data files from: /usr/local/bin/./share/nmap
Nmap done: 4 IP addresses (4 hosts up) scanned in 255.18 seconds
Raw packets sent: 16 (2.288KB) | Rcvd: 0 (0B)
```

FIGURA 40. SALIDA PROCEDIMIENTO 4 - CASO 1

```
nmap -sS -p 135-139,445,3389 -Pn -D
172.16.28.100,172.16.28.101,172.16.28.102,ME,172.16.28.103 -v dejavu

Starting Nmap 6.01 ( http://nmap.org ) at 2012-07-10 18:24 CEST
Initiating ARP Ping Scan at 18:24
Scanning dejavu (172.16.28.45) [1 port]
Completed ARP Ping Scan at 18:24, 0.03s elapsed (1 total hosts)
```

```

Initiating SYN Stealth Scan at 18:24
Scanning dejavu (172.16.28.45) [7 ports]
Discovered open port 3389/tcp on 172.16.28.45
Completed SYN Stealth Scan at 18:24, 1.24s elapsed (7 total ports)
Nmap scan report for dejavu (172.16.28.45)
Host is up (0.00081s latency).
PORT      STATE      SERVICE
135/tcp   filtered  msrpc
136/tcp   filtered  profile
137/tcp   filtered  netbios-ns
138/tcp   filtered  netbios-dgm
139/tcp   filtered  netbios-ssn
445/tcp   filtered  microsoft-ds
3389/tcp  open      ms-wbt-server
MAC Address: AA:BB:CC:DD:EE:FF (Cadmus Computer Systems)

Read data files from: /usr/local/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.42 seconds
Raw packets sent: 66 (2.888KB) | Rcvd: 2 (72B)

```

FIGURA 41. SALIDA PROCEDIMIENTO 4 - CASO 2

```

nmap -sI 172.16.28.124 -F -Pn -v dejavu

Starting Nmap 6.01 ( http://nmap.org ) at 2012-07-10 18:21 CEST
Initiating ARP Ping Scan at 18:21
Scanning dejavu (172.16.28.45) [1 port]
Completed ARP Ping Scan at 18:21, 0.03s elapsed (1 total hosts)
Initiating idle scan against dejavu (172.16.28.45) at 18:21
Idle scan using zombie 172.16.28.124 (172.16.28.124:80); Class: Incremental
Discovered open port 3389/tcp on 172.16.28.45
Completed idle scan against dejavu (172.16.28.45) at 18:21, 2.16s elapsed (100 ports)
Nmap scan report for dejavu (172.16.28.45)
Host is up (0.028s latency).
Not shown: 99 closed|filtered ports
PORT      STATE      SERVICE
3389/tcp  open      ms-wbt-server
MAC Address: AA:BB:CC:DD:EE:FF (Cadmus Computer Systems)

Read data files from: /usr/local/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 2.32 seconds
Raw packets sent: 274 (12.040KB) | Rcvd: 41 (1.628KB)

```

FIGURA 42. SALIDA PROCEDIMIENTO 4 - CASO 3

238. La Tabla 4 recoge una comparativa cuantitativa de los resultados de los casos anteriores. Cabe destacar el tiempo invertido en el Caso1, requerido en general si se desea pasar inadvertido a través de muchas herramientas de detección y prevención de intrusiones (IDS/IPS).

COMP.	Direcciones Analizadas	Puertos Analizados	Estado puertos Abiertos/Cerrados/Filtrados	Paquetes enviados	Paquetes recibidos	Tiempo empleado
CASO1	4	8	0/0/8	16	0	255,18s
CASO2	1	7	1/0/6	66	2	1,42s
CASO3	1	100	1/0/99	274	41	2,32s

Tabla 4. Comparativa resultados Procedimiento 4.

4.5. TÉCNICAS DE OPTIMIZACIÓN DEL ANÁLISIS

239. Los procedimientos anteriores mostraban técnicas de descubrimientos de equipos y servicios, centrándose en los diferentes métodos disponibles. Se aplicaban varios posibles casos en diferentes escenarios. Éstos están especialmente pensados para cubrir las situaciones más cotidianas que un Administrador se puede encontrar.
240. Sin embargo, estos entornos heterogéneos tienen unas respuestas en tiempo muy distintas entre sí. No es lo mismo escanear una red local de clase C con 254 posibles IPs que una red remota de clase B con un total de 65534 IPs a explorar. El tiempo de respuesta de cada sonda será muy diferente debido a la distancia entre equipos.

4.5.1. DESCRIPCIÓN DEL PROCEDIMIENTO

241. Este procedimiento mostrará una serie de técnicas que ayudará a determinar los parámetros óptimos que deben usarse para un escaneo a una gran red. Se tratará de este modo optimizar el tiempo de envío de sondas para analizar una red en el menor tiempo posible, mejorando así la eficiencia de la tarea.
242. Nmap dispone, como se ha descrito en la sección 3.3.3, de modificadores y plantillas temporales que permiten fijar de forma personalizada los parámetros de tiempo. Los recordamos aquí:

- *--min-parallelism, --max-parallelism <valor>*
- *--min-hostgroup, --max-hostgroup <valor>*
- *--min-rtt-timeout, --max-rtt-timeout, --initial-rtt-timeout <milisec>*
- *--max-retries <valor>*
- *--host-timeout <milisec>*
- *-T<0-5>*

Aplicaciones para Administradores

Los casos prácticos aquí expuestos puede ser extrapolados para servir de ayuda en las siguientes tareas administrativas para el mantenimiento rutinario de una red:

- Automatización de escaneos periódicos
- Reducción del tiempo de proceso de escaneo

Objetivos analizados

Se han elegido dos conjuntos de objetivos para proporcionar suficientes casos representativos a este Procedimiento. Los objetivos son los siguientes:

- **1.2.3.0/24:** subred remota con rango de 254 posibles ips
- **172.16.152.0/24:** subred local con espacio para 254 equipos

4.5.2. CONFIGURACIÓN DE NMAP

243. La correcta parametrización de Nmap implica conocer qué valores son más apropiados para ser aplicados en un escenario u otro. Por eso, el administrador debe plantearse las siguientes cuestiones:

- Los equipos a escanear, ¿están en una red local, o remota?
- ¿Hay algún hardware intermedio que pueda filtrar tráfico, como cortafuegos o balanceadores?
- ¿Se encuentra la red a escanear saturada?
- ¿Cuál es la latencia entre un equipo remoto y la máquina que va a realizar el escaneo?

244. Un método simple para conocer la latencia de un equipo es averiguar el RTT (Round Trip Time) del tráfico entre las dos máquinas a escanear haciendo un ping a la IP de un equipo remoto:

```
$ ping 1.2.3.4
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
64 bytes from 1.2.3.4: icmp_req=1 ttl=64 time=181.8 ms
64 bytes from 1.2.3.4: icmp_req=2 ttl=64 time=193.0 ms
64 bytes from 1.2.3.4: icmp_req=3 ttl=64 time=189.2 ms
64 bytes from 1.2.3.4: icmp_req=4 ttl=64 time=186.3 ms

--- 1.2.3.4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3998ms
rtt min/avg/max/mdev = 181.8/187.6/193.0/0.018 ms
```

245. Es posible que la prueba anterior no haya dado ningún resultado, bien porque haya un cortafuegos que haya filtrado el tráfico ICMP o porque la máquina tenga deshabilitada la respuesta a la petición “ICMP echo request”. En este caso se puede comprobar el retardo utilizando la herramienta hping3. Para ello será necesario conocer un equipo con un servicio disponible, y realizar un “ping” a ese servicio. En el caso del ejemplo siguiente, se comprueba el retardo en el puerto 80.

```
# hping3 -S -p 80 8.9.10.11
HPING 8.9.10.11 (eth0 8.9.10.11): S set, 40 headers + 0 data bytes
len=46 ip=8.9.10.11 ttl=235 sport=80 flags=SA seq=0 win=1608 rtt=310.6 ms
len=46 ip=8.9.10.11 ttl=236 sport=80 flags=SA seq=1 win=1608 rtt=305.2 ms
len=46 ip=8.9.10.11 ttl=236 sport=80 flags=SA seq=2 win=1608 rtt=299.0 ms
len=46 ip=8.9.10.11 ttl=236 sport=80 flags=SA seq=3 win=1608 rtt=320.9 ms
^C
--- 8.9.10.11 hping statistic ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 299.0/308.9/320.9 ms
```

246. Para comparar la efectividad de la configuración de los parámetros específicos respecto a los valores por defecto, se van a realizar dos escaneos. El primero se mantendrán los valores por defecto, usando únicamente la plantilla de tiempo -T4. En el segundo escaneo se usarán parámetros personalizados conforme a la red que se quiere analizar.

- **CASO 1: subred local.** La red a escanear se trata de 1.2.3.0/24. Las pruebas de latencias descritas en el apartado anterior han dado como resultado 25 ms de rtt medio. Se sospecha que puede haber un cortafuegos filtrando contenido y al no haber encontrado ningún pico de latencia o paquetes perdidos en las pruebas realizadas, no hay indicios de que la red esté saturada.

```
# nmap -T4 172.16.152.0/24

[...]
# Nmap done at Tue Nov 29 14:45:36 2011 256
IP addresses (229 hosts up) scanned in 347.76 seconds

# nmap -T4 -initial-rtt-timeout 30 -max-rtt-timeout 50 -max-retries 2 -min-parallelism 70 172.16.152.0/24

[...]
# Nmap done at Wed Nov 30 11:30:57 2011 256
IP addresses (229 hosts up) scanned in 105.46 seconds
```

247. Se recomienda usar el doble de `max-rtt-timeout` respecto al rtt medio calculado, y una cantidad ligeramente superior al rtt medio para `initial-rtt-timeout`. Si la red no está saturada, `max-retries` configurado a 2 intentos es suficiente.

248. Se puede observar como el escaneo con parámetros personalizados ha tardado un tiempo considerablemente inferior al escaneo por defecto. El resultado de los análisis ha sido omitido por simplicidad, aunque ha de indicarse que fueron idénticos, confirmando que la configuración ganó en velocidad sin perder eficacia.

- **CASO 2: red remota.** En este caso, las pruebas de latencia han dado como resultado 300ms de rtt medio.

```
# nmap -T4 1.2.3.0/24

[...]
# Nmap done at Tue Nov 29 15:48:32 2011 256
IP addresses (41 hosts up) scanned in 3911.43 seconds
```

249. Se sigue usando el mismo criterio de usar un valor ligeramente superior al RTT medio como `initial-rtt-timeout`, y el doble para `max-rtt-timeout`. En caso de una red remota que se encuentre a muchos saltos de la red inicial, y dependiendo de si hay saturación en algún tramo, puede ser conveniente incrementar el valor de `max-retries`.

```
# nmap -T4 -initial-rtt-timeout 350 -max-rtt-timeout 600 -max-retries 4 -min-parallelism 70 1.2.3.0/24

[...]
# Nmap done at Tue Nov 29 18:11:12 2011 256
IP addresses (41 hosts up) scanned in 1546.11 seconds
```

250. La optimización de los valores ha dado como resultado un tiempo de realización del escaneo notablemente inferior al realizado con los valores por defecto.

5. NMAP SCRIPTING ENGINE

5.1. INTRODUCCIÓN

251. *Nmap Scripting Engine* (NSE) es una característica recientemente añadida a las funcionalidades de Nmap, y una de las más potentes. Permite al usuario ejecutar tareas automáticas por medio de scripts, con toda la velocidad y eficacia de la que Nmap dispone. Estos scripts amplían las funcionales originales, pudiendo utilizarse como detector y explotación de vulnerabilidades, detector avanzado de versión de servicios, ataques por fuerza bruta, o denegación de servicio, entre otros muchos. Aunque lo verdaderamente potente es la posibilidad de escribir scripts personalizados a las necesidades puntuales que se requieran. Esta libertad facilita la colaboración de la comunidad, que aporta sus nuevas herramientas, y que muchas de ellas son añadidas a la suite de Nmap.

252. Los scripts están basados en el lenguaje interpretado Lua²⁰. Fue elegido por ser sencillo de utilizar, rápido, escalable, y eficiente. La web oficial dispone de un buen manual de referencia²¹ y una guía de usuario²² que cubre los conocimientos necesarios para desarrollar scripts en NSE.

253. El siguiente código muestra un ejemplo de uso de NSE:

```
# nmap -sC -T4 192.168.100.121

Starting Nmap ( http://nmap.org )
Interesting ports on flog (192.168.100.121):
PORT      STATE SERVICE
22/tcp    open  ssh
|  ssh-hostkey: 1024 b2:37:0e:30:51:dd:14:97:b3:6f:35:3a:0e:9c:1b:39 (DSA)
|_ 2048 78:d1:21:1d:45:10:88:a1:31:ab:86:c0:e9:cb:4d:12 (RSA)
139/tcp    open  netbios-ssn

Host script results:
smb-os-discovery:
| OS: Unix (Samba 3.6.4)
| Computer name: pc01
| Domain name: domain.local
| FQDN: pc01.domain.local
| NetBIOS computer name:
|_ System time: 2012-04-23 10:55:41 UTC+2
```

254. La funcionalidad se activa con el modificador `-sC`, utilizando los scripts por defecto para cada servicio descubierto. También puede ejecutarse con el modificador `-script <nombre>` especificando qué script se quiere lanzar.

```
# nmap -T4 -p80 -script http-headers 192.168.100.124
PORT      STATE SERVICE
80/tcp    open  http
| http-headers:
| Date: Tue, 24 Apr 2012 10:49:16 GMT
| Server: Apache/2.2.22 (Debian)
| Last-Modified: Wed, 13 Jul 2011 11:32:34 GMT
| Content-Length: 1105
| Vary: Accept-Encoding
|_ Connection: close
```

²⁰ <http://www.lua.org>

²¹ <http://www.lua.org/manual/>

²² <http://www.lua.org/pil>

```
| Content-Type: text/html
|_ (Request type: HEAD)
```

255. Algunos scripts requieren parámetros de entrada. La forma de introducirlos es con el modificador `--script-args`:

```
# nmap --script asn-query.nse --script-args dns=8.8.8.8 111.222.111.222
```

256. La carpeta donde se encuentran los scripts que incluye la instalación de Nmap es `/usr/share/nmap/scripts`. En el momento de la redacción de este documento, el paquete incluye 404 scripts, además de 98 librerías, que se describen más adelante. La lista completa, actualizada y comentada puede encontrarse en su documentación oficial²³.

5.2. CATEGORÍAS

257. Cada script está clasificado en una o más categorías. Éstas proporcionan información según el propósito o funcionamiento del script. La siguiente tabla describe cada una de ellas:

Auth	Scripts que intentan gestionar un proceso de autenticación
Default	Se ejecuta un conjunto básico de scripts por defecto. Son los que se lanzan cuando no se le indica parámetro.
Discovery	Scripts con el propósito de descubrir información de un objetivo
DOS	Tratan de ejecutar un ataque por denegación de servicio
Exploit	Intentan explotar activamente una vulnerabilidad
External	Utilizan algún recurso externo de consulta
Fuzzer	Contiene scripts que intenta forzar fallos con datos no esperados
Intrusive	Scripts cuyo uso puede suponer un riesgo para la máquina a explorar
Malware	Comprueba si el objetivo está infectado
Safe	Scripts cuyo uso se considera seguro
Vuln	Informan si existe una vulnerabilidad conocida específica

258. Estos grupos permiten ejecutar scripts de forma que solo se lancen una o varias categorías, según la que más convenga en cada caso.

```
nmap -T4 -p80 -script discovery 192.168.100.124
```

259. También pueden usarse condicionales, modificadores o comodines:

```
nmap -T4 -p80 -script "discovery, safe" 192.168.100.124
```

260. Lanzaría cualquier script dentro de la categoría `discovery` o `safe`

```
nmap -T4 -p80 -script "discovery and safe" 192.168.100.124
```

261. Lanzaría cualquier script que esté dentro de la categoría `discovery` y `safe`

```
nmap -T4 -p80 -script "not safe" 192.168.100.124
```

262. Ejecutaría cualquier script que no estuviera en la categoría `safe`

²³ <http://nmap.org/nsedoc/>

```
nmap -T4 -p80 -script http-* 192.168.100.124
```

263. Se ejecutarían todos los scripts que comenzasen por “http-“

5.3. FORMATO

264. Cada uno de los scripts siguen un formato concreto, con varios campos que definen como y cuando debe ser lanzado.

5.3.1. CAMPO *description*

265. El campo *description* contiene la información relevante sobre el funcionamiento del script. Aquí debería encontrarse todo lo necesario para que el usuario sepa que hace el script y como usarlo.

```
description = [[
Queries a VNC server for its protocol version and supported security types.
]]
```

5.3.2. CAMPO *categories*

266. Este campo especifica a que categorías pertenece. Pueden ser una, o varias categorías, no diferencia entre mayúsculas y minúsculas, y deben estar listadas en el formato array de Lua:

```
categories = {"exploit", "intrusive"}
```

5.3.3. CAMPO *author*

267. Campo opcional que identifica el autor del script, con el fin de tener un medio para contactar con el desarrollador, si procede.

```
Author = "John Doe jdoe@example.com"
```

5.3.4. CAMPO *license*

268. Campo opcional que describe el tipo de licencia con la que se ha publicado el script. Útil para conocer que permisos legales se dispone para utilizarse o distribuirse.

```
license = "Same as Nmap--See http://nmap.org/book/man-legal.html"
```

5.3.5. CAMPO *dependencies*

269. Este campo opcional si hay algún script que debe ejecutarse previamente. Se utiliza cuando un script hace uso del resultado de otro.

```
dependencies = {"smb-brute.nse"}
```

5.3.6. REGLAS

270. Nmap utiliza unas reglas para determinar cuando debe ejecutarse un script. Éstas evalúan si se cumplen las condiciones para que el script sea lanzado. Existen cuatro tipos de reglas:

- `prerule()`: se ejecuta una única vez, antes de que se escanee ningún host.
- `hostrule(host)`: se ejecuta justo después de que se ha escaneado un host.

- `portrule(host, port)`: se ejecuta justo después de haberse escaneado un host. Ambas reglas se ejecutan tantas veces como host se escaneen. Aceptan las estructuras de datos `host` y `port`, que se explicarán más adelante en el apartado API.
- `postrule()`: la regla es ejecutada después de que se hayan escaneado todos los host pendientes.

271. Estos scripts devuelven `true` o `false`. Si uno de ellos devuelve `true`, significa que la condición de funcionamiento se cumple y se lanza el script.

5.3.7. FUNCIÓN ACTION

272. Esta función recoge todas las instrucciones que se ejecutarán cuando se cumplan las condiciones definidas en las reglas anteriores. Es la función principal donde se va a llevar a cabo el proceso por el que el script ha sido lanzado.

5.4. LIBRERÍAS

273. Las librerías son módulos adicionales implementados que permiten al desarrollador crear scripts más potentes. Se pueden adjuntar al programa original para aprovechar las funcionalidades añadidas que aportan.

274. En el momento de la edición de la guía existen 98 librerías disponibles. Enumerarlas una a una sería poco práctico por el gran número de librerías, además de ser una cifra en constante crecimiento. Se puede encontrar el listado completo y actualizado en la documentación oficial de Nmap²⁴.

5.5. API

275. NSE implementa unas funciones internas que proveen detalles sobre la máquina objetivo, como el estado de los puertos o el resultado de la detección de versión. También ofrece funciones para usar contra Nsock, la librería de sockets de Nmap. Dispone de mecanismos de manejo de excepciones para crear scripts robustos.

5.5.1. ESTRUCTURAS *host* Y *port*

276. Nmap pone a disposición del desarrollador estas dos estructuras que facilita la recogida de información de la máquina objetivo contra la que se está ejecutando el script. La lista siguiente detalla cada variable de estas dos estructuras.

- `host`: esta estructura contiene información sobre la máquina objetivo. La disponibilidad de esta información dependerá si el escaneo se ha ejecutado con la opción de OS fingerprinting `-O`
- `host.os`: contiene un array de cadenas de texto con los posibles sistemas operativos que pueden estar ejecutándose en el equipo objetivo. Si no se ejecuta con la opción `-O`, el contenido será `nil`.
- `host.ip`: contiene una cadena de texto con la dirección IP del objetivo.
- `host.name`: contiene una cadena de texto con la resolución DNS inversa del objetivo. Si el objetivo no tiene resolución inversa, el valor de la cadena será vacía.
- `host.targetname`: contiene el nombre del objetivo especificado en la línea de comandos.

²⁴ <http://nmap.org/nsedoc/>

- `host.directly_connected`: valor booleano que indica si la máquina objetivo se encuentra en el mismo segmento de red.
- `host.mac_addr`: dirección MAC del equipo destino, o nil si el equipo no está directamente conectado.
- `host.mac_addr_next_hop`: dirección MAC del próximo dispositivo en la ruta hacia el host objetivo.
- `host.mac_addr_src`: dirección MAC de la máquina que está lanzando el escaneo, que puede ser falsa si se ha usado la opción `-spooof-mac`
- `host.interface_mtu`: indica el valor de la unidad máxima de transferencia, o 0 si se desconoce.
- `host.interface`: cadena de texto que contiene la interfaz de red por la que se están enviando las sondas hacia el equipo objetivo.
- `host.bin_ip`: dirección IP de la máquina objetivo
- `host.bin_ip_src`: valor del “*round trip time*” medio del objetivo
- `host.times.rttvar`: valor de la varianza del “*round trip time*” del objetivo
- `host.times.timeout`: valor del tiempo máximo de respuesta de la máquina objetivo
- `port`: estructura de datos que se pasa por parámetro a los scripts de reglas, al igual que la estructura `host`. Contiene información del puerto al que Nmap está analizando en ese momento.
- `port.number`: contiene el número de puerto del equipo objetivo al que se está analizando
- `port.protocol`: define el protocolo del puerto de la máquina objetivo. Los valores válidos deben ser “tcp” o “udp”
- `port.service`: contiene una cadena de texto con la estimación del servicio detectado.
- `port.version`: estructura de datos con información detallada del servicio. Si no se ha utilizado la opción de detección de versión `-sV` los campos serán nil.
- `port.version.name`: contiene el nombre del servicio, detectado a través del número de puerto.
- `port.version.name_confidence`: grado de confianza del nombre detectado en el campo anterior, con una fiabilidad de 1 a 10.
- `port.version.product`: nombre del fabricante o distribuidor del servicio
- `port.version.version`: cadena de texto con la versión de la aplicación
- `port.version.extrainfo`: Información adicional recopilada respecto a la versión del servicio.
- `port.version.hostname`: cadena de texto con el nombre del equipo detectado a través de la funcionalidad de detección de versión de servicio. En algunos casos puede diferir del nombre del equipo que se detecta por una resolución DNS inversa.
- `port.version.ostype`: cadena de texto con el sistema operativo detectado a través de la funcionalidad de detección de versión de servicio. Puede ser diferente a la información recopilada con la funcionalidad de detección de versión de sistema operativo `-O`
- `port.version.devicetype`: tipo de dispositivo que está ejecutando el servicio.

- `port.version.service_tunnel`: indica si Nmap está usando SSL para detectar el servicio
- `port.version.service_fp`: cadena de texto con el fingerprint del servicio, si está disponible
- `port.version.rpc_status`: cadena de texto que informa si se ha podido comprobar el número de programa de un servicio RPC.
- `port.state`: contiene información sobre el estado del puerto. Los posibles estados pueden ser “*open*” u “*open|filtered*” puesto que un script no se ejecutará sobre un puerto que se encuentre cerrado

5.5.2. API DE RED

277. La API de Nmap ofrece funciones de tráfico de información a través la red. Un usuario puede crear un socket, conectarlo a un puerto, enviar, recibir información, y cerrar el socket, con la ventaja que la librería de sockets de Nmap está optimizada para ser paralelizable y eficiente.

278. Actualmente esta API cuenta con más de 50 funciones. La lista actualizada y completa se encuentra en <http://nmap.org/nsedoc/lib/nmap.html>.

279. El siguiente ejemplo describe una prueba de concepto sobre el uso de funciones de la API de red de Nmap.

```
require ("nmap")
local sock = nmap.new_socket()
sock:set_timeout(1000)
try = nmap.new_try(function() sock:close() end)
try(sock:connect(host.ip, port.number))
try(sock:send("GET / HTTP/1.1\nHost:www.example.com\n\n"))
response = try(sock:receive())
sock:close()
```

280. Se puede observar en el ejemplo como se crea un socket, se conecta a una IP y un puerto, envía una petición HTTP y recoge la respuesta en una variable.

5.5.3. MANEJO DE EXCEPCIONES

281. NSE incluye un mecanismo de manejo de excepciones que le proporcionan robustez a los scripts. Se tiene el método `nmap.new_try` que permite crear un manejador de excepción. Este método tiene como parámetro la función que se quiere que se ejecute en el caso de que el código que se quiere controlar falla. El valor resultante del método es la función que capturaré la excepción. Esta función tiene como parámetro las funciones que se querrá controlar su correcto funcionamiento. Si una función falla, devolverá el valor `false` y la ejecución del código se detendrá, lanzándose la función definida como parámetro en el método.

282. Usando el ejemplo anterior:

```
try = nmap.new_try(function() sock:close() end)
```

283. Se ha creado un manejador de excepción `try`, que lanzará la función definida implícitamente como parámetro si captura una excepción. La función definida cerrará el socket.

```
try(sock:connect(host.ip, port.number))
```


284. La función `try` tiene como parámetro el código que se está controlando su correcto funcionamiento. En caso de fallo, la función `connect` devolverá `false`, por lo que la ejecución se detendrá y se lanzará la función que se definió al crearse el manejador de excepción, en este caso, como se ha descrito anteriormente, se cerrará el socket.

5.6. EJEMPLO

285. Hasta ahora se ha descrito una gran parte de las funcionalidades disponibles para desarrollar scripts para el motor NSE. A continuación se va a mostrar un ejemplo de script incluido en la herramienta, describiendo cada sección y su propósito.

5.6.1. CABECERA

286. Este apartado engloba básicamente metainformación referente al script. La descripción de los campos que se pueden usar en este apartado están descritos en la sección 5.3.

```
description = [[
Queries a VNC server for its protocol version and supported security types.
]]
author = "Patrik Karlsson"
license = "Same as Nmap--See http://nmap.org/book/man-legal.html"
categories = {"discovery", "safe"}
```

287. También se recomienda incluir un ejemplo de resultado por pantalla, con la etiqueta `@output`

```
---
-- @output
-- PORT      STATE SERVICE
-- 5900/tcp  open  vnc
-- | vnc-info:
-- |   Protocol version: 3.889
-- |   Security types:
-- |     Mac OS X security type (30)
-- |_    Mac OS X security type (35)
```

5.6.2. LAS REGLAS

288. Esta sección define qué requisitos son necesarios para que la acción del script se ejecute.

```
require 'shortport'
require 'vnc'

portrule = shortport.port_or_service( {5900, 5901, 5902} , "vnc", "tcp", "open")
```

289. La librería `shortport` se utiliza para crear funciones de reglas de forma cómoda.

5.6.3. LA ACCIÓN

290. Finalmente se implementa el código que va a ser ejecutado en caso de que las reglas se cumplan.

```
action = function(host, port)

    local vnc = vnc.VNC:new( host.ip, port.number )
    local status, data
    local result = {}

    status, data = vnc:connect()
```

```

    if ( not(status) ) then    return " \n ERROR: " .. data end

    status, data = vnc:handshake()
    if ( not(status) ) then    return " \n ERROR: " .. data end

    status, data = vnc:getSecTypesAsStringTable()
    if ( not(status) ) then    return " \n ERROR: " .. data end

    table.insert(result, ("Protocol version:
%s"):format(vnc:getProtocolVersion()) )

    if ( data and #data ~= 0 ) then
        data.name = "Security types:"
        table.insert( result, data )
    end

    if ( vnc:supportsSecType(vnc.sectypes.NONE) ) then
        table.insert(result, "WARNING: Server does not require
authentication")
    end

    return stdnse.format_output(status, result)
end

```

5.7. EJEMPLO DE USO

291. El siguiente ejemplo es un excelente caso de uso de la potencia y versatilidad de Nmap, extendiendo su funcionalidad de escaner de puertos a un verdadero identificador de vulnerabilidades.
292. El script a utilizar será “smb-check-vulns”. Este comprueba si el servicio samba de un equipo es vulnerable, realizando varias pruebas sobre el puerto de servicio. El script devolverá los resultados obtenidos, en función de si las pruebas realizadas han dado positivo o negativo.
293. Es conveniente remarcar que algunas de las pruebas que realiza el script se consideran peligrosas, y pueden probar la inestabilidad del equipo auditado, puesto que intenta aprovechar alguna de las vulnerabilidades conocidas del servicio samba, por lo que el script permite restringir las pruebas realizadas a únicamente las que son seguras. Para ello hay que añadir el parámetro “safe=1”
294. El uso es el siguiente:

```
nmap -sS -T4 -p445,139 --script=smb-check-vulns --script-args safe=1 172.16.28.51
```

```

PORT      STATE SERVICE
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
Host script results:
| smb-check-vulns:
|   MS08-067: CHECK DISABLED (remove 'safe=1' argument to run)
|   Conficker: Likely CLEAN
|   regsvc DoS: CHECK DISABLED (add '--script-args=unsafe=1' to run)
|   SMBv2 DoS (CVE-2009-3103): CHECK DISABLED (add '--script-args=unsafe=1' to
run)
|   MS06-025: CHECK DISABLED (remove 'safe=1' argument to run)
|_  MS07-029: CHECK DISABLED (remove 'safe=1' argument to run)

```

295. Se observa como el script comprueba que el equipo escaneado no está infectado con el malware Conficker. Las otras comprobaciones no han sido lanzadas puesto que se consideran inseguras.
296. Puede ser realmente útil si se realizan escaneos periódicos contra segmentos de red completos, si se combinan con otros scripts de funcionalidad similar.

5.8. DEFENSAS CONTRA NMAP

297. Hasta ahora se ha visto la utilidad de la herramienta Nmap desde el lado del administrador de sistemas. Sin embargo, esta aplicación puede ser utilizada por atacantes contra redes bajo nuestra responsabilidad, como herramienta de descubrimiento de máquinas y servicios vulnerables. En esta sección se describirán algunas técnicas para dificultar o confundir un escaneo Nmap y la tecnología necesaria a usar.

5.8.1. ESCANEOS PROACTIVOS

298. La principal contramedida a un escaneo hacia la red a proteger es realizar los mismos posibles escaneos que se podrían recibir procedentes de un atacante potencial. Mediante auditorías de visibilidad externas e internas, se recopila una información valiosa acerca del entorno de la red. Con estos datos, es posible hacer una valoración de qué servicios o equipos no deberían ser visibles desde ciertas redes, debido a una incorrecta configuración de los cortafuegos o una posible regla obsoleta que haya podido suponer un posible punto de entrada para un usuario externo.

5.8.2. CORTAFUEGOS CON POLÍTICA POR DEFECTO DROP

299. Los cortafuegos tienen dos modos de funcionamiento básicos: por defecto ALLOW y por defecto DROP. En el primer caso, el cortafuegos permite todo el tráfico entrante, excepto una lista negra de puertos no permitidos. En el segundo caso, el cortafuegos filtraría todo el tráfico entrante, excepto una lista blanca de puertos autorizados.

300. La consecuencia aplicada a un escaneo de Nmap es que si hay un cortafuegos en medio con política ALLOW, el cortafuegos dejará pasar todas las sondas que envíe Nmap durante el escaneo, excepto las pocas sondas que se haya filtrado porque el cortafuegos esté configurado para denegar ciertos puertos concretos. Ocurre entonces que cuando llega un intento de conexión a un puerto cerrado de un equipo, éste responde de forma activa que el puerto está cerrado. Nmap recibe este dato y comprende que debe continuar su escaneo con el siguiente puerto.

301. Con un cortafuegos con política DROP, todas las sondas que envíe Nmap quedarán filtradas, exceptuando las que estén definidas en la lista blanca. Por tanto, cada vez que se envíe una sonda hacia un puerto no alcanzable, el paquete se descartará. Entonces Nmap esperará un tiempo hasta dar por perdida la sonda y la reenviará otra vez, hasta un número máximo de intentos para darse por vencido y pasar al siguiente puerto.

302. Al no haber una respuesta a cada sonda que ha sido enviada y filtrada por el cortafuegos, el Nmap lanzado por el atacante consumiría mucho más tiempo en escanear un único equipo debido a los tiempos de espera entre sondas filtradas y puertos inaccesibles.

5.8.3. SERVICIOS OCULTOS

303. Nmap dispone de una lista de los puertos más usuales. Un escaneo por defecto, sin especificar qué rango de puertos se quiere auditar, utiliza esta lista y escoge los 1000 puertos más utilizados para intentar encontrar alguno abierto. Esta lista

usualmente está en */usr/share/nmap/nmap-services*. En algunos casos puede resultar útil cambiar el puerto a la escucha de un servicio por un puerto “oscuro” que no sea muy conocido. Puede evitar un escaneo rápido y bastante malware que busca servicios en puertos usuales para intentar aprovechar vulnerabilidades conocidas o lanzar ataques por fuerza bruta.

ANEXOS

5.9. ANEXO A.- ANÁLISIS DE RESULTADOS

Este anexo realiza una interpretación de los resultados obtenidos en los Procedimientos de Análisis del Capítulo 4, con los que poder extraer las conclusiones oportunas. Las trazas y tablas con valores cuantitativos se analizan pormenorizadamente con el fin de facilitar su lectura así como para destacar aquellos aspectos que deban llamar la atención al administrador interesado en explotar el potencial de *Nmap*.

Los resultados a analizar por cada procedimiento son los siguientes:

- **Procedimiento 1: descubrimiento de equipos de una subred**
 - CASO 1: obtención de nombres de máquina de los objetivos
 - CASO 2: determinación del estado de las máquinas

- **Procedimiento 2: escaneo de puertos**
 - CASO 1: enumeración de los puertos TCP abiertos y cerrados en una máquina local.
 - CASO 2: enumeración de los puertos TCP abiertos y cerrados en una máquina perteneciente a la misma subred local.
 - CASO 3: enumeración de los puertos TCP abiertos y cerrados en una máquina externa.
 - CASO 4: descubrimiento de las reglas de una herramienta cortafuegos.
 - CASO 5: identificación de servidores en una red.

- **Procedimiento 3: detección de servicios y de sistema operativo**
 - CASO 1: identificación de versiones de servicios.
 - CASO 2: identificación precisa de servidores en una red.
 - CASO 3: determinación precisa del estado de puertos UDP.
 - CASO 4: inventario de plataformas instaladas en una red.

- **Procedimiento 4: evasión de cortafuegos y herramientas de detección y prevención de intrusos**
 - CASO 1: evasión sigilosa.
 - CASO 2: evasión mediante falsos positivos.
 - CASO 3: evasión mediante ocultación del origen. Identificación de máquinas susceptibles de ser utilizadas como *zombie* en un *Iddle Scan*.

- **Procedimiento 5: técnicas de optimización del análisis**
 - CASO 1: análisis en una red local.
 - CASO 2: análisis en una red remota.

5.9.1. A1.- ANÁLISIS DE RESULTADOS DEL PROCEDIMIENTO 1

El primer procedimiento de análisis, consistente en descubrimiento de equipos, se ha dividido en dos propósitos principales: recuperar el nombre de los equipos de la red solicitada, y mostrar los diferentes resultados obtenidos según el tipo de escaneo de descubrimiento realizado.

En el apartado de resultados de este procedimiento (ver 4.1.3), se puede observar los nombres de algunos equipos procedentes de las redes `www.universidad.es/24` y `www.empresa.es/24`. Se puede observar como este análisis, que ni siquiera ha requerido enviar ninguna sonda a los equipos descubiertos, se ha podido averiguar información interesante sobre el propósito de algunas direcciones IP. Por ejemplo, observando los nombres `internet.universidad.es` y `correo.empresa.es` puede deducirse que el desempeño de estas máquinas es de cortafuegos perimetral o servidor de correo, respectivamente. El nombre `xuquer.empresa.es` también puede ser de utilidad, ya que, sabiendo que Xuquer es el término en valenciano de uno de los ríos que pasa por la Comunidad Valenciana, se podría hacer un listado con otros ríos similares, y tratar de obtener otros equipos en Internet pertenecientes a la empresa, en esta u otras subredes.

Ante esta información obtenida, un administrador de sistemas debe valorar si la política de asignación de nombres de los equipos es adecuada, por la información que se pudiera revelar. Evidentemente, se debe encontrar el compromiso entre usabilidad y seguridad de los dispositivos de red.

Otra información relevante obtenida en el descubrimiento de la red de `www.empresa.es` es que el rango de nombres de equipo recuperados no pertenece en su totalidad a la empresa, sino que también se observan equipos de otras empresas. Además, se encuentran intercaladas entre direcciones IP que parecen pertenecer a un proveedor de servicios de Internet (ISP), por lo que se puede deducir que ambas empresas comparten el mismo proveedor de Internet. Resulta útil además para delimitar la pertenencia de un rango de direcciones IP a una entidad o a otra.

Esto también ocurre con el análisis a `www.organiza.org`, donde se han encontrado dominios de otras empresas, así como subdominios de la empresa de alojamiento Hosting, lo que nos indica que su web está alojada por una empresa externa (este resultado no se incluye en este procedimiento, pero se puede ver un ejemplo en la Figura 37).

La siguiente parte del análisis muestra una tabla de resultados (Tabla 1) con interesantes resultados estadísticos obtenidos según el tipo de escaneo y la red hacia donde iba dirigido.

Se puede observar primeramente la diferencia de tiempos de respuesta del listado de nombres (-sL) según lo lejano que esté el servidor DNS al que se le está realizando las consultas.

También se observa que el análisis con el método Ping TCP ACK (ver 3.1.6) no devuelve resultados válidos para ninguna red que no sea la local. Esto nos indica que los sistemas cortafuegos que separan la red local de Internet guardan estado y no permiten, por tanto, el funcionamiento de este tipo de pruebas.

Otro dato interesante es el encontrado en el Ping ICMP Echo. Se puede ver que no se devuelve ningún resultado para la red `www.universidad.es/24` y que, además, el análisis tarda mucho más que para las otras redes. Esto se debe a que los administradores de dicha red, como muchos otros, cortan este tipo de peticiones y sus respuestas, para inhabilitar una

de las herramientas más populares para conocer el estado de un equipo: el comando ping. En cualquier caso, como ha quedado patente en este procedimiento, Nmap es mucho más potente e implementa muchos otros tipos de análisis, intentando maximizar el número de equipos encontrados.

5.9.2. A2.- ANÁLISIS DE RESULTADOS DEL PROCEDIMIENTO 2

El segundo procedimiento de análisis aplica las técnicas de análisis de puertos sobre uno o varios objetivos, según el fin buscado. Tal y como se ha podido ver, es recomendable combinar estas técnicas con las analizadas del anterior procedimiento de descubrimiento de equipos, ya que, al reducir el número de equipos sobre los que se realiza el análisis de puertos, sin duda pueden ayudar a obtener los mismos resultados pero en mucho menos tiempo. Este es el comportamiento por defecto en los análisis con Nmap. Esto es así debido a que las máquinas no alcanzables no serán analizadas, aumentando notablemente el rendimiento del proceso, en ocasiones muy costoso.

A la vista de los resultados de los distintos casos comparados en este procedimiento (ver Tabla 2), deben llamar la atención dos hechos importantes que afectan al rendimiento de las técnicas: el primero que, cualquier análisis de puertos completo realizado contra un objetivo que no sea la propia máquina, consumirá probablemente varios minutos. El segundo que, la tasa de escaneo contra máquinas externas respecto a internas, puede llegar a ser varios ordenes de magnitud menor. Es evidente que un escaneo masivo en esas condiciones tendría un muy elevado coste.

Se puede tomar como referencia el Caso 2, escaneo a una máquina perteneciente a la misma subred local, para cual Nmap ha invertido 17,96 segundos. Todo ello en el entorno descrito en el Banco de Pruebas, el cual se puede considerar muy rápido.

La tasa de escaneo para el Caso 2 es razonablemente elevada, en torno a 3600 puertos/segundo, considerando que el objetivo carece de cortafuegos (0% de puertos filtrados) y por lo tanto responderá rápidamente a las sondas evitando que expiren los temporizadores y se produzcan retransmisiones. El Caso4 realiza un escaneo parecido, también completo, pero contra un objetivo que está tras un cortafuegos (99,99% de puertos filtrados), disminuyendo la tasa a un 11%, en torno a 400 puertos/segundo.

Respecto al análisis de la cantidad y calidad de información que es posible obtener con estas técnicas, se debe destacar la utilidad del Caso 1, escanear la propia máquina, en casos en los que se dude de la información que proporcionan los propios sistemas operativos sobre el estado de los puertos (p.ej. comando *netstat -a*). Existen algunos programas maliciosos que consiguen ocultar sus puertos de comunicaciones al sistema operativo, pero esta técnica de Nmap haría posible revelar su presencia. Los siguientes dos casos obtienen similares resultados pero contra un objetivo local y externo respectivamente. Se observa cómo la máquina externa está tras un cortafuegos que en principio sólo permite tráfico web y ssh. Utilizando estos resultados es posible extraer y analizar las reglas de un cortafuegos (Caso 4), observándose cómo filtra de facto todo el tráfico que no vaya dirigido al puerto del servidor de escritorio remoto.

El Caso 5 es especialmente interesante por la calidad de la información que ofrece. Llama la atención cómo en apenas medio minuto sea posible descubrir más de un centenar de potenciales servidores muy comunes (http, ftp, smtp, telnet, etc). Este caso es un buen ejemplo de cómo conociendo los argumentos adecuados, Nmap demuestra un enorme potencial como herramienta de análisis y control de redes. El único detalle a tener en cuenta en este análisis lo tenemos en los casos 5.2 y 5.3, donde se observa que todos los

puertos SMTP se detectan como filtrados. Esto se debe a la existencia de un cortafuegos en la salida de la red del equipo que realiza el análisis que filtra todo el tráfico SMTP saliente, permitiendo únicamente el tráfico SMTP a los servidores de correo de la organización.

5.9.3. A3.- ANÁLISIS DE RESULTADOS DEL PROCEDIMIENTO 3

El tercer procedimiento muestra un conjunto de modificadores aplicables sobre las técnicas de análisis de puertos destinados a aumentar la fiabilidad y la utilidad de los resultados. El coste de estas opciones en tiempo de escaneo y el volumen de tráfico generado son en general relativamente elevados, lo que hace que sean distintas las posibles aplicaciones de este procedimiento.

Los resultados obtenidos en el Caso 1 exponen una situación de interés o preocupación habitual de los administradores de red: conocer exactamente qué servicio(s) está proporcionando un sistema. En este caso es posible ver cómo en el objetivo, entre otros puertos a la escucha, hay un servidor SSH cuando en principio un análisis de puertos normal, sin el modificador -sV, hubiera detectado en su lugar uno telnet (23/tcp). De hecho, Nmap detecta este servicio a la escucha en el puerto habitualmente destinado a conexiones telnet (23/tcp).

De forma similar, los resultados del Caso 2 ayudan además a comprender el impacto de esta opción sobre el tiempo de escaneo. Los casos 5.1 y 5.2 del anterior procedimiento realizan la misma prueba pero sin detección de versiones con un rendimiento de 21 y 53 puertos/s, mientras que, en el homólogo de este procedimiento, la velocidad del análisis es de 3 y 7,69 puertos/s respectivamente, alrededor de 7 veces más lento. A esta velocidad tan baja, el usuario debe considerar detenidamente la elección de puertos objetivo a escanear (modificador -p).

El caso 2 nos muestra otra cosa interesante. En el resultado del análisis se muestra un servicio que devuelve información pero que no ha sido correctamente identificado, por lo que se pide al usuario que, si conoce el servicio que está tras ese puerto, envíe información a los desarrolladores para que incluyan detección para dicha aplicación en versiones posteriores de la aplicación.

Los resultados del Caso3 muestran cómo Nmap puede ayudar, aunque no siempre, a determinar el estado de los puertos UDP cuando hay duda de que estén abiertos o filtrados. Los puertos UDP abiertos no responden implícitamente a los intentos de conexión, para un analizador de puertos parecen como si estuvieran filtrados. Nmap envía sondas intentando establecer un diálogo que provoque una respuesta. En las pruebas se preparó un servidor con el puerto DNS (53/udp) a la escucha. Sería por tanto sencillo identificar automáticamente servidores DNS en un conjunto de objetivos más grande. Nuevamente se debe llamar la atención de la muy baja tasa de escaneo que genera este tipo de opciones.

En el caso 4, se ha realizado una prueba de detección en una red con múltiples equipos y dispositivos con distintas configuraciones. A la vista de los resultados Nmap demuestra una notable precisión en esta tarea, sobre todo desde sus más recientes versiones en las que se ha mejorado notablemente su base de datos de sistemas operativos y servicios.

Es recomendable usar la detección de SO y la detección de versiones de forma conjunta, ya que el proceso de detección de SO no es perfecto, ya que indica que existen varios equipos con sistema “Comau C4G robot control unit”, mientras que en la fase de detección de versiones, comprobamos que realmente son equipos FreeBSD. De esta forma podemos evitar pequeñas discrepancias al realizar el análisis.

5.9.4. A4.- ANÁLISIS DE RESULTADOS DEL PROCEDIMIENTO 4

El cuarto procedimiento muestra las aplicaciones que podría tener para un administrador el uso de los modificadores de Nmap que podrían permitir pasar inadvertido a través de cortafuegos y herramientas IDS/IPS. Para el caso de filtros de red como las herramientas cortafuegos la opción es única, el sigilo. Para las herramientas de detección y prevención de intrusiones, podrían funcionar también las prácticas que generen mucho ruido y de este modo pasar inadvertido el verdadero origen.

El Caso 1 utiliza varias técnicas de evasión, centrándose en el sigilo para evitar ser detectado por IDS/IPS o conseguir traspasar las restricciones de los cortafuegos. A continuación se describen las técnicas utilizadas:

- Fragmentación de paquetes (*-f*): puede ser útil contra algunos filtros de paquetes o IDS, que tengan configurado no reensamblar los fragmentos, por consumo excesivo de recursos que supondría.
- Sondas no vacías (*--data length n*): los IDS tienen en cuenta el tamaño del paquete para evaluar si lo que está analizando es una posible sonda de un escaneo. Añadiendo datos aleatorios a cada sonda, puede confundir al IDS y evitar la detección del escaneo.
- Falsificación del puerto de origen (*-g n*): suele resultar útil contra reglas mal configuradas de cortafuegos
- Esquema de tiempo lento (*-T0/I*): el uso de largos tiempos de respuesta entre sondas hacen más complicada la detección de barrido de puertos por los IDS y cortafuegos.

Los resultados muestran que el uso de esquemas de tiempo bajos ralentizan extremadamente el tiempo de finalización del escaneo. Por eso es conveniente utilizar esta técnica en escaneos de puertos concretos y en grupos de ips reducidos.

El caso 2 utiliza señuelos con direcciones IP falsas para dificultar el reconocimiento del origen real del ataque. La finalidad de este método se basa principalmente en camuflar la IP de donde procede el escaneo, mediante la generación de sondas con origen falso. De este modo es posible confundir a los sistemas de detección de intrusos y pasar desapercibido de entre todas las alertas generadas.

Destacable mencionar el parámetro ME, indicando la posición donde debe usar la IP propia. Las posibilidades de evasión aumentan conforme la lista de direcciones IP señuelo sea mayor, y la posición del parámetro ME sea lo más posterior posible.

El caso 3, que utiliza la técnica de *Idle scan*, es un complejo método que utiliza un equipo intermedio zombie y gracias a éste, se puede realizar un escaneo a un equipo sin tener que enviar directamente ninguna sonda. El éxito de esta técnica depende si el equipo zombie es vulnerable, es decir, si genera secuencias de paquetes IP con un campo IPID predecible. De esta forma sería posible hacer un escaneo a una máquina a la que no se puede alcanzar de forma directa, pero que se tiene conocimiento que es alcanzable desde el equipo zombie. Por ello, es muy importante que el administrador de la red sea consciente de este tipo de escaneos para evitar que un posible atacante use alguno de los equipos de su responsabilidad para realizar escaneos a redes aparentemente inalcanzables por una red externa.

El procedimiento utilizado demuestra como ha sido posible determinar el servicio de Terminal Server disponible en un equipo que no es alcanzable, sin que sea necesario enviar ninguna sonda directamente a la máquina analizada.

5.9.5. A5.- ANÁLISIS DE RESULTADOS DEL PROCEDIMIENTO 5

El procedimiento 5 se centra en la optimización de parámetros de tiempos de respuesta de sondas para realizar escaneos eficientes. Estas mejoras resultan imprescindibles si se piensa hacer análisis a grandes rangos de direcciones IP, debido a que el tiempo de finalización es notablemente inferior.

Se ha dividido el procedimiento en dos casos de pruebas: un entorno de red local, y una red remota, con el fin de comparar los tiempos de respuesta y buscar los mejores parámetros de optimización, según sea el caso.

En el caso primero, perteneciente a la red local, se puede observar como con la configuración por defecto, y realizando un escaneo a una red de 254 ips, se obtienen resultados después de 347 segundos. Después de obtener los valores de los parámetros con las pruebas de latencia descritas, se realiza mismo escaneo con los tiempos optimizados y se obtienen los resultados después de 105 segundos, lo que implica un 70% de mejoría de rendimiento.

En cuanto al segundo caso, donde se muestra un ejemplo en una red remota, se realiza un escaneo igualmente a una red de 254 equipos, finalizándose después de 3911 segundos. De nuevo calculándose los parámetros para la optimización de tiempo, finaliza el escaneo después de 1546 segundos. Se consigue así una mejora del 69% con los tiempos de respuesta.

5.10. ANEXO B.- LISTADO DE COMANDOS NMAP

Las siguientes tablas sintetizan el funcionamiento de las técnicas descritas en la anterior sección, sirviendo a su vez de una valiosa guía avanzada de recordatorio de las opciones de *Nmap*.

ESPECIFICACIÓN DE OBJETIVOS			
Opción	Nombre	Funcionamiento	Observaciones
-iL <fich>	Objetivos en fichero	Se pasan los objetivos en un fichero, cada uno en una línea ²⁵ .	
-iR <num>	Objetivos aleatorios	Elige los objetivos de forma aleatoria.	
--exclude <hosts>	Lista exclusión	Indica equipos a excluir del análisis.	
--excludefile <fich>	Fichero exclusión	Se pasan en un fichero los equipos a excluir del análisis ²⁵ .	

DESCUBRIMIENTO DE EQUIPOS			
Opción	Nombre	Funcionamiento	Observaciones
-Pn	No ping	No realiza ninguna técnica de descubrimiento. Pasa directamente al análisis de puertos.	Considera a todos los objetivos como aptos para un análisis de puertos.
-sL	List Scan	Sólo lista equipos. No envía ningún paquete a los objetivos.	Hace resolución inversa DNS.
-sn	Ping Sweep	Implica un -PE + -PA 80 + -PS 443. Si misma subred, también -PR. No hace análisis de puertos posterior.	Si usuario sin privilegios: connect() a 80 y 443. Hace resolución inversa DNS.
-PR	Ping ARP	Sólo para objetivos de nuestra red local (activo por defecto). Envía un ARP Request.	<i>Host Up</i> : Se recibe un ARP Reply. <i>Host Down</i> : Expira el <i>timeout</i> .
-PS <ports>	Ping TCP SYN	Envía un SYN, por defecto al puerto 80. Acepta lista de puertos. Se ejecuta este si usuario sin privilegios.	<i>Host Up</i> : Se recibe un SYN/ACK o RST. <i>Host Down</i> : Expira el <i>timeout</i> .
-PA <ports>	Ping TCP ACK	Envía un ACK vacío, por defecto al puerto 80. Acepta lista de puertos. Traspasa cortafuegos sin estado.	<i>Host Up</i> : Se recibe un RST. <i>Host Down</i> : Expira el <i>timeout</i> .
-PU <ports>	Ping UDP	Envía un UDP vacío al puerto 31338. Acepta lista de puertos. Traspasa cortafuegos que sólo filtran TCP.	<i>Host Up</i> : Se recibe un ICMP port unreachable. <i>Host Down</i> : Otros ICMPs, expira el <i>timeout</i> .
-PY <ports>	Ping SCTP	Envía un paquete SCTP INIT al puerto 80. Acepta lista de puertos. Solo usuarios privilegiados.	<i>Host Up</i> : Se recibe ABORT o INIT-ACK. <i>Host Down</i> : Expira el <i>timeout</i> .
-PE	Ping ICMP Echo	Envía un ICMP Echo Request. Poco fiable. Filtrado en la mayoría de cortafuegos.	<i>Host Up</i> : Se recibe ICMP Echo Reply. <i>Host Down</i> : Expira el <i>timeout</i> .
-PP	Ping ICMP Timestamp	Envía un ICMP Timestamp Request. Muchos cortafuegos no filtran este ICMP.	<i>Host Up</i> : Se recibe ICMP Timestamp Reply. <i>Host Down</i> : Expira el <i>timeout</i> .
-PM	Ping ICMP Address mask	Envía un ICMP Address Mask Request. Muchos cortafuegos no filtran este ICMP.	<i>Host Up</i> : Se recibe ICMP AddMask Reply. <i>Host Down</i> : Expira el <i>timeout</i> .
-PO <proto>	IP Protocol Ping	Envía sondas IP con protocolo 1, 2 y 4. Acepta lista de protocolos.	<i>Host Up</i> : Respuesta o ICMP Prot. Unreachable. <i>Host Down</i> : Expira el <i>timeout</i> .
Modificadores			
-n	DNS	No realiza nunca resolución inversa de DNS.	Más sigiloso y más rápido.
-R		Realiza la resolución inversa de DNS incluso a los objetivos que aparecen como <i>Down</i> .	
--dns-servers <srv>		Especifica la lista de servidores DNS a utilizar para hacer la resolución	
--system-dns		Utiliza el sistema de resolución DNS del sistema operativo	
--tracetoute	Ruta	Descubre la ruta seguida por los paquetes hasta el equipo objetivo.	

²⁵ Los objetivos, ya sean nombres de máquina, direcciones IP, o cualquier otro formato aceptado por Nmap, deben aparecer cada uno en una línea distinta.

ANÁLISIS DE PUERTOS			
Opción	Nombre	Funcionamiento	Observaciones
-sT	Connect	Envía un SYN, luego un RST para cerrar conexión. Puede utilizarse sin privilegios de root . Se utilizan llamadas del SO. Menos eficiente que SYN Stealth.	<i>Closed</i> : Recibe RST. <i>Open</i> : Recibe SYN/ACK. <i>Filtered</i> : ICMP unreachable o expira el <i>timeout</i> .
-sS	SYN Stealth	Envía un SYN. Es la técnica usada por defecto. Rápida, fiable y relativamente sigilosa. También denominada <i>half-open scan</i> .	<i>Closed</i> : Recibe RST. <i>Open</i> : Recibe SYN/ACK. <i>Filtered</i> : ICMP unreachable o expira el <i>timeout</i> .
-sU	UDP Scan	Envía UDP vacío. Más lento que un análisis TCP. Se puede realizar en paralelo a otras técnicas. Para diferenciar entre <i>Open</i> y <i>Filtered</i> se puede usar el detector de versiones (-sV).	<i>Closed</i> : Recibe ICMP port unreachable. <i>Filtered</i> : Recibe otros ICMP unreachable. <i>Open</i> : Ha habido una respuesta. <i>Open/Filtered</i> : Expira el <i>timeout</i> .
-sI <zombie[:port]>	Idle Scan	Compleja. Usa IP origen de un equipo intermedio (Zombie) para analizar el objetivo. Según los cambios en el IPID del zombie, se deduce el estado de los puertos del objetivo.	Técnica muy avanzada y sigilosa. No queda registro de ningún paquete directo al objetivo.
-sA	TCP ACK	Envía ACK vacío. Sólo determina si los puertos están o no filtrados.	<i>Unfiltered</i> : Recibe RST. <i>Filtered</i> : ICMP error; expira el <i>timeout</i> .
-sN	TCP NULL	Envía TCP con todos los <i>flags</i> a 0.	<i>Closed</i> : Recibe RST. <i>Filtered</i> : Recibe ICMP unreachable.
-sF	TCP FIN	Envía TCP con el <i>flag</i> FIN a 1.	<i>Open/Filtered</i> : expira el <i>timeout</i> .
-sX	XMas Scan	Envía TCP con los <i>flags</i> FIN, PSH y URG a 1.	
-sM	TCP Maimon	Envía ACK con el <i>flag</i> FIN a 1.	
-sW	TCP Window	Envía ACK vacío. Muy parecido a ACK Stealth. Diferencia entre puertos open y closed. No siempre es fiable.	<i>Open</i> : Recibe RST con <i>Window size</i> positivo. <i>Closed</i> : Recibe RST con <i>Window size</i> cero. <i>Filtered</i> : ICMP error; expira el <i>timeout</i> .
--scanflags <flags>	TCP Personal.	Envía TCP con los <i>flags</i> que se indiquen. Por defecto, trata estado de puertos como lo hace -sS, pero se puede especificar otro <i>scan</i> .	<i>Flags posibles</i> : URG, ACK, PSH, RST, SYN, y FIN. Sin espacios.
-sO	IP Protocol	Envía paquetes IP con la cabecera vacía (excepto para TCP, UDP e ICMP) iterando sobre el campo <i>IP Protocol</i> . Determina los protocolos de transporte soportados por el objetivo.	<i>Open</i> : Recibe cualquier respuesta (no error). <i>Closed</i> : Recibe ICMP protocol unreachable. <i>Filtered</i> : Recibe otros ICMP unreachable. <i>Open/Filtered</i> : expira el <i>timeout</i> .
-sY	SCTP INIT	Envía paquetes SCTP INIT (inicio conexión). Equivalente a TCP SYN.	<i>Open</i> : Recibe SCTP INIT-ACK. <i>Closed</i> : Recibe SCTP ABORT. <i>Filtered</i> : Recibe ICMP unreachable o expira el <i>timeout</i> .
-sZ	SCTP Cookie Echo	Envía paquetes SCTP Cookie Echo (3ª fase conexión). Útil si hay cortafuegos sin estado.	<i>Closed</i> : Recibe SCTP ABORT. <i>Open/Filtered</i> : Expira <i>timeout</i> . <i>Filtered</i> : Recibe ICMP Unreachable.
-b <ftpsrv>	FTP Bounce	Usa la funcionalidad Proxy-FTP para recorrer puertos del objetivo. Las respuestas FTP indican estado del puerto. Parámetro: <i>username:pwd@server:port</i>	Explota las conexiones <i>Proxy-FTP, poco extendidas</i> . Se usa para traspasar cortafuegos.

ESPECIFICACIÓN DE PUERTOS

Opción	Funcionamiento	Observaciones
-F	Limita el análisis a los 100 puertos más comunes (ver archivo <i>nmap-services</i>).	Por defecto, se usan los 1000 puertos más comunes.
-r	Los puertos se analizan en orden secuencial creciente.	Por defecto, la lista de puertos se recorre aleatoriamente.
-p<rango>	Especifica el rango de puertos a analizar. -p- escanea todos los puertos. U: indica sólo UDP; T: sólo TCP; S: sólo SCTP.	Ej: -p U:53,111,T:21-25,80,139,S:9 Sin espacios.
--top-ports <num>	Analiza los <num> puertos más comunes, según clasificación de Nmap.	
--port-ratio <ratio>	Analiza los puertos cuyo ratio de uso sea superior a <ratio>.	

DETECCIÓN DE VERSIONES		
Opción	Funcionamiento	Observaciones
-sV	Interroga al conjunto de puertos abiertos detectados para tratar de descubrir servicios y versiones en puertos abiertos.	También usado para distinguir entre puertos marcados como <i>open/filtered</i> .
--allports	Incluye todos los puertos en la fase de detección de versiones. Por defecto se excluyen algunos.	
--version-intensity <num>	Intensidad con que se realizan pruebas para comprobar servicios y versiones disponibles. Valores de 0 (ligera) a 9 (todas pruebas disponibles).	
--version-light	Alias de --version-intensity 2	
--version-all	Alias de --version-intensity 9	
--version-trace	Muestra traza de actividad del análisis de versiones y servicios.	Útil para tareas de depuración.

DETECCIÓN DE SISTEMA OPERATIVO		
Opción	Funcionamiento	Observaciones
-O	Envía paquetes TCP y UDP al objetivo. Analiza las respuestas para conocer qué tipo de implementación de la pila TCP/IP tiene el objetivo.	Muy efectivo si al menos existe un puerto abierto y otro cerrado.
--osscan-limit	Limita la detección del SO a objetivos prometedores.	
--osscan-guess	Realiza un proceso más agresivo para la detección del SO.	Alias: --fuzzy
--max-os-tries	Fija máximo de intentos para detectar el SO.	Por defecto, 5 intentos.

EVASIÓN DE CORTAFUEGOS/IDS Y SPOOFING			
Opción	Nombre	Funcionamiento	Observaciones
-f	Fragmentar paquetes	Divide los paquetes en fragmentos de 8 bytes después de la cabecera IP. Cada f extra aumenta en 8 bytes más el tamaño de los fragmentos.	Usado para dividir las cabeceras TCP y complicar su análisis.
--mtu		Especifica el tamaño deseado. En múltiplos de 8 bytes.	
--data-length	Tamaño del paquete	Añade datos aleatorios a los paquetes enviados. Por defecto, las sondas se envían vacías.	Usado debido a que un paquete no vacío es menos sospechoso.
--randomize-hosts	Objetivos aleatorios	Divide la lista de objetivos en grupos de hasta 16384 equipos y los analiza en orden aleatorio.	Evita flujo de paquetes hacia IP consecutivas (suele ser sospechoso).
-D <host1>[, <hostN>]	Señuelos	Permite especificar un conjunto de IPs válidas que se usarán como dirección origen en el análisis a modo de señuelos. Las respuestas de los objetivos llegarán también a los señuelos.	Usado para enmascarar la propia IP en el escaneo y dificultar la traza del origen. Los señuelos deben estar activos.
-S <IP>	Falsear dirección/ puerto origen	Envía paquetes IP con la dirección origen especificada.	Usado para hacer creer al objetivo que hay otra persona escaneándolo. Algunos ISP filtran las IP origen falseadas. No se reciben respuestas.
--spooof-mac <mac>		Envía tramas Ethernet con la dirección origen especificada. Si no se especifica completa, el resto se completa de forma aleatoria.	

-g <port> --sourceport <port>		Envía paquetes usando el puerto especificado, cuando sea posible.	Usado porque muchos cortafuegos aceptan conexiones entrantes a puertos típicos como p.e. TCP20 ó UDP53.
-e <iface>	Definir interfaz	Define la interfaz de red, en caso de existir múltiples, por la que Nmap lanzará el análisis.	
--ip-options <opts>	Opciones IP	Permite fijar opciones del protocolo IP. Routers bloquean muchas de ellas. Útil para definir o reconocer rutas.	Más info y ejemplos en: http://seclists.org/nmap-dev/2006/q3/52
--ttl <valor>	TTL	Fija el tiempo de vida de las sondas enviadas.	
--badsum	Checksums incorrectos	Usa checksums inválidos para TCP, UDP y SCTP.	Usado porque muchos Cortafuegos/IDSs no procesan este campo y los objetivos sí.
--adler32	SCTP Checksum	Utiliza método de cálculo de resumen Adler32, en lugar del actual CRC-32C, para paquetes SCTP.	Útil para obtener respuestas de implementaciones SCTP antiguas.

TEMPORIZACIÓN Y RENDIMIENTO²⁶

Opción	Nombre	Funcionamiento	Observaciones
--min-hostgroup <num> --max-hostgroup <num>	Objetivos en paralelo	Establece los límites mínimo y máximo de objetivos que se pueden analizar de forma concurrente.	
--min-parallelism <num> --max-parallelism <num>	Pruebas en paralelo	Establece los límites mínimo y máximo de pruebas que pueden estar llevándose a cabo de forma concurrente. Por defecto valor dinámico basado en el rendimiento de la red.	Útil en redes o equipos lentos. Valor demasiado alto puede afectar precisión.
--min-rtt-timeout <time> --max-rtt-timeout <time> --initial-rtt-timeout <time>	Tiempo de respuesta de las sondas	Modifica el tiempo de espera de respuestas a sondas enviadas. Si vence el tiempo de espera, Nmap considera que no hay respuesta y sigue con la siguiente sonda. Por defecto valor dinámico basado en tiempo de sondas anteriores.	Útil en redes rápidas o cuando muchos puertos están cerrados.
--max-retries <num>	Retransmisiones	Especifica el número de retransmisiones para cada sonda, en caso de no recibir respuesta.	Por defecto 10 reintentos.
--host-timeout <time>	Tiempo de análisis de equipo	Especifica el tiempo máximo que ocupa Nmap en el análisis de un equipo completo. Si vence este tiempo, no se muestra nada sobre el mismo en el análisis final.	Útil para análisis grandes en redes poco fiables o lentas, a costa de perder algunos resultados.
--scan-delay <time> --max-scan-delay <time>	Tiempo entre sondas	Define el tiempo inicial y máximo que espera Nmap entre cada prueba. Nmap trata de ajustar ese tiempo de forma dinámica.	Útil si la red limita la tasa de transferencia o de respuestas. P.ej. equipos que solo envían 1 respuesta ICMP por segundo.
--min-rate <num> --max-rate <num>	Tasa de envío de sondas	Controla la tasa de envío de sondas. Ámbito global del análisis, no por objetivo.	
--defeat-rst-ratelimit	Límite de respuestas RST	Muchos equipos limitan, además del número de ICMP, el número de RST que envían. Por defecto Nmap se ajusta al límite. Este parámetro hace que Nmap no tenga en cuenta este límite.	Puede reducir precisión.
--nsock-engine <motor>	Motor E/S nsock	Fuerza el uso de un motor de control de entrada salida.	Valores posibles: epoll y select.
-T <plant>	Plantillas de tiempo	Define una plantilla genérica de tiempos, que configura varios de los parámetros vistos anteriormente (ver 3.3.4 para información detallada).	Valores (de + a - lento): paranoid, sneaky, polite, normal, aggressive, insane. Alias: 0 a 5 (p.ej. -T4). Por defecto: Normal (3).

²⁶ Los tiempos se miden por defecto en segundos. Se pueden utilizar otras unidades de medida añadiendo los sufijos 'ms' (milisegundos), 's' (segundos), 'm' (minutos), o 'h' (horas). Por ejemplo 30m, 500ms o 2h.

SCRIPTING		
Opción	Funcionamiento	Observaciones
-sC	Incluye en el análisis actual el conjunto por defecto de scripts (algunos pueden ser intrusivos).	Equivalente a: <i>--script default</i>
--script <valor>	Define el/los script a utilizar. Valor puede ser un nombre de fichero, categoría, directorio, expresión, etcétera. Alias <i>all</i> ejecuta todos los script (peligroso).	Valores separados por comas. Prefijo + hace que se ejecuten aunque no corresponda.
--script-args <args>	Argumentos a pasar a los scripts. Formato: <nombre>=<valor>	Argumentos separados por comas. Prioridad sobre los definidos en fichero.
--script-args-file <file>	Carga argumentos de un fichero.	Por defecto, 5 intentos.
--script-help <valor>	Muestra ayuda sobre los scripts. Valores como --script.	
--script-trace	Símil de --packet-trace una capa ISO por encima. Muestra todas las comunicaciones realizadas por un script.	
--script-updatedb	Actualiza la BBDD de scripts existente.	Útil si se realizan cambios a la carpeta de scripts por defecto.

SALIDA			
Opción	Nombre	Funcionamiento	Observaciones
-oN <file>	Salida normal	Registra en un fichero una salida muy similar a la mostrada por pantalla en modo interactivo.	Debe definirse la extensión deseada (nmap para salida normal y gmap para la "grepable").
-oX <file>	Salida XML	Crea un fichero XML con los detalles del análisis. Se puede usar la plantilla XST incluida o cualquier reconocedor de XML para procesarla.	
-oS <file>	Salida Script Kiddie	Salida muy similar a la del modo interactivo, pero sustituyendo caracteres y capitalización para ajustarse al lenguaje utilizado por estos grupos en Internet como sello de identidad.	Se pueden aplicar formatos de tiempo al estilo <i>strftime</i> : %H, %M, %S, %m, %d, %y, %Y, %T, %R, %D.
-oG <file>	Salida "grepable"	Salida con formato especial que es fácilmente tratable con herramientas de consola como <i>grep</i> . Obsoleta.	
-oA <patrón>	Salida en todos los formatos	Crea un fichero para los tipos de salida normal, XML y "grepable", definidos anteriormente.	Sin extensión. Nmap usa el patrón definido y añade cada extensión.
-v[<nivel>]	Verbosidad	Aumenta la cantidad de información sobre el progreso del análisis que muestra Nmap por pantalla.	Para aumentar verbosidad se pueden añadir más v o incluir un número (p. ej. -vvv o -v3).
-d[<nivel>]	Depuración	Añade información de depuración a la salida que Nmap muestra por pantalla.	Se pueden añadir más d o incluir un número (p.ej -ddd o -d3) para aumentar el nivel de depuración.
--reason	Razón	Indica la razón por la que se ha concluido el estado de un puerto o equipo.	Permite diferenciar el tipo de respuestas que ha generado un puerto cerrado.
--stats-every <time>	Estadísticas	Indica cada cuanto tiempo se imprimen estadísticas sobre el tiempo restante del análisis.	Se imprime tanto por pantalla como en la salida XML.
--packet-trace	Traza de paquetes	Hace que Nmap imprima información sobre cada paquete que envía o recibe.	Incluye información de <i>--version-trace</i> y <i>--script-trace</i> .
--open	Puertos abiertos	Muestra en la salida los puertos identificados como (posiblemente) abiertos, obviando aquellos con otros estados (filtrados o cerrados).	Útil en grandes análisis para obtener listado de puertos alcanzables.
--iflist	Interfaces y rutas	Muestra únicamente el listado de interfaces y de rutas detectado por Nmap.	Útil para depuración.
--log-errors	Errores	Guarda los errores generados durante la ejecución del análisis en los ficheros de salida, además de mostrarlos por pantalla.	
--append-output	Ficheros de salida	Instruye a Nmap para añadir los resultados del análisis actual a un fichero de salida existente, en lugar de borrar el contenido de dicho fichero.	Puede causar problemas de tratamiento automático con ficheros XML.
--resume <file>	Continuar	Continúa un análisis Nmap en el punto en que se quedó, si se indica como parámetro un fichero generado con los modificadores -oN o -oG.	Interesante para análisis muy largos o que necesitan ser interrumpidos por causas de fuerza mayor.

--stylesheet <file>	Hoja de estilos	Indica que hoja de estilos XSL incrustar en la salida XML. Requiere ruta o URL completa.	Las hojas de estilos XSL definen como se traduce un fichero XML a uno HTML. Útil para visualizar informes en XML a través de un navegador web si el cliente no tiene Nmap instalado.
--webxml		Alias para <i>--stylesheet</i> http://nmap.org/svn/docs/nmap.xsl	
--no-stylesheet		Indica que no se incruste ningún enlace a hoja de estilos en la salida XML.	

MISCELÁNEA			
Opción	Nombre	Funcionamiento	Observaciones
-6	Ipv6	Habilita el análisis en redes IPv6.	
-A	Análisis agresivo	Alias para <i>-O -sV -sC --traceroute</i>	
--datadir <dir>	Directorio de datos	Indica el directorio de donde Nmap lee algunos ficheros que necesita para su uso (nmap-service-probes, nmap-services, nmap-protocols, nmap-rpc, nmap-mac-prefixes y nmap-os-db).	
--servicedb <file>	Fichero de servicios	Indica una localización personalizada para el fichero de donde Nmap obtiene la información sobre servicios.	Fichero nmap-services. Más prioritario que --datadir. Activa -F.
--versiondb <file>	Fichero de versiones	Indica la ubicación del fichero de donde Nmap obtiene las sondas que debe enviar para detectar servicios (-sV).	Fichero nmap-service-probes. Más prioritario que --datadir.
--send-eth	Escribir en tramas ethernet	Escribe directamente tramas a nivel Ethernet sin usar el API de red ni transporte. Por defecto se decide de forma dinámica el tipo de tramas a enviar.	Usado para evitar limitaciones de algunas implementaciones de la pila TCP/IP. Activada por defecto en la versión para Windows.
--send-ip	Escribir tramas IP	Escribe paquetes a nivel IP, y los pasa al sistema operativo para que este se encargue de enviarlos.	Complementaria de la opción anterior.
--privileged	Modo privilegiado	Asume que tiene suficientes permisos para realizar operaciones que requieren elevación de privilegios, como apertura de sockets RAW y captura de paquetes, entre otros.	Útil si se permite a usuarios sin privilegios realizar dichas acciones. Alternativa: Fijar la variable de entorno NMAP_PRIVILEGED.
--unprivileged	Modo sin privilegios	Opuesto al anterior. Asume que no se tienen privilegios para realizar operaciones privilegiadas.	Útil para pruebas o depuración.
--release-memory	Liberar memoria	Hace que Nmap libere toda su memoria antes de finalizar su ejecución. Normalmente es el SO quien hace esta tarea.	Facilita descubrimiento de filtraciones de memoria.
-V --version	Versión	Imprime la versión de Nmap y finaliza la ejecución.	
-h --help	Ayuda	Imprime la página de ayuda resumida.	Alias: Lanzar nmap sin argumentos.

INTERACCIÓN EN TIEMPO DE EJECUCIÓN	
Comando	Funcionamiento
d / V	Aumenta / Disminuye el nivel de verbosidad.
d / D	Aumenta / Disminuye la cantidad de información de depuración que se muestra.
p / P	Activa / Desactiva la traza de paquetes (--packet-trace).
?	Muestra la pantalla de ayuda de interacción en tiempo de ejecución.
Cualquier otro	Imprime mensaje con estado actual del análisis.