"SEGURIDAD EN LA FAMILIA DE PROTOCOLOS TCP/IP Y SUS SERVICIOS ASOCIADOS" Primera parte.

(Autor: Lic. JUAN JOSE TALENTI **–** correo electrónico: jjtalent@arnet.com.ar // Egresado del CMN como Subt A en 1986 - Docente en el IAC Filial Santa Fe - Beta Tester de Microsoft - Participa del Programa "Desarrollador Cinco Estrellas" que capacita a programadores de la plataforma .NET).

Resumen

En esta primera parte del trabajo se realiza una introducción al mundo de la seguridad de los sistemas de información desde la perspectiva y el enfoque que proporciona una de las familias de protocolos de comunicaciones más extendidas actualmente: TCP/IP. Se mencionan las tres leyes imperantes en la Seguridad Informática, aquellos equipos que tienen una implementación de la pila de protocolos TCP/IP nucleados en los Sistemas y los Dispositivos de Red, expresando que desde el punto de vista de la seguridad, la familia de protocolos TCP/IP puede ser vulnerada en base a dos conceptos inherentes a su diseño:

1. El formato de los paquetes de los diferentes protocolos y
2. El modo de funcionamiento de los protocolos, y

por último se explican brevemente los procedimientos posibles en el transcurso de una comunicación TCP/IP, como lo son el establecimiento y la terminación de la misma.

Luego se desarrolla una pequeña historia de las vulnerabilidades, donde se han definido y clasificado las tres generaciones de ataques en las redes existentes a lo largo del tiempo:

1. Primera generación - Ataque Físico,
2. Segunda generación - Ataque Sintáctico *(objeto del presente trabajo) y*
3. Tercera generación - Ataque Semántico.

Para finalizar esta primera parte se inicia el desarrollo del tema VULNERABILIDADES GENÉRICAS, explicándose una serie de ellas como lo son: Footprinting, Fingerprinting, Escaneo de Puertos, Escaneo basado en el protocolo ICMP, Sniffing, Eavesdropping, Snooping, IP Spoofing y SMTP Spoofing y Spamming.

Desarrollo

1. INTRODUCCIÓN.

El presente trabajo pretende únicamente ser una introducción al mundo de la seguridad de los sistemas de información desde la perspectiva y el enfoque que proporciona una de las familias de protocolos de comunicaciones más extendidas actualmente: TCP/IP.

Hoy en día, el uso de TCP/IP se ha extendido prácticamente a la totalidad de las redes de comunicaciones de datos, potenciado fundamentalmente por la expansión de Internet, así como de las redes corporativas y de cooperación asociadas a esta tecnología: *Intranets* y *Extranets*.

Habitualmente el diseño de las redes se basa en características como la funcionalidad o la eficiencia, pero no en la seguridad. Para la realización del análisis y diseño de una red segura es necesario conocer los detalles y características de los protocolos de comunicaciones subyacentes, que serán los encargados de transportar la información y datos que desean distribuirse. A su vez, deberán analizarse los servicios que se proporcionan en dicha red y sus detalles de funcionamiento.

Para sintetizar el problema, se pueden obtener 3 leyes imperantes en la Seguridad Informática actualmente:

- **Todo software tiene *bugs*.**
- **Todo software de seguridad tiene *bugs* de seguridad.**
- **Si el software no es utilizado, no se sabrá que *bugs* tiene realmente.**

La filosofía de seguridad de los *firewalls* se basa en ésta regla: **"*security through obscurity*"**.

La familia de protocolos TCP/IP (*Transport Control Protocol / Internet Protocol*) caracteriza un estándar *ad-hoc* de protocolos de comunicaciones entre sistemas informáticos. El protocolo TCP/IP surgió alrededor de 1960 como base de un sistema de comunicación basado en redes de conmutación de paquetes desarrollado por el gobierno estadounidense y la agencia de defensa, ARPA. Actualmente constituye la infraestructura tecnológica más extendida y desarrollada sobre la que circulan las comunicaciones electrónicas (datos, voz, multimedia...). Su expansión se ha debido principalmente al desarrollo exponencial de la red mundial INTERNET.

Dentro de los equipos que poseen una implementación de la pila de protocolos TCP/IP, se distinguen de forma más detallada dos grupos, todos ellos objetivo de los potenciales ataques:

Sistemas: son los equipos que engloban tanto a los clientes de un servicio o comunicación, ya sean PCs de escritorio o estaciones de trabajo (que ejecutarán un sistema operativo cliente: Windows, Unix, MacOS...) así como dispositivos móviles (PDAs, teléfonos móviles...), como a los servidores que proporcionan el servicio, típicamente ejecutando un sistema operativo servidor: Unix (incluyendo todas sus variantes: HP-UX, Linux, Solaris, AIX...), AS/400, Windows NT/2000/2003, Novell Netware. Principalmente serán estos últimos el objetivo principal de los atacantes, porque contienen información relevante.

Dispositivos de red: son los encargados de que el tráfico de red fluya dentro o entre redes. Por tanto engloban a los repetidores, puentes o *bridges*, concentradores o *hubs*, conmutadores o *switches*, encaminadores o *routers*, cortafuegos o *firewalls*, servidores de terminales y acceso (RAS) (que contienen un conjunto de módems o accesos RDSI), dispositivos

de almacenamiento (*storage appliance*).Los principales fabricantes son Cisco, 3Com, Lucent, Nortel, y HP.

Desde el punto de vista de la seguridad, la familia de protocolos TCP/IP puede ser vulnerada en base a dos conceptos inherentes a su diseño:

El formato de los paquetes de los diferentes protocolos: Aparte de la propia información trasportada, la información contenida en cada uno de los campos de las cabeceras de los protocolos proporciona una fuente muy valiosa de conocimiento.

El modo de funcionamiento de los protocolos: Las etapas asociadas a cada proceso en los protocolos, así como el método de actuación en las diferentes situaciones posibles, ofrecen la información necesaria para analizar la existencia de vulnerabilidades.

El protocolo más complejo es TCP, ya que se encarga de controlar el estado en todo momento de la comunicación mediante el concepto de conexión. El proceso determinista que debe llevarse a cabo dentro de una conexión viene condicionado por **el diagrama de estados de TCP**. Éste especifica las reglas que debe seguir toda implementación de TCP durante la transmisión de información, y denota los estados posibles de una conexión en cada momento, así como los paquetes que deben recibirse o enviarse para transitar de un estado a otro.

Estudiando los procedimientos posibles en el transcurso de una comunicación TCP, los principales son el establecimiento y terminación de la misma.

Establecimiento de una conexión TCP: Una conexión TCP se establece en tres pasos, lo que se denomina el three-way handshake, en el que el sistema que inicia la conexión o cliente (TCP A), envía un paquete de SYN (con su número de secuencia inicial asociado a esta conexión) al destinatario o servidor (TCP B); éste le responde con un paquete SYN-ACK, confirmándole la recepción del SYN inicial y enviándole su propio número de secuencia. Finalmente, el cliente reconoce la recepción del SYN del servidor mediante un ACK. En este momento la conexión se ha establecido y puede tener lugar toda la transferencia de datos.

Finalización de una conexión TCP: De igual modo, la finalización de la conexión se lleva a cabo mediante el intercambio de paquetes TCP, en este caso cuatro. El sistema que desea terminar la conexión envía un paquete de FIN notificándolo. El sistema remoto reconoce su recepción (mediante un ACK), e indica su interés de terminar con la conexión también mediante un paquete de FIN, que debe ser reconocido por el otro extremo (de nuevo con un ACK) para que la misma se de por cerrada.

Hecha éstas aclaraciones voy a proceder a explicar en detalle las vulnerabilidades que existen en TCP/IP.

2. HISTORIA DE LAS VULNERABILIDADES.

En los primeros años, los ataques involucraban poca sofisticación técnica. Los ataques internos se basaban en utilizar los permisos para alterar la información. Los externos se basaban en acceder a la red simplemente averiguando una clave válida. A través de los años se han desarrollado formas cada vez más sofisticadas de ataque para explotar vulnerabilidades en el diseño, configuración y operación de los sistemas. Esto permitió a los nuevos atacantes tomar control de sistemas completos, produciendo verdaderos desastres que en muchos casos llevaron a la desaparición de aquellas organizaciones o empresas con altísimo grado de dependencia tecnológica (bancos, servicios automáticos, etc).

Estos nuevos métodos de ataque han sido automatizados, por lo que en muchos casos sólo se necesita un conocimiento técnico básico para realizarlos. El aprendiz de intruso, *script-kiddie* o *ankle biter*, o aprendiz de *hacker*, *lamer* o *wannabee*, tiene acceso hoy en día a numerosos programas y *scripts* (*exploits*) que se aprovechan de las vulnerabilidades, disponibles desde numerosas fuentes *underground*, como *hacker newsgroups, mailing-lists* y *web sites*, donde además encuentra todas las instrucciones para ejecutar ataques con las herramientas disponibles.

Bruce Schneier, en numerosos artículos, ha definido y clasificado las generaciones de ataques en la redes existentes a lo largo del tiempo:

La primera generación - Ataque Físico: Ataques que se centraban en los componentes electrónicos: ordenadores y cables. El objetivo de los protocolos distribuídos y de la redundancia es la tolerancia frente a un punto único de fallo. Son mayormente problemas para los que actualmente se conoce la solución.

La segunda generación - Ataque Sintáctico *(objeto del presente trabajo):* Las pasadas décadas se han caracterizado por ataques contra la lógica operativa de los ordenadores y las redes, es decir, pretenden explotar las vulnerabilidades de los programas, de los algoritmos de cifrado y de los protocolos, así como permitir la denegación del servicio prestado. En este caso se conoce el problema, y se está trabajando en encontrar soluciones cada vez más eficaces.

La tercera generación - Ataque Semántico: Se basan en la manera en que los humanos asocian significado a un contenido. El hecho es que en la sociedad actual la gente tiende a creerse todo lo que lee (medios informativos, libros, la Web...). El inicio de este tipo de ataques surgió con la colocación de información falsa en boletines informativos o *e-mails*, por ejemplo, para beneficiarse de las inversiones dentro de la bolsa financiera, obtener un rédito político, etc. También pueden llevarse a cabo modificando información caduca.

Esta generación de ataques se lleva a su extremo si se modifica el contenido de los datos de los programas de computadora, que son incapaces de cotejar o sospechar de su veracidad, como por ejemplo la manipulación del sistema de control de tráfico aéreo, el control de un coche inteligente, la base de datos de los libros más vendidos o de índices bursátiles como el NASDAQ. Lo más curioso es que estos ataques han existido fuera del entorno informático desde hace muchos años como estadísticas manipuladas, falsos rumores, pero es la tecnología la que potencia su difusión.

Su solución pasará no sólo por el análisis matemático y técnico, sino también por el humano.

La conclusión tras el análisis de las vulnerabilidades desde un punto de vista operacional es que para evitarlas pueden definirse las tareas a realizar dentro de un sistema de seguridad en tres etapas:

- **Prevención**: implementada por dispositivos como los *firewalls*.
- **Detección**: a través de sistemas como los IDS.
- **Respuesta**: las acciones a tomar deben ser dirigidas por la parte humana, típicamente los administradores de la red.

3. VULNERABILIDADES GENÉRICAS.

Las vulnerabilidades pretenden describir las debilidades y los métodos más comunes que se utilizan para perpetrar ataques a la seguridad de la familia de protocolos TCP/IP (confidencialidad, integridad y disponibilidad de la información).

Los ataques pueden estar motivados por diversos objetivos, incluyendo fraude, extorsión, robo de información confidencial, venganza, acceso no autorizado a un sistema, anulación de un servicio o simplemente el desafío de penetrar un sistema.

Éstos pueden provenir principalmente de dos fuentes:

Usuarios autentificados: al menos a parte de la red, como por ejemplo empleados internos o colaboradores externos con acceso a sistemas dentro de la red de la empresa. También denominados *insiders*.

Atacantes externos: a la ubicación física de la organización, accediendo remotamente. También denominados *outsiders*.

Los métodos de ataque descritos se han dividido en categorías que pueden estar relacionadas entre sí, ya que el uso de un método permite o facilita el uso de otros, en ocasiones, complementarios. Un ejemplo de ataque podría ser la realización del análisis de un sistema, mediante *fingerprinting*, tras el cual es posible explotar una vulnerabilidad como un *buffer-overflow* de un servicio TCP/IP, enviando paquetes que parecen válidos mediante *IP spoofing*. Dentro de los métodos no se han incluído ataques de alto nivel, como por ejemplo la distribución y ejecución de virus a través del correo electrónico (protocolo SMTP), ya que afectan a vulnerabilidades particulares de las aplicaciones y los lenguajes de programación soportados por éstas.

En numerosas ocasiones utilizo inicialmente el término inglés para nombrar la vulnerabilidad, ya que es como se conoce comúnmente, para posteriormente asociarle su posible traducción al español.

Las vulnerabilidades pueden clasificarse según dos criterios:

Número de paquetes a emplear en el ataque:

Atomic: se requiere un único paquete para llevarla a cabo.

Composite: son necesarios múltiples paquetes.

Información necesaria para llevar a cabo el ataque:

Context: se requiere únicamente información de la cabecera del protocolo.

Content: es necesario también el campo de datos o *payload*.

Context	*Ping of death* *Land attack* *WinNuke*	*Port scan* *SYN Flood* *TCP hijacking*
Content	*DNS attack* *Proxied RPC* *IIS attack*	*SMTP attacks* *String matches* *Sniffing*
	Atomic	**Composite**

3.1. Footprinting.

La regla número uno antes de planificar o analizar un posible ataque a un sistema, o red, es conocer el objetivo, es decir, obtener su huella identificativa o *footprinting* – el arte de extraer toda la información posible de la red objetivo del ataque. Por tanto la primera tarea a realizar pasa por dedicar un esfuerzo considerable a obtener y recolectar ésta información. Existen numerosas utilidades para obtener la información de un sistema: ping, whois, finger, rusers, nslookup, rcpinfo, telnet, dig, nmap.

El atacante podría comenzar por ejecutar un ping contra el sistema a atacar:

```
$ ping www.sistema.ar
```

Para comprobar su existencia (también el uso de traceroute o de nslookup permiten obtener información si los paquetes ICMP están deshabilitados). Posteriormente podría intentar extraer información del sistema y sus usuarios mediante la utilidad finger:

```
$ finger root@www.sistema.ar
```

Pudiendo realizar diversas pruebas tratando de descubrir algún usuario válido, para posteriormente intentar adivinar su clave y disponer de acceso remoto al sistema mediante telnet.

La primera etapa sería buscar información general de la empresa en INTERNET. Los pasos que podrían realizarse a continuación irían desde realizar búsquedas en ICANN para obtener el rango de redes asociado a la organización,

como descubrir qué sistemas activos existen (mediante *pings*), para posteriormente conocer los posibles servicios vulnerables, empleando para ello la técnica de escaneo de puertos. Finalmente, a través de una conexión, por ejemplo mediante *telnet*, al puerto seleccionado se podrá obtener información en la mayoría de los casos, como cadenas de texto (*banner grabbing*), que identifiquen el servicio, y que permitirán conocer el tipo de servidor y su versión. Está técnica puede aplicarse mediante el uso de utilidades como *telnet* o *netcat*.

Ejemplos de cadenas identificativas:

$ ftp 10.10.10.10

Connected to 10.10.10.10

220 *hostname* FTP server (Version 1.1.214.2 Mon May 11 12:21:14 GMT 1998) ready.

User (10.10.10.10:(none)):

$ telnet 10.10.10.10

HP-UX *hostname* B.11.00 A 9000/712 (t0)

login:

La información a obtener de un objetivo se puede clasificar en 4 grupos principales: Internet, Intranet, Extranet y Acceso Remoto. A continuación se analizan diversos métodos empleados:

A través de una búsqueda en los grupos de noticias (USENET), un atacante puede obtener información emitida por los usuarios de una organización, y así conocer detalles de los sistemas existentes en la misma, de las tecnologías empleadas y de la relevancia de la seguridad por parte de los administradores, con el objetivo de obtener el perfil de la organización. Para ello basta con buscar por la cadena "@dominio.com" en www.dejanews.com. Asimismo, es recomendable buscar información relacionada con la organización en meta buscadores o en los grandes buscadores de Internet.

El análisis tanto de las páginas Web como de las fuentes correspondientes a éstas páginas HTML pueden proporcionar información interesante, principalmente las etiquetas de comentarios:<, ! o --. Para realizarlo se podrán emplear herramientas que permite realizar una copia completa de un servidor, como *Wget*, *Teleport* o *Web Snake*.

La información relativa a los dominios asociados a una organización así como sus subredes correspondientes puede obtenerse en los servicios WHOIS.

Otro de los servicios que proporciona información muy útil es el servicio de nombres o DNS. Si el servicio no se ha configurado adecuadamente, será posible realizar una consulta de transferencia de zona completa, lo que permitirá obtener toda la información de traducción de direcciones IP a nombres de máquinas. Este tipo de consulta puede realizarse con la utilidad *Nslookup*:

$ nslookup

Default Server: dns.dominio.com

Address: 30.1.1.1

> server 100.100.100.1

Default Server: [100.100.100.1]

Address: 100.100.100.1

> set type=any

> ls –d dominio_objetivo.com. > /tmp/fichero_zona

La información de la zona puede permitir obtener relaciones entre sistemas, el propósito para el que se emplean los mismos, el sistema operativo que ejecutan o el tipo de sistema que ocultan tras la nomenclatura: todo ello en base al nombre o a ciertos registros propios del servicio DNS.

Por otro lado, el DNS permite conocer los servidores de correo asociados a una organización a través de los registros MX. Para obtenerlos basta con recurrir de nuevo a la utilidad *Nslookup*:

$ nslookup

Default Server: dns.dominio.com

Address: 30.1.1.1

> server 100.100.100.1

Default Server: [100.100.100.1]

Address: 100.100.100.1

> set type=mx

> dominio_objetivo.com.

Por ejemplo, los servidores encargados de la recepción del correo del dominio *yahoo.com* son:

> yahoo.com

Name Server: dnsserver.dominio.com

Address: 100.100.100.1

Trying DNS

Non-authoritative answer:

yahoo.com preference = 1, mail exchanger = mx2.mail.yahoo.com

yahoo.com preference = 1, mail exchanger = mx3.mail.yahoo.com

yahoo.com preference = 9, mail exchanger = mta-v18.mail.yahoo.com

yahoo.com preference = 1, mail exchanger = mx1.mail.yahoo.com

Authoritative answers can be found from:

yahoo.com nameserver = NS3.EUROPE.yahoo.com

yahoo.com nameserver = NS1.yahoo.com

yahoo.com nameserver = NS5.DCX.yahoo.com

mx3.mail.yahoo.com internet address = 216.115.107.17

mx3.mail.yahoo.com internet address = 216.136.129.16

mx3.mail.yahoo.com internet address = 216.136.129.17

NS3.EUROPE.yahoo.com internet address = 217.12.4.71

NS1.yahoo.com internet address = 204.71.200.33

NS5.DCX.yahoo.com internet address = 216.32.74.10

Simplemente añadir que existen numerosas utilidades para automatizar la extracción de información del DNS, como por ejemplo, *Host, Sam Spade, Axfr, Dig*.

3.1.1. Traceroute.

Toda red está caracterizada por una topología o distribución, tanto física como lógica, concreta. Existe una herramienta que ayuda a la obtención de ésta: *Traceroute*, creada originalmente para solucionar problemas en una red. Esta técnica permite saber todos los sistemas existentes en un camino entre dos equipos.

Su funcionamiento se basa en el manejo del campo TTL de la cabecera IP de un paquete, de forma que es capaz de determinar uno a uno los saltos por los que un determinado paquete avanza en la red. El campo TTL actúa como un contador de saltos, viéndose decrementado en uno al ser reenviado por cada *router*. Por tanto, mediante esta utilidad de diagnóstico se podrá obtener una lista de los elementos de red recorridos desde una ubicación origen hasta un sistema destino. Los paquetes de comprobación son enviados de tres en tres.

El primer datagrama enviado tiene un TTL de valor 1, por lo que generará en el primer salto un paquete *ICMP Time Exceded*. El siguiente datagrama verá el valor del TTL incrementado en uno, por lo que será capaz de llegar un salto más en la red que el datagrama anterior. Debe tenerse en cuenta que el comportamiento de la herramienta puede variar en función de la plataforma: el *Traceroute* de Unix, por ejemplo Linux o HP-UX, utiliza el protocolo UDP (pudiendo usar ICMP mediante la opción “-I”). En el caso de Windows NT (*Tracert*), se emplea el protocolo ICMP. Por tanto, en función de los filtros existentes en los dispositivos de red que deben atravesarse, será necesario usar uno u otro protocolo. En el caso de UDP, el mensaje generado por el sistema final es *ICMP Port Unreachable*, mientras que en los sistemas intermedios, al igual que en todos los sistemas en el caso de ICMP, es un *ICMP Echo Reply*.

En el caso de la implementación basada en UDP, el número de puerto UDP se incrementa en cada iteración, por lo que es necesario conocer el puerto UDP con el que se debe enviar en base a una fórmula: **(Puerto objetivo – (nº. de saltos * nº. de pruebas)) –1**.

El principal problema de esta versión es que si existe un *firewall*, éste filtrará los paquetes UDP al ir con diferentes puertos. Existe una implementación de Michael Shiffman que evita estos incrementos en los puertos; presenta el problema de que si en ese puerto UDP hay un servicio disponible, no se generará el paquete ICMP de vuelta.

Ejemplo de *Traceroute*:

traceroute 15.13.120.190

traceroute to 15.13.120.190 (15.13.120.190), 30 hops max, 20 byte packets

1 pop (15.128.104.12)		5 ms	
15.128.104.1 (15.128.104.11)	6 ms		3 ms
2 15.191.224.13 (15.191.224.113)	126 ms	141 ms	133 ms
3 15.191.40.1 (15.191.40.11)		134 ms	152 ms 129 ms
4 172.16.8.37 (172.16.8.137)		248 ms	273 ms 253 ms

```
5 172.17.192.49 (172.17.192.149)          314 ms  324 ms  302 ms
6 15.73.152.2 (15.73.152.12)                      313 ms  304 ms  308 ms
7 15.68.136.3 (15.68.136.13)                      304 ms  325 ms  326 ms
8 15.61.211.83 (15.61.211.183)            347 ms  369 ms  312 ms
9 15.75.208.38 (15.75.208.138)            296 ms  273 ms  309 ms
10 15.13.120.187 (15.13.120.190)          305 ms  285 ms  282 ms
#
```

Existen herramientas gráficas con una funcionalidad similar a *Traceroute*, que permiten visualizar el mapa mundial con las correspondientes asociaciones de cada elemento IP y su ubicación física.

Asimismo, mediante la utilización de paquetes ICMP (ECHO y REPLY, *Ping Sweep*) se puede obtener la lista de dispositivos IP activos. Existen herramientas que facilitan la obtención de este tipo de información: *Fping*, *Gping* y *Pinger*. Asimismo, mediante la utilidad "*Nmap*" con la opción "-sP" se obtienen resultados similares.

Este protocolo permite obtener también información de otro tipo, como la franja horaria del sistema destino o la máscara de subred empleada en los diferentes subinterfaces (paquetes ICMP de tipo 13 y 17 respectivamente).

Por último, en el caso de disponer del demonio *Fngerd* (puerto TCP 79), el sistema operativo del *host* puede ser identificado en muchos casos a través de una petición *Finger* del tipo "root@host, bin@host o daemon@host". Este servicio se creó en los comienzos de Internet para obtener información de contacto de los usuarios de un sistema, cuando la seguridad no era un hecho a tener en cuenta.

Otros comandos que permiten información de este servicio son: "finger –l @objetivo.com y finger 0@objetivo.com". Por ejemplo:

```
# finger root@host.dominio.com
[host.dominio.com]
Login name: root     (messages off)
Directory: /                        Shell: /sbin/sh
On since May 23 13:49:41 on pts/ta from sistema.dominio.com
New mail received Thu May 10 16:18:39 2004;
          unread since Sun May 23 13:49:42 2004
No Plan.
#
# finger -l @ host.dominio.com
[host.dominio.com]
Login name: root (messages off)
Directory: /              Shell: /sbin/sh
On since May 23 13:49:41 on pts/ta from sistema.dominio.com
1 minute 38 seconds Idle Time
New mail received Thu May 10 16:18:39 2001;
unread since Wed May 23 13:49:42 2001
No Plan.
#
```

Otra técnica empleada sobre los servidores de correo electrónico para obtener información del sistema y la red destino es mediante la ejecución del comando SMTP **"expn <user>"**.

Existen herramientas integradas cuyo objetivo es aunar las diferentes técnicas presentadas en este apartado inicial, de forma que se obtenga toda la información posible de un entorno de red.

3.2. Fingerprinting.

Una técnica más específica que permite extraer información de un sistema concreto es el *fingerprinting*, es decir, la obtención de su huella identificativa respecto a la pila TCP/IP. El objetivo primordial suele ser obtener el sistema operativo que se ejecuta en la máquina destino de la inspección. Esta información junto con la versión del servicio o servidor facilitará la búsqueda de vulnerabilidades asociadas al mismo. Gran parte de la información de la pila TCP/IP puede obtenerse en base al intercambio en tres pasos propio del protocolo TCP/IP (*TCP three-way handshake*). La probabilidad de acierto del sistema operativo remoto es muy elevada, y se basa en la identificación de las características propias de una implementación de la pila TCP/IP frente a otra, ya que la interpretación de los RFCs no concuerda siempre. Para poder aplicar esta técnica con precisión es necesario disponer de un puerto abierto (TCP

y/o UDP). Las diferentes pruebas a realizar para diferenciar los sistemas operativos son:

FIN probe: Al enviarse un paquete de FIN el sistema remoto no debería responder, aunque implementaciones como la de Windows NT devuelven un FIN-ACK.

Bogus flag probe: Se activa un *flag* TCP aleatorio en un paquete SYN. La respuesta de implementaciones como Linux devuelven un SYN-ACK con el mismo *flag* activo.

ISN sampling: Pretende encontrarse un patrón empleado por la implementación para seleccionar los números iniciales de secuencia (ISN) de una conexión TCP.

Monitorización del "Don´t fragment bit": Se analiza si el sistema operativo establece por defecto el *bit* de no fragmentación (DF) como activo o no.

Tamaño de ventana TCP inicial: El tamaño de ventana empleado por defecto en cada implementación es muy particular y ayuda a descubrir de cual puede tratarse.

Valor de ACK: El valor del número de secuencia asignado en el campo ACK diferencia también la implementación, ya que algunas devuelven el valor recibido como número de secuencia mientras que otras lo incrementan en uno.

Mensaje de error de ICMP quenching: El RFC 1812 determina que el control de flujo de mensajes de error debe limitarse. Al enviar un paquete UDP a un número elevado de puerto, aleatoriamente, se puede medir el número de mensajes de tipo *unreachable* por unidad de tiempo.

ICMP message quoting: Los comentarios añadidos a los mensajes de error ICMP varían en función del sistema operativo.

Mensajes de error ICMP-integridad: Las cabeceras IP pueden ser alteradas por las diferentes implementaciones al devolver mensajes de error ICMP. Un análisis exhaustivo de los cambios en las cabeceras puede permitir determinar el S.O.

TOS (Tipo de Servicio): Ante los mensajes "*ICMP port unreachable*" puede examinarse el campo TOS, que suele ser cero pero puede variar.

Gestión de la fragmentación: El manejo de los paquetes fragmentados que se superponen es gestionado de forma particular por cada pila: al reensamblar los fragmentos, algunas sobrescriben los datos más antiguos con los nuevos y viceversa.

Opciones TCP: Los RFCs 793 y 1323 definen las opciones TCP posibles. Mediante el envío de paquetes con muchas opciones avanzadas activas (*no operation*, MSS, *Window scale factor*, *timestamps*.) puede descubrirse el comportamiento de cada S.O.

Dos de las herramientas que facilitan esta tarea son NMAP y QUESO. Mientras que la funcionalidad de la primera es muy amplia, la segunda sólo se aplica a la aplicación de esta técnica (identificación de sistemas a través del comportamiento de la pila TCP/IP).

Dentro de las técnicas de identificación de un sistema existen otras, denominadas pasivas, que no se basan en enviar paquetes al sistema a atacar. Para ello monitorizan el tráfico asociado al sistema, y en función de los atributos y características de los paquetes, principalmente de las cabeceras TCP, determinan su origen:

TTL: ¿cuál es el valor del campo *Time To Live* (TTL) en los paquetes salientes?

Tamaño de ventana: ¿cuál es el valor fijado por el S.O.?

TOS: ¿se fija algún valor para el campo Tipo de Servicio, TOS?

DF: ¿se activa o no el *bit* de no fragmentación?

3.3. Escaneo de puertos.

Una vez que se dispone de los dispositivos a nivel IP activos en una red (por ejemplo, mediante ICMP), puede aplicarse a cada uno de ellos una técnica, centrada en la posterior búsqueda de vulnerabilidades, basada en una exploración de escaneo de puertos abiertos, tanto UDP como TCP.

El escaneo es la determinación de las características de una red o sistema remotos, con el objetivo de identificar los equipos disponibles y alcanzables desde Internet, así como los servicios que ofrece cada uno. Permite saber los sistemas existentes, los servicios ofrecidos por ellos, cómo están organizados los equipos, que sistemas operativos ejecutan, cual es el propósito de cada uno.

De forma general, entre los métodos de escaneo se incluyen técnicas como:

- **Ping sweep.**
- **Escaneo de puertos.**
- **Firewalking.**
- **Trace routing.**
- **Identificación de Sistema Operativo.**

Al escanear los puertos de los sistemas se descubren puntos de entrada a los mismos, que abren las puertas a nuevas vulnerabilidades potenciales en base a la implementación del servidor que escucha tras cada puerto. Además, esta técnica también permite identificar el tipo de sistema existente, así como su sistema operativo, y las aplicaciones

que ofrecen un servicio en la red, así como su versión asociada.

La herramienta por excelencia para realizar un escaneo de puertos es NMAP. Las técnicas existentes en el proceso de escaneo emplean diferentes procedimientos para descubrir la información del servicio:

TCP connect scan: Mediante el establecimiento de una conexión TCP completa (3 pasos).

TCP SYN scan: Se abren conexiones a medias, ya que simplemente se envía el paquete SYN inicial, determinando la existencia de un servicio si se recibe del sistema objetivo un SYN-ACK. Si, por el contrario, se recibe un RST-ACK, es que no existe un servicio. En el caso de la existencia de un servicio, se envía un RST-ACK para no establecer conexión alguna, y no ser registrados por el sistema objetivo, a diferencia del caso anterior.

Estos dos tipos funcionarán en todos los sistemas con implementaciones TCP/IP, mientras que los siguientes variarán según la implementación particular:

TCP FIN scan: Al enviar un FIN a un puerto, RFC 793, debería recibirse como resultado un paquete de *reset* sí está cerrado (se aplica principalmente a las pilas TCP/IP de Unix).

TCP Xmas Tree scan: Esta técnica es similar a la anterior, obteniéndose como resultado también un RST si el puerto está cerrado. En este caso se envían paquetes FIN, URG y PUSH.

TCP Null Scan: En el caso de poner a cero todos los *flags* de la cabecera TCP, debería recibirse de nuevo como resultado un paquete RST en los puertos no activos.

TCP ACK scan: Mediante este procedimiento puede determinarse si un *firewall* es simplemente de filtro de paquetes (manteniendo el tráfico de sesiones abiertas, caracterizadas por el *flag* ACK) o si mantiene el estado, con un sistema de filtro de paquetes avanzado.

TCP window scan: Mediante una anomalía en ciertas implementaciones en como se muestra el tamaño de ventana TCP, puede saberse si un puerto está abierto o si es o no filtrado.

TCP RPC scan: Es una técnica propia de sistemas Unix que permite conocer puertos de llamadas a procedimientos remotos (RPCs), junto al programa asociado a los mismos y su versión.

UDP scan: Al enviarse un paquete UDP a un puerto destino, puede obtenerse como resultado un paquete ICMP de puerto inalcanzable (*port unreachable*), con lo que se determina que el puerto no está activo. En caso contrario, no se recibirá ese mensaje. Debido a que UDP es no orientado a conexión, la fiabilidad de éste método depende de numerosos factores (más aún en Internet), como son la utilización de la red y sus recursos, la carga existente, la existencia de filtros complejos. Asimismo y a diferencia de los escaneos TCP, se trata de un proceso lento, ya que la recepción del mencionado paquete se rige por el vencimiento de temporizadores. Mediante pruebas UDP es posible determinar si un sistema está o no disponible, así como sus servicios UDP. Para ello se envían datagramas UDP con 0 *bytes* en el campo de datos. En el caso de que el puerto esté cerrado, se recibirá un mensaje *ICMP Port Unreachable*.

Si está abierto el puerto, no se recibirá ninguna respuesta. En el caso en el que se detecten un elevado número de puertos UDP abiertos, podrá indicar que existe un dispositivo de filtrado entre el atacante y el objetivo. Para confirmar esta última posibilidad, se puede enviar un paquete UDP al puerto cero, lo que debería generar una respuesta ICMP como la ya comentada. Si no se recibe ninguna respuesta quiere decir que hay un dispositivo filtrando el tráfico.

Este tipo de prueba suele ser detectada por los IDS. Existen otras comprobaciones más centradas en el nivel de aplicación, como pueden ser la extracción de los servicios RPC existentes a través del *portmapper*, la obtención de listados de sistemas de archivos compartidos a través de *nfsd*, *Samba* o *NetBios*, el escaneo de vulnerabilidades de CGIs en los servidores Web, así como de versiones conocidas de servicios típicos: *Sendmail, IMAP, POP3, RPC status y RPC mountd*. Algunas herramientas existentes (las más populares y reconocidas a lo largo del tiempo), aparte de NMAP que permiten aplicar algunas de estas técnicas se muestran a continuación.

El análisis de todas ellas supondría la elaboración de un extenso estudio, por lo que más información sobre las características y capacidades de cada una puede ser obtenida de las referencias asociadas.

Nmap (Unix): analiza tanto UDP como TCP, además de poseer numerosas funcionalidades.

NmapNT (Windows): versión de NMAP para Windows.

SuperScan (Windows): escáner de puertos TCP.

NetScan Tools Pro 2000 (Windows): incluye utilidades no solo para el escaneo de puertos.

Strobe (Unix): solo permite el análisis de servicios TCP.

Udp_scan (Unix): añade a la anterior el soporte de UDP (originalmente contenida en SATAN).

Netcat o nc (Unix y NT): además de otras funcionalidades añadidas es aplicable a UDP y TCP.

WinScan (Windows): escáner de puertos TCP, en modo texto y gráfico.

WUPS (Windows): escáner de puertos UDP.

ADMHack.

Security Analyzer (Windows): NetIQ.

Por ejemplo, mediante Nmap pueden realizarse las siguientes acciones de identificación de sistemas:

Descubrimiento de direcciones IP activas mediante un escaneo de la red:

nmap –sP <<rango_direcciones_IP>>

Escaneo de puertos TCP activos:

nmap –sT <<rango_direcciones_IP>>

Escaneo de puertos UDP activos:

nmap –sU <<rango_direcciones_IP>>

Intento de obtención del sistema operativo de un equipo en red:

nmap –O <<rango_direcciones_IP>>

Utilidades de escaneo de vulnerabilidades:

NESSUS (Unix y Win): Permite el análisis de vulnerabilidades conocidas sobre un sistema.

SATAN (Security Administrator Tool for Analyzing Networks, Unix).

SAINT (nuevas versiones de SATAN).

SARA: Security Auditor's Research Assistant.

TITAN: Escaner de vulnerabilidades para Solaris.

Este tipo de utilidades permite comprobar si un sistema es vulnerable a un conjunto muy amplio de problemas de seguridad encontrados en el pasado e incluidos en la base de datos de las diferentes aplicaciones, alertándonos sobre su existencia y su posible solución.

Las comprobaciones afectan a un gran número de servicios asociados a la pila TCP/IP. Algunas utilidades, como SATAN, fueron las precursoras inicialmente de los sistemas de ataque y protección actuales, pero hoy en día no tienen utilidad al ser más lentas e intentar explotar vulnerabilidades reparadas actualmente.

Mediante la ayuda de la utilidad Nmap pueden analizarse todas las técnicas empleadas por la herramienta, que por su innovación y complejidad, representan las principales utilizadas hoy en día. Una vez que el atacante dispone de la lista de sistemas externos e internos a su alcance, empleará las herramientas mencionadas para analizar su comportamiento y uso.

3.4. Escaneo basado en el protocolo ICMP.

Una vez conocido el propósito original del protocolo ICMP, notificar errores y condiciones inusuales que requieren atención respecto del protocolo IP, y el formato de sus paquetes, es necesario analizar los usos indebidos que se le pueden dar, todos asociados al escaneo de un sistema remoto.

De manera excepcional, se incluirán en cada una de las técnicas basadas en ICMP los métodos para su detección.

3.4.1. ICMP Echo (Ping sweep).

Mediante esta técnica se pretenden identificar los equipos existentes en las redes objetivo de un ataque, típicamente accesibles desde Internet. Constituye uno de los pasos principales en la obtención de información.

Empleando para ello los paquetes ICMP de tipo *echo* (8) y *echo reply* (0), se sabrá si una determinada dirección IP está o no activa. Se envía un paquete de tipo *echo*, y si se recibe el paquete de *echo reply* es que dicha dirección está siendo utilizada. La técnica envía numerosos paquetes de este estilo para conocer todos los equipos disponibles en una subred.

Existen numerosas herramientas que implementan este proceso, como por ejemplo, *Fping*, *Gping*, el propio *Nmap*, o la utilidad *Pinger* de Rhino9. Para detectar este tipo de paquetes enviados de forma masiva puede analizarse el log del servidor DNS asociado al dominio escaneado, ya que aparecerán múltiples intentos de resolución de nombres de direcciones IPs consecutivas. Asimismo, se podrá obtener la dirección IP del atacante. Los sistemas IDS también permiten su detección, tanto cuando se usa de forma secuencial, como cuando se lanzan los *pings* en paralelo.

3.4.2. ICMP Broadcast.

Cuando se envía un paquetes ICMP *echo* a la dirección de *broadcast* o a la dirección de red, con un único paquete enviado se consigue que todos los equipos respondan con su *echo reply* asociado.

Las implementaciones de los diferentes sistemas operativos se comportan de manera diferente. Esta técnica puede emplearse en las variantes de Unix, pero los SO de Microsoft, Windows, no responden a este tipo de paquetes. El RFC1122 especifica que este comportamiento mencionado en último lugar debería ser el correcto: "*if we send an ICMP echo request to an IP Broadcast or IP multicast addresses it may be silently discarded by a host*".

Existen técnicas de escaneo más avanzadas basadas en ICMP, pero NO en los paquetes de tipo *echo*. Podrían considerarse técnicas tanto de *ICMP Sweep* como de *ICMP Broadcast*, pero con otros tipos de paquetes ICMP, no *echo*. Estos paquetes se van a a analizar a continuación.

3.4.3. ICMP Timestamp.

Mediante el envío de un paquete ICMP de tipo *timestamp*, si un sistema está activo, se recibirá un paquete de *timestamp reply* indicando que implementa este tipo de transferencia de información que permite conocer la referencia de tiempo en el sistema destino. Tal y como denota el RFC 1122, la decisión de responder a estos

paquetes depende de la implementación. Algunos sistemas Windows sí responden mientras que otros no, sin embargo la mayoría de los Unix sí que lo implementan.

3.4.4. ICMP Information.

El propósito de los paquetes ICMP de información y su respuesta asociada, *information reply*, es permitir que ciertos equipos que no poseían disco del que extraer su propia configuración, pudieran autoconfigurarse en el momento de su arranque, principalmente para obtener su dirección IP. En el paquete, tanto la dirección origen como destino tienen el valor cero.

Tanto el RFC 1122 como el 1812 indican que los sistemas no deberían generar ni responder a este tipo de paquetes, pero la realidad de las implementaciones existentes es otra. Algunos sistemas operativos responderán cuando la dirección IP destino del paquete tiene el valor de una dirección IP específica.

En la respuesta, en lugar de tener la dirección IP de la red en el campo de dirección origen, se tiene la dirección IP del *host*. Algunos UNIX comerciales y equipos Cisco implementan la respuesta ante este tipo de paquetes.

3.4.5. ICMP Address Mask.

El propósito de los paquetes de tipo *Address Mask* y *Address Mask Reply*, era que los equipos o estaciones de trabajo sin disco pudiesen obtener la máscara de red asociada a la subred en la que estaban conectados en el momento de arrancar.

Se supone que un sistema no debería responder con un paquete de este tipo salvo que fuera un agente autorizado para notificar la máscara, típicamente el *router* de la subred. Mediante esta información, un atacante puede conocer la estructura interna de la red y el esquema de enrutamiento empleado.

Asimismo, permite identificar los *routers* existentes en el camino que une al atacante con la red objetivo. Un ejemplo de SO que “ayuda” mucho en este aspecto es Sun Solaris y versiones personales de Windows. Es posible emplear técnicas de detección de equipos más avanzadas, no ya en función del tipo de paquete ICMP, sino en base al comportamiento de las implementaciones del protocolo ICMP. Para ello se analizarán los mensajes de error de ICMP, generados desde las máquinas que sirven como prueba, lo que nos permitirá saber si existe algún dispositivo de filtrado presente, así como descubrir la configuración de las listas de acceso empleadas.

De manera general, los métodos a emplear incluyen:

- Modificación maliciosa de la cabecera IP de un paquete, por ejemplo cambiando el campo de la longitud de la cabecera, o los campos de opciones del protocolo IP.
- Uso de valores inválidos en los campos de la cabecera IP.
- Posibilidad de abusar de la fragmentación.
- Emplear el método de escaneo basado en el protocolo UDP: es el protocolo ICMP el que se encarga de notificar las anomalías de éste.

3.4.6. IP Bad Headers Fields:

Fijando un valor incorrecto de los campos de la cabecera IP, se pretende obtener de la máquina objetivo un mensaje ICMP de error: *ICMP Parameter Problem*. Este mensaje se obtiene cuando un *router* o sistema procesa un paquete y encuentra un problema con los parámetros de la cabecera IP que no está contemplado en ningún otro mensaje de error ICMP. Es enviado sólo si el paquete es descartado debido al error. Los *routers* deberían generar este tipo de error, pero no todos ellos comprueban que ciertos campos de la cabecera IP son correctos.

Las comprobaciones varían en función del *router*, por lo que es posible según su comportamiento identificar el fabricante del mismo. Por ejemplo, típicamente comprueban el *checksum*, y si no es correcto, descartan el paquete. Los sistemas implementan generalmente la verificación de versión IP, y si no es 4, descartan el paquete. Igualmente, comprueban el valor del *checksum*, para protegerse frente a errores producidos en el transporte de los datos por la red. Un atacante empleará esta funcionalidad para escanear el rango de IPs completo asociado a una subred.

En el caso de que exista un *firewall* protegiéndola, mediante el envío de paquetes falsos a los puertos que supuestamente deberían estar abiertos, como HTTP(80), FTP(21), DNS(53), Sendmail(25)..., se podrá saber si hay algún equipo. Los sistemas IDS deberían avisar de este tipo de tráfico anormal.

En el caso de querer descubrir la configuración de ACLs existente, deberá escanearse todo el rango de IPs con todos los protocolos y puertos posibles, de forma que se obtenga la visión más detallada posible de la topología y servicios de la red. Esto permitirá saber las ACLs empleadas, al poder visualizar que tráfico pasa y cual no. Una red se puede proteger frente a este ataque si los *firewalls* o *screening routers* se encargan de verificar y descartar este tipo de errores, no permitiendo este tipo de tráfico. Asimismo, si el dispositivo de filtrado no implementa esta característica, es posible filtrar los paquetes *ICMP Parameter Problem* en su camino de vuelta. Existe una herramienta, ISIC: *IP Stack Integrity Check*, de Mike Frantzen, disponible para este tipo de pruebas, que permite poner a prueba la pila TCP/IP, encontrar debilidades en un *firewall*, y comprobar la implementación de *firewalls* e IDS. Permite especificar si los paquetes se fragmentan, sus opciones IP, las opciones TCP y el bit URG.

3.4.7. IP Non-Valid Field Values.

Es posible modificar un paquete IP para que contenga valores no válidos en algunos de sus campos. Cuando un equipo recibe un paquete de este estilo modificado, generará un mensaje *ICMP Destination Unreachable*. Por ejemplo, es posible fijar un valor en el campo que especifica el protocolo, que no represente un protocolo válido.

Cuando el sistema objetivo reciba este paquete generará el error ICMP. Si no se recibe esta respuesta, podemos asumir que existe un dispositivo de filtrado que no permite que el paquete llegue a su destino, salvo en algunos Unix, como AIX o HP-UX. Es posible, por tanto, realizar un escaneo probando todos los valores de protocolo posibles, 256 (8 bits). Esta funcionalidad está implementada en la utilidad Nmap.

Sí se detectan muchos protocolos abiertos, indicará que existe un dispositivo de filtrado. Si el dispositivo filtra los mensajes de *ICMP Protocol Unreachable*, entonces parecerá que los 256 protocolos existen y están disponibles. Existen dos opciones para protegerse frente a este ataque.

Por un lado, comprobar que el *firewall* bloquea los protocolos que no están soportados, denegando todo por defecto salvo lo permitido. Por otro, el *firewall* puede bloquear la salida de los paquetes *ICMP Protocol Unreachable* como respuesta al paquete falso.

3.4.8. IP Fragmentation.

Cuando un sistema recibe un fragmento de un paquete IP y algunos de los fragmentos del datagrama total se han perdido, y no son recibidos en un período de tiempo determinado, el sistema descartará ese paquete. Asimismo, generará un mensaje *ICMP Fragment Reassembly Time Exceded* hacia el origen de esa comunicación.

Si se realiza un escaneo hacia todos los puertos TCP y UDP del rango de direcciones IPs de la subred a atacar, es posible determinar la configuración de las ACLs existentes que filtran el tráfico. Si se recibe el paquete ICMP de tiempo excedido al reconstruir los fragmentos, quiere decir que el puerto está disponible y sin filtrar; en caso contrario, o está filtrado o cerrado.

Los fragmentos empleados para este escaneo no pueden tener un tamaño menor al de la cabecera IP más la cabecera TCP o UDP. De nuevo, la contramedida frente a este ataque es no permitir la salida hacia el exterior de los paquetes ICMP de este tipo.

3.5. Sniffing.

Un ataque realmente efectivo, ya que permite la obtención de gran cantidad de información sensible enviada sin encriptar, como por ejemplo usuarios, direcciones de e-mail, claves, números de tarjetas de crédito..., es emplear *sniffers* u olfateadores en entornos de red basados en difusión, como por ejemplo *ethernet* (mediante el uso de concentradores o *hubs*). El análisis de la información trasmitida permite a su vez extraer relaciones y topologías de las redes y organizaciones. Los *sniffers* operan activando una de las interfaces de red del sistema en modo promiscuo.

En este modo de configuración, el *sniffer* almacenará en un *log* todo el tráfico que circule por la tarjeta de red, ya sea destinado o generado por el propio sistema o desde/hacia cualquiera de los sistemas existentes en el entorno de red compartido (segmento *ethernet*). Asimismo, pueden ser instalados tanto en sistemas como en dispositivos de red. La efectividad de esta técnica se basa en tener acceso (habitualmente es necesario además disponer de dicho acceso como administrador o *root*) a un sistema interno de la red; por tanto, no puede ser llevado a cabo desde el exterior.

Antes de la instalación de un *sniffer*, normalmente se instalarán versiones modificadas (*troyanos*) de comandos como “ps” o “netstat” (en entornos Unix), para evitar que las tareas ejecutadas con el *sniffer* sean descubiertas. Cuando los *sniffers* se emplean para la obtención de *passwords*, en ocasiones éstos no son necesarios, ya que los administradores de sistemas descuidan los equipos dejándolos configurados con las *passwords* que por defecto proporcionan los fabricantes. Aparte de los programas independientes existentes para ésta tarea, los sistemas operativos poseen *sniffers* en las distribuciones comerciales, típicamente utilizados por el administrador de red para resolver problemas en las comunicaciones:

Software general:

- Network Associates Sniffer.
- NetXray.
- HP Internet Advisor (dispositivo hardware de escaneo).

Software incluido en múltiples sistemas operativos:

- Unix: ethereal.
- HP-UX: nettl.
- Solaris: snoop.
- Linux: tcpdump.
- Windows: Microsoft Network Monitor.
- Cisco IOS: comandos debug.

3.6. Eavesdropping.

El *Eavesdropping* es una variante del *Sniffing* caracterizada porque únicamente contempla la adquisición o

intercepción del tráfico que circula por la red de forma pasiva, es decir, sin modificar el contenido de la misma.

3.7. Snooping.

De nuevo, ésta es otra variante dentro del *Sniffing* basada en el almacenamiento de la información obtenida en el ordenador del atacante (*downloading*).

También se asocia a la obtención de la información existente en un sistema y no sólo a la extraída del tráfico de red. En este caso, tampoco se modifica la información incluida en la transmisión.

3.8. IP Spoofing.

El *Spoofing* como tal, se basa en actuar en nombre de otro usuario tal y como si se fuese él mismo (*impersonation*). En el caso que se está analizando, TCP/IP, se basa en la generación de paquetes IP con una dirección origen falsa.

El motivo para realizar el envío de paquetes con esa IP puede ser, por ejemplo, que desde la misma se disponga de acceso hacia un sistema destino objetivo, porque existe un dispositivo de filtrado (*screening router* o *firewall*) que permite el tráfico de paquetes con esa dirección IP origen, o porque existe una relación de confianza entre esos dos sistemas.

En los equipos Cisco es muy sencillo implementar este ataque, ya que puede configurarse un *interface* de *loopback* (*interface* lógico interno al *router*), al que se le puede asociar la dirección IP deseada. Por ejemplo:

```
router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
router(config)#int loopback 0
router(config-if)#ip address 31.31.31.33 255.255.255.255
router(config-if)#
!
interface Loopback0
ip address 31.31.31.33 255.255.255.255
!
```

En los S.O. actuales también es posible la configuración de interfaces virtuales que permiten asociarle al sistema más de una dirección IP. Las direcciones IP seleccionadas para emplear esta técnica deben ser direcciones libres en Internet, ya que si el sistema con la dirección IP falseada existe, el funcionamiento normal del mismo será enviar un paquete de *reset* al recibir un SYN-ACK para el cual no envió un SYN (el SYN fue enviado por la herramienta de *Spoofing*). Por tanto, la conexión falseada (*spoofeada)* finalizará.

Otra solución es aplicar alguna técnica de DoS sobre ese sistema, para inhabilitarlo e imposibilitar la respuesta de RST. Existe una modalidad denominada *Blind Spoofing*, que permite llevar a cabo este ataque sin realizar *sniffing* de los datos que se intercambian por la red.

3.9. SMTP Spoofing y Spamming.

En un nivel superior, concretamente a nivel de aplicación, en el protocolo SMTP (puerto TCP 25) es posible falsear la dirección fuente de un correo o e-mail, enviando por tanto mensajes en nombre de otra persona. Es así porque el protocolo no lleva a cabo ningún mecanismo de autenticación cuando se realiza la conexión TCP al puerto asociado.

El *spamming* consiste en el envío masivo de un mensaje de correo a muchos usuarios destino, pudiendo llegar a saturarse los servidores de correo. Suele emplearse para el envío no deseado de publicidad o información.

Aquí finaliza la primera parte de este trabajo.

En la Segunda y última parte, continuaré con el desarrollo de otras **VULNERABILIDADES GENÉRICAS**, entre ellas: Denial of Service, Net Flood, Smurf, TCP Syn Flood, Ddos, Trinoo, Ping of death, Loki, Land, Routing Protocols, Souerce Routing, Caballos de Troya, etc.
Además, una reflexión sobre el **FUTURO** de las redes bajo TCP/IP y las **CONCLUSIONES FINALES**.