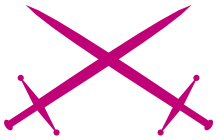

LOS ATAQUES DDOS

Por Jaime Gutierrez y Mateu Llull

Grado de dificultad

A lo largo del tiempo, los ataques de denegación de servicio han sido un medio que junto con otras aplicaciones (recordemos el caso *MyDOOM*) han sido un verdadero quebradero de cabeza para administradores de sistemas y profesionales del sector.

Se caracterizan por su fácil ejecución y su principal rasgo es la dificultad con la que pueden ser mitigados DoS es el acrónimo the *Denial of Service* (*Denegación de Servicio*). Un ataque DoS a un servidor conectado a Internet tiene como objetivo agotar sus recursos, ya sean de ancho de banda o de procesamiento, para que sea (prácticamente) imposible acceder a él. En principio, el realizar un ataque DoS está a disposición de cualquiera que disponga de mayor ancho de banda que el servidor atacado *y/o* haya descubierto alguna vulnerabilidad del sistema operativo que gestiona el servidor (o los routers). Está claro que la primera opción no está al alcance de cualquiera por lo que los ataques DoS suelen ser del segundo tipo. Quizá el ataques DoS por excelencia es y será el conocido ping de la muerte, que consiste en enviar un ping con un paquete de más de 56k que colapsa el sistema, este ataque fué usado en los noventa. Otros tipos básicos de ataque son :

DoS mediante paquetes ICMP (ping)

Es una técnica DoS que pretende agotar el ancho de banda de la víctima. Consiste en enviar de forma continuada un número elevado de paquetes ICMP echo request (ping (1)) de tamaño considerable a la víctima, de forma que esta ha de responder con paquetes ICMP echo reply (pong) lo que supone una sobrecarga tanto en la red como en el sistema de la víctima. Dependiendo de la relación entre capacidad de procesamiento de la víctima y atacante, el grado de sobrecarga varía, es decir, si un atacante tiene una capacidad mucho mayor, la víctima no puede manejar el tráfico generado.

¿QUE ES EL PING?

Se trata de una utilidad que comprueba el estado de la conexión con uno o varios

En este artículo aprenderás...

- todo sobre los ataques DoS;
- sus distintos tipos de ataques y como intentar mitigarlos;
- conocer como funciona un programa para hacer DoS.

Lo que deberías saber...

- nociones básicas sobre las redes TCP/IP;
- conocimientos de programación.

equipos remotos, por medio de los paquetes de solicitud de eco y de respuesta de eco (definidos en el protocolo de red ICMP) para determinar si un sistema IP específico es accesible en una red. Es útil para diagnosticar los errores en redes o enrutadores IP:

```
C:\>ping 192.168.0.1
```

Estadísticas de ping para 192.168.0.1:

```
    Paquetes: enviados = 4,
  recibidos = 4, perdidos = 0
    (0% perdidos),
```

Lo que vemos en la pantalla es una respuesta mostrando la cantidad de bytes que se están enviando y el tiempo que se demoran dichos paquetes.

Al final del test se muestra un resumen con las estadísticas de la prueba.

Algunos comandos PING

Ataque LAND

Un ataque LAND se produce al enviar un paquete *TCP/SYN* falsificado con la dirección del servidor objetivo como si fuera la dirección origen y la dirección destino a la vez. Esto causa que el servidor se responda a sí mismo continuamente acabe desbordándose y al final falle.

Connection FlooD

Todo servicio de Internet orientado a conexión (la mayoría) tiene un límite máximo en el número de conexiones simultaneas que puede tolerar. Una vez que se alcanza ese límite, no se admitirán conexiones nuevas.

Listado 1.

Haciendo ping a 192.168.0.1 con 32 bytes de datos:

```
Respuesta desde 192.168.0.1: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.0.1: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.0.1: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.0.1: bytes=32 tiempo<1m TTL=128
```

Listado 2. *Algunos comandos PING*

```
-t    Ping el host especificado hasta que se pare.
   Para ver estadísticas y continuar - presionar Control-Inter;
   Parar - presionar Control-C.
-a    Resolver direcciones en nombres de host.
-n cuenta    Número de peticiones eco para enviar.
-l tamaño    Enviar tamaño del búfer.
-f    Establecer el indicador No fragmentar en los paquetes.
-i TTL    Tiempo de vida.
-v TOS    Tipo de servicio.
```

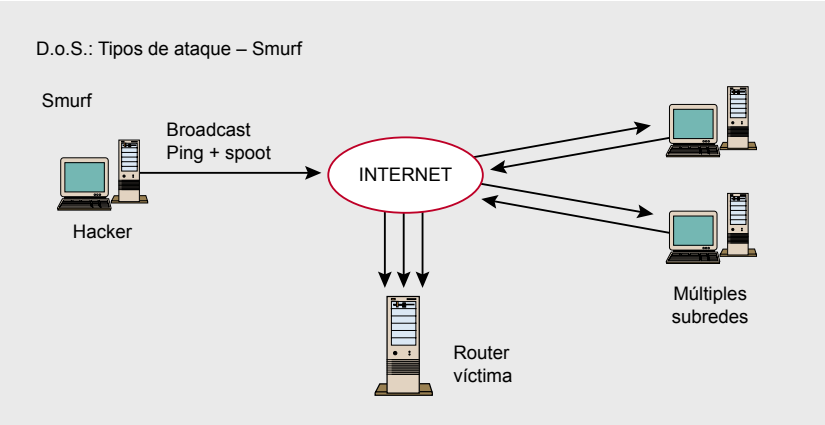

Figura 1. *Estructura de un ataque Smurf*

Así, por ejemplo, un servidor Web puede tener capacidad para atender a mil usuarios simultáneos. Si un atacante establece mil conexiones y no realiza ninguna petición sobre ellas, monopolizará la capacidad del servidor. Las conexiones van caducando por inactividad poco a poco, pero el atacante sólo necesita intentar conexiones nuevas constantemente, como ocurre con el caso del *syn flood*.

Afortunadamente este ataque implica que la conexión tiene lugar o, lo que es lo mismo, que se completa la negociación en tres pasos que comentábamos en la sección anterior. Debido a ello la máquina atacada tiene constancia de la identidad real del atacante. Al menos, si sus administradores merecen su sueldo y saben qué comandos utilizar.

Ataques Smurf

El ataque *smurf* utiliza una característica de Internet: broadcast. Toda red tiene lo que se denomina una dirección de *broadcast*. Los datagramas enviados a esa dirección son recibidos por todas las máquinas en la red local. Ello permite, por ejemplo, que una máquina localice un servidor proporcionando un servicio haciendo una pregunta a la red, no preguntando máquina por máquina.

El problema de la dirección *broadcast* es que suele estar disponible también para usuarios de fuera de la red local, en particular para todo Internet. Ello permite, por ejemplo, que un atacante envíe un pequeño datagrama a toda una red remota, y que las máquinas de dicha red respondan todas a la vez, posiblemente con un datagrama de mayor tamaño. Si la red sondeada tiene 150 máquinas activas, la respuesta es 150 veces más intensa. Es decir, se consigue un efecto *multiplicador*.

Ataques DNS y de enrutamiento

La mayoría de los protocolos de enrutamiento como RIP (*Routing Information Protocol*) o BGP (*Border Gateway Protocol*) carecen de autenticación, o tienen una muy sencilla.

Se trata por tanto de un escenario perfecto para que cualquier atacante pueda alterar las rutas correctas y, falsificando su IP origen, crear una condición DoS. Las víctimas de estos ataques verán como su tráfico se dirige por ejemplo hacia un agujero negro: a una red que no existe.

Tabela 1. *Lista de puertos usados generalmente para el ataque DoS y sus correspondientes fallas*

Servicio	Puerto/Protocolo	Vulnerabilidad
Ftp	21/TCP	Wu-ftpd "site-exec" , FTP bounce, problemas de validación en el FTPD
Telnet	23/TCP	Demonio telnet IRIX
Ssh	22/TCP	Buffer overflow en el demonio SSH
Domain	53/UPD	Vulnerabilidades en el BIND
LinuxConf	98/TCP	Vulnerabilidades en el servicio linuxconf
Pop2	109/TCP	Ipop2d buffer overflow
Pop3	110/TCP	Qpopper buffer overflow y vulnerabilidades en IMAP
Runrpc	111/TCP	Rpc.statd buffer overflow en amd, mountd , rpc.cmsd.
Netbios-ns	137/UPD	Recursos de windows no protegidos
Netbios-dgm	138/UPD	
Netbios-ssn	139/TCP	
Imap	143TCP	Buffer overflow en algunas implementaciones IMAP
ObjectServer	5135/TCP	Vulnerabilidad en Objectserver IRIX 5.3 y 6.2

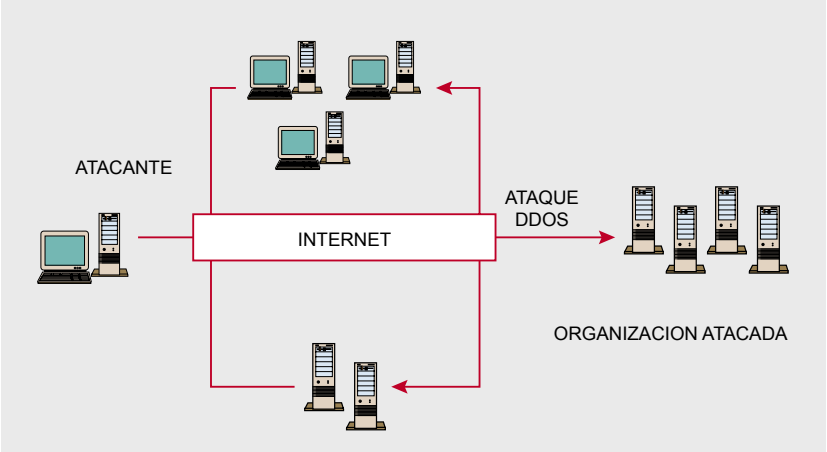

Figura 2. *Estructura de un ataque DDoS*

Los ataques DoS sobre servidores de nombres de dominios (DNS) son tan problemáticos como los anteriores. Estos ataques intentan convencer al servidor DNS, por ejemplo, para almacenar direcciones falsas: cuando un servidor DNS realiza una búsqueda el atacante puede redireccionar a su propio servidor o bien a un *agujero negro*.

Ataques DDOS

Los ataques DDoS son ataques DoS, pero distribuidos, es decir mediante mas host remotos.

ANALIZANDO LOS ATAQUES (DdoS)

Como ya hemos mencionado antes, estos tipos de ataques se basan en coordinar una serie de host conectados a Internet de tal forma que concentren su ataque simultáneamente y desde diferentes lugares, sobre un único objetivo. Los distintos tipos de ataques DDoS, como por ejemplo, colapsar un servicio simulando conexiones de miles de usuarios a la vez, o haciendo que el tiempo de espera sea elevado, aumentado de esta forma el tiempo de respuesta, generando así una gran cantidad de tráfico que como consecuencia del gran consumo de ancho de banda (bandwicht) agota los recursos de una *aplicación/servidor*. Algunas de las causas que hacen que el DDoS sea un ataque tan extendido es,por ejemplo, que el protocolo *TCP/IP* no tenga seguridad: ya que se diseñó para su empleo en una comunidad abierta y confiada y la versión que actualmente se usa tiene defectos inherentes y graves. No puede modificarse o implementarse otro tipo de seguridad por el simple motivo de que Internet dejaría de funcionar. Igualmente muchas implementaciones del *TCP/IP* en sistemas operativos e incluso dispositivos físicos de red, tienen defectos que debilitan su capacidad para resistir ataques.

ESTRUCTURANDO UN ATAQUE DDOS

Un usuario ilegitimo, busca con el ataques DDoS agotar los recursos

de una aplicación o tomar este como un ataque secundario, para proceder a vulnerar otro sistema. La estructura de un ataque conjunto DoS, es casi siempre la misma. El primer paso que busca el atacante, es infectar al mayor numero de víctimas (lo que significa mas host remotos) posible, esto se conseguía mediante worms, troyanos, y otros tipos de bichos que principalmente busca en el sistema un puerta por la cual acceder. Una vez el sistema infectado, el atacante instala en las máquinas remotas, una aplicación la cual le va a permitir realizar este tipos de ataques. Un usuario consigue la implantación de estos en miles de máquinas, lo que le producirá una red botnet (como ya hemos tratados en artículos anteriores).

ESTRUCTURANDO UNA DEFENSA CONTRA UN DdoS

Un ataque de DDoS hacia tu web puede ser letal para colapsar el servidor, y así gastar todo tu *bandwicht*. Por eso la defensa empieza en los principales proveedores y los ISP. Podemos soportar el ataque de varias ips (proveniente de varios host remotos) pero ¿qué pasa cuando un Botnet de más de 5000 zombis ataca nuestro servidor?, estaríamos indefensos y perplejos sintiéndonos impotentes, al no poder realizar nada. Para evitar esto debemos llevar una política de seguridad estricta. Vamos a ver unos métodos de defensa de un servidor (tomándolo como referencia). El ataque de un DDoS contra un servidor, se concentra en gastar su ancho de banda, y colapsarlo, para ello vamos a analizar posibles subidas anormales de trafico analizado los *archivos.logs*. Instalación de módulos de seguridad (en nuestro caso el *mod_evasive*) un modulo destinado a intentar mitigar el DDoS, se configura para servidores Apache veamos una configuración básica :

```
<IfModule mod_evasive.c>
DOSHashTableSize 3097
DOSPageCount 5
DOSSiteCount 100
DOSPageInterval 2
DOSSiteInterval 2
DOSBlockingPeriod 600
</IfModule>
```

DOSPageCount 5

DOSSiteCount 100

Si la ip, recarga 100 páginas a menos de 2 páginas por segundo, esta será banneada desde el servidor, deteniendo el floodeo que provenía por parte de esa dirección.

No permitir el tráfico *broadcast* desde fuera de nuestra red:

De esta forma evitamos ser empleados como *multiplicadores* durante un ataque *smurf*.

Filtrar el tráfico *IP spoof*:

Esto es algo que todo *carrier* o *backbone* debería hacer, y que permitiría localizar y solucionar el problema con una gran rapidez. En pocas palabras, estos filtros evitan que un atacante se pueda hacer pasar por otro usuario.

Listado 1. *Listado de un ataque DDoS, el archivo log del servidor*

```
Jul 14 13:05:30 lang kernel:NET: 1594 messeges suppressed.
Jul 14 13:05:30 lang kernel:TCP drop open request from 20.14.237.83/3876
Jul 14 13:05:36 lang kernel:NET: 633 messeges suppressed.
Jul 14 13:05:36 lang kernel:TCP drop open request from 86.212.167.27/1116
Jul 14 13:05:39 lang kernel:NET: 970 messeges suppressed.
Jul 14 13:05:39 lang kernel:TCP drop open request from 81.38.172.161/3040
Jul 14 13:05:45 lang kernel:NET: 548 messeges suppressed.
Jul 14 13:05:45 lang kernel:TCP drop open request from 81.203.228.102/2119
Jul 14 13:05:50 lang kernel: 421 messeges suppressed.
Jul 14 13:05:50 lang kernel:TCP drop open request from 81.203.228.102/2478
Jul 14 13:05:56 lang kernel:379 messeges suppressed.
Jul 14 13:05:56 lang kernel:TCP drop open request from 81.203.228.102/4005
Jul 14 13:05:59 lang kernel:891 messeges suppressed.
Jul 14 13:05:59 lang kernel:TCP drop open request from 81.38.172.161/3568
```

No tener proxies abiertos a todo Internet:

Algunos administradores tienen sus proxies, wingates, open sesame, *SOCKS*, *SQUIDs*, etc. abiertos a todo el mundo, sin ser conscientes de ello. Esto permite que cualquier usuario de Internet pueda atacar cualquier sistema responsabilizando a esa red intermedia mal administrada.

- Afinamiento en TCP/IP;
- Aumentar backlog;
- Disminuir el timeout.
- Teoría de como hacer un DoS a una Web

El ataque consistirá en sobrecargar el servidor de la Web mediante una conexión TCP/IP para asi colapsarlo y provocar un DoS.

Ahora que ya tenemos una pequeña idea de como vamos a trabajar empecemos a tocar el Visual Basic.

Para hacer una conexión *TCP/IP* se tendrá que agregar el control Winsock que usa la dll *WS2_32*, para C++ es la misma.

Bien, puesto que presupongo que hay personas de los que lean este texto no saben Visual Basic voy a explicar que el código.

Ahora que lo tenemos agregado vamos a empezar a codear, primero pondré el código y luego lo explicaré.

Lo que tenemos que agregar sea lo siguiente:

- 2 CommandButton;
- 1 timer;
- 1 label.

Recalco que los nombres de los comandos deben de ser los que vienen por defecto.

Ahora iremos por partes, primero las declaraciones globales:

```
Dim Host As String
Dim Puerto As String
```

Como vemos declaramos dos cadenas (*Strings*), una que contendrá la IP de la pagina web y la otra el puerto a atacar.

En este ejemplo siempre trabajaremos por el puerto 80, aunque se puede conectar a otros, tantos como la web tenga abiertos.

Bien, ahora veremos el código que tiene que ir dentro del Timer, que es el más importante:

```
Private Sub Timer1_Timer()
On Error Resume Next
Winsock1.Close
Winsock1.Connect
Winsock1.SendData''
YYYYYYYYYYYYYYY
YYYYYYYYYY
YYYYYYYYY''
End Sub
```

Primero, con el On `Error Resume Next` evitamos que si se produce un error nos cierre el programa, así que con esto si hay algún error lo omitirá y no nos cerrará el programa.

`El Winsock1.Close` es obligatorio ponerlo, ya que si estamos en un bucle, para volver a conectar se tiene que cerrar primero el *socket*.

Bien, ahora con el `Winsock1.Connect` conectamos al servidor y con el siguiente comando (`Winsock1.SendData`) le enviamos varios caracteres, para aumentar así el consumo de ancho de Banda. Cabe destacar que la frase que hay dentro de las comillas se puede substituir por la que queráis, esta es solo de ejemplo.

En las propiedades tenéis que poner en el Intervalo del `Timer el n° 1`, ya que esto indica que lo que hay dentro del `Timer1` se va a repetir cada milésima de segundo.

Ahora pasemos al código del Formulario en el evento Load (es decir, cuando el formulario se cargue):

```
Private Sub Form_Load()
Host = ''64.125.10.23''
Puerto = ''80''
End Sub
```

Aquí vemos que cuando cargue introduciremos dentro de la variable *Host* la IP del Website y en Puerto el puerto por el cual conectaremos.

En el evento *Error* del *Winsock* podemos poner un *MessageBox* que nos avise de que la conexión a fallado, esto no es obligatorio, pero si es recomendable para saber el estado de la conexión.

En el *Evento Conect* del *Winsock* podemos copiar unas cuantas veces el Comando `Winsock1.SendData` y la frase que le persigue, esto aumentara más aun el ancho de banda consumido, aunque es recomendable no abusar, con ponerlo unas 5 veces ya vale.

Y bueno, con este comando tan simple ya nos valdría para hacer un DoS a un *Website* que no tenga mucho ancho de banda, aunque esto no solo afecta al ancho de banda, sino que afecta al propio procesador del *Servidor*, que tiene que procesar los datos que le enviamos, las conexiones nuestras y la de los otros visitantes y esto provoca el colapso del *Servidor*.

Como aquí se pretende ayudar a todo el mundo les diré que estos ataques se pueden parar configurando correctamente las *IPTables*.

Teoría sobre como hacer un DoS en un servidor mediante un Worm

Otra manera de hacer un DoS es mediante un Gusano, lo que suelen hacer los gusanos que hacen DoS a websites es que a una determinada hora envían todos una petición a un Servidor. Supongamos que el gusano a infectado a 10.000 Pc's, y estos realizaran una petición a al vez al servidor. Si el ancho de banda del servidor es pequeño el efecto seria como un *Tsunami*, una *avalancha* de peticiones para un solo servidor al mismo tiempo.

Teoría sobre como hacer un DoS en un servidor de IRC

Lo que se tiene que hacer para colapsar un servidor de IRC es conectarse a el mediante Bots, ¿que son Bots? Pues son programas que se conectan a un canal de IRC automáticamente, están programados para ello. Con ello se puede conseguir desde inundar un Canal haciendo hablar a todos los bots a la vez, esto probablemente produciría una sobrecarga de recursos en el servidor para procesar todos los mensajes y podría desde sacar a todos los usuarios del IRC del Canal a provocar un DoS en el servidor. Evidentemente todo depende del número de Bots que programemos. Normalmente se suelen programar con un Array de sockets, que cada uno trabaja independientemente.

Teoría de como hacer un DoS en un servidor de juegos

Ahora vamos con el DoS mas *complicados* de hacer que estén en los que e explicado.

Estos juegos suelen utilizar el protocolo UDP, por lo tanto es imposible hacer un ataque de IP Spoofing, ahora verán porque.

Bien, para ver el protocolo que utiliza el Juego vamos a hacer que el cliente del servidor se conecte a nosotros mismos para ver las cabeceras del protocolo, es decir, el server solo responde a un tipo de respuestas que solo conoce el cliente y el servidor, si el protocolo interno no es el mismo que el establecido se ignoraran las peticiones por parte del servidor. Bien, para sacar el protocolo podemos usar el Netcat, ya que es bastante útil para esto. Ponemos en escucha el netcat en un puerto aleatorio (el que queramos) y luego en el Cliente establecemos que se conecte al netcat, en la IP ponemos 127.0.0.1 (lcalhost) y de puerto ponemos el puerto que habíamos puesto en el netcat, si no nos hemos equivocado en la pantalla del Netcat saldrá una frase, esta frase es la que envía el cliente al Servidor para identificarse.

Ahora cogemos Visual Basic y con el Winsock programamos un programa que se conecte al Servidor enviándole la frase que habíamos sacado del Netcat, luego, si todo funciona bien, el servidor nos enviara otra frase distinta y al final de todo, un código (un numero o una frase). Aquí esta el problema para hacer un IP Spoofing (2), ya que el atacante no recibe los pa-

quetes enviados desde el Servidor a la IP spoofeada.

IP SPOOFING

IP *SPOOFING*: Es la suplantación de IP. Consiste básicamente en sustituir la dirección IP origen de un paquete *TCP/IP* por otra dirección IP a la cual se desea suplantar. Esto se consigue generalmente gracias a programas destinados a ello y puede ser usado para cualquier protocolo dentro de *TCP/IP* como ICMP, UDP o TCP. Hay que tener en cuenta que las respuestas del host que reciba los paquetes irán dirigidas a la IP falsificada. Por ejemplo si enviamos un ping (paquete icmp *echo request*) spoofeado, la respuesta será recibida por el host al que pertenece la IP legalmente. Este tipo de spooofing unido al uso de peticiones *broadcast* a diferentes redes es usado en un tipo de ataque de *flood* conocido como *smurf* ataque. Para poder realizar IP *SPOOFING* en sesiones TCP, se debe tener en cuenta el comportamiento de dicho protocolo con el envío de paquetes SYN y ACK con su ISN específico y teniendo en cuenta que el propietario real de la IP podría (si no se le impide de alguna manera) cortar la conexión en cualquier momento al recibir paquetes sin haberlos solicitado. También hay que tener en cuenta que los *routers* actuales no admiten el envío de paquetes con IP origen no perteneciente a una de las redes que administra (los paquetes spoofeados no sobrepasarán el *router*).

En general se usa el termino *SPOOFING*, como falseo de cualquier terminología, como por ejemplo: *MAC SPOOFING* (falseo de una dirección mac) *EMAIL SPOOFING* (falseo de un e-mail).

Bien, ahora otra vez con el Visual Basic ponemos un puerto a la escucha con el Winsock, esta vez conectaremos el Cliente del juego a nuestro programa y al recibir los datos le enviaremos lo que el servidor nos a enviado.

Si lo hacemos bien veremos que el servidor nos da otra frase, en la cual se puede ver el nick del Jugador (en el caso del juego Quake III) y demás configuraciones.

Ahora ya al enviar esto al servidor desde nuestra aplicación podremos entrar como si fuésemos un Cliente del Quake normal.

Ahora que sabemos como diseñar un Bot para el juego lo que podemos hacer es crear mas bots con un array de Sockets(3), lo que sucederá sera igual que lo que pasa en el servidor del IRC, que al haber tantos bots hablando a la misma vez pasara que el servidor sacara a todos los usuarios del juego fuera.

QUE ES UN SOCKET

Un *socket* (*enchufe*), es un método para la comunicación entre un programa del cliente y un programa del servidor en una red. Un *socket* se define como el punto final en una conexión. Los *sockets* se crean y se utilizan con un sistema de peticiones o de llamadas de función a veces llamados interfaz de programación de aplicación de *sockets* (API, *application programming interface*).

Un *socket* también puede ser una dirección de Internet, combinando una dirección IP (la dirección numérica única de cuatro partes que identifica a un ordenador particular en Internet) y un número de puerto (el número que identifica una aplicación de Internet particular, como FTP, Gopher, o WWW).

CONCLUSIÓN Y FINAL

Como hemos visto a lo largo del artículo los ataques de denegación de servicio, se han convertido en un medio deseado para atacantes, debido a que su ejecución no es un proceso complicado. Estos temas son tratados como secundarios, es decir, estiman poca atención a los administradores y profesionales, pero debemos tenerlos muy presentes a la hora de estructurar una política de seguridad. En un principio los ataques DoS son medidas de sabotajes, de las que el atacante no puede esperar obtener nada a cambio como el acceso a un servidor, datos, o otro tipo de beneficio. Sin embargo, el pasado más reciente ha demostrado que los ataques DoS si pueden traer beneficios. De esta forma, los ataques a importantes ofertas de Internet como Yahoo! O eBay ofrecieron un fuerte tirón a la cotización de las empresas de la competencia. Tras finalizar los ataques se recuperaron rápidamente las cotizaciones. Puede que todo fuese obra de un *CRACKER* con un depósito de acciones. El estudio de un plan a medida de seguridad, la rápida intervención de los medios provisto, el continuo análisis de la red. Ayudan a minimizar y detectar precozmente este tipo de ataques ●

En la red

- *http://www2.axent.com/swat/news/ddos.html* – Información y prevención de riesgos;
- *http://www.ackstorm.es/security_analisis_dos.cfm* – Ataques DdoS;
- *http://en.wikipedia.org/wiki/Denial-of-service_attack* – Denial OF Servicie Atacck.

Sobre el Autor

Jaime Gutierrez es un joven, que se interesa por la seguridad informática en general, y en especial el estudio de métodos de ataques contra aplicaciones en red. En sus ratos libres se los pasa programando aplicaciones en Java.

Mateu Llull es un joven programador en varios lenguajes, interesado por la seguridad Web y la Criptografía. Pasas sus ratos programado aplicaciones orientada a la seguridad en redes.